

ADAPTATION OF A CONTROL SYSTEM  
TO  
VARYING MISSILE CONFIGURATIONS

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖZGÜR EKİNCİ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
AEROSPACE ENGINEERING

DECEMBER 2009

Approval of the thesis:

**ADAPTATION OF A CONTROL SYSTEM TO VARYING MISSILE CONFIGURATIONS**

submitted by **ÖZGÜR EKİNCİ** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering, Middle East Technical University** by,

Prof. Dr. Canan Özgen  
Dean, Graduate School of **Natural and Applied Sciences**

\_\_\_\_\_

Prof. Dr. Ozan Tekinalp  
Head of Department, **Aerospace Engineering**

\_\_\_\_\_

Assist. Prof. Dr. İlkay Yavrucuk  
Supervisor, **Aerospace Engineering Dept., METU**

\_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. Ozan Tekinalp  
Aerospace Engineering Dept., METU

\_\_\_\_\_

Assist. Prof. Dr. İlkay Yavrucuk  
Aerospace Engineering Dept., METU

\_\_\_\_\_

Prof. Dr. Serkan Özgen  
Aerospace Engineering Dept., METU

\_\_\_\_\_

Assist. Prof. Dr. Ali Türker Kutay  
Aerospace Engineering Dept., METU

\_\_\_\_\_

Dr. Osman Merttopçuoğlu  
Chief Systems and Control Engineer, ROKETSAN

\_\_\_\_\_

**Date:**

\_\_\_\_\_

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last name :

Signature :

# **ABSTRACT**

## **ADAPTATION OF A CONTROL SYSTEM TO VARYING MISSILE CONFIGURATIONS**

Ekinci, Özgür

M.S. Department of Aerospace Engineering

Supervisor : Assist. Prof. Dr. İlkey Yavrucuk

December 2009, 90 pages

Varying missile configurations may create uncertainty for a missile control algorithm developed with linear control theory, for instance the control system performance requirements may not be satisfied anymore. Missile configuration may change during the missile design period due to variations in subsystem locations, subsystem weights and missile geometry. Likewise, burning propellant, deployment of aerodynamic surfaces and wings with varying sweep angle can be considered as in-flight missile configuration changes. This thesis study addresses development and analysis of an adaptive missile control algorithm to account for the uncertain effects caused by varying missile configuration.

Control algorithms, designed using pole placement, are augmented with adaptive neural networks. The resulting controller is a type of model reference adaptive controller. Adaptation characteristics of the augmented control algorithms are investigated to changing center of pressure location and missile geometry. Analyses are performed for three different missile configurations using simulation.

Keywords: Missile, Neural Networks, Adaptive Augmentation,

# ÖZ

## DEĞİŞEN FÜZE KONFIGÜRASYONLARINA KONTROL SİSTEMİ ADAPTASYONU

Ekinci, Özgür

Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü

Tez Yöneticisi : Yar. Doç. Dr. İlkey Yavrucuk

Aralık 2009, 90 sayfa

Değişen füze konfigürasyonları, lineer kontrol teorisi ile geliştirilen füze kontrol algoritmaları için, kontrol sistemi performans gereksinimlerinin karşılanamayacağı şekilde belirsizlik yaratabilmektedir. Füze konfigürasyonu, tasarımı süreci sırasında alt sistem yerleşiminin, alt sistem ağırlıklarının ve füze geometrisinin farklılaşması sebebi ile değişebilir. Ayrıca, yanan yakıt, açılan aerodinamik yüzeyler ve değişen kontrol yüzeyi süpürme açısı, uçuş sırası füze konfigürasyon değişiklikleri olarak değerlendirilebilir. Bu tez çalışması, değişen füze konfigürasyonlarının neden olduğu belirsiz etkilerin telafi edilmesi amacı ile adaptif füze kontrol algoritmalarının geliştirilmesi ve analizini gösterir.

Kök yerleştirme ile tasarlanan kontrol algoritmaları, adaptif sinir ağları ile geliştirilmiştir. Ortaya çıkan kontrolcü, bir çeşit model referans adaptif kontrolcüdür. Geliştirilmiş kontrol algoritmalarının, değişen ağırlık merkezi ve füze geometrisine adaptasyon karakteristiği incelenmiştir. Üç değişik füze konfigürasyonu için benzetim aracı kullanılarak analizler yapılmıştır.

Anahtar Kelimeler: Füze, Sinir Ağı, Adaptif Geliştirme

~ *To My Family* ~

## ACKNOWLEDGMENT

I wish to express my deepest gratitude to my supervisor Assist. Prof. Dr. İlkey Yavrucuk for his guidance, advice, criticism, encouragements and insight throughout the research.

I would also like to thank Dr. Osman Merttopçuoğlu for his advice and criticism on the organization of this thesis.

Valuable advice, comments and help of Assist. Prof. Dr. Ali Türker Kutay on the research subject is gratefully acknowledged.

I would like to thank to Mr. Hakan Tiftikçi for his guidance and support on the organization, missile aerodynamics and missile control.

I would like to thank to ROKETSAN A.Ş. for providing the engineering experience to improve myself on the subject.

Assistance of anyone who has a contribution somehow is also gratefully acknowledged.

Lastly, I would like to thank to my family who always loved, supported and encouraged me.

# TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZ .....	v
ACKNOWLEDGMENT.....	vii
TABLE OF CONTENTS.....	viii
LIST OF TABLES.....	x
LIST OF FIGURES.....	xi
LIST OF SYMBOLS.....	xiv
CHAPTERS	
1 INTRODUCTION.....	1
1.1 Problem Statement.....	1
1.2 Literature Survey.....	2
1.3 Motivation and Contribution of this Thesis.....	4
1.4 Thesis Outline.....	4
2 PRELIMINARIES.....	5
2.1 Pole Placement Method [4].....	5
2.2 Existing Control System Augmentation [10].....	8
2.3 Frames of Reference and Coordinate Axes.....	14
2.3.1 Earth Fixed Reference Frame $F_E$ (Axes, $O_E X_E Y_E Z_E$ ).....	14
2.3.2 Missile Body Fixed Reference Frame $F_B$ (Axes, $O_B X_B Y_B Z_B$ ).....	15
2.3.3 Thrust Reference Frame $F_F$ (Axes, $O_F X_F Y_F Z_F$ ).....	15
2.3.4 Wind Reference Frame $F_A$ (Axes, $O_A X_A Y_A Z_A$ ).....	15
2.4 Coordinate Axes Transformations.....	17
2.4.1 Euler Angles Formulation.....	18
2.4.2 Direction Cosine Matrix Formulation.....	19
2.4.3 Euler-Rodrigues Quaternion Formulation.....	20



3	CONFIGURATION SELECTION AND MATHEMATICAL MODELING..	22
3.1	Missile Configuration Selection.....	22
3.1.1	Desired Missile Properties .....	22
3.1.2	Missile Baseline Configuration Alternatives .....	24
3.1.3	Varying Missile Configurations.....	25
3.2	Missile Aerodynamics.....	28
3.2.1	Aerodynamics of Configurations C1, C2 and C3 .....	31
3.2.2	Non-Linear Parameters in Missile Aerodynamics .....	35
3.3	Missile Dynamics.....	36
3.4	Thrust Model.....	43
3.5	Control Actuation System (CAS) Model .....	44
4	AUTOPILOT DESIGN.....	46
4.1	Control Requirements .....	46
4.2	Linear Missile Models.....	48
4.2.1	Linear Missile Model for Pitching Motion .....	49
4.2.2	Linear Missile Model for Yawing Motion.....	51
4.2.3	Linear Missile Model for Rolling Motion.....	52
4.3	Missile Acceleration Response Characteristics .....	53
4.4	Full-State Feedback Autopilot Design Using Pole Placement.....	54
4.5	Adaptive Augmentation of Existing Autopilots.....	58
5	RESULTS AND DISCUSSION .....	61
5.1	Analysis 1 .....	63
5.2	Analysis 2.....	66
5.3	Analysis 3.....	70
5.4	Analysis 4.....	74
5.5	Analysis 5.....	79
6	CONCLUSIONS.....	83
	REFERENCES.....	85
	APPENDIX A AERODYNAMIC ANALYSIS .....	88
A.1	Missile DATCOM.....	88

# LIST OF TABLES

## TABLES

Table 3.1 Aerodynamic Surface Arrangements .....	23
Table 3.2 Aerodynamic Surface Alignments .....	23
Table 3.3 Plus and Cross Orientation Control Forces.....	23
Table 3.4 Alternative Missiles [19].....	24
Table 3.5 Aerodynamic Database Coefficients.....	29
Table 4.1 Transient Response Performance Requirements.....	47
Table 4.2 Open Loop Poles for Pitch and Yaw Autopilots.....	56
Table 4.3 Open Loop Poles for Roll Autopilot.....	56
Table 4.4 Pitch, Yaw and Roll Autopilot Gains for Calculated for C1 .....	57
Table 4.5 Closed Loop Poles for Pitch and Yaw Autopilots (using Gains for C1) .	57
Table 4.6 Closed Loop Poles for Roll Autopilot (using Gains for C1).....	57
Table 4.7 Linear CAS Model Closed Loop Poles.....	58
Table 5.1 Description of Analyses Cases.....	62
Table 5.2 Parameters for Analysis 1 .....	63
Table 5.3 Parameters for Analysis 2 .....	66
Table 5.4 Parameters for Analysis 3 .....	70
Table 5.5 Parameters for Analysis 4 .....	74
Table 5.6 Parameters for Analysis 5 .....	79
Table A.1 Input and Output Files for DATCOM.....	88
Table A.2 DATCOM Input Parameters .....	89

# LIST OF FIGURES

## FIGURES

Figure 2.1 Pole Placement Method for a Type 1 Plant .....	5
Figure 2.2 Pole Placement Method for a Type 0 System Model .....	6
Figure 2.3 Closed Loop Augmentation Architecture .....	8
Figure 2.4 SHL Neural Network Structure .....	10
Figure 2.5 Earth Axes .....	14
Figure 2.6 Missile Body Axes.....	15
Figure 2.7 Thrust Vector Axes.....	16
Figure 2.8 Wind Axes .....	16
Figure 2.9 Body Axes Orientations [13].....	17
Figure 3.1 Baseline Configuration AIM 9D (Modeled in SolidWorks) .....	24
Figure 3.2 Configuration C1 – Baseline Configuration .....	25
Figure 3.3 Configuration C2 – Variation 1 .....	26
Figure 3.4 Configuration C3 – Variation 2 .....	27
Figure 3.5 Flow Diagram for Aerodynamic Database Generation .....	28
Figure 3.6 Configuration C1 Center of Pressure Locations.....	31
Figure 3.7 Configuration C2 Center of Pressure Locations.....	32
Figure 3.8 Configuration C3 Center of Pressure Locations.....	32
Figure 3.9 $C_M$ and $C_Z$ versus Alpha at 1.8 Mach for $0^\circ$ Delta .....	33
Figure 3.10 $C_M$ Coefficient at Mach 1.8 for C1 .....	34
Figure 3.11 $C_Z$ Coefficient at Mach 1.8 for C1.....	34
Figure 3.12 $C_M$ and $C_N$ versus Mach at $10^\circ$ Alpha .....	35
Figure 3.13 Axes and Vectors Used to Derive the Equations of Motion.....	36
Figure 3.14 Variable Thrust Engine Model .....	43
Figure 3.15 Non-linear Actuator Model Response to Step Input.....	45

Figure 4.1 Definitions of Transient-Response Specifications.....	46
Figure 4.2 Configuration C1 Acceleration Transfer Function Root Locus .....	53
Figure 4.3 Pitch Autopilot Architecture with Pole Placement Approach.....	55
Figure 4.4 Yaw Autopilot Architecture with Pole Placement Approach.....	55
Figure 4.5 Roll Autopilot Architecture with Pole Placement Approach .....	55
Figure 4.6 Architecture for Adaptive Augmentation of Pitch Autopilot .....	59
Figure 4.7 Architecture for Adaptive Augmentation of Yaw Autopilot.....	60
Figure 5.1 Analysis 1 – Elevator Deflection vs Time.....	64
Figure 5.2 Analysis 1 – Angle of Attack vs Time.....	64
Figure 5.3 Analysis 1 – Pitch Acceleration Response .....	65
Figure 5.4 Analysis 2 – Pitch A/P Rise Time with Mach.....	67
Figure 5.5 Analysis 2 – Yaw A/P Rise Time with Mach.....	67
Figure 5.6 Analysis 2 – Roll A/P Rise Time with Mach .....	68
Figure 5.7 Analysis 2 – Pitch A/P Percent Overshoot with Mach.....	68
Figure 5.8 Analysis 2 – Yaw A/P Percent Overshoot with Mach.....	69
Figure 5.9 Analysis 2 – Roll A/P Percent Overshoot with Mach .....	69
Figure 5.10 Analysis 3 – Pitch Acceleration Response .....	71
Figure 5.11 Analysis 3 – Pitch Acceleration Response (5s – 6s).....	71
Figure 5.12 Analysis 3 – Pitch Rate.....	72
Figure 5.13 Analysis 3 – Elevator Control Inputs.....	72
Figure 5.14 Analysis 3 – Pitch NN Input – Hidden Layer Weights .....	73
Figure 5.15 Analysis 3 – Pitch NN Hidden – Output Layer Weights.....	73
Figure 5.16 Analysis 4 – Pitch Acceleration Response .....	75
Figure 5.17 Analysis 4 – Pitch Acceleration Response (5s – 6s).....	75
Figure 5.18 Analysis 4 – Pitch Acceleration Response (65s – 66s).....	76
Figure 5.19 Analysis 4 – Pitch Rate.....	76
Figure 5.20 Analysis 4 – Elevator Control Inputs.....	77
Figure 5.21 Analysis 4 – Pitch NN Input – Hidden Layer Weights .....	77
Figure 5.22 Analysis 4 – Pitch NN Hidden – Output Layer Weights.....	78

Figure 5.23 Analysis 5 – Pitch Acceleration Response .....	80
Figure 5.24 Analysis 5 – Pitch Acceleration Response (2.5s – 3s).....	80
Figure 5.25 Analysis 5 – Pitch Rate.....	81
Figure 5.26 Analysis 5 – Elevator Control Inputs.....	81
Figure 5.27 Analysis 5 – Pitch NN Input – Hidden Layer Weights .....	82
Figure 5.28 Analysis 5 – Pitch NN Hidden – Output Layer Weights.....	82

## LIST OF SYMBOLS

$x$	State vector
$u$	Control vector
$y$	Output vector
$A$	System matrix
$B$	Input matrix
$C$	Output matrix
$D$	Feed-forward matrix
$K$	Feedback vector
$E$	Error vector
$\Delta(.,.)$	System modeling errors
$\hat{V}$	Input – Hidden layer learning rate
$\hat{W}$	Hidden – Output layer learning rate
$F_A$	Reference frame A
$O_A$	Origin of reference frame A
$X_A, Y_A, Z_A$	axes of reference frame A
$\phi, \theta, \psi$	Missile Euler Angles
$x, y, z$	Missile position vector components
$a_x, a_y, a_z$	Missile accelerations
$\alpha$	Missile angle of attack
$\beta$	Missile sideslip angle
$L_{AB}$	A-B transformation matrix
$L_1(A)$	Rotation matrix for a degree rotation along first axis
$C_{12}$	12 Direction cosine angle
$S_\phi$	Sine of angle $\phi$

$C_\phi$	Cosine of angle $\phi$
$e_0, e_x, e_y, e_z$	Quaternion parameters
$E_X, E_Y, E_Z$	Euler axis
$p, q, r$	Missile Angular velocity components
$\omega_M$	Missile angular velocity vector
$\tilde{\omega}_M$	Cross product operator for missile angular velocity vector
$WL$	Wing loading
$W$	Missile weight
$b$	Wing span
$c$	Wing chord
$S$	Wing area
$d$	Root chord – Tip chord
$X, Y, Z$	Aerodynamic force components
$L, M, N$	Aerodynamic moment components
$T$	Thrust force
$Q$	Dynamic pressure
$S_{ref}$	Reference area
$l_{ref}$	Reference length
$V_{MA}$	Missile free stream velocity
$\delta_a, \delta_e, \delta_r$	Virtual CAS deflections
$\delta_1, \delta_2, \delta_3, \delta_4$	Actual CAS deflections
$F_A$	Force vector expressed w.r.t. Frame A
$L_A$	Linear momentum vector expressed w.r.t. Frame A
$H_A$	Moment of momentum vector expressed w.r.t. Frame A
$M_A$	Moment vector expressed w.r.t. Frame A
$r_{A,B}$	Position vector components expressed w.r.t. Frame B
$V_{A,B}$	Velocity vector components expressed w.r.t. Frame B
$m$	Mass of the missile

$I_B$	Inertia Matrix
$\omega_{CAS}$	CAS model natural frequency
$\zeta_{CAS}$	CAS model damping coefficient
PID	Proportional Integral Derivative
LQR	Linear Quadratic Regulator
LQG	Linear Quadratic Gaussian
SHL	Single Hidden Layer
NN	Neural Network
CG	Center of Gravity
C1	Configuration 1
C2	Configuration 2
C3	Configuration 3
DCM	Direction Cosine Matrix
STT	Skid to Turn
CAS	Control Actuation System



# CHAPTER 1

## INTRODUCTION

### 1.1 Problem Statement

The overall performance of a missile is a strong function of the control system performance. Missile control algorithms developed with linear methods are often subject to model uncertainties originated by varying missile configuration, unmodeled dynamics, low fidelity aerodynamic models, and actuator failures etc. Uncertain effects may cause degradation in performance of a missile control system. Within the scope of this thesis, changes in configuration of a missile system are studied. Missile configuration may change during missile design period due to variations in subsystem locations, subsystem weights and missile geometry. Also, missile configuration may change during the flight because of the burning propellant, deployment of aerodynamic surfaces and wings with varying sweep angle.

This thesis research applies an adaptive control algorithm to account for the effects caused by varying configuration, such that the system satisfies the performance requirements. Therefore, existing control algorithms are augmented with neural networks. Performance of the augmented control algorithms are analyzed for changing center of gravity location and changing missile geometry using simulations for three different missile configurations. First missile configuration is selected to be the baseline configuration and the second and third missile configurations are modified by varying the center of gravity location and shape of the missile aerodynamic surfaces respectively.

## 1.2 Literature Survey

Most missile control systems are designed using classical control methods which are based on well known PID and three loop control architectures. PID algorithms have shown to be suitable and perform adequately for missile acceleration control [1]. Likewise, acceleration control systems based on three loop control is a common control technique including the rate stabilization loop, synthetic stabilization loop and acceleration loop [2]. With three loop control method acceleration response of missiles is found out to be quite well [3]. For both PID and three loop architectures, measurable outputs like acceleration (with accelerometers) and angular rates (with gyroscopes) are used as feedback [3]. This is an important property since it alleviates the need for observers to estimate unmeasured states. Implementations of both control methods require gain scheduling to maintain flight stability and control [1] & [3].

Another common approach to control missile acceleration is pole placement. Theory of this technique is based on assigning the closed loop poles to desired locations. To find the desired pole locations, dominant poles are placed as the desired response is satisfied and the remaining poles are placed such that their effect on closed loop dynamics are small [4]. Alternatively, LQR like methods based on optimum control theory is used to determine desired pole locations. Unfortunately, state feedback designs require knowledge of full state parameters. However, common problem for designers is that these parameters are often unmeasured and have to be estimated. There exists an output feedback theory presented in [4] for the state observer based extension of the previous approaches. Alternatively, Kalman estimator extension of LQR architecture known as LQG is a solution to full state feedback problem. LQG method is designed considering a Gaussian measurement noise and it is feasible to implement this architecture on a control system with noisy measurements. Pole placement methods usually satisfy the desired performance for missile acceleration control if designed well as is shown in this thesis and [4].

In order to analyze the problem defined in the previous section, adaptive control methods are investigated. Some application of adaptive control theory to missile

systems are based on adaptive dynamic inversion control, L1 adaptive control and reference model following adaptive control approaches.

Dynamic inversion control is a technique using the approximate model inverse to create control signals and compensate the modeling error by adding an adapting control signal. One way to calculate adaptive control signal is using neural networks as in this study. Also, angle of attack and side slip angle control of bank to turn missiles are presented in [5] and [6]. Simulation results in these references have shown that achieving both rapid learning and robustness is difficult. Also, introductory study on lateral acceleration control system for a skid to turn missile based on adaptive dynamic inversion is presented in [7].

Methods for neural network augmentation of existing control systems implement a reference model defining the desired dynamics and an adaptive structure to compensate for errors caused by modeling and linearization. There are several application examples of this method as adaptation of control gains directly or augmentation of an existing state/output feedback controller with neural networks. Some applications of this method to augment an existing controller is given in [8] [9] and [10]. An augmentation of a simple PI controller for a nonlinear system is presented in [8]. Simulation results indicate that reference model tracking for a simple system is acceptable. Similar approach as in [8] is applied to JDAM guided bomb for angle of attack control and tracking of the reference model is achieved with a time delay. Also [10] presents simulation results of an arbitrary open loop and closed loop reference model following architectures developed with the observer based output feedback extension. Simulation results in this study showed significant improvements by means of control system performance with applications to guided projectile, micro adaptive flow control actuators and several numerical examples.

Adaptive control technique called L1 is developed to filter unwanted high frequency signals on the adaptive control signal observed at some model referenced adaptive control approaches [11]. Removal of high frequency adaptive control signals that disturb the system let the designer to increase learning rates for the adaptive structures. An application of this method for unmanned combat air vehicle

X-45A is presented in [12] and it's shown that results obtained with this method are superior to ones obtained with a model reference adaptive control approach.

### **1.3 Motivation and Contribution of this Thesis**

Motivation of this thesis is to augment the existing control system of an agile missile system to adapt to configuration changes such that the performance requirements are still satisfied.

Contributions of this thesis are,

- Effect of change in missile center of gravity location on control system is analyzed.
- Effect of change in external missile geometry on control system is analyzed.
- Adaptive neural network augmentation of existing pole placement control algorithm of an agile missile system is performed using model reference adaptive control approach.
- Different adaptation characteristics with the augmented control algorithms are shown.

### **1.4 Thesis Outline**

Chapter 1 introduces the problem and gives references about the theory and applications.

Chapter 2 presents theoretical background on used control system design methods, formulations and control architectures.

Chapter 3 explains the design approaches, models and simulation related studies to be used in the analysis of the control systems.

Chapter 4 includes autopilot performance requirements, design approaches and autopilot design parameters on some design conditions.

Chapter 5 presents analysis results for several design conditions.

Chapter 6 discusses analysis results and presents the conclusions.

# CHAPTER 2

## PRELIMINARIES

### 2.1 Pole Placement Method [4]

To design control systems with the pole placement method, different approaches depending on the plant type are used. To design type 1 closed loop systems with this method, two different approaches are used depending on whether the plant itself is type 1 or type 0 [4]. Type of a system is determined by the order of “s” multiplier in the characteristic equation in other words with the number of integrating actions. In control engineering, type of a closed loop system is important to interpret the steady state error characteristics depending on the reference input.

A type 1 closed loop system design method is presented in Figure 2.1.

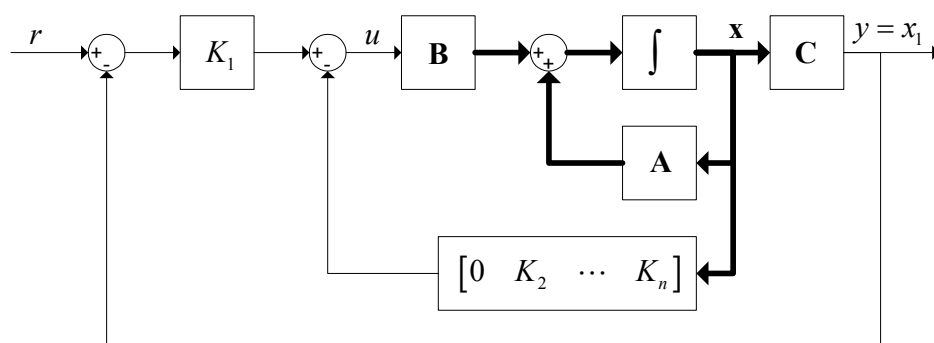


Figure 2.1 Pole Placement Method for a Type 1 Plant

Open loop system model is given in (2.1).

$$\begin{aligned}\dot{x}_{n \times 1} &= A_{n \times n} x_{n \times 1} + B_{n \times 1} u \\ y &= C_{1 \times n} x_{n \times 1}\end{aligned}\quad (2.1)$$

For the following input signal  $u$ , closed loop system dynamics are found as in (2.2)

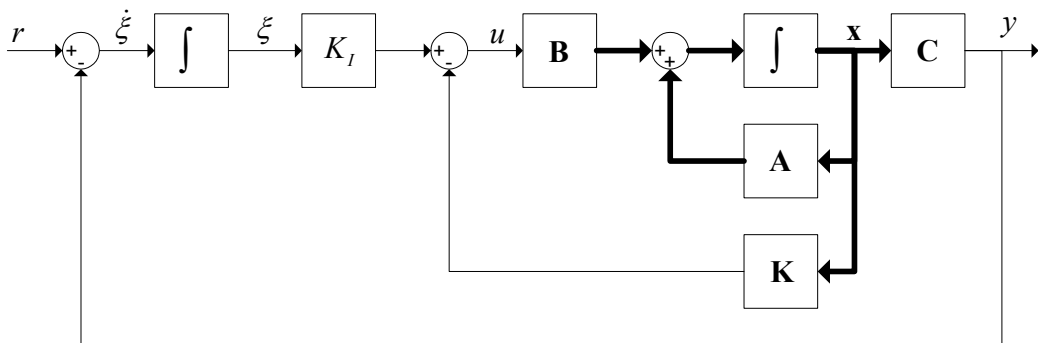
$$\begin{aligned}u &= -K_{1 \times n} x_{n \times 1} + K_1 r \\ \dot{x}_{n \times 1} &= (A_{n \times n} - B_{n \times 1} K_{1 \times n}) x_{n \times 1} + B_{n \times 1} K_1 r \\ y &= C_{1 \times n} x_{n \times 1}\end{aligned}\quad (2.2)$$

Where,

$$K_{1 \times n} = [K_1 \quad K_2 \quad \dots \quad K_n]$$

Feedback gain vector  $K$  is calculated with Ackerman's Formula such that poles of the closed loop system are assigned to desired locations on  $s$  plane.

Next architecture which is pole placement approach for type 0 plants is presented in Figure 2.2 [4]. This method adds an integrating action in order to increase the order of the closed loop system such that it is type 1. This property is desired since response of type 1 systems is better compared to type 0 systems in terms of steady state error.



**Figure 2.2 Pole Placement Method for a Type 0 System Model**

Open loop system model is given in (2.3).

$$\begin{aligned}\dot{x}_{n \times 1} &= A_{n \times n} x_{n \times 1} + B_{n \times 1} u \\ y &= C_{1 \times n} x_{n \times 1}\end{aligned}\quad (2.3)$$

For the following input signal  $u$ , closed loop system dynamics are found as (2.4).

$$\begin{aligned}u &= -\hat{K}_{1 \times (n+1)} \hat{x}_{(n+1) \times 1} \\ \hat{x}_{(n+1) \times 1} &= \hat{A}_{(n+1) \times (n+1)} \hat{x}_{(n+1) \times 1} + \hat{B}_{(n+1) \times 1} u \\ y &= \hat{C}_{1 \times (n+1)} \hat{x}_{(n+1) \times 1}\end{aligned}\quad (2.4)$$

Or in open form,

$$\begin{aligned}\begin{bmatrix} \dot{x}_{n \times 1} \\ \dot{\xi} \end{bmatrix} &= \begin{bmatrix} (A_{n \times n} - B_{n \times 1} \cdot K_{1 \times n}) & B_{n \times 1} \cdot K_l \\ -C_{1 \times n} & 0 \end{bmatrix} \begin{bmatrix} x_{n \times 1} \\ \xi \end{bmatrix} + \begin{bmatrix} 0_{n \times 1} \\ 1 \end{bmatrix} r \\ y &= [C_{1 \times n} \quad 0] \begin{bmatrix} x_{n \times 1} \\ \xi \end{bmatrix}\end{aligned}$$

Where,

$$\hat{K}_{1 \times (n+1)} = [K_{1 \times n} \quad \vdots \quad -K_l]$$

Feedback gain vector  $\hat{K}$  is calculated with Ackerman's Formula using the closed loop system matrices  $\hat{A}$  and  $\hat{B}$  such that poles of the closed loop system are assigned to desired locations on  $s$  plane as in the previous case.

## 2.2 Existing Control System Augmentation [10]

Arbitrary reference model following adaptive control approach presented in Figure 2.3 is used to augment the existing pole placement control system as presented in [9] and [10].

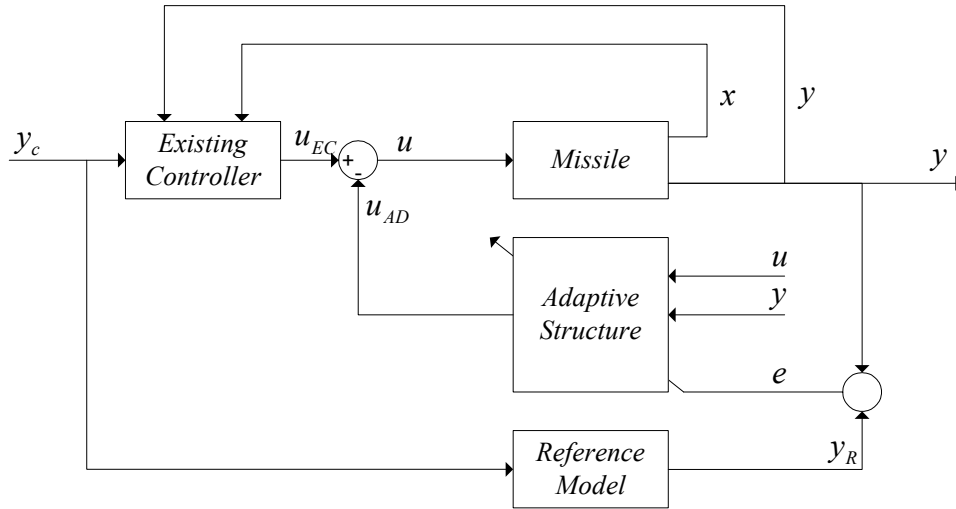


Figure 2.3 Closed Loop Augmentation Architecture

To explain the errors cancelled with adaptive structure, error dynamics is formulated. Missile dynamics can be considered as an observable and stabilizable nonlinear SISO system in normal form [13] as given in (2.5).

$$\begin{aligned}
 \dot{z} &= f(z, x) \\
 \dot{x}_1 &= x_2 \\
 &\vdots \\
 \dot{x}_{r-1} &= x_r \\
 \dot{x}_r &= h_r(z, x, u) \\
 y &= x_1
 \end{aligned} \tag{2.5}$$



Where  $\mathbf{z} \in \mathbb{R}^{n-r}$  are states of internal dynamics,  $\mathbf{x} = [x_1 \ \dots \ x_r] \in \mathbb{R}^r$  are states of output dynamics,  $u, y \in \mathbb{R}$  are control and measurement variables,  $\mathbf{f}$  and  $h$  are sufficiently smooth partially known functions, and  $r$  is the relative degree of the system.

As a reference model  $r^{\text{th}}$  order linear stable system with full relative degree in (2.6) is selected.

$$\begin{aligned} \dot{x}_{m_1} &= x_{m_2} \\ &\vdots \\ \dot{x}_{m_{r-1}} &= x_{m_r} \\ \dot{x}_{m_r} &= \mathbf{C}_r \mathbf{x}_m + D_r y_c \\ y_m &= x_{m_1} \end{aligned} \quad (2.6)$$

Or,

$$\begin{aligned} \dot{\mathbf{x}}_m &= \mathbf{A}_m \mathbf{x}_m + \mathbf{b}_m y_c \\ y_m &= \mathbf{c}_m^T \mathbf{x}_m \end{aligned} \quad (2.7)$$

Defining tracking error as  $e = y_m - y$ , the error dynamics can be expressed from (2.5) and (2.7) as,

$$\begin{aligned} \dot{\mathbf{E}} &= \mathbf{A}_m \mathbf{E} + \mathbf{b}_m (u_{AD} - \Delta(\mathbf{x}, \mathbf{z}, y_c, u_{AD})) \\ y &= e = \mathbf{c}_m^T \mathbf{E} \end{aligned} \quad (2.8)$$

Where

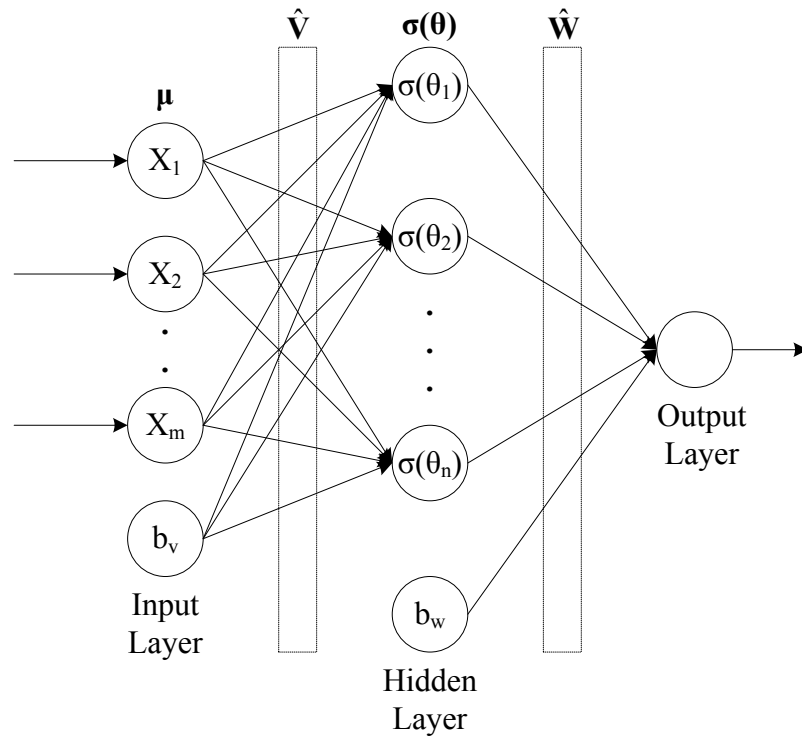
$$\mathbf{E} = [e \ \dot{e} \ \dots \ e^{r-1}]$$

$$\Delta(\mathbf{x}, \mathbf{z}, y_c, u_{AD}) = D_r^{-1} (h_r(\mathbf{z}, x_1, \dots, x_r, u) - \mathbf{C}_r \mathbf{x}_m) - y_c + u_{AD}$$

Control goal with this approach is to design an adaptive control input such that it cancels modeling errors defined by  $\Delta$ .

For the adaptive structure presented in Figure 2.3, neural networks is selected. There are various neural network architectures such as *Radial Basis Function Neural Networks* and *Single Hidden Layer Neural Networks* [9] that are used to augment the control systems.

*Single Hidden Layer Neural Network*, a nonlinear in the parameters feed forward neural network structure, is used as an adaptive element. Details of the formulation of SHLNN are given in references [8], [9] and [14].



**Figure 2.4 SHL Neural Network Structure**

Existing control system augmentation with neural networks is done as,

$$u = u_{EC} - u_{AD} \quad (2.9)$$

$$u_{AD} = \hat{W}^T \sigma(\hat{V}^T \mu) \quad (2.10)$$

Sigmoid activation function  $\sigma$  is defined as given in (2.11).

$$\sigma(\theta) = \frac{1}{1 + e^{-\theta}} \quad (2.11)$$

Also the input - hidden layer and hidden - output layer weights  $\hat{\mathbf{V}}$  and  $\hat{\mathbf{W}}$  are defined as,

$$\hat{\mathbf{V}} \triangleq \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m,1} & v_{m,2} & \cdots & v_{m,n} \\ v_{b1} & v_{b2} & \cdots & v_{bn} \end{bmatrix} \quad (2.12)$$

$$\hat{\mathbf{W}} \triangleq \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ w_b \end{bmatrix} \quad (2.13)$$

The input vector  $\boldsymbol{\mu}$  hidden layer activation function vector  $\boldsymbol{\sigma}$  is defined as

$$\boldsymbol{\mu} \triangleq \begin{bmatrix} X_1 \\ \vdots \\ X_m \\ b_v \end{bmatrix} \quad (2.14)$$

$$\boldsymbol{\sigma}(\boldsymbol{\theta}) \triangleq \begin{bmatrix} \sigma(\theta_1) \\ \vdots \\ \sigma(\theta_n) \\ b_w \end{bmatrix} \quad (2.15)$$

Derivation of adaptation laws required to adapt neural networks weights  $\hat{\mathbf{W}}$  and  $\hat{\mathbf{V}}$  in (2.10) includes the following steps.

1. Define the error dynamics.
2. Define a Lyapunov energy like function  $V(\mathbf{e}, \hat{\mathbf{V}}, \hat{\mathbf{W}})$
3. Perform Lyapunov stability analysis finding the  $\dot{V}(\mathbf{e}, \hat{\mathbf{V}}, \hat{\mathbf{W}})$
4. Conditions satisfying stability requirement ( $\dot{V}(\mathbf{e}, \hat{\mathbf{V}}, \hat{\mathbf{W}}) \leq 0$ ) gives the adaptation laws that guarantee stable error dynamics outside a compact set in the error state space also implying that all the parameters are bounded.

Performing these steps, adaptation laws [8] in (2.16) are found.

$$\begin{aligned}\dot{\hat{\mathbf{V}}} &= -\Gamma_V (\boldsymbol{\mu} \zeta \hat{\mathbf{W}}^T \boldsymbol{\sigma}(\hat{\mathbf{V}}^T \boldsymbol{\mu}) + \lambda \|\zeta\| \hat{\mathbf{V}}) \\ \dot{\hat{\mathbf{W}}} &= -\Gamma_W ((\boldsymbol{\sigma}(\hat{\mathbf{V}}^T \boldsymbol{\mu}) - \boldsymbol{\sigma}'(\hat{\mathbf{V}}^T \boldsymbol{\mu}) \hat{\mathbf{V}}^T \boldsymbol{\mu}) \zeta + \lambda \|\zeta\| \hat{\mathbf{W}})\end{aligned}\quad (2.16)$$

Where  $\Gamma_V$ ,  $\Gamma_W$  represents learning rates,  $\lambda$  represent e-modification gain and  $\zeta$  defined as,

$$\zeta = \mathbf{e}^T \mathbf{P} \mathbf{B}$$

Also  $\boldsymbol{\sigma}'$  is the Jacobian of  $\boldsymbol{\sigma}$  and defined as,

$$\boldsymbol{\sigma}'(\boldsymbol{\theta}) \triangleq \begin{bmatrix} \frac{d\sigma(\theta_1)}{d\theta_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{d\sigma(\theta_n)}{d\theta_n} \\ 0 & 0 & 0 \end{bmatrix}\quad (2.17)$$

Where,

$$\frac{d\sigma(\theta_i)}{d\theta_i} = a \cdot \sigma(\theta_i) \cdot (1 - \sigma(\theta_i)) \quad i = 1 \dots n$$

To show the boundedness of the signals in the closed loop control algorithms the following assumptions are made.

**Assumption 2.1:** The true plant given in (2.5) is minimum phase, i.e. internal dynamics  $\dot{z} = f(z, x)$  with  $x = \mathbf{0}$  are asymptotically stable.

**Assumption 2.2:** The existing controller is bounded input – bounded output stable. Adaptive control signal introduced in (2.9) is computed with the neural network as follows.

Then, Theorem 2.1 guarantees the boundedness of the signals in the control system.

**Theorem 2.1:** Consider the system (2.5) under the control of existing control algorithm and neural network augmenting controller. Let Assumption 2.1 and Assumption 2.2 hold. Then all the signals in the augmented missile system are *uniformly ultimately bounded* [10].

One design restriction is that parameters and errors may grow outside the allowable bounds in the compact set discussed above (derivative of Lyapunov function is positive in that set). So the neural network parameters should be tuned such that the size of the compact set is allowable. However, finding the optimum parameters for the augmentation scheme may require many simulations.

### 2.3 Frames of Reference and Coordinate Axes

Right handed coordinate axes are used as a convention. Since the missile to be used in analysis is a short range missile, non-rotating, non-accelerating and flat earth is assumed. Hence, any reference frame defined on the earth is assumed to be an inertial frame of reference relative to which Newton's laws of motion are valid.

#### 2.3.1 Earth Fixed Reference Frame $F_E$ (Axes, $O_E X_E Y_E Z_E$ )

Earth-fixed reference frame is defined on the ground plane as shown in Figure 2.5. Origin of the earth-fixed reference frame is defined at a point on the ground from which the initial position vector to missile is perpendicular to the ground plane.  $O_E X_E$  axis is defined in any convenient direction.  $O_E Y_E$  axis is also on the ground plane to the right of  $O_E X_E$  direction and  $O_E Z_E$  axis is downwards.

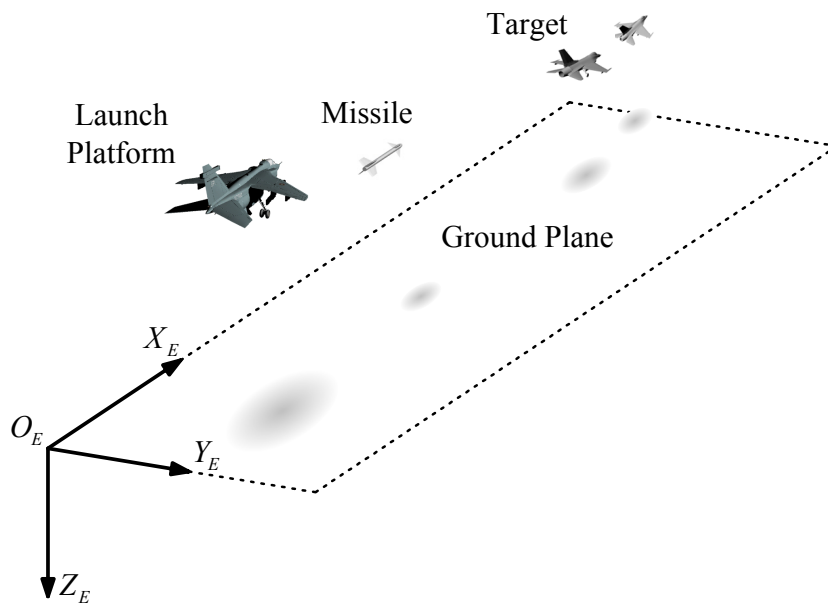


Figure 2.5 Earth Axes

### 2.3.2 Missile Body Fixed Reference Frame $F_B$ (Axes, $O_B X_B Y_B Z_B$ )

Missile body-fixed reference frame is attached to center of gravity of the missile as shown in Figure 2.6.  $O_B X_B$  axis is longitudinal axis pointing forward direction of the missile.  $O_B Y_B$  axis is lateral axis to the right and  $O_B Z_B$  axis is vertical axis to the down, at the missile back view.

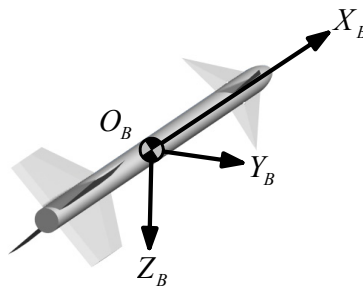


Figure 2.6 Missile Body Axes

### 2.3.3 Thrust Reference Frame $F_F$ (Axes, $O_F X_F Y_F Z_F$ )

Origin of thrust reference frame is attached to the nozzle as presented in Figure 2.7.  $O_F X_F$  axis is aligned to thrust vector.  $O_F Y_F$  axis is always in  $X_B Y_B$  plane perpendicular and to the right of  $O_F X_F$ .  $O_F Z_F$  axis is defined such that it's orthogonal to  $X_F Y_F$  plane completing a right handed coordinate axes.

### 2.3.4 Wind Reference Frame $F_A$ (Axes, $O_A X_A Y_A Z_A$ )

Origin of wind reference frame is center of gravity of the missile and coincides with  $O_B$  as shown in Figure 2.8.  $O_A X_A$  axis is always aligned with wind vector.  $O_A Z_A$  axis is defined such that it is always in  $X_B Z_B$  plane so called symmetry plane of the missile.  $O_A Y_A$  axis is defined such that it is completing an orthogonal right handed coordinate axes.

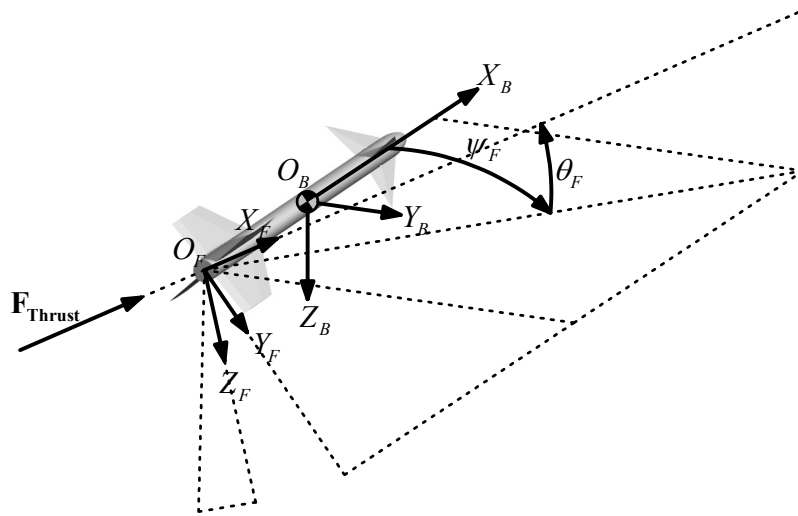


Figure 2.7 Thrust Vector Axes

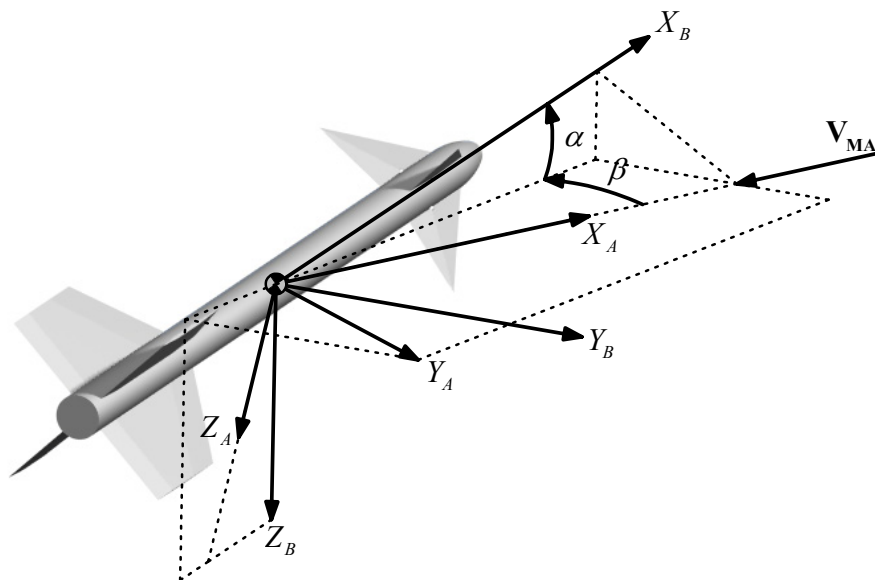


Figure 2.8 Wind Axes



## 2.4 Coordinate Axes Transformations

Orientation of coordinate axes with respect to another can be defined in different ways. One of them is as three rotation angles called Euler angles. One other method is using a magnitude of rotation and an axis of rotation called Euler axis or Eigen axis.

Based on these methods to relate coordinate axes, formulations used in aerospace literature for coordinate axes transformation of vector components are as follows [13];

1. Euler angles formulation
2. Direction cosine matrix formulation
3. Euler-Rodrigues quaternion formulation

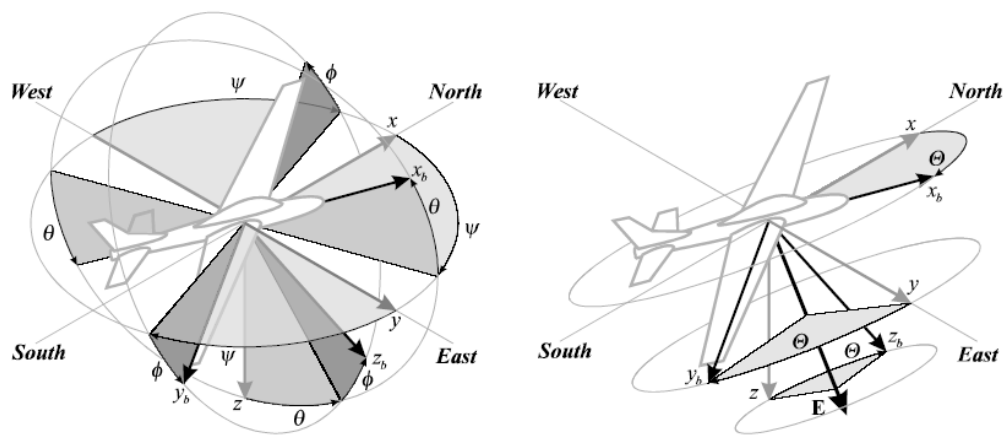


Figure 2.9 Body Axes Orientations [13]

Coordinate axes transformations are used to define quantities and solve equations of motion in a simpler way. As an example, for a missile moving in inertial space, it is more convenient to express equations of motion on body coordinate axes since the

moment equations include inertia terms which are constant with respect to body coordinate axes.

For the case where one of the coordinate axes is not stationary, as body-fixed coordinate axes in the missile, transformation matrix is a function of time. Thus transformation matrix and its parameters have to be calculated instantaneously using rate of change of those parameters (Euler angle rates, direction cosine rates, quaternion rates).

Formulations listed before is given in between missile body-fixed reference frame and earth-fixed reference frame in the following sections and is adapted for any coordinate system transformation needed in simulation.

#### 2.4.1 Euler Angles Formulation

Euler angles defining the orientation of  $F_B$  with respect to  $F_E$  is given as,

$$\Phi_{M,E} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

Transformation matrices for individual rotations on  $O_E X_E$ ,  $O_E Y_E$  and  $O_E Z_E$  axis is defined as,

$$\mathbf{L}_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{bmatrix} \quad \mathbf{L}_2(\theta) = \begin{bmatrix} C_\theta & 0 & -S_\theta \\ 0 & 1 & 0 \\ S_\theta & 0 & C_\theta \end{bmatrix} \quad \mathbf{L}_3(\psi) = \begin{bmatrix} C_\psi & S_\psi & 0 \\ -S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where,

$$S_x = \sin(x)$$

$$C_x = \cos(x)$$

Although it is possible to use twelve [13] different rotation sequences to calculate Euler angle transformation matrices, aerospace engineering convention is “321” rotations or “Yaw-Pitch-Roll” rotations. 321 rotations can be visualized as a rotation of magnitude  $\psi$  on yawing axis  $O_E Z_E$ , then a rotation of magnitude  $\theta$  on the rotated pitching axis  $O_E Y'_E$  and lastly a rotation of  $\phi$  on twice rotated rolling axis  $O_E X''_E$ .

Transformation matrix from  $F_E$  to  $F_B$  for 321 rotations is then;

$$\mathbf{L}_{BE} = \mathbf{L}_1(\phi) \cdot \mathbf{L}_2(\theta) \cdot \mathbf{L}_3(\psi)$$

$$\mathbf{L}_{BE} = \begin{bmatrix} C_\theta \cdot C_\psi & C_\theta \cdot S_\psi & -S_\theta \\ S_\phi \cdot S_{\theta_M} \cdot C_\psi - C_\phi \cdot S_\psi & S_\phi \cdot S_\theta \cdot S_\psi + C_\phi \cdot C_\psi & S_\phi \cdot C_\theta \\ C_\phi \cdot S_\theta \cdot C_\psi + S_\phi \cdot S_\psi & C_\phi \cdot S_\theta \cdot S_\psi - S_\phi \cdot C_\psi & C_\phi \cdot C_\theta \end{bmatrix} \quad (2.18)$$

Also for the rotation in (2.18), Euler angle rates  $\dot{\boldsymbol{\phi}}_{M,E}$  is given with angular velocity of the missile  $\boldsymbol{\omega}_{M,B}$  are presented with (2.19) and the derivation is given in [13] and [16].

$$\dot{\boldsymbol{\phi}}_{M,E} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \frac{S_\phi \cdot S_\theta}{C_\theta} & \frac{C_\phi \cdot S_\theta}{C_\theta} \\ 0 & C_\phi & -S_\phi \\ 0 & \frac{S_\phi}{C_\theta} & \frac{C_\phi}{C_\theta} \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.19)$$

It should be noted that Euler angle formulation has a singularity called ‘‘Gimbal Lock’’ since Euler angle rates are undefined for  $\pm\pi/2$ . Although the formulation is easy to understand and interpret it should be used for applications in which attitude change is not close to singular points. This problem is solved with other formulations explained in the following sections.

#### 2.4.2 Direction Cosine Matrix Formulation

Elements of a direction cosine matrix are called the direction cosines and they represent orientation of  $F_B$  with respect to  $F_E$ . Direction cosines are defined as the angles in between for all possible combinations of axis couples. Direction cosine matrix for the transformation is expressed as,

$$\mathbf{L}_{BE} = \mathbf{DCM} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (2.20)$$

Since for any formulation, transformation of a vector between same coordinate axes should result the same vector components, all Euler angle transformation matrices based on different sequences are equal to the one in (2.20).

Rate of change of DCM is given by the following formula known as Poisson's kinematic equations derived in [13] and [17].

$$\dot{\mathbf{L}}_{\mathbf{BE}} = \begin{bmatrix} \dot{C}_{11} & \dot{C}_{12} & \dot{C}_{13} \\ \dot{C}_{21} & \dot{C}_{22} & \dot{C}_{23} \\ \dot{C}_{31} & \dot{C}_{32} & \dot{C}_{33} \end{bmatrix} = - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \cdot \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}$$

$$\dot{\mathbf{L}}_{\mathbf{BE}} = -\tilde{\boldsymbol{\omega}}_{\mathbf{M},\mathbf{B}} \cdot \mathbf{L}_{\mathbf{BE}} \quad (2.21)$$

Also rate for reverse transformation [17] is given as (2.22).

$$\dot{\mathbf{L}}_{\mathbf{EB}} = \mathbf{L}_{\mathbf{EB}} \cdot \tilde{\boldsymbol{\omega}}_{\mathbf{M},\mathbf{B}} \quad (2.22)$$

Time rate of change of DCM expressed in (2.21) has no singularity problem. But note that the degree of freedom for describing the orientation is three which means that six of the nine equations given in (2.21) are redundant. Numerical integration of these redundant equations may cause orthogonality problems with time. Also this formulation needs high computational power compared to other transformation formulations which is no longer a problem for today's processor technology.

### 2.4.3 Euler-Rodrigues Quaternion Formulation

Transformation matrix formed using Euler-Rodrigues quaternion formulation is a method which doesn't suffer from a singularity problem and has the highest computational efficiency.

This formulation is based on Euler Axis formulation and it overcomes the singularity problem which Euler Axis formulation also suffers.

Quaternion parameters are defined as [13],

$$\mathbf{Q} = \begin{bmatrix} e_0 \\ e_x \\ e_y \\ e_z \end{bmatrix} = \begin{bmatrix} C_{\Theta/2} \\ E_X \cdot S_{\Theta/2} \\ E_Y \cdot S_{\Theta/2} \\ E_Z \cdot S_{\Theta/2} \end{bmatrix} \quad (2.23)$$

$\mathbf{E}$  is the Euler axis as given below and  $\Theta$  is the magnitude of rotation along Euler axis.

$$\mathbf{E} = \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

Quaternion also can be calculated from Euler angles as in (2.24) with the following formulation which is used in the simulation for calculating initial conditions for quaternion integrations.

$$\mathbf{Q} = \begin{bmatrix} e_0 \\ e_x \\ e_y \\ e_z \end{bmatrix} = \begin{bmatrix} C_{\phi/2} \cdot C_{\theta/2} \cdot C_{\psi/2} + S_{\phi/2} \cdot S_{\theta/2} \cdot S_{\psi/2} \\ S_{\phi/2} \cdot C_{\theta/2} \cdot C_{\psi/2} - C_{\phi/2} \cdot S_{\theta/2} \cdot S_{\psi/2} \\ C_{\phi/2} \cdot S_{\theta/2} \cdot C_{\psi/2} + S_{\phi/2} \cdot C_{\theta/2} \cdot S_{\psi/2} \\ C_{\phi/2} \cdot C_{\theta/2} \cdot S_{\psi/2} - S_{\phi/2} \cdot S_{\theta/2} \cdot C_{\psi/2} \end{bmatrix} \quad (2.24)$$

Transformation matrix derived from quaternion parameters is given in (2.25).

$$\mathbf{L}_{\mathbf{BE}} = \begin{bmatrix} e_x^2 + e_0^2 - e_y^2 - e_z^2 & 2 \cdot (e_x \cdot e_y - e_z \cdot e_0) & 2 \cdot (e_x \cdot e_z + e_y \cdot e_0) \\ 2 \cdot (e_x \cdot e_y + e_z \cdot e_0) & e_y^2 + e_0^2 - e_x^2 - e_z^2 & 2 \cdot (e_y \cdot e_z - e_x \cdot e_0) \\ 2 \cdot (e_x \cdot e_z - e_y \cdot e_0) & 2 \cdot (e_y \cdot e_z + e_x \cdot e_0) & e_z^2 + e_0^2 - e_x^2 - e_y^2 \end{bmatrix} \quad (2.25)$$

Above expression of quaternion parameters is also equal to the DCM found in (2.18) and (2.20).

Quaternion rates for the integration of quaternion parameters are given as,

$$\dot{\mathbf{Q}} = \begin{bmatrix} \dot{e}_0 \\ \dot{e}_x \\ \dot{e}_y \\ \dot{e}_z \end{bmatrix} = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} e_0 \\ e_x \\ e_y \\ e_z \end{bmatrix} \quad (2.26)$$

Or alternatively,

$$\dot{\mathbf{Q}} = \begin{bmatrix} \dot{e}_0 \\ \dot{e}_x \\ \dot{e}_y \\ \dot{e}_z \end{bmatrix} = \begin{bmatrix} -e_x & -e_y & -e_z \\ e_0 & -e_z & e_y \\ e_z & e_0 & -e_x \\ -e_y & e_x & e_0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.27)$$

# CHAPTER 3

## CONFIGURATION SELECTION AND MATHEMATICAL MODELING

### 3.1 Missile Configuration Selection

Configuration selection of the missile is performed such that the missile characteristics and configuration which affect the missile performance is as desired. For this purpose, desired properties are defined, alternative baseline configurations are searched and geometry of baseline and varying configurations are selected.

#### 3.1.1 Desired Missile Properties






In order to observe the effects of varying configurations on the control system, neutrally stable dynamics for most of the flight regime is desired. For neutral stability, center of pressure should be around the center of gravity.

STT maneuvering is desired in which missile is non-rolling and pitching and yawing motion is possible at any instant [18]. For STT maneuvering agility and maneuverability is higher than other strategies. Moreover, aerodynamic analysis with this maneuvering type is simpler due to no rolling motion and aerodynamic couplings.


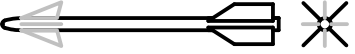
A canard-controlled missile is desired, since tail controlled systems have unstable zero dynamics (non-minimum phase systems) for lateral and transverse acceleration. This property complicates the problem and creates a challenge for some of nonlinear autopilot design approaches that are beyond the purpose of this thesis.

Four control surfaces and four lifting surfaces (Table 3.1) that are aligned in-line (Table 3.2) are desired since four panel configurations have symmetric aerodynamics [18].

**Table 3.1 Aerodynamic Surface Arrangements**







2 Surface	3 Surface	4 Surface	6 Surface	8 Surface
				

**Table 3.2 Aerodynamic Surface Alignments**

In-Line Alignment	
Interdigitated Alignment	

For the cruciform aerodynamic surface arrangements, aerodynamic surface orientation can be a plus or cross orientation. It is desired to orient the aerodynamic surfaces in a cross configuration since they have statically stable roll characteristics. Also aerodynamic effectiveness in pitching and yawing planes is higher than plus orientations for the same aerodynamic surface area. For plus and cross configurations control forces needed to maneuver a missile achieving steady trimmed flight are given in Table 3.3.


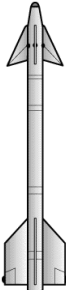
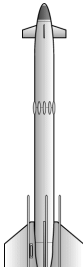
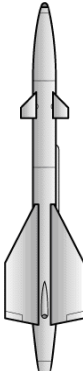
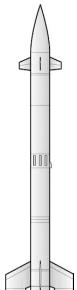
**Table 3.3 Plus and Cross Orientation Control Forces**

	Pitch Maneuver (Nose Up)	Yaw Maneuver (Nose Left)	Roll Maneuver (CCW )
Plus Orientation (Canard C.)			
Cross Orientation (Canard C.)			

**3.1.2 Missile Baseline Configuration Alternatives**

For the desired properties defined, short range agile missile systems are searched to establish a baseline configuration. Potential baseline missile configurations that seem reasonable to perform the analysis of control systems are given in Table 3.4.

**Table 3.4 Alternative Missiles [19]**

				
<b>CM</b>	USA AIM 9D	Russia SA-15 Gauntlet	Russia AA-3 Anab	Russia SA - 8 Gecko

As a baseline configuration, AIM 9D missile is selected since its known characteristics are as desired.



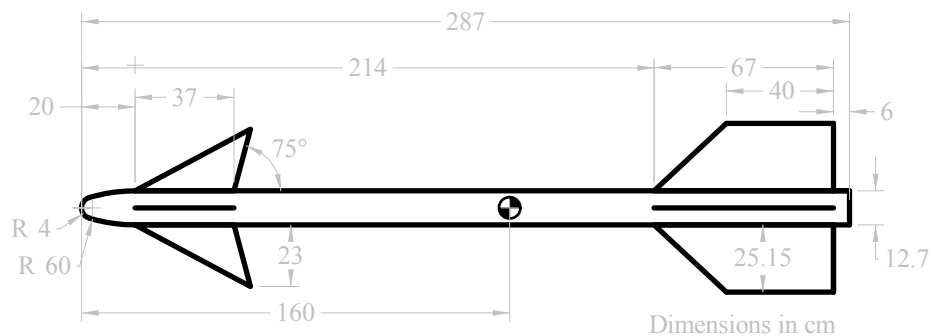
**Figure 3.1 Baseline Configuration AIM 9D (Modeled in SolidWorks)**



### 3.1.3 Varying Missile Configurations

In this section various configurations from the baseline configuration is obtained such that the aerodynamic characteristics are different from each other. Aerodynamic differences on configurations are considered as an uncertainty for the control system of baseline configuration. Then control systems are augmented using the adaptive augmentation strategy explained in Section 2.2.

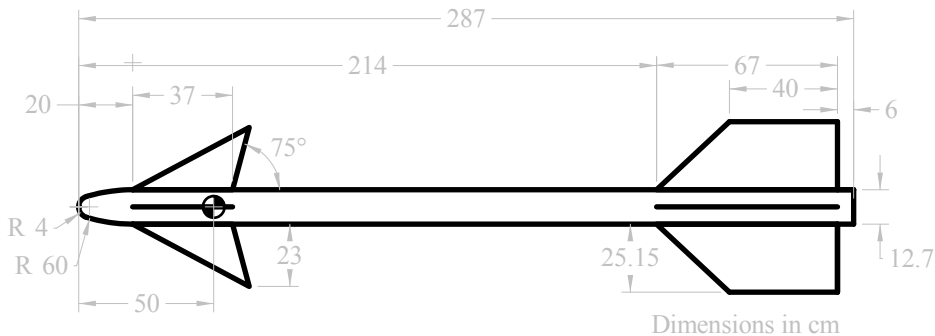
Baseline configuration called “C1” is obtained from AIM 9D. Approximate dimensions are derived from distributed specifications and digitizing images from the internet as presented with Figure 3.2.



**Figure 3.2 Configuration C1 – Baseline Configuration**

Center of gravity location is assigned to a location along the longitudinal axis considering the desired stability characteristics. For this purpose, center of pressure locations are estimated using the tool explained in section 3.2 which automates missile DATCOM. Then, a proper center of gravity location is selected accordingly such that the desired stability characteristics are met.

Next configuration called “C2” is obtained by varying center of gravity location of the baseline configuration as in Figure 3.3. Purpose of such a variation is to simulate the effects of burning propellant and change in subsystem locations or weights on the control system.



**Figure 3.3 Configuration C2 – Variation 1**

Third configuration called “C3” is obtained by changing the shape of fixed aerodynamic surfaces as given in Figure 3.4. For a missile system, such a variation often occurs during the development phase. Also it is possible to have such an effect with aerodynamic surfaces extracting from the missile body during the flight. It is also possible see applications that have wings with changing sweep angle during the flight which would also affect the control system in a similar manner. In order to keep the change in maneuverability of configuration C3 minimal, wing loading is kept constant. Then, dimensions for the new tail surface are found with the following approach.

Wing loading formula for C1 and C3 is,

$$WL_{C1} = \frac{W}{S_{1,tail}} = \frac{2 \cdot W}{(c_{1,tip} + c_{1,root}) \cdot b_1} \quad (3.1)$$

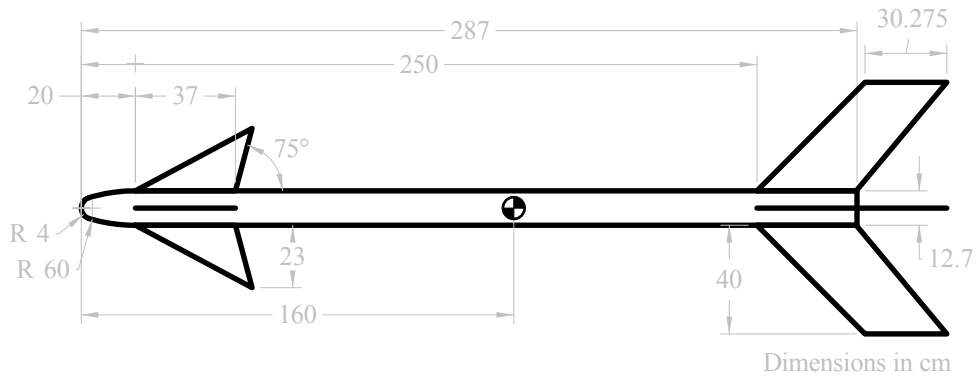
$$WL_{C3} = \frac{W}{S_{3,tail}} = \frac{2 \cdot W}{(c_{3,tip} + c_{3,root}) \cdot b_3} \quad (3.2)$$

Then, equating (3.1) to (3.2),

$$c_{3,tip} = \frac{(c_{1,tip} + c_{1,root}) \cdot b_1}{b_3} - c_{3,root} \quad (3.3)$$

Finally, C3 tail dimensions are found as,

$$\begin{aligned} c_{1,tip} &= 40 \text{ cm} & b_3 &= 40 \text{ cm} \\ c_{1,root} &= 67 \text{ cm} & c_{3,root} &= 37 \text{ cm} & \Rightarrow & c_{3,tip} \cong 30.275 \text{ cm} \\ b_1 &= 25.15 \text{ cm} \end{aligned}$$



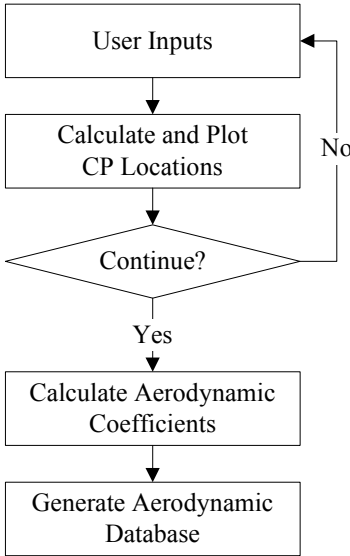
**Figure 3.4 Configuration C3 – Variation 2**

### 3.2 Missile Aerodynamics

To simulate and establish a mathematical model of a missile, aerodynamic forces and moments have to be determined.

Missile DATCOM is used for aerodynamic analysis considering the ease of use and availability. User manual [20] gives detailed instructions about the configuration of the program.

To create the aerodynamic database, an automated database generation tool is developed in MATLAB which configures the input file, run executable and read output file included in missile DATCOM. The aerodynamics database generation process with this tool is given with Figure 3.5.



**Figure 3.5 Flow Diagram for Aerodynamic Database Generation**

Necessary aerodynamic parameters to simulate missile motion are selected and output from Missile DATCOM. In order to have a small size database, aerodynamic

coefficients' dependence on some parameters as presented in Table 3.5 is assumed to be small and neglected. Also, missile DATCOM accepts actual control surface deflections  $\delta_1, \delta_2, \delta_3$  and  $\delta_4$  as input. A convenient way to express the aerodynamic coefficients dependence is using virtual control deflections  $\delta_a, \delta_e, \delta_r$  defined on rolling, pitching and yawing axes respectively.

**Table 3.5 Aerodynamic Database Coefficients**

Coeff.	Dependence						Description
	M	$\alpha$	$\beta$	$\delta_a$	$\delta_e$	$\delta_r$	
$C_X$	•	•	•		•	•	Axial Force Coefficient
$C_Y$	•		•			•	Side Force Coefficient
$C_Z$	•	•			•		Normal Force Coefficient
$C_L$	•	•	•	•			Rolling Moment Coefficient
$C_M$	•	•			•		Pitching Moment Coefficient
$C_N$	•		•			•	Yawing Moment Coefficient
$C_{Lp}$	•						Rolling moment coefficient derivative with roll rate
$C_{Mq}$	•						Pitching moment coefficient derivative with pitch rate
$C_{Nr}$	•						Yawing moment coefficient derivative with yaw rate
$C_{M\dot{\alpha}}$	•						Pitching moment derivative with rate of change of angle of attack
$C_{N\dot{\beta}}$	•						Yawing moment derivative with rate of change of side slip

Virtual control surface deflection mapping explained in [21] is applied as in (3.4). The same mapping is also used in order to convert the autopilot commands calculated as virtual deflection commands to actual CAS deflection commands.

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} \quad (3.4)$$

Also inverse mapping which is given in (3.5) is used to convert real CAS deflections to virtual deflections in simulation.

$$\begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & -0.25 & -0.25 \\ 0.25 & -0.25 & -0.25 & 0.25 \end{bmatrix} \cdot \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} \quad (3.5)$$

Aerodynamic forces and moments are determined from database coefficients in Table 3.5 with (3.6).

$$\mathbf{F}_{\text{Aero},B} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Q \cdot S_{ref} \cdot \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix}$$

$$\mathbf{M}_{\text{Aero},B} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = Q \cdot S_{ref} \cdot l_{ref} \cdot \left( \begin{bmatrix} C_L \\ C_M \\ C_N \end{bmatrix} + \frac{l_{ref}}{2 \cdot V_{MA}} \cdot \begin{bmatrix} p \cdot C_{Lp} \\ q \cdot C_{Mq} \\ r \cdot C_{Nr} \end{bmatrix} + \frac{l_{ref}}{2 \cdot V_{MA}} \cdot \begin{bmatrix} 0 \\ \dot{\alpha} \cdot C_{M\dot{\alpha}} \\ \dot{\beta} \cdot C_{N\dot{\beta}} \end{bmatrix} \right) \quad (3.6)$$

Aerodynamic coefficients in (3.6) are functions of  $M, \alpha, \beta, \delta_a, \delta_e, \delta_r$  (Table 3.5).

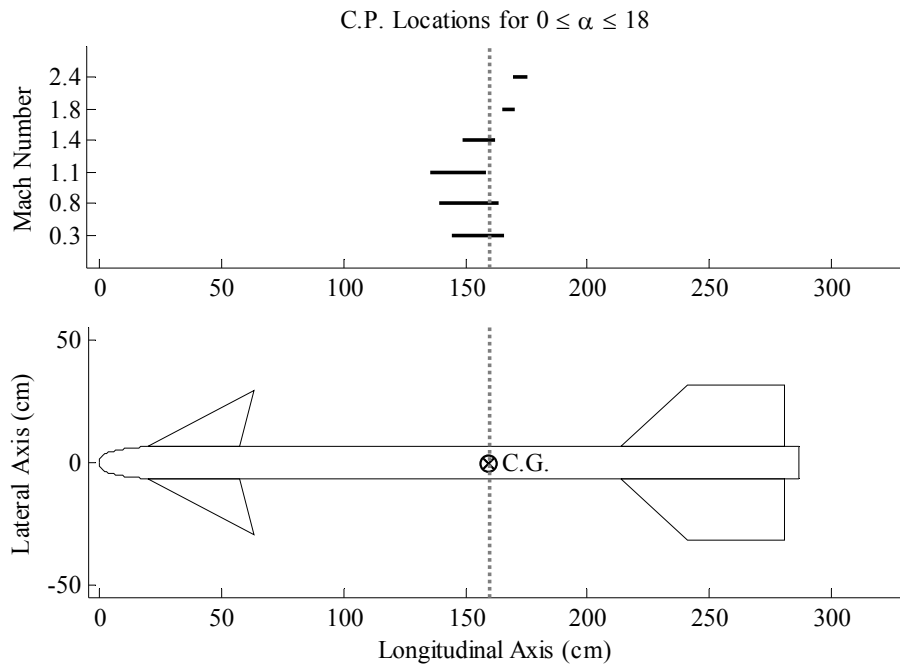
Aerodynamic database is generated for discrete values of these parameters and 1D, 2D, 3D or 4D linear interpolation algorithms are implemented in order to calculate aerodynamics of the missile in between.

### 3.2.1 Aerodynamics of Configurations C1, C2 and C3

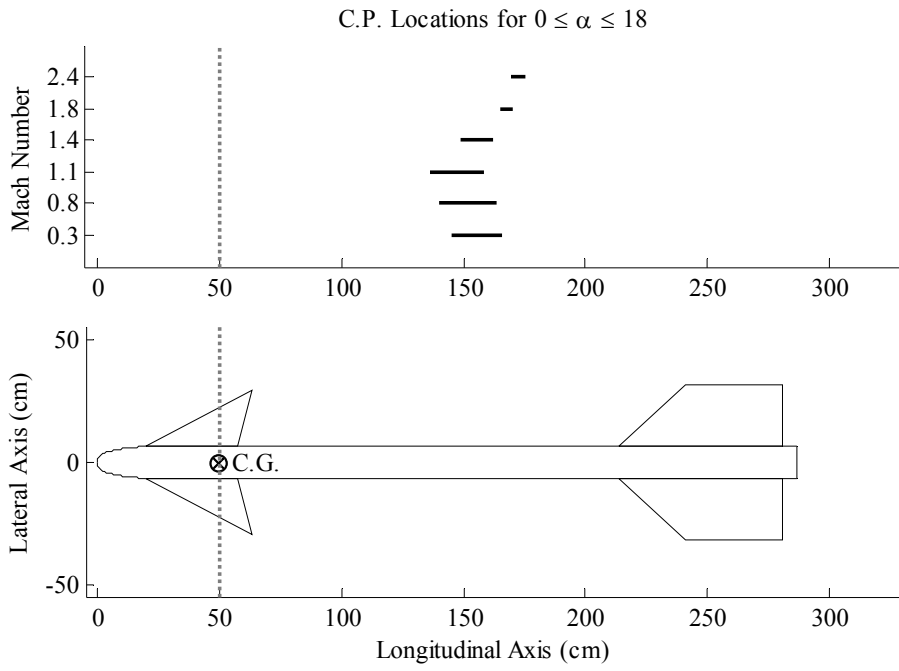
Aerodynamics of C1, C2 and C3 are calculated as discussed in the preceding section. Roll aerodynamics for C2 and C3 are assumed to be the same with C1 since problem defined in this thesis mainly affect pitch and yaw dynamics.

Also due to the symmetry, missile aerodynamics in pitch and yaw planes are found to be the same. For this reason only the pitch aerodynamics is presented in this part since the yaw aerodynamics is also the same in magnitude.

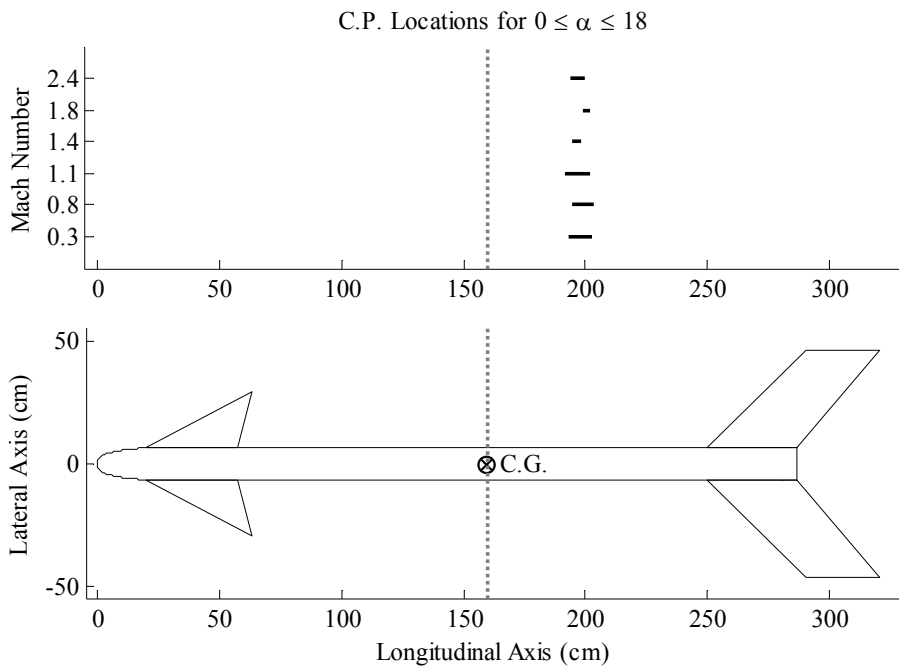
Change of center of pressure for the whole Mach interval and an angle of attack interval is given in Figure 3.2, Figure 3.3 and Figure 3.4. This is shown for fixed control surfaces ( $\delta_i = 0^\circ$ ,  $i = 1, \dots, 4$ ).



**Figure 3.6 Configuration C1 Center of Pressure Locations**



**Figure 3.7 Configuration C2 Center of Pressure Locations**



**Figure 3.8 Configuration C3 Center of Pressure Locations**



Previous figures show that C1 is “nearly” neutrally stable whereas C2 and C3 are stable while the controls surfaces are fixed. These stability characteristics also observed from Figure 3.9. Since the geometry for C1 and C2 are the same, normal force coefficients  $C_Z$  coincide in Figure 3.9.

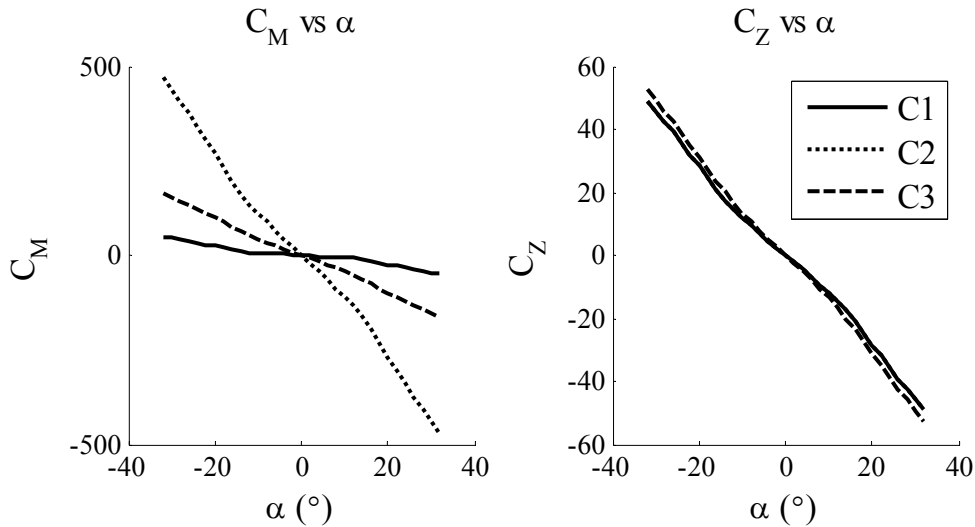
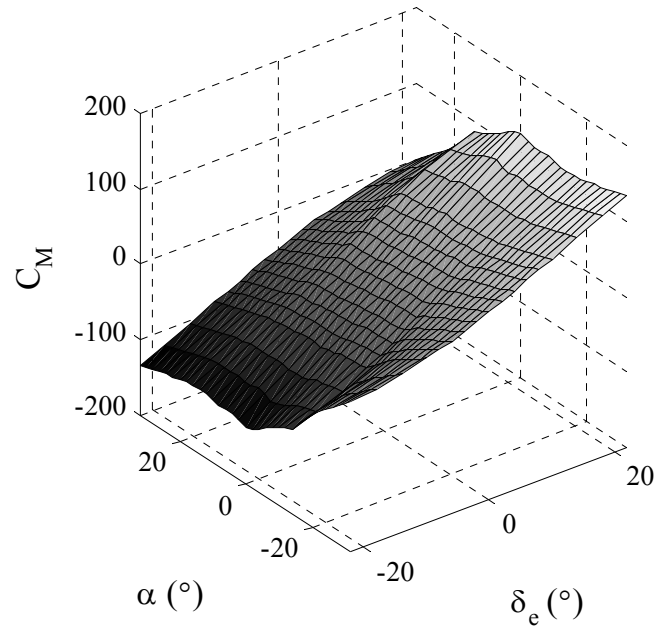
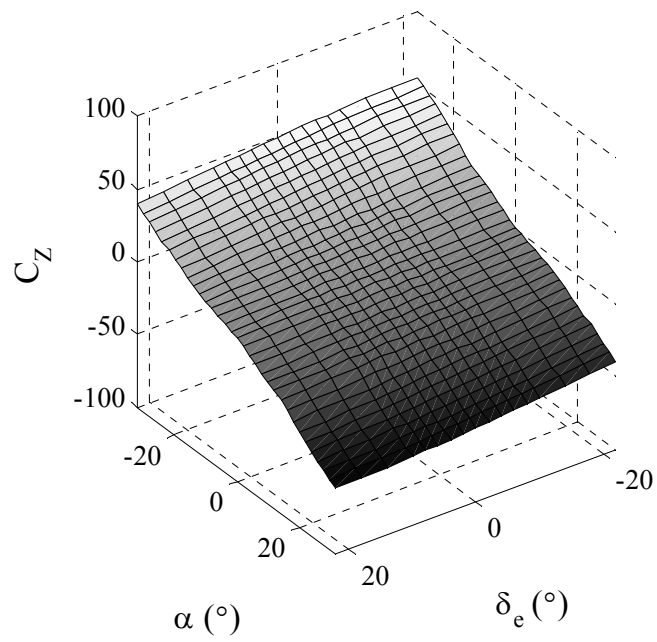


Figure 3.9  $C_M$  and  $C_Z$  versus Alpha at 1.8 Mach for  $0^\circ$  Delta

$C_M$  and  $C_Z$  are given in 3 dimensions (see Table 3.5 for dependencies) in Figure 3.10 and Figure 3.11 for C1.



**Figure 3.10  $C_M$  Coefficient at Mach 1.8 for C1**



**Figure 3.11  $C_Z$  Coefficient at Mach 1.8 for C1**

### 3.2.2 Non-Linear Parameters in Missile Aerodynamics

Aerodynamic non-linearity is one of the issues that make linearization process necessary. Nonlinearity may occur on any dimension of aerodynamic coefficient data depending on the missile configuration. To illustrate, in Figure 3.9,  $C_M$  and  $C_Z$  coefficients of C1 is nonlinear for varying alpha as shown below,

$$C_{M\alpha}(|\alpha| \leq 15^\circ) \neq C_{M\alpha}(|\alpha| \geq 15^\circ)$$

Similarly, aerodynamic coefficients are nonlinear with Mach number variation as explained in Figure 3.12. Normal force coefficients for C1 and C2 again coincide since the external geometry is the same.

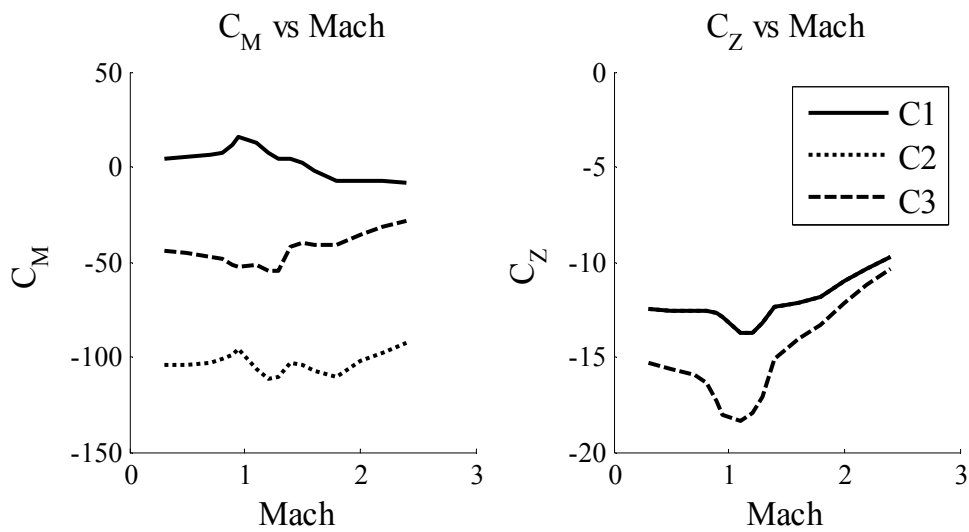


Figure 3.12  $C_M$  and  $C_N$  versus Mach at  $10^\circ$  Alpha

### 3.3 Missile Dynamics

In order to derive the governing equations of motion for a missile, Newton's laws of motion are used as given in reference [17].

Following is assumed while deriving equations of motion for a missile.

1. Constant mass is assumed.
2. Origin of  $F_B$  is at center of gravity, CG of the missile.
3. Rigid body is assumed.
4. Missile is symmetric in  $X_B Y_B$  and  $X_B Z_B$  planes.

For all vectors, subscript defines the reference frame in which the components are given.

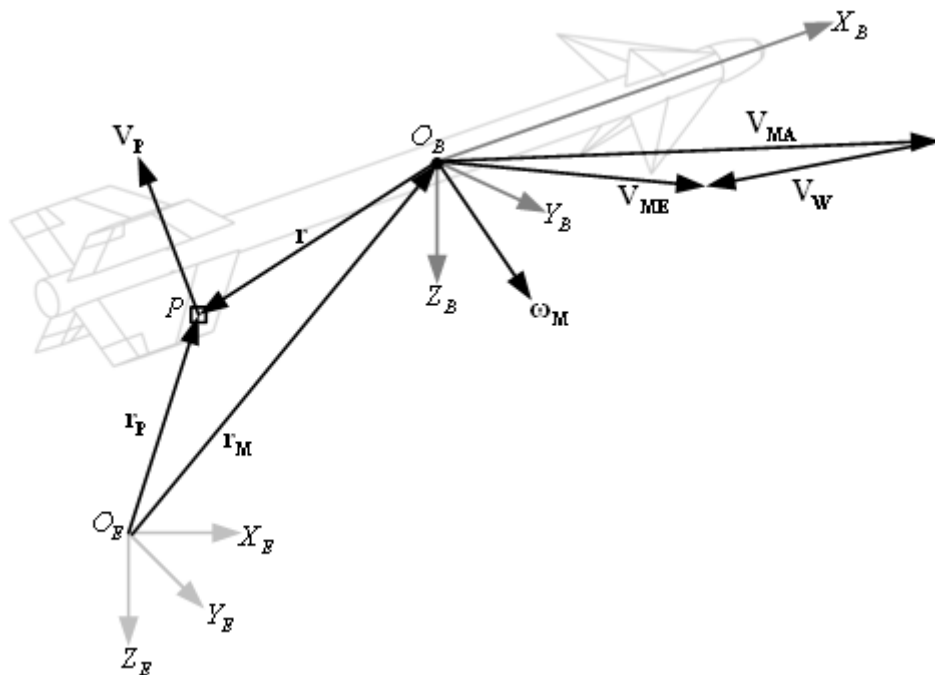


Figure 3.13 Axes and Vectors Used to Derive the Equations of Motion.

Translational equations of motion are derived for a missile starting from particles and applying Newton's second and third law with the assumptions above.

Newton's second law is applied to a particle in inertial space as,

$$d\mathbf{F}_E = \frac{d}{dt}(d\mathbf{L}_E) \quad (3.7)$$

Linear momentum of a particle is,

$$d\mathbf{L}_E = \mathbf{V}_{P,E} \cdot dm \quad (3.8)$$

Then linear momentum of the missile is,

$$\int d\mathbf{L}_E = \int \mathbf{V}_{P,E} dm \quad (3.9)$$

In order to formulate velocity of a particle P in terms of velocities defined,

$$\mathbf{r}_{P,E} = \mathbf{r}_{M,E} + \mathbf{r}_E \quad (3.10)$$

Taking the time derivative of the above equation gives,

$$\dot{\mathbf{r}}_{P,E} = \dot{\mathbf{r}}_{M,E} + \dot{\mathbf{r}}_E \quad (3.11)$$

Noting that in (3.11)  $\dot{\mathbf{r}}_{M,E} = \mathbf{V}_{ME,E}$  and  $\dot{\mathbf{r}}_{P,E} = \mathbf{V}_{P,E}$ ,

$$\mathbf{V}_{P,E} = \mathbf{V}_{ME,E} + \dot{\mathbf{r}}_E \quad (3.12)$$

Then the linear momentum expression in (3.9) is,

$$\int d\mathbf{L}_E = \mathbf{V}_{ME,E} \cdot \int dm + \int \dot{\mathbf{r}}_E dm \quad (3.13)$$

$\int \dot{\mathbf{r}}_E dm = 0$  in (3.13) since second assumption states that  $O_B$  is the mass center,

$$\int d\mathbf{L}_E = \int \mathbf{V}_{P,E} dm = \mathbf{V}_{ME,E} \cdot m \quad (3.14)$$

The integration of (3.7) gives the Newton's second law applied to the missile as,

$$\mathbf{F}_E = \frac{d}{dt}(\mathbf{V}_{ME,E} \cdot m) \quad (3.15)$$

Following remarks are valid for (3.15):

1. (3.15) is not valid when the first assumption does not hold.
2.  $\mathbf{F}_E$  is summation of external forces only since the internal forces that the particles apply to each other cancel each other obeying Newton's third law.
3. This equation is valid even though the third assumption does not hold.

In order to express (3.15) in  $F_B$ , coordinate system transformation has to be performed.

$$\mathbf{L}_{EB} \cdot \mathbf{F}_B = m \cdot \frac{d}{dt} (\mathbf{L}_{EB} \cdot \mathbf{V}_{ME,B}) \quad (3.16)$$

$$\mathbf{L}_{EB} \cdot \mathbf{F}_B = m \cdot (\dot{\mathbf{L}}_{EB} \cdot \mathbf{V}_{ME,B} + \mathbf{L}_{EB} \cdot \dot{\mathbf{V}}_{ME,B}) \quad (3.17)$$

Also derivative of a coordinate transformation matrix is given in (3.18) as;

$$\dot{\mathbf{L}}_{EB} = \mathbf{L}_{EB} \cdot \tilde{\boldsymbol{\omega}}_{M,B} \quad (3.18)$$

Where,

$$\boldsymbol{\omega}_{M,B} = [p \quad q \quad r]^T \quad (3.19)$$

$$\tilde{\boldsymbol{\omega}}_{M,B} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (3.20)$$

Substituting (3.18) into (3.17),

$$\mathbf{L}_{EB} \cdot \mathbf{F}_B = m \cdot (\mathbf{L}_{EB} \cdot \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{V}_{ME,B} + \mathbf{L}_{EB} \cdot \dot{\mathbf{V}}_{ME,B}) \quad (3.21)$$

$$\mathbf{F}_B = m \cdot (\dot{\mathbf{V}}_{ME,B} + \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{V}_{ME,B}) \quad (3.22)$$

Rearranging terms in (3.22) give the translational equation of motion as,

$$\dot{\mathbf{V}}_{ME,B} = \frac{1}{m} \cdot (\mathbf{F}_B) - \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{V}_{ME,B} \quad (3.23)$$

In a case where wind exists, (3.26) has to be used with (3.23),

$$\mathbf{V}_{ME,E} = \mathbf{V}_{MA,E} + \mathbf{V}_{W,E} \quad (3.24)$$

$$\mathbf{L}_{EB} \cdot \mathbf{V}_{ME,B} = \mathbf{L}_{EB} \cdot \mathbf{V}_{MA,B} + \mathbf{V}_{W,E} \quad (3.25)$$

$$\mathbf{V}_{ME,B} = \mathbf{V}_{MA,B} + \mathbf{L}_{EB}^{-1} \cdot \mathbf{V}_{W,E} \quad (3.26)$$

Note that transformation matrices satisfy the ‘‘orthogonality condition’’ and the following formulas are valid for any transformation matrices.

$$\mathbf{L}_{EB}^T \cdot \mathbf{L}_{EB} = \mathbf{I} \quad (3.27)$$

$$\mathbf{L}_{EB}^T = \mathbf{L}_{EB}^{-1} = \mathbf{L}_{BE} \quad (3.28)$$

Hence (3.26) can also be written as,

$$\mathbf{V}_{ME,B} = \mathbf{V}_{MA,B} + \mathbf{L}_{EB}^T \cdot \mathbf{V}_{W,E} \quad (3.29)$$

Where,

$$\mathbf{V}_{MA,B} = [u \quad v \quad w]^T \quad (3.30)$$

$$\mathbf{V}_{ME,B} = [u_E \quad v_E \quad w_E]^T \quad (3.31)$$

$$\mathbf{V}_{W,E} = [W_X \quad W_Y \quad W_Z]^T \quad (3.32)$$

Rotational dynamics of a missile is also derived in a similar manner to translational dynamics. Moment of forces acting on a particle is integrated for the whole missile.

Moment of momentum of  $dm$  with respect to  $O_M$  is,

$$d\mathbf{H}_E = \tilde{\mathbf{r}}_E \cdot \mathbf{V}_{P,E} \cdot dm \quad (3.33)$$

Taking the time derivative,

$$\frac{d}{dt}(d\mathbf{H}_E) = \dot{\tilde{\mathbf{r}}}_E \cdot \mathbf{V}_{P,E} \cdot dm + \tilde{\mathbf{r}}_E \cdot \dot{\mathbf{V}}_{P,E} \cdot dm \quad (3.34)$$

Also from (3.12),

$$\dot{\tilde{\mathbf{r}}}_E = \tilde{\mathbf{V}}_{P,E} - \tilde{\mathbf{V}}_{ME,E} \quad (3.35)$$

Also moment of  $d\mathbf{F}_E$  acting on the particle with respect to  $O_M$  is,

$$d\mathbf{M}_E = \tilde{\mathbf{r}}_E \cdot d\mathbf{F}_E \quad (3.36)$$

Substituting (3.7) and (3.8) in (3.36),

$$d\mathbf{M}_E = \tilde{\mathbf{r}}_E \cdot \dot{\mathbf{V}}_{P,E} \cdot dm \quad (3.37)$$

Then using (3.35) and (3.37) in (3.34) and arranging terms,

$$d\mathbf{M}_E = \frac{d}{dt}(d\mathbf{H}_E) - (\tilde{\mathbf{V}}_{P,E} - \tilde{\mathbf{V}}_{ME,E}) \cdot \mathbf{V}_{P,E} \cdot dm \quad (3.38)$$

Since  $-\tilde{\mathbf{V}}_{P,E} \cdot \mathbf{V}_{P,E} = 0$ , (3.38) reduces to,

$$d\mathbf{M}_E = \frac{d}{dt}(d\mathbf{H}_E) + \tilde{\mathbf{V}}_{ME,E} \cdot \mathbf{V}_{P,E} \cdot dm \quad (3.39)$$

Integration of (3.39) give,

$$\int d\mathbf{M}_E = \frac{d}{dt}(\int d\mathbf{H}_E) + \tilde{\mathbf{V}}_{ME,E} \cdot \int \mathbf{V}_{P,E} dm \quad (3.40)$$

Substituting (3.14) for the last integral in (3.40),

$$\mathbf{M}_E = \frac{d}{dt}(\mathbf{H}_E) + \tilde{\mathbf{V}}_{ME,E} \cdot \mathbf{V}_{ME,E} \cdot m \quad (3.41)$$

Noting that  $\tilde{\mathbf{V}}_{ME,E} \cdot \mathbf{V}_{ME,E} = 0$ , (3.41) reduces to moment equation defining the rotational dynamics of a missile,

$$\mathbf{M}_E = \frac{d}{dt}(\mathbf{H}_E) \quad (3.42)$$

Following remarks are valid for (3.42):

1. (3.42) is not valid when the first assumption does not hold.
2.  $\mathbf{M}_E$  is summation of external moments only since the moments of internal forces cancel each other obeying Newton's third law.
3. This equation is valid even though the third assumption does not hold.

In order to express the moment equation (3.42) governing the rotational dynamics in  $F_B$ , coordinate system transformation has to be performed.

$$\mathbf{L}_{EB} \cdot \mathbf{M}_B = \frac{d}{dt}(\mathbf{L}_{EB} \cdot \mathbf{H}_B) \quad (3.43)$$

Taking the time derivative of the right hand side,

$$\frac{d}{dt}(\mathbf{L}_{EB} \cdot \mathbf{H}_B) = \dot{\mathbf{L}}_{EB} \cdot \mathbf{H}_B + \mathbf{L}_{EB} \cdot \dot{\mathbf{H}}_B \quad (3.44)$$

Substituting (3.18) in (3.44) and dividing by  $\mathbf{L}_{EB}$ ,

$$\mathbf{M}_B = \dot{\mathbf{H}}_B + \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{H}_B \quad (3.45)$$

In order to calculate the angular momentum of the missile with respect CG in  $F_B$ , angular momentum of particles in it have to be integrated for the whole missile as,

$$\mathbf{H}_E = \int \tilde{\mathbf{r}}_E^B \cdot \mathbf{V}_{P,E} dm \quad (3.46)$$

In (3.10), position of the particle with respect to CG is written in  $F_B$  as,

$$\mathbf{r}_{P,E} = \mathbf{r}_{M,E} + \mathbf{L}_{EB} \cdot \mathbf{r}_B \quad (3.47)$$

Time derivative of (3.47) gives,

$$\dot{\mathbf{r}}_{P,E} = \dot{\mathbf{r}}_{M,E} + \dot{\mathbf{L}}_{EB} \cdot \mathbf{r}_B + \mathbf{L}_{EB} \cdot \dot{\mathbf{r}}_B \quad (3.48)$$



Note that  $\dot{\mathbf{r}}_{M,E} = V_{ME,E}$  and for a rigid missile  $\dot{\mathbf{r}}_B = 0$ . Also substituting (3.18) in (3.48),

$$\mathbf{V}_{P,E} = \mathbf{L}_{EB} \cdot \mathbf{V}_{ME,B} + \mathbf{L}_{EB} \cdot \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{r}_B \quad (3.49)$$

Then (3.46) is written relative to  $F_M$  using (3.49) as,

$$\mathbf{H}_B = \mathbf{L}_{BE} \cdot \mathbf{H}_E = \int \mathbf{L}_{BE} \cdot \tilde{\mathbf{r}}_E \cdot (\mathbf{L}_{EB} \cdot \mathbf{V}_{ME,B} + \mathbf{L}_{EB} \cdot \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{r}_B) dm \quad (3.50)$$

Using the rule following rule of matrix transformation derived in [17]

$$\tilde{\mathbf{r}}_B = \mathbf{L}_{BE} \cdot \tilde{\mathbf{r}}_E \cdot \mathbf{L}_{EB} \quad (3.51)$$

Rearranging terms in (3.50) yields,

$$\mathbf{H}_B = \left( \int \tilde{\mathbf{r}}_B dm \right) \cdot \mathbf{V}_{ME,B} + \int \tilde{\mathbf{r}}_B \cdot \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{r}_B dm \quad (3.52)$$

The first integral in (3.52) is zero since 1<sup>st</sup> assumption holds.

When the last integral in (3.52) is expanded the following is found,

$$\mathbf{H}_B = \mathbf{I}_B \cdot \boldsymbol{\omega}_{M,B} \quad (3.53)$$

Where,

$$\mathbf{I}_B = \begin{bmatrix} I_{XX} & -I_{XY} & -I_{XZ} \\ -I_{YX} & I_{YY} & -I_{YZ} \\ -I_{ZX} & -I_{ZY} & I_{ZZ} \end{bmatrix} \quad (3.54)$$

$$\begin{aligned} I_{XX} &= \int (y^2 + z^2) dm & I_{XY} &= I_{YX} = \int (x \cdot y) dm \\ I_{YY} &= \int (x^2 + z^2) dm & I_{XZ} &= I_{ZX} = \int (x \cdot z) dm \\ I_{ZZ} &= \int (x^2 + y^2) dm & I_{YZ} &= I_{ZY} = \int (y \cdot z) dm \end{aligned} \quad (3.55)$$

Noting that missile is symmetric relative to  $X_B Y_B$  and  $X_B Z_B$ , off-diagonal terms of the inertia tensor, so called products of inertia are,

$$I_{XY} = I_{XZ} = I_{YZ} = 0 \quad (3.56)$$

Because of the above condition, body-fixed coordinate axes are also the principal axes of rotation. In other words, for angular velocity of a missile in any of the body-fixed axis, the angular momentum vector is also on that axis.

Substituting the angular momentum expression given in (3.53) into rotational dynamics equation (3.45),

$$\mathbf{M}_B = \frac{d}{dt}(\mathbf{I}_B \cdot \boldsymbol{\omega}_{M,B}) + \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{I}_B \cdot \boldsymbol{\omega}_{M,B} \quad (3.57)$$

Carrying time derivation on,

$$\mathbf{M}_B = \dot{\mathbf{I}}_B \cdot \boldsymbol{\omega}_{M,B} + \mathbf{I}_B \cdot \dot{\boldsymbol{\omega}}_{M,B} + \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{I}_B \cdot \boldsymbol{\omega}_{M,B} \quad (3.58)$$

Since third assumption yields  $\dot{\mathbf{I}}_B = 0$ , (3.58) reduces to,

$$\mathbf{M}_B = \mathbf{I}_B \cdot \dot{\boldsymbol{\omega}}_{M,B} + \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{I}_B \cdot \boldsymbol{\omega}_{M,B} \quad (3.59)$$

Expanding (3.59) and arranging the terms if  $\mathbf{I}_B$  is non-singular, give rotational equation of motion as,

$$\dot{\boldsymbol{\omega}}_{M,B} = (\mathbf{I}_B)^{-1} \cdot (\mathbf{M}_B - \tilde{\boldsymbol{\omega}}_{M,B} \cdot \mathbf{I}_B \cdot \boldsymbol{\omega}_{M,B}) \quad (3.60)$$

Total forces and moment used in translational and rotational equations are defined as follows.

$$\mathbf{F}_B = \begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix} = \mathbf{F}_{\text{Aero},B} + \mathbf{F}_{\text{Thrust},B} + \mathbf{F}_{\text{Gravity},B} \quad (3.61)$$

Where,

$$\mathbf{F}_{\text{Aero},B} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \mathbf{F}_{\text{Thrust},B} = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} = \mathbf{L}_{\text{BF}} \cdot \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{F}_{\text{Gravity},B} = \mathbf{L}_{\text{EB}}^T \cdot \begin{bmatrix} 0 \\ 0 \\ m \cdot g \end{bmatrix}$$

As stated before  $\mathbf{M}_B$  term in (3.60) is the sum of all external moments and is given as (3.62).

$$\mathbf{M}_B = \begin{bmatrix} M_X \\ M_Y \\ M_Z \end{bmatrix} = \mathbf{M}_{\text{Aero},B} + \mathbf{M}_{\text{Thrust},B} \quad (3.62)$$

Where

$$\mathbf{M}_{\text{Aero},B} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} \quad \mathbf{M}_{\text{Thrust},B} = \tilde{\mathbf{r}}_{\text{N},B} \cdot \mathbf{F}_{\text{Thrust},B} \quad \mathbf{r}_{\text{N},B} = \begin{bmatrix} -x_N \\ 0 \\ 0 \end{bmatrix}$$

### 3.4 Thrust Model

A jet engine like varying thrust engine model is developed to maintain the missile velocity as constant as possible during the analysis in simulation. For this purpose following assumptions are made.

1. Mass is constant, as always in this thesis,
2. Wind velocity is negligibly small compared to the missile air velocity and  $V_{MA} \cong V_{ME}$
3. Air velocity vector magnitude is approximately equal to its component on  $O_B X_B$  axis ( $\alpha \cong 0, \beta \cong 0$ ) and  $u \cong V_{MA}$ ,
4. Thrust misalignment is negligible and thrust is aligned with  $O_B X_B$  axis,
5. Gravity vector component on  $O_B X_B$  axis is negligible compared to the missile axial acceleration.

Translational dynamics of a missile is given in Section 3.3. Considering the given assumptions, translational equation along  $O_B X_B$  axis reduces to (3.63).

$$\sum F_{X,B} \cong T + X \cong m \cdot \frac{dV_{MA}}{dt} \quad (3.63)$$

Variable thrust engine model for missile velocity control is implemented using reduced equations above.

While designing a velocity hold control algorithm, following assumptions are made,

1. Drag force changes linearly with speed,
2. It's assumed that engine is commanded with thrust input.

Architecture used to control the missile velocity is given in Figure 3.14.

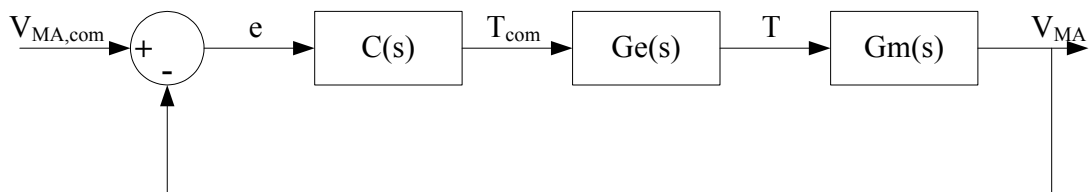


Figure 3.14 Variable Thrust Engine Model

In order to determine missile transfer function  $G_m$  defining translational dynamics, drag force is formulated in terms of velocity.

$$X = K_X \cdot V_{MA} \quad (3.64)$$

Substituting (3.64) into (3.63),

$$T + K_X \cdot V_{MA} \cong m \cdot \frac{dV_{MA}}{dt} \quad (3.65)$$

Taking the Laplace transformation of (3.65),

$$T(s) + K_X \cdot V_{MA}(s) \cong m \cdot V_{MA}(s) \cdot s \quad (3.66)$$

Then  $G_m$  is found to be as,

$$G_m(s) = \frac{V_{MA}}{T} = \frac{1}{m \cdot s - K_X} \quad (3.67)$$

To find engine transfer function  $G_e$ , time constant of the engine from idle to full throttle is defined as follows,

$$\tau_e = 0.5 \text{ s}$$

First order model for the engine is given in (3.68),

$$G_e(s) = \frac{T}{T_{com}} = \frac{w_{n,e}}{s + w_{n,e}} \quad (3.68)$$

Where,

$$w_{n,e} = \frac{1}{\tau_e} = 2 \text{ s}^{-1}$$

### 3.5 Control Actuation System (CAS) Model

For CAS model, non-linear model included in *MATLAB Simulink Aerospace Blockset* is referred. This model is based on a regular second order actuator as presented in (3.69) plus the following extensions,

- Position integral saturation modeling mechanical limit,
- Actuator input saturation in terms of velocity, modeling the limited input (voltage for electro-mechanical systems, oil and gas pressure for hydraulic and pneumatic systems).

$$\frac{\delta}{\delta_C} = \frac{\omega_{CAS}^2}{s^2 + 2 \cdot \zeta_{CAS} \cdot \omega_{CAS} \cdot s + \omega_{CAS}^2} \quad (3.69)$$

Response of this model for 20° fin deflection command for different velocity limits is presented in Figure 3.15.

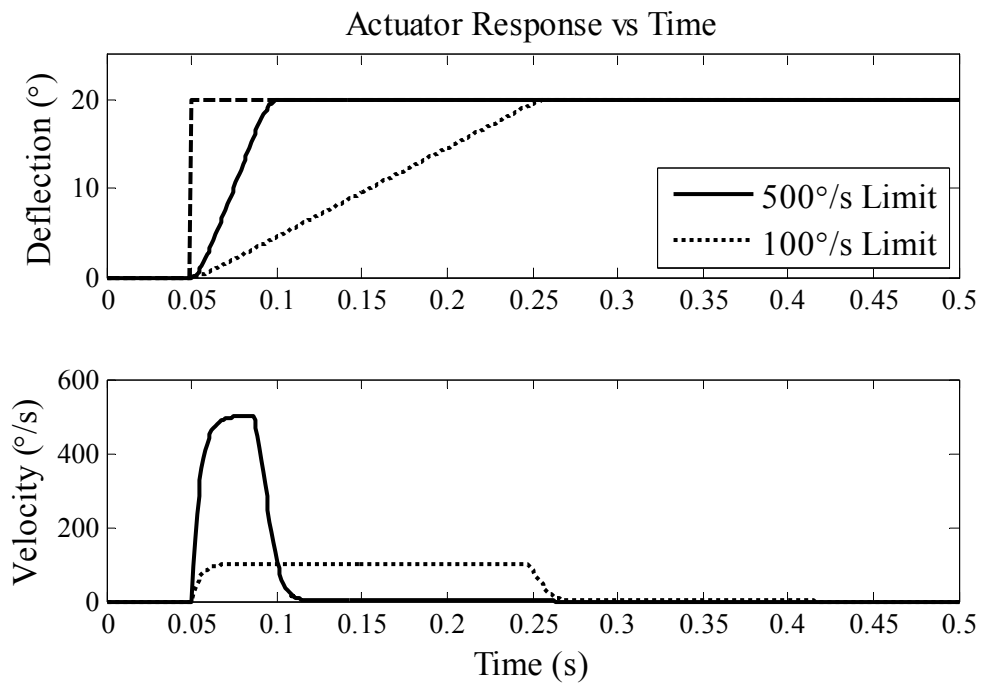


Figure 3.15 Non-linear Actuator Model Response to Step Input

In the simulation, following CAS performance parameters are used.

$$\begin{aligned} \omega_{CAS} &= 30 \text{ Hz} & \delta_{MAX} &= 22^\circ \\ \zeta_{CAS} &= 0.8 & \dot{\delta}_{MAX} &= 500^\circ/s \end{aligned}$$

# CHAPTER 4

## AUTOPILOT DESIGN

### 4.1 Control Requirements

Closed loop system performance requirements have to be defined before designing control systems. Requirements are defined at different design points in time domain considering the limits of aerodynamics and CAS dynamics

Some of the time domain transient-response specifications in Figure 4.1 are used to define control requirements [4].

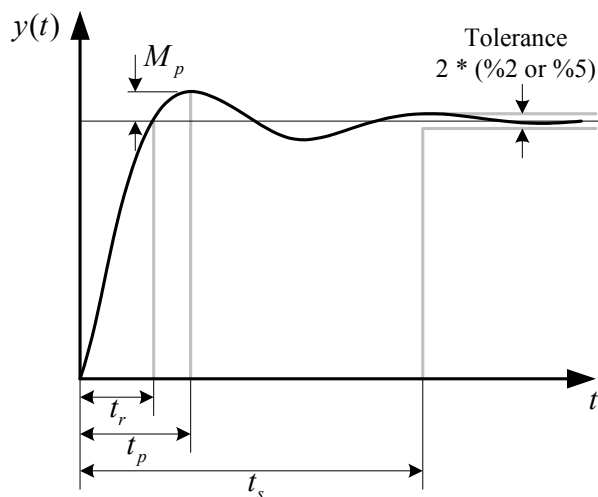


Figure 4.1 Definitions of Transient-Response Specifications

Requirements are defined considering the open loop missile characteristics and CAS characteristics. Using the definitions above performance requirements are found as in Table 4.1. A similar approach is also proposed in [9].

**Table 4.1 Transient Response Performance Requirements**

Mach	Pitch & Yaw Channel		Roll Channel	
	$t_r$ (s)	M.P. (%)	$t_r$ (s)	M.P. (%)
0.3	$\leq 1.50$	$\leq 20$	$\leq 0.40$	$\leq 20$
0.8	$\leq 1.21$	$\leq 20$	$\leq 0.33$	$\leq 20$
1.1	$\leq 1.04$	$\leq 20$	$\leq 0.29$	$\leq 20$
1.4	$\leq 0.87$	$\leq 20$	$\leq 0.24$	$\leq 20$
1.8	$\leq 0.64$	$\leq 20$	$\leq 0.19$	$\leq 20$
2.4	$\leq 0.30$	$\leq 20$	$\leq 0.10$	$\leq 20$

For any design condition in between Mach 0.3 and 2.4, performance requirements are assumed to change in a linear manner and can be interpolated with Mach number.

## 4.2 Linear Missile Models

Linear missile model is derived for pitch, yaw and roll channel respectively using the non-linear equations of motion and Euler angle rates given in section 3.3, and 2.4.1 respectively. While deriving the linear models, CAS dynamics are neglected. Effects of CAS on missile dynamics are compensated by re-arranging the closed loop pole locations if missile response is not as desired.

Equations of motion in all three axes are given as below,

$$\begin{aligned} F_X &= m \cdot \dot{u}_E + m \cdot (q \cdot w_E - r \cdot v_E) \\ F_Y &= m \cdot \dot{v}_E + m \cdot (r \cdot u_E - p \cdot w_E) \\ F_Z &= m \cdot \dot{w}_E + m \cdot (p \cdot v_E - q \cdot u_E) \end{aligned} \quad (4.1)$$

$$\begin{aligned} M_X &= I_{XX} \cdot \dot{p} - (I_{YY} - I_{ZZ}) \cdot q \cdot r - I_{YZ} \cdot (q^2 - r^2) - I_{ZX} \cdot (\dot{r} + p \cdot q) - I_{XY} \cdot (\dot{q} - r \cdot p) \\ M_Y &= I_{YY} \cdot \dot{q} - (I_{ZZ} - I_{XX}) \cdot r \cdot p - I_{ZX} \cdot (r^2 - p^2) - I_{XY} \cdot (\dot{p} + q \cdot r) - I_{YZ} \cdot (\dot{r} - p \cdot q) \\ M_Z &= I_{ZZ} \cdot \dot{r} - (I_{XX} - I_{YY}) \cdot p \cdot q - I_{XY} \cdot (p^2 - q^2) - I_{YZ} \cdot (\dot{q} + r \cdot p) - I_{ZX} \cdot (\dot{p} - q \cdot r) \end{aligned} \quad (4.2)$$

For the case that thrust vector is along the  $O_B X_B$  axis ( $T_Y = T_Z = 0$ ), forces and moments in (4.1) and (4.2) are,

$$\begin{aligned} F_X &= X + T_X - m \cdot g \cdot \sin \theta & M_X &= L \\ F_Y &= Y + m \cdot g \cdot \cos \theta \cdot \sin \phi & M_Y &= M \\ F_Z &= Z + m \cdot g \cdot \cos \theta \cdot \cos \phi & M_Z &= N \end{aligned}$$

The Euler angle rates are,

$$\begin{aligned} \dot{\phi} &= p + q \cdot \sin \phi \cdot \tan \theta + r \sin \phi \cdot \tan \theta \\ \dot{\theta} &= q \cdot \cos \phi - r \cdot \sin \phi \\ \dot{\psi} &= q \cdot \sin \phi \cdot \sec \theta + r \cdot \cos \phi \cdot \sec \theta \end{aligned} \quad (4.3)$$

In order to derive a linear model of missile dynamics, assumption called ‘‘short period approximation’’ is used (see [22] and [23]). This approach is commonly used in missile autopilot design process and simply states that rate of change of missile velocity is negligible compared to rates of change of velocities along other axes ( $\dot{u} \cong 0$ ). With this approximation, simple linear models as explained in the following sections are derived to define short period missile dynamics to be used in autopilot design.



### 4.2.1 Linear Missile Model for Pitching Motion

Equations defining 3 degree of freedom pitch plane motion is extracted from (4.1), (4.2) and (4.3) with the assumptions and conditions below,

1. Thrust vector is along  $O_B X_B$  axis,
2. For only pitching motion,  $v = p = r = Y = L = N = \phi = \psi = 0$
3. Atmosphere is stationary,  $\mathbf{V}_{ME} = \mathbf{V}_{MA}$

$$Z + m \cdot g \cdot \cos \theta = m \cdot \dot{w} - m \cdot (q \cdot u) \quad (4.4)$$

$$M = I_{YY} \cdot \dot{q} \quad (4.5)$$

$$\dot{\theta} = q \quad (4.6)$$

Also as a convention, pitch dynamics are based on  $\alpha$  instead of  $w$  using the following definitions for small angles and short period approximation above,

$$\alpha = \tan \frac{w}{u_0} \cong \frac{w}{V_{MA}} \quad (4.7)$$

$$\dot{\alpha} = \tan \frac{\dot{w}}{u_0} \cong \frac{\dot{w}}{V_{MA}} \quad (4.8)$$

Since the missile is an air to air missile, it has to have higher acceleration capability compared to target whose lateral acceleration is limited 10-15 g's considering the blackout problem of the pilot. In order to provide a precision hit capability missile lateral acceleration limit is about 2-3 times higher than the targets', i.e. around 30-50 g's. Then non-linear gravitational force term  $m \cdot g \cdot \cos \theta$  in (4.4) is neglected compared to the  $Z$  aerodynamic force in body z-axis.

Lastly before formulating the linear model, aerodynamic forces and moments have to be expressed with a linear combination of dominant database parameters.

$$Z = Q \cdot S_{ref} \cdot C_{Z\alpha} \cdot \alpha + Q \cdot S_{ref} \cdot C_{Z\delta_E} \cdot \delta_E \quad (4.9)$$

$$M = Q \cdot S_{ref} \cdot l_{ref} \cdot C_{M\alpha} \cdot \alpha + \frac{Q \cdot S_{ref} \cdot l_{ref}^2 \cdot C_{Mq}}{2 \cdot V_{MA}} \cdot q + Q \cdot S_{ref} \cdot l_{ref} \cdot C_{M\delta_E} \cdot \delta_E \quad (4.10)$$

Re-arranging with preceding equations through (4.4) to (4.10), remaining linear equations defining missile pitching motion are given as follows.

$$\dot{\alpha} = \frac{Q \cdot S_{ref} \cdot C_{Z\alpha}}{m \cdot V_{MA}} \cdot \alpha + q + \frac{Q \cdot S_{ref} \cdot C_{Z\delta_E}}{m \cdot V_{MA}} \cdot \delta_E \quad (4.11)$$

$$\dot{q} = \frac{Q \cdot S_{ref} \cdot l_{ref} \cdot C_{M\alpha}}{I_{YY}} \cdot \alpha + \frac{Q \cdot S_{ref} \cdot l_{ref}^2 \cdot C_{Mq}}{I_{YY} \cdot 2 \cdot V_{MA}} \cdot q + \frac{Q \cdot S_{ref} \cdot l_{ref} \cdot C_{M\delta_E}}{I_{YY}} \cdot \delta_E \quad (4.12)$$

For simplicity,

$$Z_\alpha \triangleq \frac{Q \cdot S_{ref} \cdot C_{Z\alpha}}{m \cdot V_{MA}} \quad Z_{\delta_E} \triangleq \frac{Q \cdot S_{ref} \cdot C_{Z\delta_E}}{m \cdot V_{MA}}$$

$$M_\alpha \triangleq \frac{Q \cdot S_{ref} \cdot l_{ref} \cdot C_{M\alpha}}{I_{YY}} \quad M_q \triangleq \frac{Q \cdot S_{ref} \cdot l_{ref}^2 \cdot C_{Mq}}{I_{YY} \cdot 2 \cdot V_{MA}} \quad M_{\delta_E} \triangleq \frac{Q \cdot S_{ref} \cdot l_{ref} \cdot C_{M\delta_E}}{I_{YY}}$$

Lateral acceleration needed as an output for autopilot design is defined as,

$$a_z \cong \frac{Z}{m} = V_{MA} \cdot Z_\alpha \cdot \alpha + V_{MA} \cdot Z_{\delta_E} \cdot \delta_E \quad (4.13)$$

Equations (4.11) and (4.12) are the state equations and given in a matrix form as,

$$\underbrace{\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} Z_\alpha & 1 \\ M_\alpha & M_q \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \alpha \\ q \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} Z_{\delta_E} \\ M_{\delta_E} \end{bmatrix}}_{\mathbf{B}} \underbrace{\delta_{ECOM}}_{\mathbf{u}} \quad (4.14)$$

$$\underbrace{\begin{bmatrix} a_z \\ \alpha \\ q \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} V_{MA} \cdot Z_\alpha & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \alpha \\ q \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} V_{MA} \cdot Z_{\delta_E} \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{D}} \underbrace{\delta_{ECOM}}_{\mathbf{u}} \quad (4.15)$$

#### 4.2.2 Linear Missile Model for Yawing Motion

Equations defining 3 degree of freedom yaw plane motion are extracted in a similar manner to pitching equations using assumptions and conditions below,

1. Thrust vector is along the  $O_B X_B$  axis,  $T_Y = T_Z = 0$
2. For only yawing motion,  $w = p = q = Z = L = M = \theta = \phi = 0$
3. Atmosphere is stationary,  $\mathbf{V}_{ME} = \mathbf{V}_{MA}$

$$\begin{aligned} Y &= m \cdot \dot{v} + m \cdot (r \cdot u) \\ N &= I_{ZZ} \cdot \dot{r} \\ \dot{\psi} &= r \end{aligned} \quad (4.16)$$

If same methodology is applied as in pitch plane, state space linear yaw model is found as,

$$\underbrace{\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} Y_{\beta} & -1 \\ N_{\beta} & N_r \end{bmatrix}}_A \underbrace{\begin{bmatrix} \beta \\ r \end{bmatrix}}_x + \underbrace{\begin{bmatrix} Y_{\delta_R} \\ N_{\delta_R} \end{bmatrix}}_B \underbrace{\delta_{RCOM}}_u \quad (4.17)$$

And the output equation is,

$$\underbrace{\begin{bmatrix} a_y \\ \beta \\ r \end{bmatrix}}_y = \underbrace{\begin{bmatrix} V_{MA} \cdot Y_{\beta} & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_C \underbrace{\begin{bmatrix} \beta \\ r \end{bmatrix}}_x + \underbrace{\begin{bmatrix} V_{MA} \cdot Y_{\delta_R} \\ 0 \\ 0 \end{bmatrix}}_D \underbrace{\delta_{RCOM}}_u \quad (4.18)$$

Where,

$$\begin{aligned} Y_{\beta} &\triangleq \frac{Q \cdot S_{ref} \cdot C_{Y\beta}}{m \cdot V_{MA}} & Y_{\delta_R} &\triangleq \frac{Q \cdot S_{ref} \cdot C_{Y\delta_R}}{m \cdot V_{MA}} \\ N_{\beta} &\triangleq \frac{Q \cdot S_{ref} \cdot l_{ref} \cdot C_{N\beta}}{I_{ZZ}} & N_r &\triangleq \frac{Q \cdot S_{ref} \cdot l_{ref}^2 \cdot C_{Nr}}{I_{ZZ} \cdot 2 \cdot V_{MA}} & N_{\delta_R} &\triangleq \frac{Q \cdot S_{ref} \cdot l_{ref} \cdot C_{N\delta_R}}{I_{ZZ}} \end{aligned}$$

### 4.2.3 Linear Missile Model for Rolling Motion

Equations of motion defining rolling motion are found with a similar approach as in pitching and yawing planes.

1. For only rolling motion,  $v = w = q = r = Y = Z = M = N = \theta = \psi = 0$
2. Atmosphere is stationary,  $\mathbf{V}_{ME} = \mathbf{V}_{MA}$

$$\begin{aligned} L &= I_{XX} \cdot \dot{p} \\ \dot{\phi} &= p \end{aligned} \quad (4.19)$$

For the roll model, rolling moment has to be expressed in a linear composition of state parameters as,

$$L = \frac{Q \cdot S_{ref} \cdot l_{ref}^2 \cdot C_{Mp}}{2 \cdot V_{MA}} \cdot p + Q \cdot S_{ref} \cdot l_{ref} \cdot C_{L\delta_A} \cdot \delta_A \quad (4.20)$$

After some substitution, roll equations are found as,

$$\begin{aligned} \dot{\phi} &= p \\ \dot{p} &= \frac{Q \cdot S_{ref} \cdot l_{ref}^2 \cdot C_{Mp}}{I_{XX} \cdot 2 \cdot V_{MA}} \cdot p + \frac{Q \cdot S_{ref} \cdot l_{ref} \cdot C_{L\delta_A}}{I_{XX}} \cdot \delta_A \end{aligned} \quad (4.21)$$

For simplicity,

$$L_p \triangleq \frac{Q \cdot S_{ref} \cdot l_{ref}^2 \cdot C_{Mp}}{I_{XX} \cdot 2 \cdot V_{MA}} \quad L_{\delta_A} \triangleq \frac{Q \cdot S_{ref} \cdot l_{ref} \cdot C_{L\delta_A}}{I_{XX}}$$

Then the state and output equations are defined as,

$$\underbrace{\begin{bmatrix} \dot{\phi} \\ \dot{p} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & L_p \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \phi \\ p \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \\ L_{\delta_A} \end{bmatrix}}_{\mathbf{D}} \underbrace{\delta_{ACOM}}_{\mathbf{u}} \quad (4.22)$$

And the output equation is,

$$\underbrace{\begin{bmatrix} \phi \\ p \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \phi \\ p \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\mathbf{D}} \underbrace{\delta_{ACOM}}_{\mathbf{u}} \quad (4.23)$$

### 4.3 Missile Acceleration Response Characteristics

In order to examine the missile acceleration response characteristics, the open loop transfer function from elevator deflection to normal acceleration is found using (4.14) and (4.15) as in (4.24).

$$\frac{a_Z(s)}{\delta_E(s)} = V_{MA} \cdot \frac{Z_{\delta_E} \cdot s^2 + (M_{\delta_E} - M_q \cdot Z_{\delta_E}) \cdot s + (M_{\delta_E} \cdot Z_\alpha - M_\alpha \cdot Z_{\delta_E})}{s^2 - (M_q + Z_\alpha) \cdot s + (M_q \cdot Z_\alpha - M_\alpha)} \quad (4.24)$$

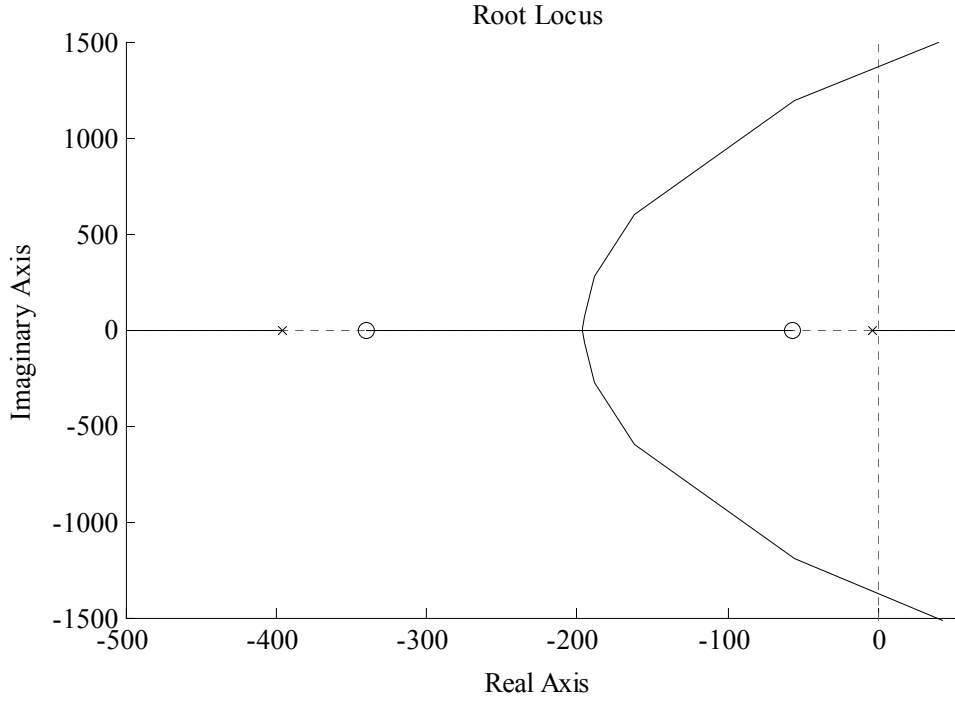


Figure 4.2 Configuration C1 Acceleration Transfer Function Root Locus

The acceleration transfer function of configuration C1 is found using the aerodynamic derivatives calculated at Mach number 1.8 and is given with (4.25).

$$\frac{a_Z(s)}{\delta_E(s)} = \frac{-206.4 \cdot s^2 + 8182s + 3977000}{s^2 + 400 \cdot s + 1677} \quad (4.25)$$

The root locus plot in Figure 4.2 shows that the open loop acceleration response to elevator deflection is minimum phase. Therefore, the Assumption 2.1 for boundedness of the augmented system is verified.

#### **4.4 Full-State Feedback Autopilot Design Using Pole Placement**

Full state feedback autopilot algorithms are designed using the data for configuration C1. Then effects of changing the configurations specified with C2 and C3 are analyzed using the autopilot designed for C1.

Nonlinearities in aerodynamics are handled with gain scheduling during the design. Gain scheduling stands for determining some design points for flight parameters and designing linear controllers at the design points. Between consecutive design points (if design points selected properly), changes in aerodynamics are assumed to be small and linear.

For systems like missiles, gain scheduling is necessary to compensate changes in dynamic pressure and schedule linear controllers to the conditions at that instant. The controllers in this thesis are scheduled for Mach number only. Normally scheduling is done for dynamic pressure or both Mach number and altitude.

For missile systems of this kind, lateral and transverse acceleration transfer functions, which can be easily derived from state space representations given in (4.14), (4.15), (4.17) and (4.18), are type 0 and proper transfer functions. On the other hand, bank angle transfer function that can be derived from (4.22) and (4.23) is type 1 and strictly proper. Using a suitable method presented in Section 2.1 autopilots are designed in a decoupled manner.

The pitch, yaw and roll autopilot architectures are presented in Figure 4.3, Figure 4.4 and Figure 4.5 respectively.

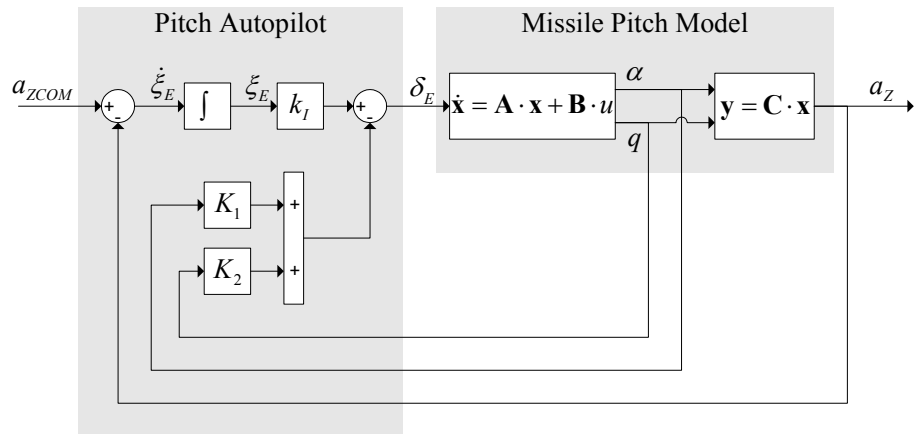


Figure 4.3 Pitch Autopilot Architecture with Pole Placement Approach

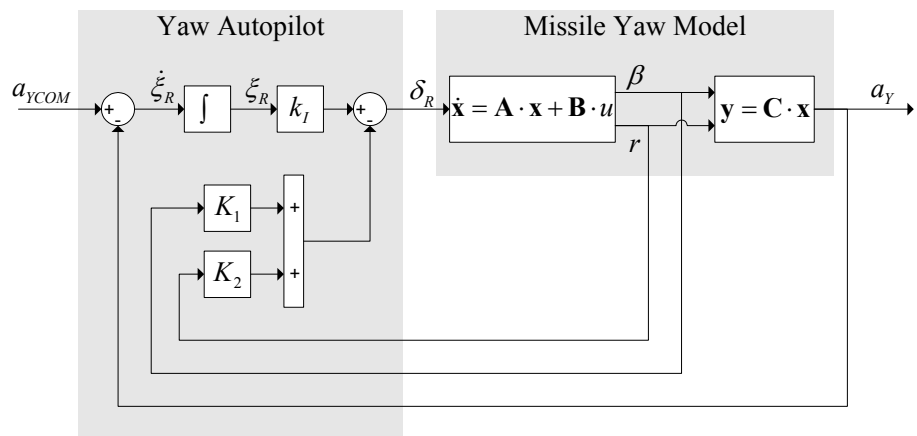


Figure 4.4 Yaw Autopilot Architecture with Pole Placement Approach

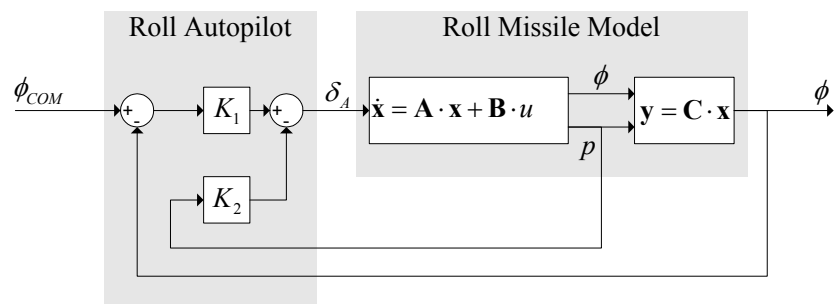


Figure 4.5 Roll Autopilot Architecture with Pole Placement Approach

Poles of closed loop systems are placed to proper locations using the approach explained in section 2.1. Dominant poles are placed to the desired locations on the s-plane. Remaining poles of the closed loop system are placed to other locations where interference with dominant dynamics is negligible. However as stated in section 4.2, CAS model is not included in linear missile models. Interferences to the missile dynamics due to CAS is compensated by re-arranging closed loop pole locations if necessary. Missile response with CAS dynamics for different pole locations is analyzed in simulation until the desired response is achieved.

Pitch, yaw and roll open-loop poles for all configurations are presented in Table 4.2 and Table 4.3.

**Table 4.2 Open Loop Poles for Pitch and Yaw Autopilots**

Open Loop Poles (Pitch & Yaw)						
	Configuration C1		Configuration C2		Configuration C3	
Mach	Pole 1	Pole 2	Pole 1	Pole 2	Pole 1	Pole 2
0.3	-0.6	-75.7	-1.1	-87.9	-1.1	-110.0
0.8	-1.4	-189.1	-3.0	-213.2	-3.3	-285.9
1.1	-2.0	-305.1	-4.4	-310.1	-4.7	-480.4
1.4	-2.7	-345.9	-5.6	-351.4	-5.3	-506.8
1.8	-4.2	-395.7	-7.8	-392.4	-6.4	-580.2
2.4	-5.2	-410.3	-9.6	-440.1	-7.1	-560.9

**Table 4.3 Open Loop Poles for Roll Autopilot**

Open Loop Poles (Roll)						
	Configuration C1		Configuration C2		Configuration C3	
Mach	Pole 1	Pole 2	Pole 1	Pole 2	Pole 1	Pole 2
0.3	0.0	-221.7	0.0	-221.7	0.0	-221.7
0.8	0.0	-524.9	0.0	-524.9	0.0	-524.9
1.1	0.0	-320.1	0.0	-320.1	0.0	-320.1
1.4	0.0	-868.9	0.0	-868.9	0.0	-868.9
1.8	0.0	-1595.0	0.0	-1595.0	0.0	-1595.0
2.4	0.0	-2419.0	0.0	-2419.0	0.0	-2419.0



The gains calculated using C1 aerodynamic models are presented in Table 4.4. Using the gains given in Table 4.4, the closed loop poles for pitch and yaw autopilots using linear models for C1, C2 and C3 are presented in Table 4.5 and Table 4.6.

**Table 4.4 Pitch, Yaw and Roll Autopilot Gains for Calculated for C1**

Mach	Feedback Gains (Pitch & Yaw)			Feedback Gains (Roll)	
	K1	K2	-K1	K1	K2
0.3	5.795	-0.752	0.271	-2.463	-0.042
0.8	1.178	-0.355	0.009	-0.466	0.023
1.1	0.727	-0.264	0.003	-0.271	-0.003
1.4	0.642	-0.215	0.002	-0.272	0.015
1.8	0.666	-0.178	0.003	-0.352	0.028
2.4	2.833	-0.100	0.015	-0.864	0.026

**Table 4.5 Closed Loop Poles for Pitch and Yaw Autopilots (using Gains for C1)**

Mach	Closed Loop Poles (Pitch & Yaw)								
	Configuration C1			Configuration C2			Configuration C3		
	Pole 1	Pole 2	Pole 3	Pole 1	Pole 2	Pole 3	Pole 1	Pole 2	Pole 3
0.3	-5.7 +1.8i	-5.7 -1.8i	-22.9	-1.0 +1.3i	-1.0 -1.3i	-71.9	-3.1 +3.0i	-3.1 -3.0i	-58.4
0.8	-7.0 +2.3i	-7.0 -2.3i	-28.3	-3.3	-0.8	-160	-5.5	-4	-108
1.1	-8.2 +2.7i	-8.2 -2.7i	-32.9	-5.5	-0.7	-207.4	-8.8	-2.5	-169.7
1.4	-9.8 +3.2i	-9.8 -3.2i	-39.4	-6.8	-0.9	-238.7	-8.9	-3.2	-185.2
1.8	-13.3 +4.3i	-13.3 -4.3i	-53.4	-9.4	-1.4	-268.8	-6.4 +0.9i	-6.4 -0.9i	-251.4
2.4	-28.6 +9.4i	-28.6 -9.4i	-114.5	-7.8 +4.8i	-7.8 -4.8i	-349.4	-12.9+11.6i	-12.9-11.6i	-308.4

**Table 4.6 Closed Loop Poles for Roll Autopilot (using Gains for C1)**

Mach	Closed Loop Poles (Roll)					
	Configuration C1		Configuration C2		Configuration C3	
	Pole 1	Pole 2	Pole 1	Pole 2	Pole 1	Pole 2
0.3	-10.7	-257.7	-10.7	-257.7	-10.7	-257.7
0.8	-13.0	-313.7	-13.0	-313.7	-13.0	-313.7
1.1	-15.0	-360.8	-15.0	-360.8	-15.0	-360.8
1.4	-17.6	-424.5	-17.6	-424.5	-17.6	-424.5
1.8	-23.1	-555.1	-23.1	-555.1	-23.1	-555.1
2.4	-42.9	-1031.0	-42.9	-1031.0	-42.9	-1031.0

For the nonlinear CAS model presented in section 3.5, closed loop poles for small signals for which the system behaves linearly is given in Table 4.7.

**Table 4.7 Linear CAS Model Closed Loop Poles**

Pole 1	Pole 2
-150.8 + 113.1i	-150.8 - 113.1i

For the pitch and yaw autopilots, CAS dynamics is at least 6 times faster than the closed loop missile dynamics. However for the roll autopilot, interference from CAS to missile dynamics is significant since the roll channel closed loop poles is at least 2 times faster. Response of the roll autopilots with the desired pole locations is found to be more oscillatory than expected due to the CAS. This oscillatory response is compensated by placing the closed loop poles of the roll autopilot on the real axis of s-plane as presented in Table 4.6 such that the desired performance requirements in Table 4.1 is satisfied.

Closed loop dynamics of the configurations presented in Table 4.5 and Table 4.6 show that the rise time of the missile configurations for pitch and yaw channels are increasing for C1, C3 and C2 respectively which will also be analyzed in the next chapter with simulation.

**4.5 Adaptive Augmentation of Existing Autopilots**

Existing pole placement control system is augmented with neural network in pitch and yaw channel according to the theory presented in section 2.2.

For varying configurations, response of the missile changes due to the varying aerodynamics which causes the existing autopilots designed for C1 do not produce the necessary control input. Neural network augmentation is used to produce the missing control input portion to satisfy the requirements defined with reference model.

For pitch and yaw autopilots augmentation architectures are given in Figure 4.6 and Figure 4.7.

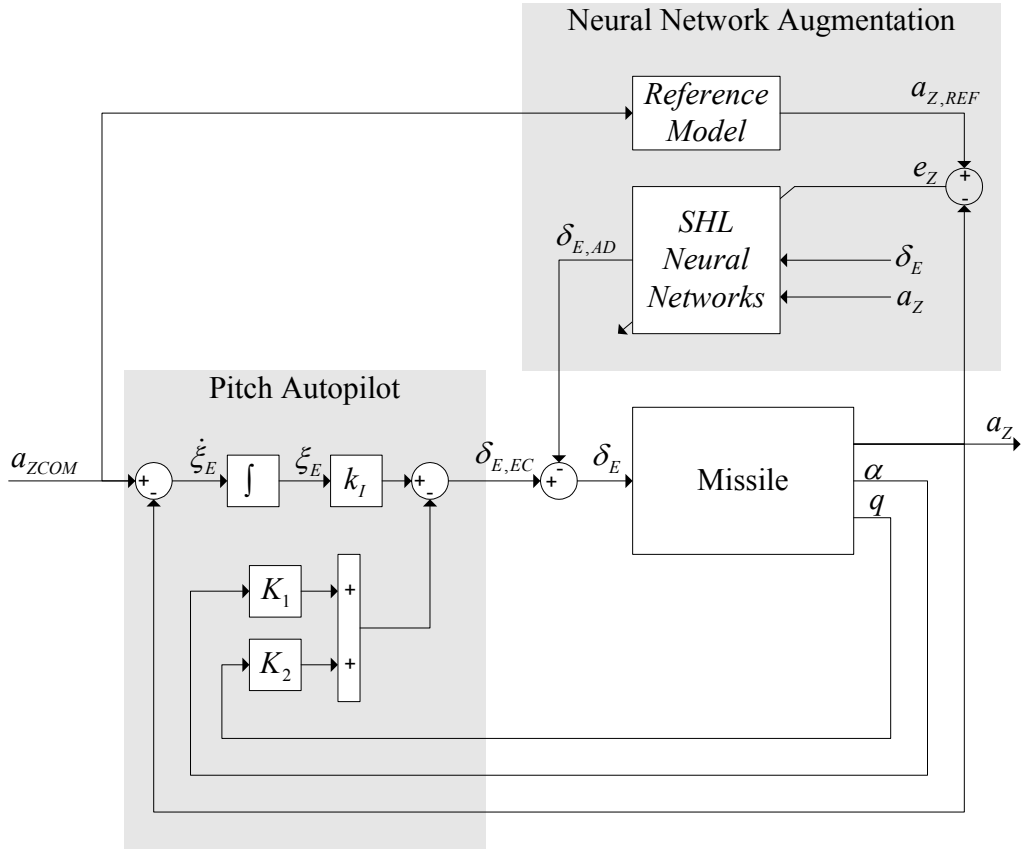
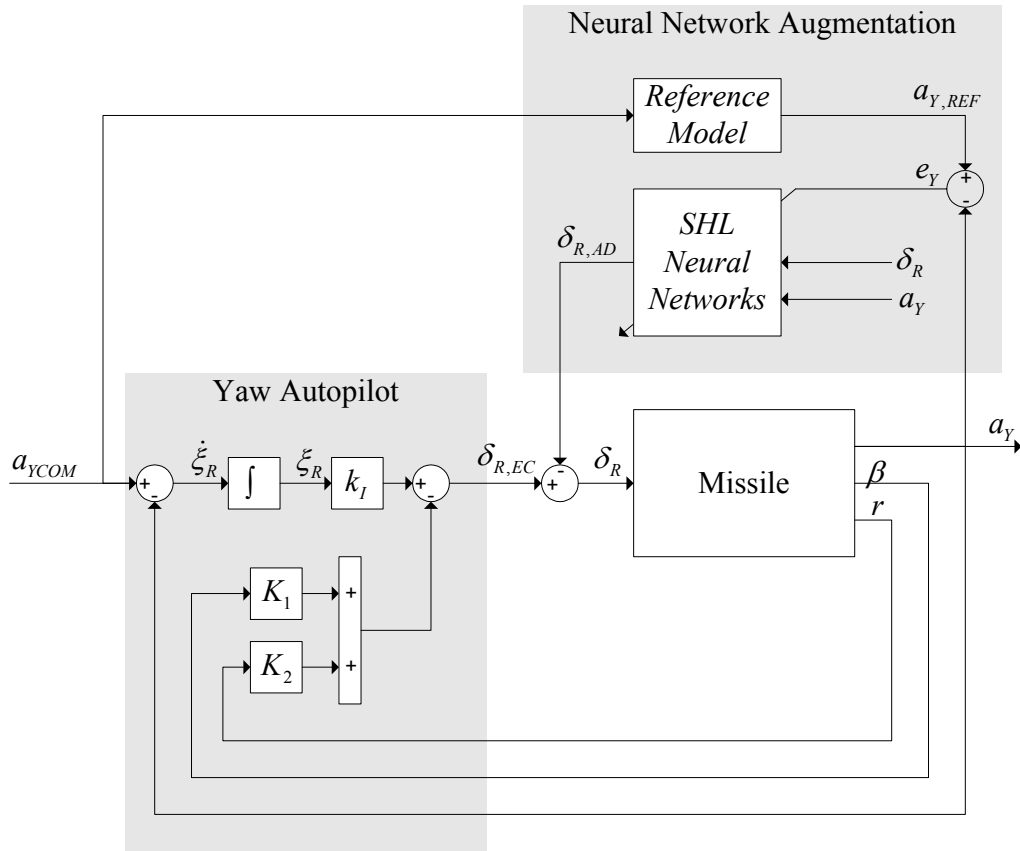


Figure 4.6 Architecture for Adaptive Augmentation of Pitch Autopilot



**Figure 4.7 Architecture for Adaptive Augmentation of Yaw Autopilot**

To find convenient values for the neural network parameters as number of neurons, input scaling, and learning rates simulation is used. Finding specific parameters that satisfy adequate performance for the augmented autopilots is not easy to achieve since the method includes trial and error design approach.

# CHAPTER 5

## RESULTS AND DISCUSSION

As explained in the previous chapter, the autopilots are designed using the base configuration C1. Then the same autopilots are also used for varying configurations C2 and C3 which have different pitch and yaw aerodynamics creating an uncertainty for the pitch and yaw autopilots. The decrease in performance is compensated for by augmenting existing autopilots with adaptive networks.

Five different time domain analysis case are set up to show the source of uncertainty for existing autopilots, the performance of existing autopilots and the performance improvement with augmented pitch and yaw autopilots as presented in Table 5.1.

The analysis results in cases 1, 3, 4 and 5 are only presented in pitch plane since the pitch and yaw aerodynamics, autopilot architectures and the desired performance are same.

In the first case, source of uncertainty on autopilots caused by varying configurations are explicitly shown and discussed using simulation results for the pitch autopilot. Then in the second case, performance of unaugmented autopilots for the base and varying configurations is analyzed considering the performance requirements defined in Table 4.1. Lastly, with the remaining three cases, performance of augmented pitch autopilot is presented for configurations C1, C2 and C3 respectively.

**Table 5.1 Description of Analyses Cases**

<b>Analysis</b>	<b>Config.</b>	<b>Description</b>
1	C1–C2–C3	Pitch plane open loop missile response analysis
2	C1–C2–C3	Pitch/Yaw/Roll autopilot time domain performance analysis
3	C1	Pitch autopilot performance analysis with/without NN
4	C2	Pitch autopilot performance analysis with/without NN
5	C3	Pitch autopilot performance analysis with/without NN

## 5.1 Analysis 1

Open loop missile dynamics are analyzed for the case defined in Table 5.2 to show why the variations on missile configuration degrade the performance of the controlled missile.

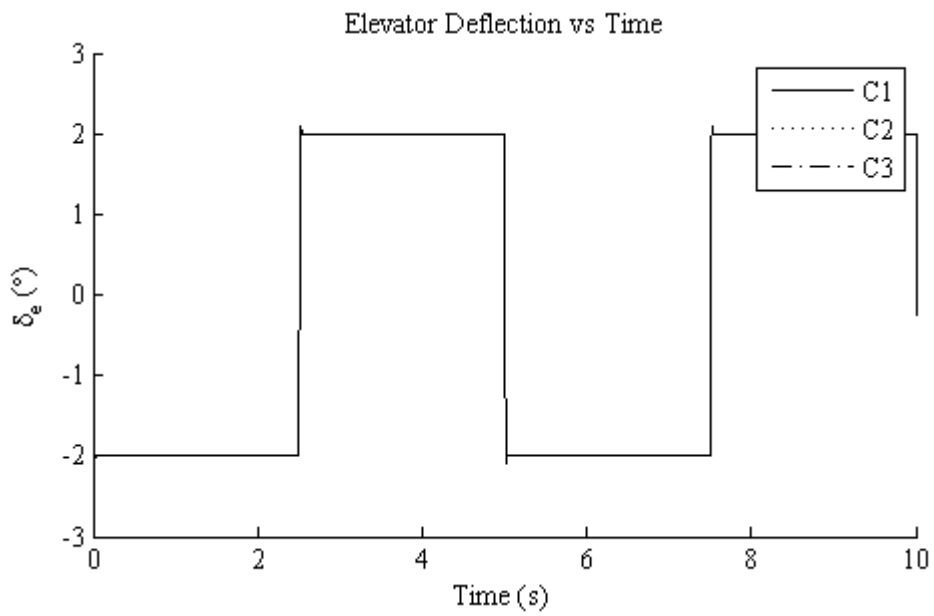
**Table 5.2 Parameters for Analysis 1**

<b>Property</b>	<b>Symbol</b>	<b>Value</b>
Missile Configuration		C1-C2-C3
Mach No	M	1.8

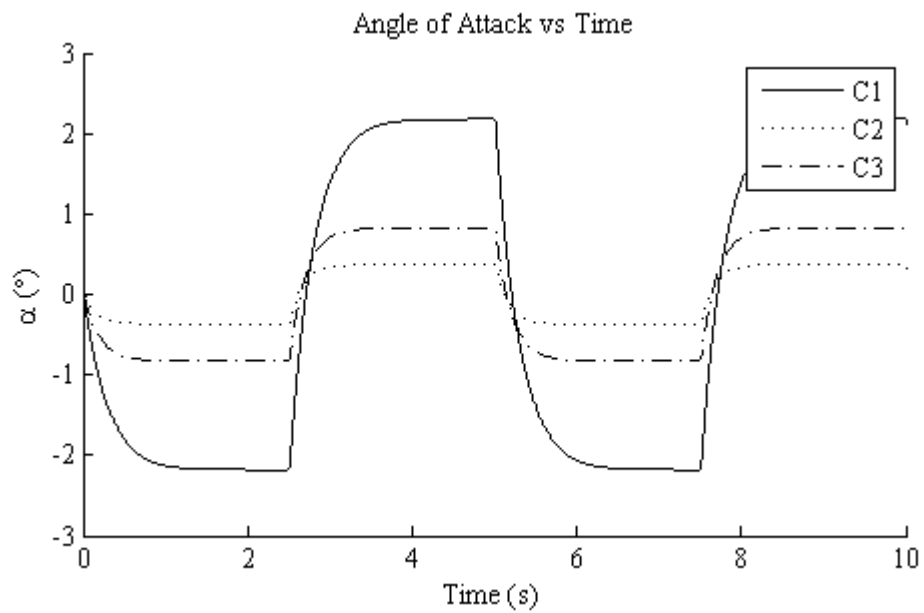
The cause of performance degradation for missile autopilots is the change in control effectiveness for various configurations. For varying configurations, missile aerodynamic moments are different causing change in control effectiveness. Control effectiveness is defined as the ratio of angle of attack to control input. For the same control input (control force), angle of attack at trim condition are different. This condition is so, since the body + wing aerodynamic moments with respect to center of gravity are different.

For all the configurations, 2 degrees elevator deflections as presented in Figure 5.1 are used. Then control effectiveness for configurations can be calculated using the angle of attack responses in Figure 5.2 for the commands in Figure 5.1.

Control effectiveness is decreasing as C1, C3 and C2 as observed in Figure 5.1, Figure 5.2 and Figure 5.3. Then physical interpretation is that the gains calculated for configuration C1 whose controls are more effective, cannot create the necessary control authority for configurations C2 and C3 resulting an increase in autopilot rise time.

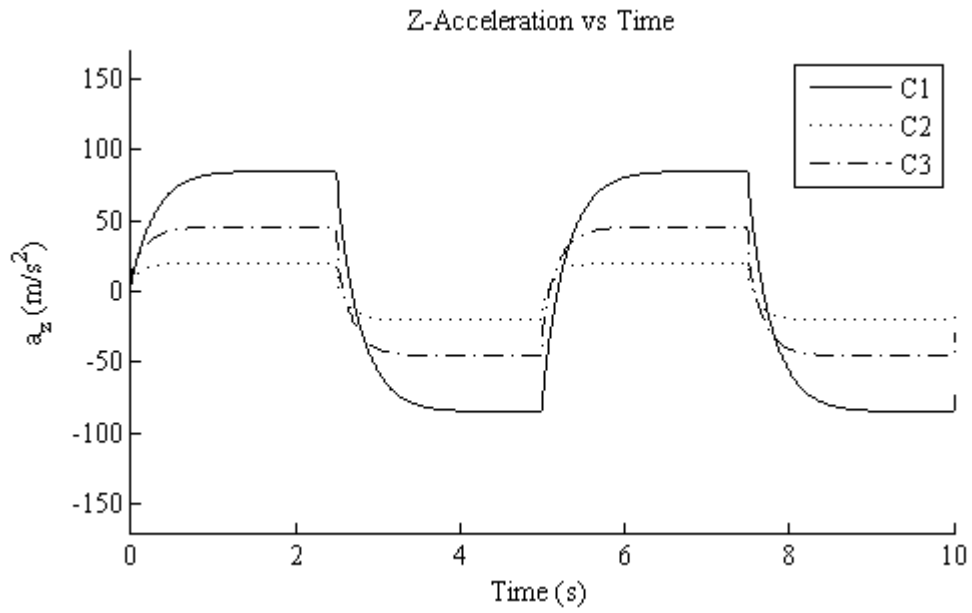


**Figure 5.1 Analysis 1 – Elevator Deflection vs Time**



**Figure 5.2 Analysis 1 – Angle of Attack vs Time**





**Figure 5.3 Analysis 1 – Pitch Acceleration Response**

## 5.2 Analysis 2

Designed un-augmented pitch, yaw and roll autopilots are analyzed for the case defined in Table 5.3 to show whether the requirements defined in Table 4.1 are satisfied for the alternative configurations.

**Table 5.3 Parameters for Analysis 2**

<b>Property</b>	<b>Symbol</b>	<b>Value</b>
Missile Configuration		C1-C2-C3
Mach No	M	0.3 – ... – 2.4

Results show that performance for pitch and yaw autopilots for any configuration are found to be the same since the pitch and aerodynamics are the same. Also, as explained in section 3.2, the roll aerodynamics is assumed to be the same for all configurations which caused the performance analysis for roll autopilots to be the same.

As seen from Figure 5.4 and Figure 5.5, pitch and yaw autopilot rise time performance for C1 is as required. However, C2 and C3 cannot satisfy the rise time requirements for most of the flight region. The roll autopilot rise time performance is as required for all configurations which are presented in Figure 5.6.

For all the configurations, maximum percent overshoot requirements are satisfied as in Figure 5.7, Figure 5.8 and Figure 5.9.

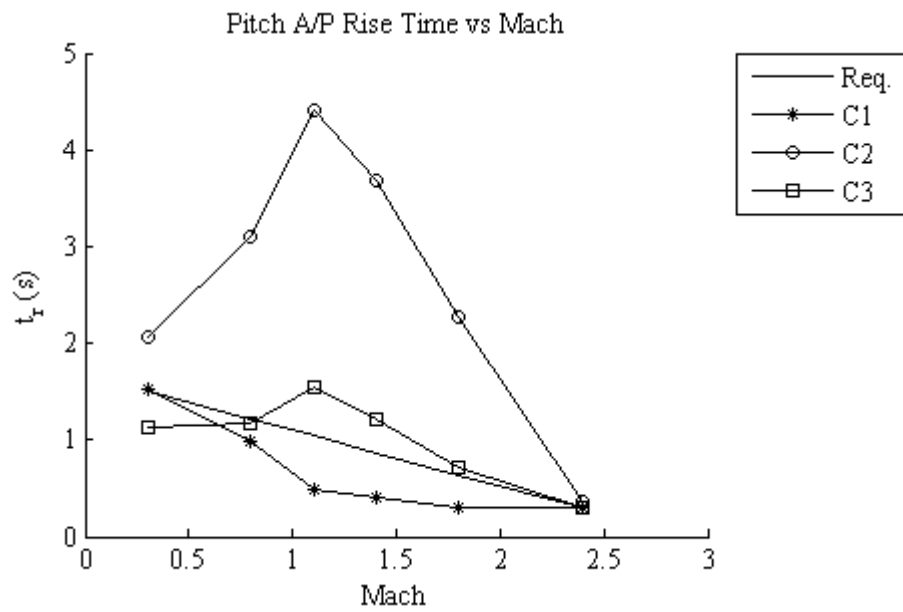


Figure 5.4 Analysis 2 – Pitch A/P Rise Time with Mach

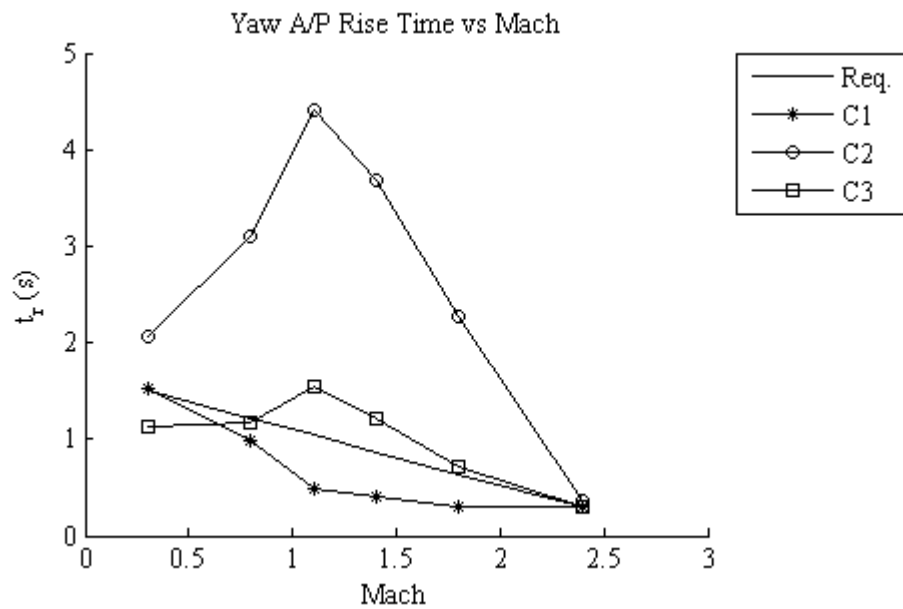


Figure 5.5 Analysis 2 – Yaw A/P Rise Time with Mach

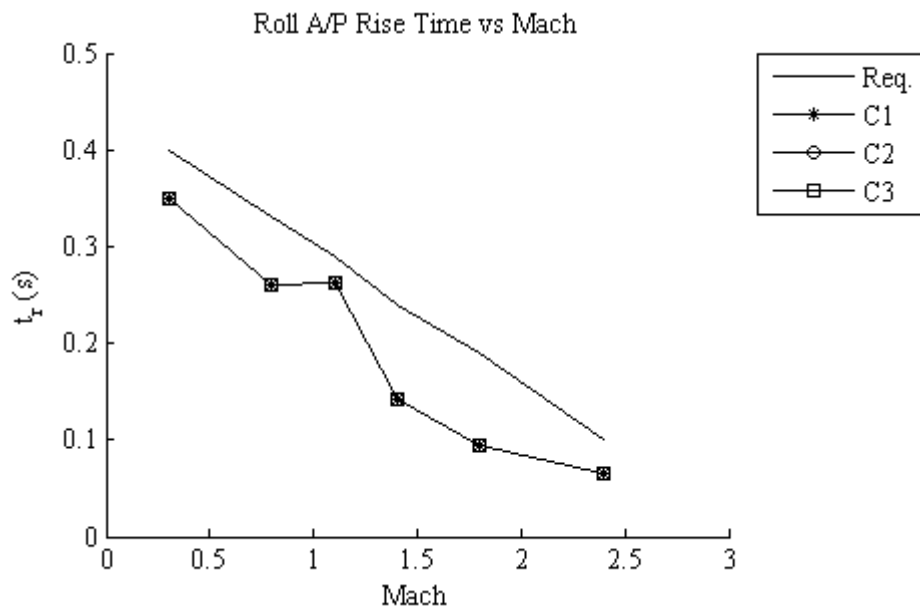


Figure 5.6 Analysis 2 – Roll A/P Rise Time with Mach

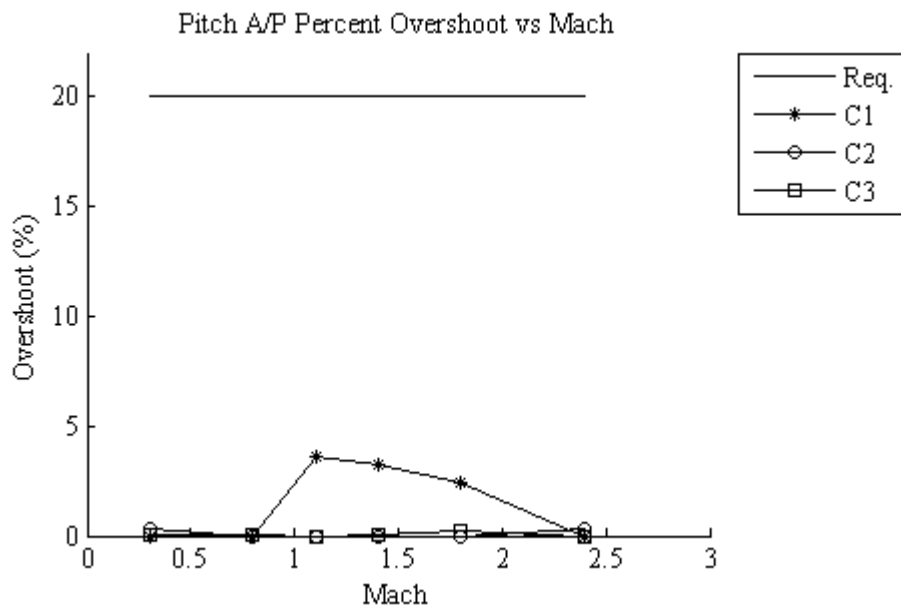


Figure 5.7 Analysis 2 – Pitch A/P Percent Overshoot with Mach

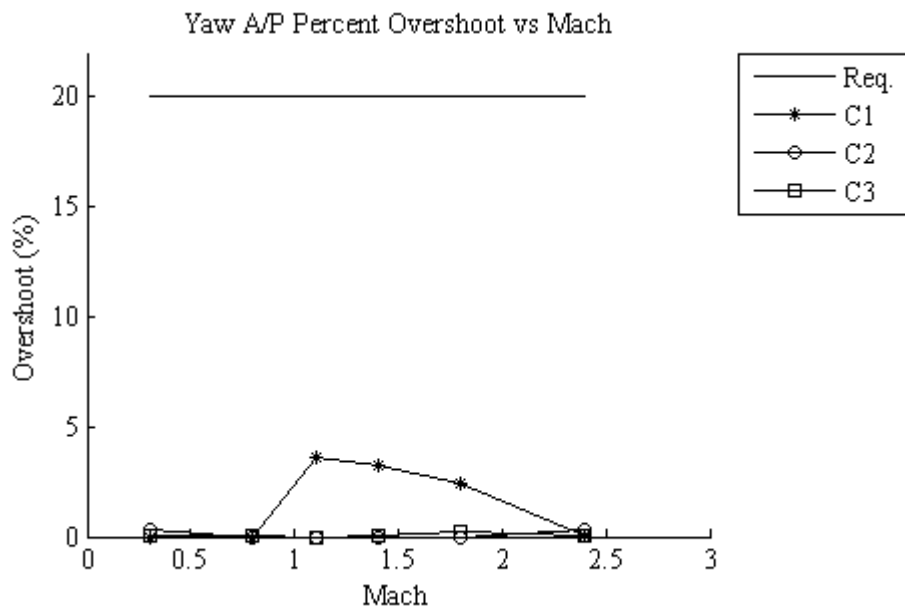


Figure 5.8 Analysis 2 – Yaw A/P Percent Overshoot with Mach

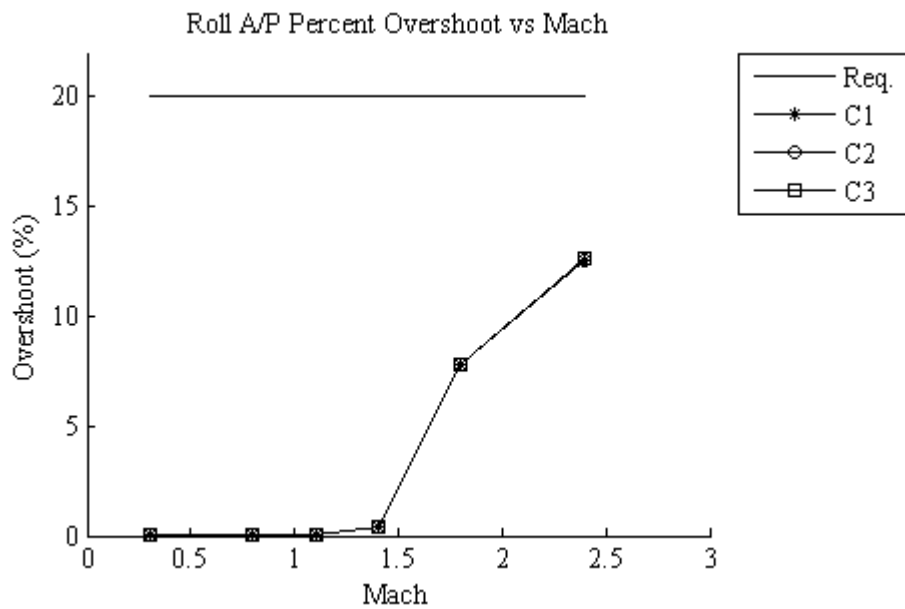


Figure 5.9 Analysis 2 – Roll A/P Percent Overshoot with Mach

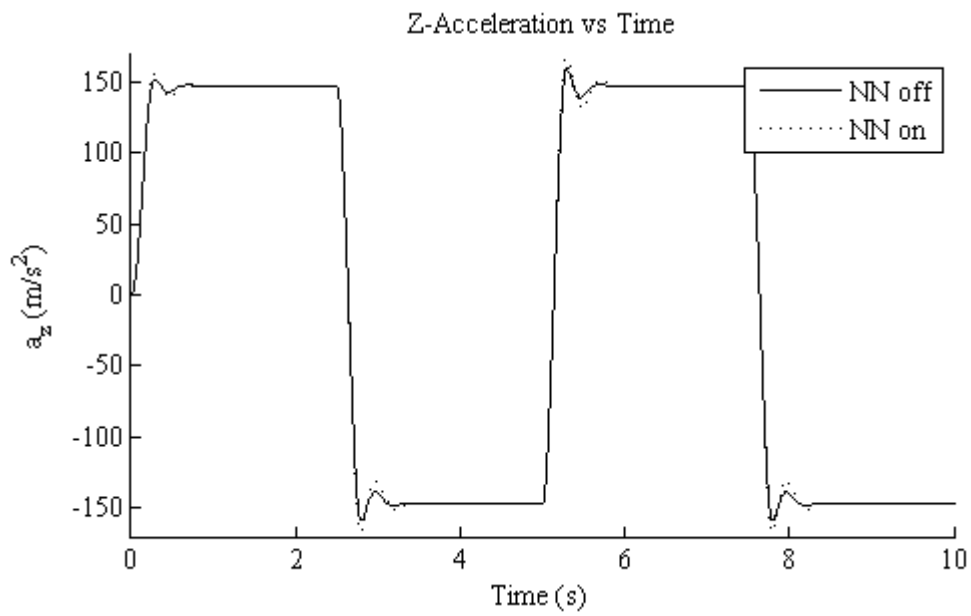
### 5.3 Analysis 3

The effect of adaptive augmentation on missile pitch and yaw autopilots is analyzed for the case in Table 5.4. The purpose of this analysis is to show that if the modeling error caused by linearization is small, interference of adaptive augmentation on the existing controller is small.

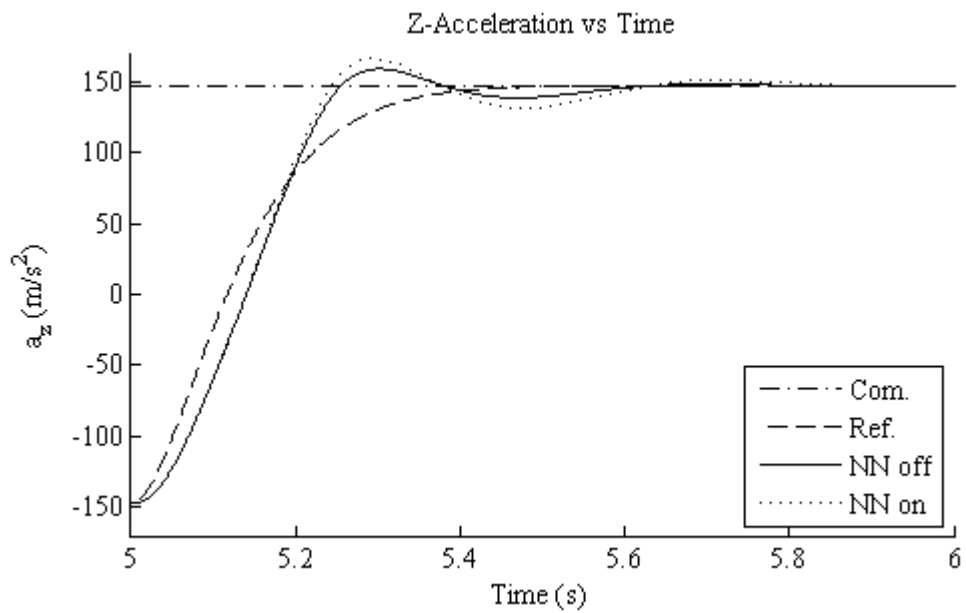
**Table 5.4 Parameters for Analysis 3**

<b>Property</b>	<b>Symbol</b>	<b>Value</b>
Missile Configuration		C1
Mach No	M	1.8
Number of Neurons	N	5
Input – Hidden Layer Weights Learning Rate	$\Gamma_v$	1
Hidden – Output Layer Weights Learning Rate	$\Gamma_w$	0.001
e-modification Coefficient	$\lambda$	0.001
Activation Potential	a	0.1 – ... – 1
Total Simulation Time	s	10

Results of this case show that the response of augmented pitch autopilot not much altered by neural networks. This result can be observed from transverse acceleration presented with Figure 5.10 and Figure 5.11, pitch rate given in Figure 5.12 and control inputs Figure 5.13. In Figure 5.13, it is observed that the order of magnitude of neural network control input is very low compared to the existing autopilot control input. Also weight histories in Figure 5.14 and Figure 5.15 show that instantaneous adaptation for this configuration is observed which means that the neural network weights did not converge.



**Figure 5.10 Analysis 3 – Pitch Acceleration Response**



**Figure 5.11 Analysis 3 – Pitch Acceleration Response (5s – 6s)**

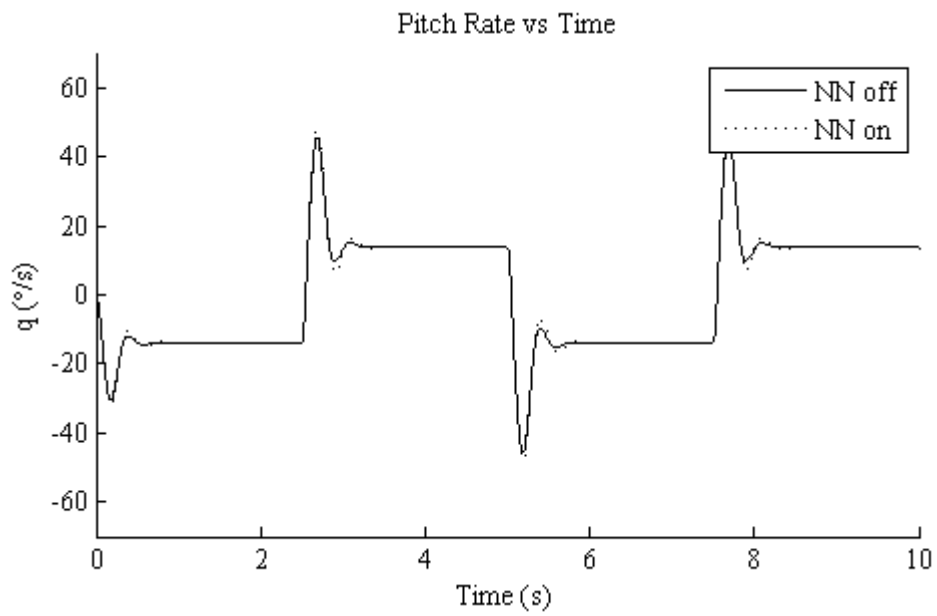


Figure 5.12 Analysis 3 – Pitch Rate

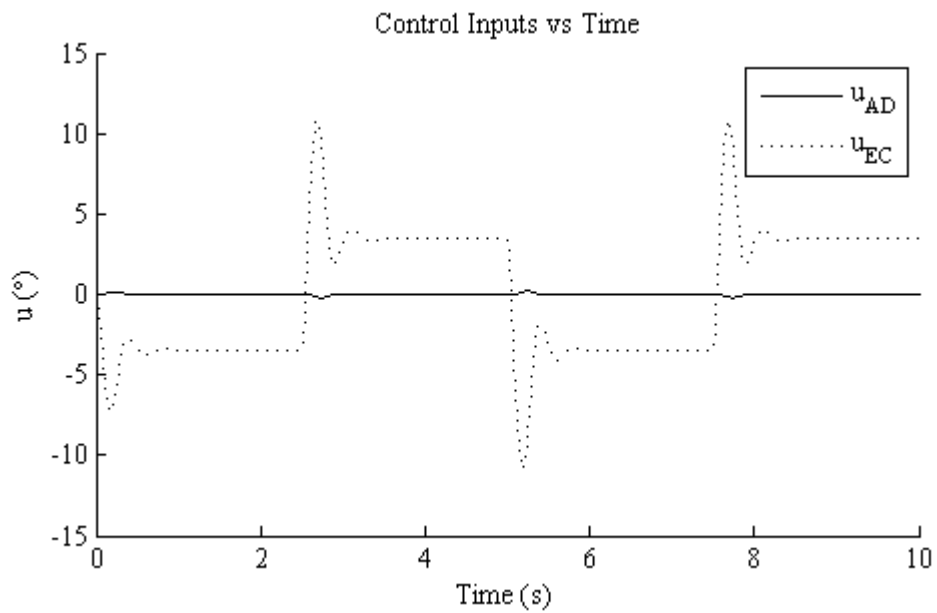
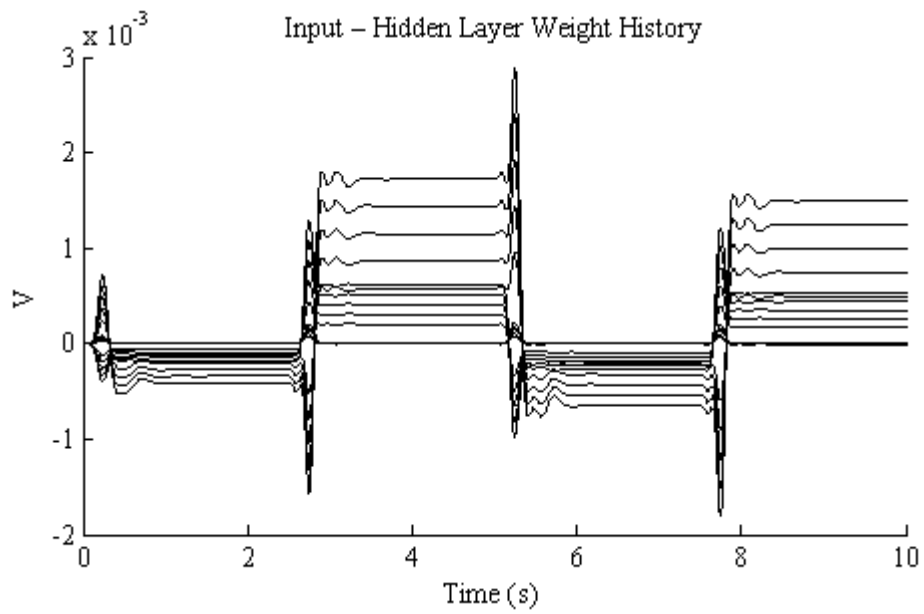
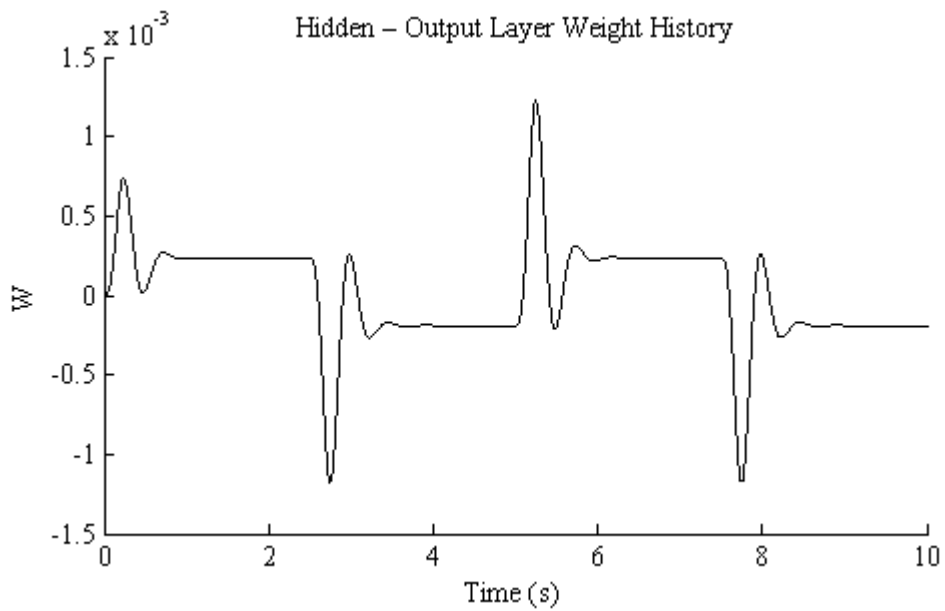


Figure 5.13 Analysis 3 – Elevator Control Inputs





**Figure 5.14 Analysis 3 – Pitch NN Input – Hidden Layer Weights**



**Figure 5.15 Analysis 3 – Pitch NN Hidden – Output Layer Weights**

#### 5.4 Analysis 4

The effect of adaptive augmentation on missile pitch and yaw autopilots is analyzed for the case in Table 5.5. The purpose of this analysis is to show that adaptive augmentation of existing controller can compensate for the modeling error originated from changing center of gravity location.

**Table 5.5 Parameters for Analysis 4**

<b>Property</b>	<b>Symbol</b>	<b>Value</b>
Missile Configuration		C2
Mach No	M	1.8
Number of Neurons	N	5
Input – Hidden Layer Weights Learning Rate	$\Gamma_v$	1
Hidden – Output Layer Weights Learning Rate	$\Gamma_w$	0.001
e-modification Coefficient	$\lambda$	0.001
Activation Potential	a	0.1 – ... – 1
Total Simulation Time	s	70

Results of this case show that the response of augmented pitch autopilot for this configuration is improved by neural networks. This result can be observed from transverse acceleration presented with Figure 5.16, Figure 5.17 and Figure 5.18, pitch rate given in Figure 5.19 and control inputs Figure 5.20. Figures show that augmented autopilot response satisfies the requirements defined using reference model with time. In Figure 5.20, it is shown that order of magnitude of neural network control input is comparable to existing autopilot control input. Also, weight histories in Figure 5.21 and Figure 5.22 show that adaptation for this configuration is observed which means neural network weights converge and learn this configuration.

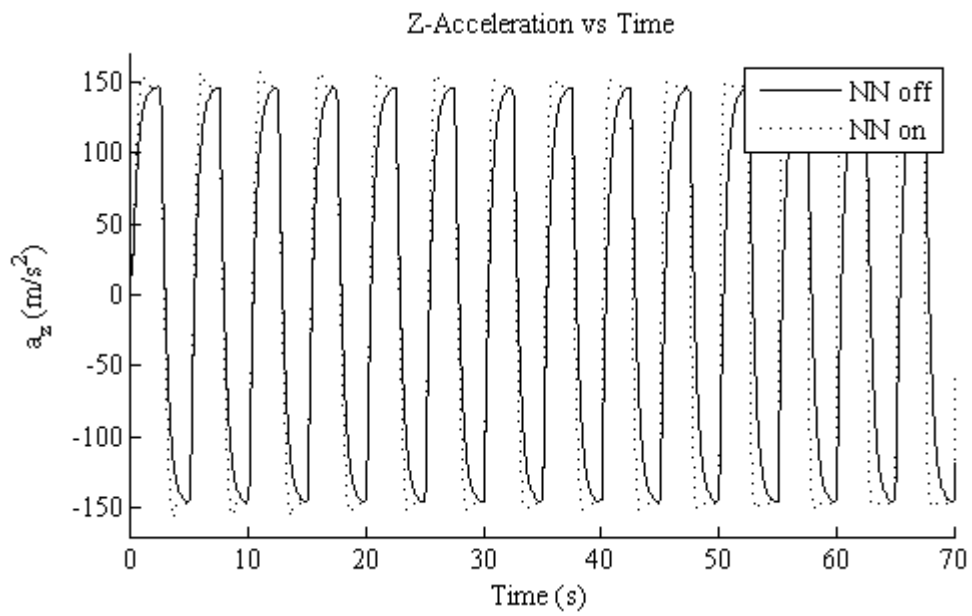


Figure 5.16 Analysis 4 – Pitch Acceleration Response

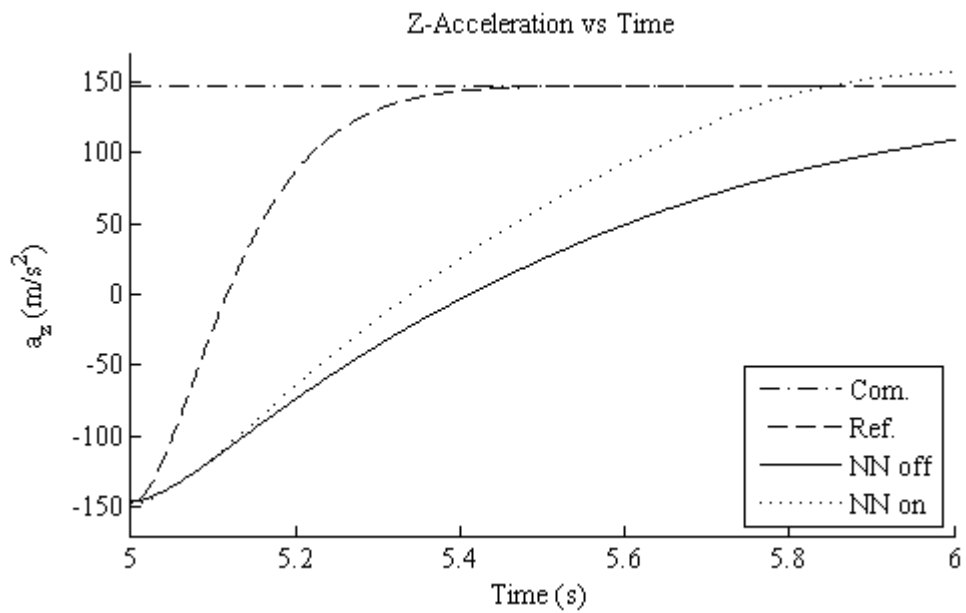


Figure 5.17 Analysis 4 – Pitch Acceleration Response (5s – 6s)

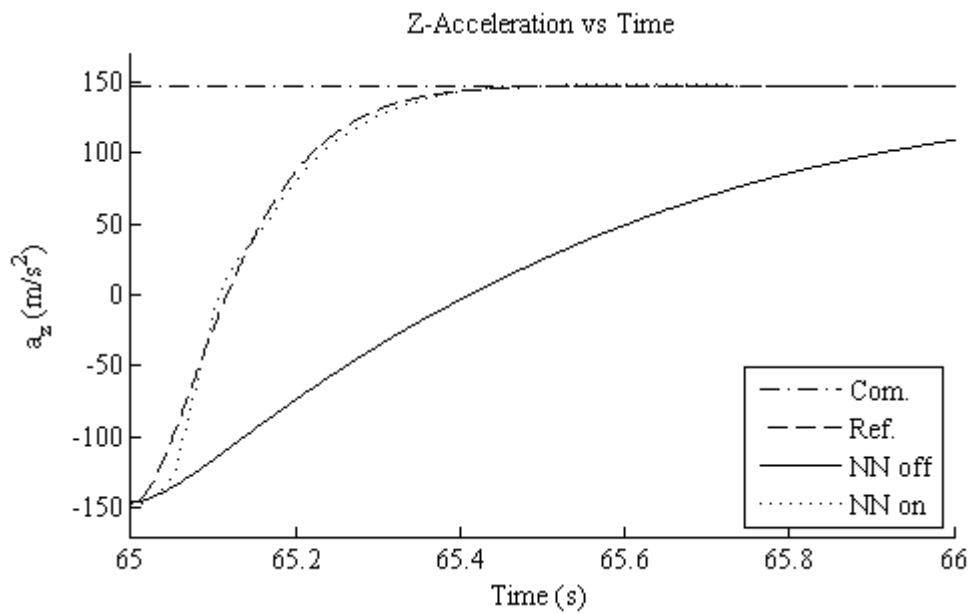


Figure 5.18 Analysis 4 – Pitch Acceleration Response (65s – 66s)

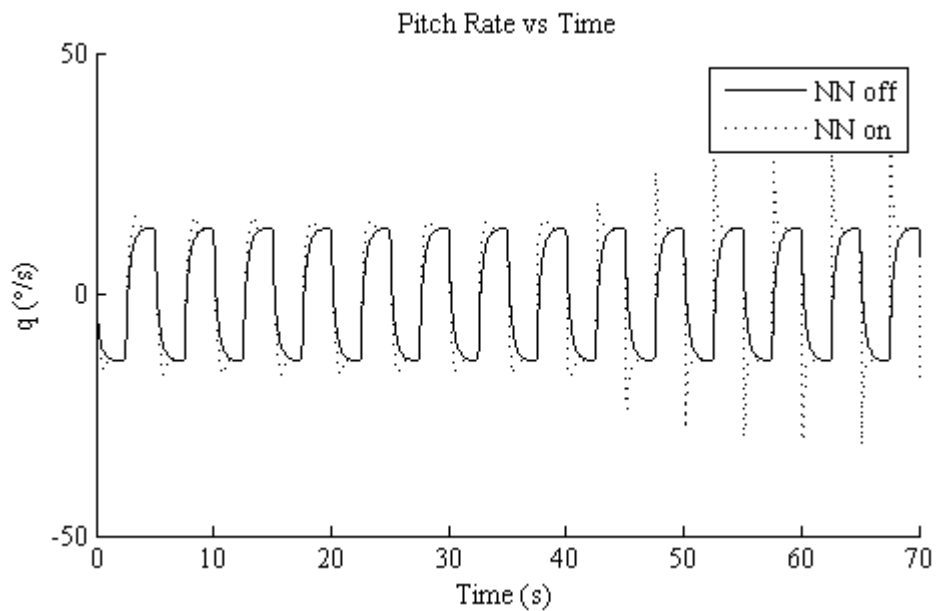


Figure 5.19 Analysis 4 – Pitch Rate

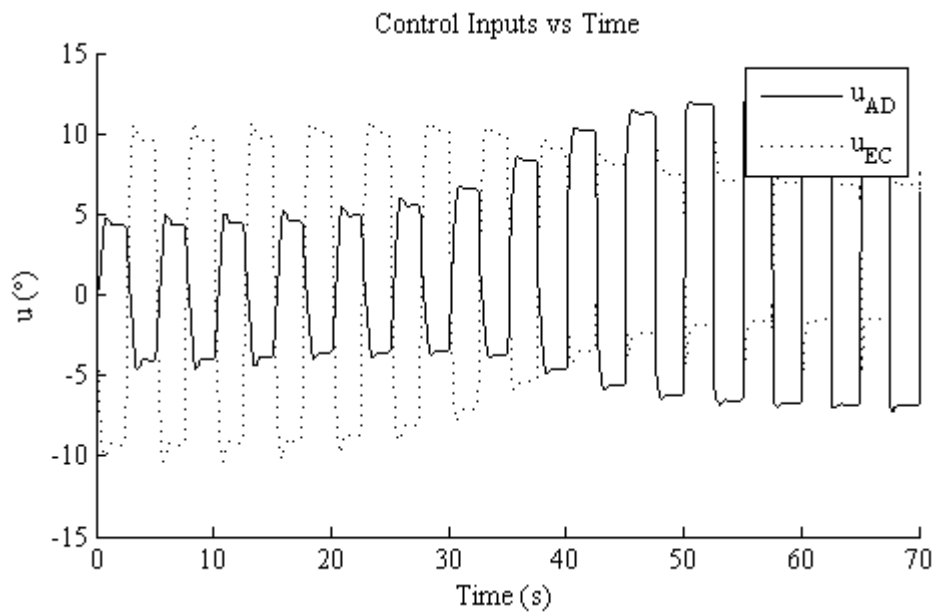


Figure 5.20 Analysis 4 – Elevator Control Inputs

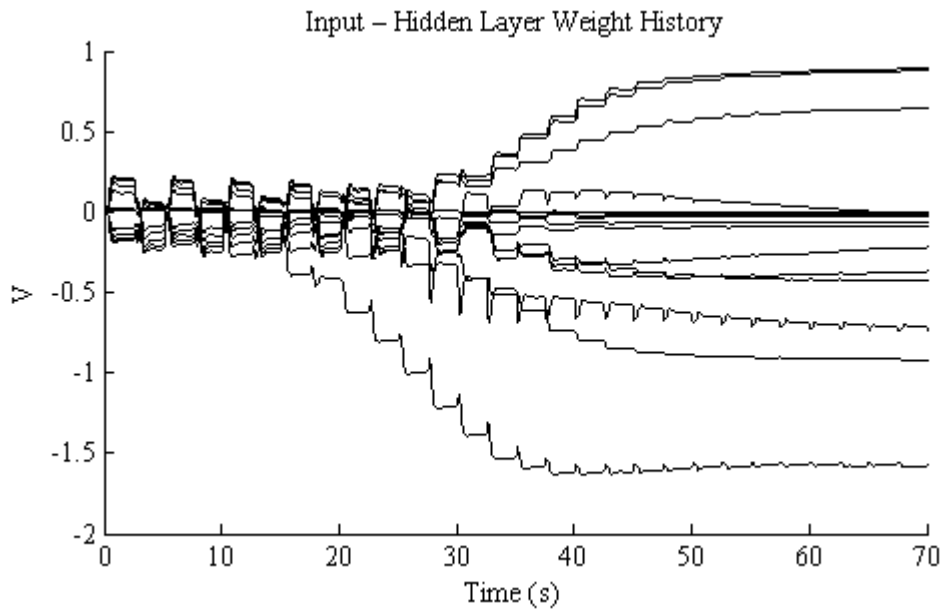
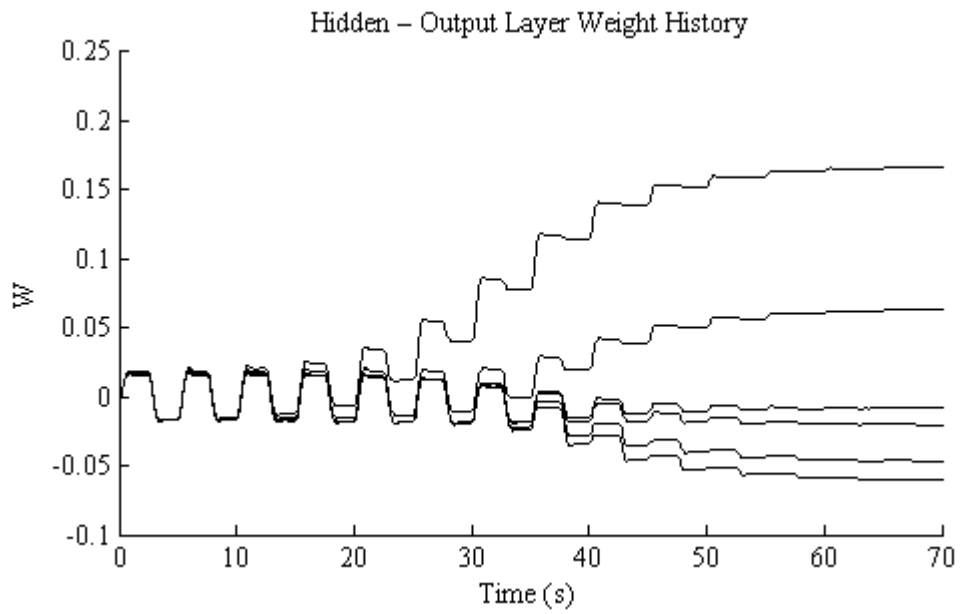


Figure 5.21 Analysis 4 – Pitch NN Input – Hidden Layer Weights



**Figure 5.22 Analysis 4 – Pitch NN Hidden – Output Layer Weights**

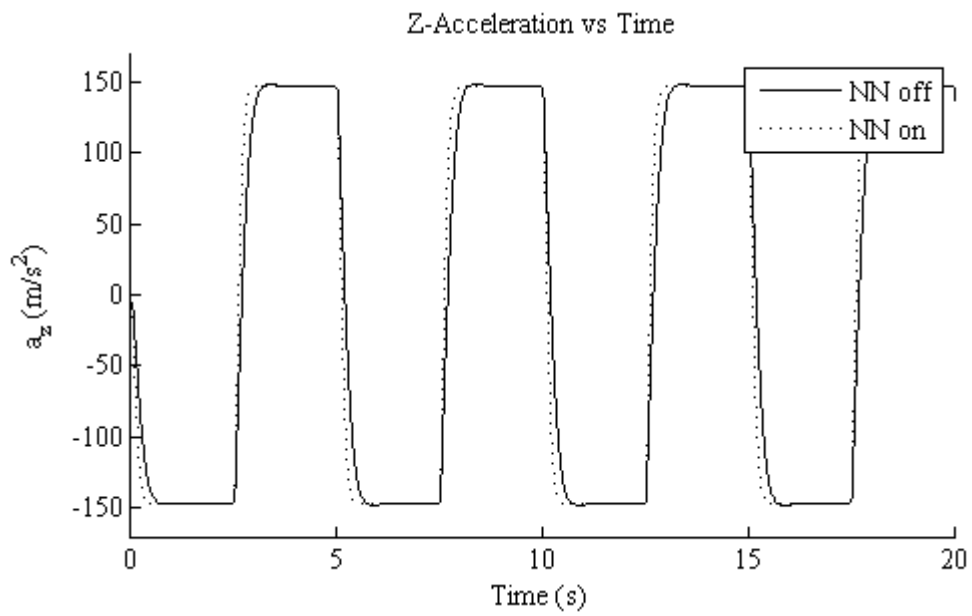
## 5.5 Analysis 5

The effect of adaptive augmentation on missile pitch and yaw autopilots is analyzed for the case in Table 5.6. The purpose of this analysis is to show that adaptive augmentation of existing controller can compensate for the modeling error originated from the varying aerodynamics due to the changing tail geometry.

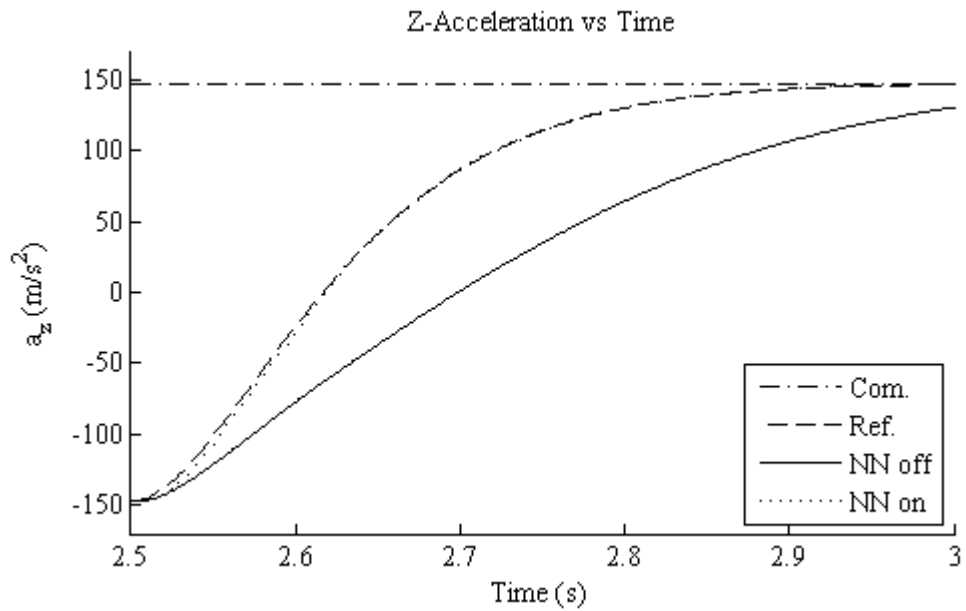
**Table 5.6 Parameters for Analysis 5**

<b>Property</b>	<b>Symbol</b>	<b>Value</b>
Missile Configuration		C3
Mach No	M	1.8
Number of Neurons	N	5
Input – Hidden Layer Weights Learning Rate	$\Gamma_v$	1
Hidden – Output Layer Weights Learning Rate	$\Gamma_w$	0.1
e-modification Coefficient	$\lambda$	0.001
Activation Potential	a	0.1 – ... – 1
Total Simulation Time	s	20

Results of this case show that the response of augmented pitch autopilot for this configuration is also improved by neural networks. This result can be seen in transverse acceleration presented with Figure 5.23 and Figure 5.24, pitch rate given in Figure 5.25 and control inputs Figure 5.26. Figures show that augmented autopilot response satisfies the requirements defined using reference model very fast. Also it is observed in Figure 5.24 that perfect tracking of the reference model is achieved with the neural network augmented autopilots. In Figure 5.26, it is shown that order of magnitude of neural network control input is also comparable to existing autopilot control input as the previous case analysis 4. Also, weight histories in Figure 5.27 and Figure 5.28 show that instantaneous adaptation for this configuration is observed meaning that neural network weights did not converge as in the case analysis 3.



**Figure 5.23 Analysis 5 – Pitch Acceleration Response**



**Figure 5.24 Analysis 5 – Pitch Acceleration Response (2.5s – 3s)**



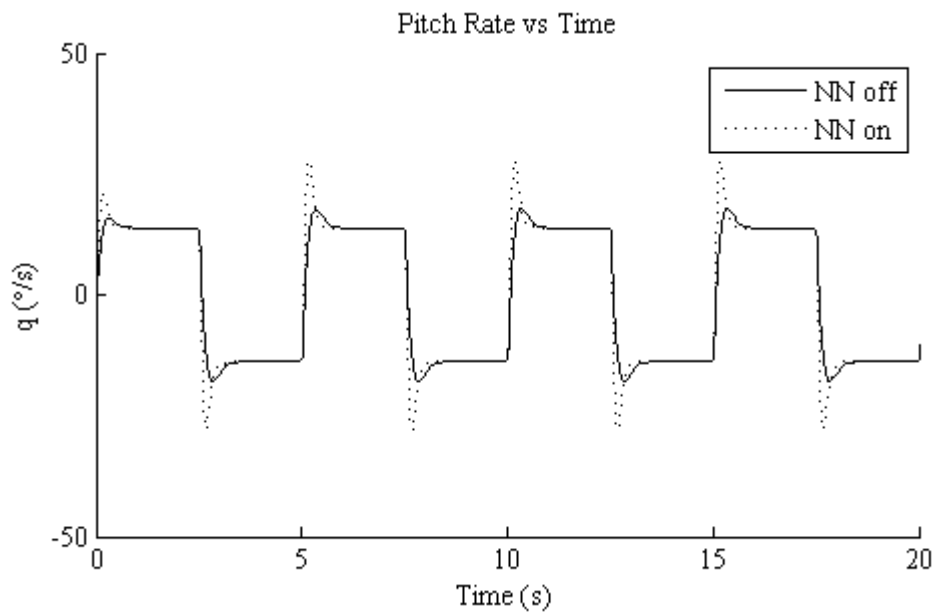


Figure 5.25 Analysis 5 – Pitch Rate

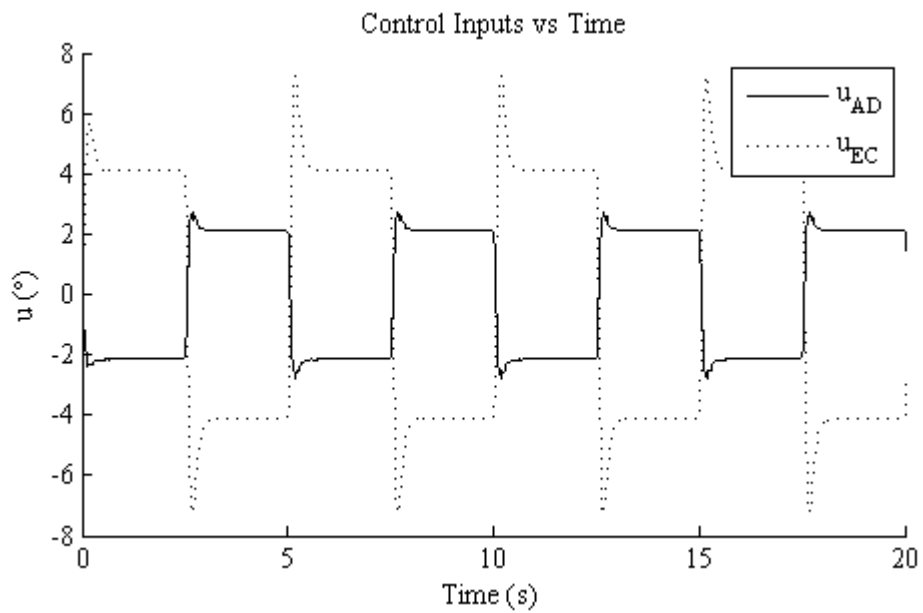
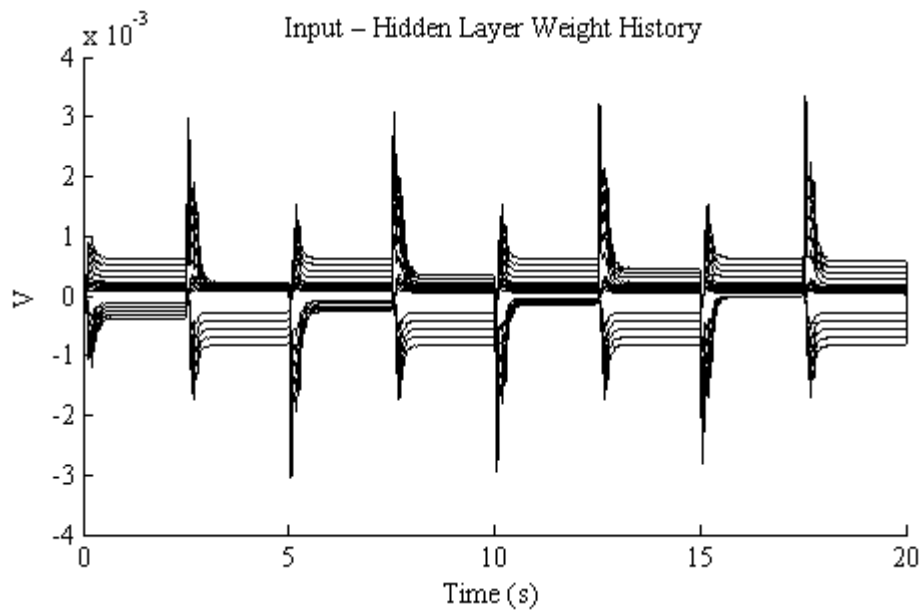
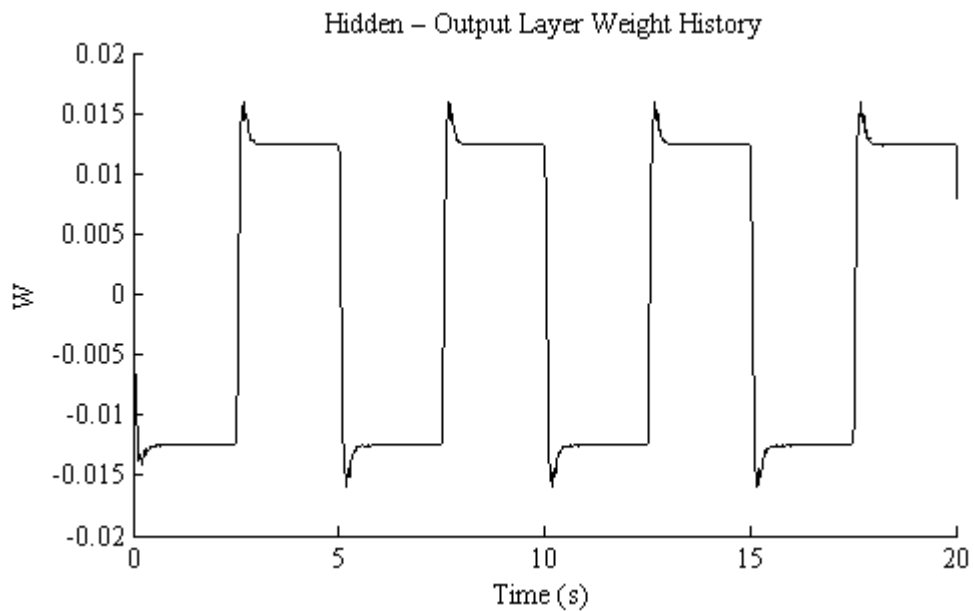


Figure 5.26 Analysis 5 – Elevator Control Inputs



**Figure 5.27 Analysis 5 – Pitch NN Input – Hidden Layer Weights**



**Figure 5.28 Analysis 5 – Pitch NN Hidden – Output Layer Weights**

## CHAPTER 6

### CONCLUSIONS

With this thesis, an automated aerodynamic analysis tool is developed and used to create different configurations with different aerodynamics, such that the performance of the control system designed for the base configuration decreases. After modeling the missile dynamics, six degree of freedom simulation is developed and used to simulate different missile configurations and analyze control system performance during the analysis. Afterwards, control systems are designed with the derivation of linear missile models considering time domain performance requirements. Then, control systems are augmented using existing control system augmentation approach. Lastly the performance of un-augmented and augmented existing controllers are analyzed and discussed.

Analysis 1 presented in previous chapter showed that the reason for autopilot performance decrease for varying configurations is the change in control effectiveness. Control effectiveness for configurations is shown to reduce with the order C1, C3 and C2. Control system performance for different configurations is also shown to decrease in the same manner as explained in analysis 2. With the adaptation of existing autopilot designed for C1, augmented pitch and yaw autopilots satisfy the required performance for base configuration C1 and varying configurations C2 and C3 as presented with analysis 3 to 5. Augmented autopilots for C2 in analysis 4 are shown to have global (convergent) adaptation and neural networks learned the system. For this analysis, augmented autopilots took some time to achieve adaptation to the uncertain aerodynamics of C2. On the other hand, augmented autopilots for C3 in analysis 5 achieve local adaptation in which the neural network weights did not converge. But the adaptation of augmented autopilots in analysis 5 is found to be instantaneous.

During the analysis, some restrictions of adaptation method for this kind of problems are discovered. With this method response characteristics depend on the flight velocity, command and amplitude of the command. This in turn required tuning process to be performed at some design points for different commands and learning rate scheduling. Furthermore, increasing the learning rates of the neural networks may cause unwanted high frequency dynamics in missile response for certain conditions.

To summarize, overall assessment of adaptive augmentation architecture presented herein work well. For an agile missile control problem, in which lateral accelerations are usually controlled, application may not increase the performance much because of the restrictions of the method. The analysis cases show that instantaneous adaptation with this method is better than the convergent adaptation which requires more time to satisfy the performance defined with requirements.

## REFERENCES

- [1] Wadley, C.E., Modeling and Sensitivity of the Pole-Placement Design of a Missile Control System, Msc. Thesis, The University of Texas at Arlington, December, 1998
- [2] Gurfil, P., “Synthesis Of Zero Miss Distance Missile Guidance Via Solution Of An Optimal Tuning Problem”, Control Engineering Practice 9, PP 1117–1130, 2001
- [3] Devaud, E., Siguerdidjane, H., Font ,S., “Some Control Strategies For A High-Angle-Of-Attack Missile Autopilot” , Control Engineering Practice 8, PP 885-892, 2000
- [4] Ogata, K., Modern Control Engineering 4<sup>th</sup> Edition, Prentice Hall, Inc., Upper Saddle River, N.J., 2002
- [5] McFarland, M. B., “Adaptive Nonlinear Control of Missiles using Neural Networks”, Phd Thesis, Georgia Institute of Technology, 1997
- [6] McFarland, M. B., “Neural-Adaptive Nonlinear Autopilot Design for an Agile Anti-Air Missile”, AIAA Guidance, Navigation, and Control Conference, San Diego, California, July, 1996
- [7] Hindman, R., Shell, W.M., “Missile Autopilot Design using Adaptive Nonlinear Dynamic Inversion”, 2005 American Control Conference, June 8-10, 2005. Portland, OR, USA
- [8] Sharma, M., “A Neuro-Adaptive Autopilot Design for Guided Munitions”, Phd Thesis, Georgia Institute of Technology, 2001
- [9] Kim, N.W., “Improved Methods in Neural Network-Based Adaptive Output Feedback Control, With Applications to Flight Control”, Georgia Institute of Technology, 2003
- [10] Kutay, A. T., “Neural Network Based Adaptive Output Feedback Control: Applications And Improvements” , Georgia Institute of Technology, 2005

- [11] Wise, K., Li D., Patel, V. V., Cao, C., Hovakimyan, N. , “Optimization of the Time-Delay Margin of L1 Adaptive Controller via the Design of the Underlying Filter”, AIAA Guidance, Navigation and Control Conference and Exhibit 20 - 23 August 2007, Hilton Head, South Carolina
- [12] Patel, V.V., Cao, C., Hovakimyan, N., Wise, K.A., Lavretsky, E., “L<sub>1</sub> Adaptive Controller for Tailless Unstable Aircraft in the Presence of Unknown Actuator Failures”, AIAA Guidance, Navigation and Control Conference and Exhibit 20 - 23 August 2007, Hilton Head, South Carolina
- [13] Isidori, A., Nonlinear control systems, Springer, Berlin, 1995.
- [14] Rysdyk, R.T., “Adaptive Nonlinear Flight Control”, Phd Thesis, , Georgia Institute of Technology, 1998
- [15] Phillips, W. F., Hailey, C.E., “A Review of Attitude Kinematics for Aircraft Flight Simulation”, Modeling and Simulation Technologies Conference, 14-17 August 2000, Denver, Colorado
- [16] Gorecki, R. M., “A Baseline 6 Degree of Freedom (DOF) Mathematical Model of a Generic Missile”, DSTO-TR-0931, July,2003
- [17] Etkin, B., Reid, L. D., Dynamics of Flight, 3<sup>rd</sup> Edition, John Wiley & Sons,Inc. , New York, 1996
- [18] Fleeman, E.L., Tactical Missile Design, 2<sup>nd</sup> Edition, AIAA, Inc., Reston, 2006
- [19] Missile Index Web Page, <http://missile.index.ne.jp/en/>, 14 March 2009
- [20] Blake, William B., “Missile Datcom User’s Manual - 1997 Fortran 90 Revision”, AFRL-VA-WP-TR-1998-3009, Feb. 1998
- [21] Corey, S., “A Comparison of Missile Autopilot Designs Using H<sub>∞</sub>, Control with Gain Scheduling and Nonlinear Dynamic Inversion” , Proceedings of the American Control Conference, Albuquerque, New Mexico, June 1997

- [22] Siouris, G. M., Missile Guidance and Control Systems, Springer-Verlag New York, Inc., 2004
- [23] Blakelock, J. H., Automatic Control of Aircraft and Missiles, 2<sup>nd</sup> Edition, John Wiley & Sons, Inc. , New York, 1991

# APPENDIX A

## AERODYNAMIC ANALYSIS

### A.1 Missile DATCOM

Missile DATCOM is a FORTRAN based executable analysis code which includes different empirical formulas, charts as well as theories and formulas depending on the flight region and geometry. It includes text files with '.dat' extension that are used for input and output. Files used during the aerodynamic analysis in this thesis and their descriptions are listed in the following table and function of other files in the program is explained in Ref. [20].

**Table A.1 Input and Output Files for DATCOM**

<b>Unit</b>	<b>Name</b>	<b>Usage</b>
4	for004.dat	Program output file for user specified parameters
5	for005.dat	User input file
6	for006.dat	Program output file

Missile DATCOM is configured using the commands called 'Control Cards' and physical properties of the geometry input are defined with subroutines called 'Namelist'. Inputs are defined using file 'for005.dat' and see Ref [20] for a complete list of control cards and namelists. Basic parameters used to define a missile are expressed in [20] and missile DATCOM input file configured for AIM9D aerodynamic analysis is presented below.



**Table A.2 DATCOM Input Parameters**

<b>Namelist</b>	<b>Parameter</b>	<b>Description</b>
REFQ	LREF	Reference length
	SREF	Reference area
	XCG	Center of gravity
FLTCON	BETA	Sideslip angle
	NALPHA	angle of alpha vector length
	ALPHA	angle of alpha vector
	NMACH	Mach vector length
	MACH	Mach vector
	ALT	Altitude vector
AXIBOD	TNOSE	Nose shape selection
	LNOSE	Nose length
	DNOSE	Nose diameter
	LCENTR	Center body length
	DCENTR	Center body diameter
	LAFT	Aft body length
	DAFT	Aft body diameter at the front
	DEXIT	Aft body diameter at the back
	BNOSE	Nose bluntness radius
FINSETn	ZUPPER	Span-wise vector for thickness to chord ratio of upper fin surface.
	LMAXU	Span-wise vector for fraction of chord from section leading edge to maximum thickness of upper surface.
	LFLATU	Span-wise vector for fraction of chord of constant thickness section of upper surface.
	SSPAN	Span-wise vector of semi-span locations
	CHORD	Span-wise vector of chord length
	XLE	Span-wise vector of nose to leading edge distance
	STA	Place of sweep angle (leading/trailing edge)
	SWEEP	Sweepback angle at each span station.
	NPANEL	Number of panels
	PHIF	Angle of the fins to vertical axis at back view
DEFLCT	DELTA <sub>n</sub>	Vector of fin set n deflection angles for each panel
	XHINGE	Vector of fin set hinge line locations relative to nose

A complete aerodynamic database has to include most of the coefficients needed to estimate aerodynamic forces and moments at any flight condition. An aerodynamic flight condition is defined by Mach number, altitude, alpha, beta and control surface deflections. However, Missile DATCOM gives aerodynamic coefficients for a vector of angles of attack at certain Mach number, altitude, beta and control surface deflection angles. So that the user has to change the parameters except the angles of attack defined in order to sweep all possible flight conditions. This procedure is done autonomously with aerodynamic database creation code developed in MATLAB. This code forms 'for005.dat' file according to the user inputs, run executable analysis code and save the aerodynamic analysis results read from 'for004.dat' to a file. Also the code generates a plot of the input geometry and potential center of pressure locations before the analysis to be able to decide on the stability characteristics of the missile.