ADAPTIVE NEURAL NETWORK APPLICATIONS ON MISSILE CONTROLLER DESIGN

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

BY

SERKAN SAĞIROĞLU

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

SEPTEMBER 2009

Approval of the thesis:

ADAPTIVE NEURAL NETWORK APPLICATIONS ON MISSILE CONTROLLER DESIGN

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

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

Prof. Dr. İsmail H. Tuncer _____
Head of Department, **Aerospace Engineering**

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

Examining Committee Members:

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

Asst. Prof. Dr. İlkay Yavrucuk _____
Aerospace Engineering Dept., METU

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

Dr. Ali Türker Kutay _____
Aerospace Engineering Dept., METU

Dr. Tayfun Çimen _____
Senior System and Control Engineer, Roketsan

Date: 08.09.2009

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, Surname: Serkan Sağıroğlu

Signature:

ABSTRACT

ADAPTIVE NEURAL NETWORK APPLICATIONS ON MISSILE CONTROLLER DESIGN

Sağıroğlu, Serkan

M.Sc., Department of Aerospace Engineering

Supervisor : Asst. Prof. Dr. İlkay Yavrucuk

September 2009, 127 pages

In this thesis, adaptive neural network controllers are designed for a high subsonic cruise missile. Two autopilot designs are included in the study using adaptive neural networks, namely an altitude hold autopilot designed for the longitudinal channel and a directional autopilot designed for heading control. Aerodynamic coefficients are obtained using missile geometry; a 5-Degree of Freedom (5-DOF) simulation model is obtained, and linearized at a single trim condition. An inverted model is used in the controller. Adaptive Neural Network (ANN) controllers namely, model inversion controllers with Sigma-Pi Neural Network, Single Hidden Layer Neural Network and Background Learning implemented Single Hidden Layer Neural Network, are deployed to cancel the modeling error and are applied for the longitudinal and directional channels of the missile. This approach simplifies the autopilot designing process by combining a controller with model inversion designed for a single flight condition with an on-line learning neural network to account for errors that are caused due to the approximate inversion.

Simulations are performed both in the longitudinal and directional channels in order to demonstrate the effectiveness of the implemented control algorithms. The

advantages and drawbacks of the implemented neural network based controllers are indicated.

Keywords: Model Inversion, Adaptive Controller, Adaptive Neural Network, Missile Controller, Missile Autopilot

ÖZ

UYARLANABİLİR YAPAY SİNİR AĞLARI UYGULAMALARIYLA
FÜZE KONTROLCÜSÜ TASARIMI

Sağıroğlu, Serkan
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi : Yar. Doç. Dr. İlkay Yavrucuk

Eylül 2009, 127 sayfa

Bu tezde bir yüksek ses altı seyir füzesi için uyarlanabilir yapay sinir ağı kontrolcüleri tasarlanmıştır. Bu çalışmada uyarlanabilir yapay sinir ağları yöntemi ile iki adet otopilot tasarımı içerilmiştir, yani düşey kanal için irtifa tutma otopilotu tasarlanmış, yatay kanal kontrolü için istikamet otopilotu tasarlanmıştır. Aerodinamik katsayılar füze geometrisi kullanılarak elde edilmiş, 5 serbestlik dereceli benzetim modeli elde edilmiş, tek bir trim koşulu için doğrusallaştırılmıştır. Kontrolcü içinde bir tersleme modeli kullanılmıştır. Uyarlanabilir yapay sinir ağı kontrolcüleri, yani Sigma-Pi yapay sinir ağı, tek gizli katmanlı yapay sinir ağı, arka planda öğrenme uygulanmış tek gizli katmanlı yapay sinir ağı, modelleme hatasını gidermek için yerleştirilmiş ve füzenin düşey ve yatay kanalları için uygulanmıştır. Bu yaklaşım tek bir uçuş koşulunda tersleme kontrolcüsü tasarımı ile çevrim içi öğrenen yapay sinir ağını birleştirip yaklaşık terslemenin sebep olduğu hataları hesaba katarak otopilot tasarım sürecini basitleştirmektedir.

Uygulanan kontrol algoritmalarının etkinliğini göstermek amacıyla düşey ve yatay kanallar için benzetimler yapılmıştır. Uygulanan yapay sinir ağı kontrolcülerinin avantajları ve eksikleri gösterilmektedir.

Anahtar Kelimeler: Model Tersleme, Uyarlanabilir Kontrolcü, Uyarlanabilir Yapay Sinir Ağı, Füze Kontrolcüsü, Füze Otopilotu

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Asst. Prof. Dr. İlkay Yavrucuk for his great care and guidance throughout my studies.

I would like to thank all jury members for their advices and corrections.

I would also like to thank Pınar Çeliker, for her endless friendship and support.

For his assistance, I am grateful to Emrah Zaloğlu. Thank you for your support and help without concerning any time limitation.

I want to state my thanks to Hakan Tiftikçi for suggestions during the period of calculating the missile aerodynamic coefficients and answering all of my questions.

Thanks to ROKETSAN, Systems Engineering Department for providing me a comfortable working and computing environment.

Thanks to all those who contributed to the completion of this work in anyway.

Finally I would like to thank my parents Yurdagül Sağıroğlu and Remzi Sağıroğlu, without them nothing could have been possible.

TABLE OF CONTENTS

ABSTRACT..... iv
ÖZ..... vi
ACKNOWLEDGEMENTS..... viii
TABLE OF CONTENTS..... ix
LIST OF ABBREVIATIONS..... xii
CHAPTERS
1 INTRODUCTION..... 1
1.1 Literature Survey and Motivation..... 1
1.2. Scope..... 10
1.3. Outline..... 11
2 MISSILE FLIGHT DYNAMICS MODELING..... 13
2.1 Assumptions..... 13
2.2 5-DOF Dynamic Model..... 15
2.2.1 Translational Motion..... 16
2.2.2 Rotational Motion..... 18
2.3 Aerodynamic Model..... 19
2.3.1 Aerodynamic Parameters..... 19
2.3.2 Aerodynamic Force and Moment Coefficients..... 22
2.4 Kinematics Model..... 23
2.5 Thrust Model..... 26
2.6 Guidance Model..... 28
2.6.1 Altitude Hold Guidance..... 28
2.6.2 Directional Guidance..... 29
2.7 Autopilot..... 31
2.8 Simulation Architecture..... 31
3 ADAPTIVE NEURAL NETWORK BASED ALTITUDE HOLD AUTOPILOT..... 34
3.1 Linearization..... 34
3.2 Neural Network Based Inversion Architecture..... 37

3.2.1 Inner Loop Design 39
3.2.2 Command Filter Design and Inner Loop Error Dynamics 42
3.2.3 Outer Loop Design 46
3.3 Online Learning Neural Network Architecture 47
3.3.1 Linear In the Parameters Sigma-Pi Neural Networks 48
3.3.2 Single Hidden Layer Neural Networks 52
4 ADAPTIVE NEURAL NETWORK BASED DIRECTIONAL AUTOPILOT . 57
4.1 Neural Network Based Inversion Architecture 57
4.1.1 Command Filter Design and Error Dynamics 59
4.2 Online Learning Neural Network Architecture 63
4.2.1 Linear In the Parameters Sigma-Pi Neural Network 63
4.2.2 Single Hidden Layer Neural Networks 65
5 BACKGROUND LEARNING IMPLEMENTED NEURAL NETWORK ADAPTATION 67
5.1 Background Learning Implemented Neural Network Architecture 67
5.1.1 Rank-1 Limitation 68
5.1.2 Combination of Online & Background Learning Law 68
5.1.3 Selection of Data Points for Background Learning 71
6 SIMULATION RESULTS 72
6.1 Comparison of LIP NN & SHL NN 72
6.1.1 Pitch Angle Response 73
6.1.2 Yaw Angle Response 75
6.1.3 Altitude Response 76
6.1.4 Missile Mach number 78
6.1.5 Adaptation Response 79
6.1.6 Actuator Deflections 81
6.2 Comparison of SHL NN & BL implemented SHL NN 84
6.2.1 Pitch Angle Response 85
6.2.2 Yaw Angle Response 87
6.2.3 Altitude Response 89
6.2.4 Missile Mach number 90
6.2.5 Adaptation Response 91
6.2.6 Actuator Deflections 95
7 CONCLUSION 98
REFERENCES 101
APPENDICES

A. AERODYNAMIC CHARACTERISTICS OF THE MISSILE 109
B. LINEARIZATION OF THE ROTATIONAL EQUATIONS OF MOTION119
C. PID CONTROLLER GAIN SELECTION..... 122

LIST OF ABBREVIATIONS

ACAH	Attitude Command Attitude Hold
ANN	Adaptive Neural Network
BL	Background Learning
BTT	Bank-to-Turn
CFD	Computational Fluid Dynamics
DOF	Degrees of Freedom
JDAM	Joint Direct Attack Munition
LOS	Line of Sight
LQR	Linear Quadratic Regulator
LIP	Linear in the Parameters
LTI	Linear Time Invariant
NN	Neural Network
PCH	Pseudo-Control Hedging
PD	Proportional Plus Derivative
SHL	Single Hidden Layer
STT	Skid-to-Turn

SYMBOLS

α	Angle of Attack
β	Side Slip Angle
γ	Flight Path Angle
Cg	Center of Gravity
δ	Fin Deflection
$\hat{\delta}$	Estimated Fin Deflection
a	Acceleration
h	Altitude
X	Axial Force
Y	Side Force
Z	Normal Force
$\dot{\gamma}$	Flight Path Angle Rate
V	Speed
a	Speed of Sound
WP	Waypoint
Kp	Proportional Gain
Kd	Derivative Gain
m	Mass of the Missile
ρ	Air Density
Q	Dynamic Pressure
λ	LOS angle
$\dot{\lambda}$	LOS rate
$N_{guidance}$	Guidance Constant
X_{cg}	Cg location on x-axis
w_{body}	Body Angular Velocity
F	Total External Forces
Φ	Euler Roll Angle
θ	Euler Pitch Angle
ψ	Yaw (Heading) Angle
p	Roll Angular Rate
q	Pitch Angular Rate
r	Yaw Angular Rate
L	Rolling Moment
M	Pitching Moment

I_{xx}	Moment of inertia with respect to missile body x-axis
I_{yy}	Moment of inertia with respect to missile body y-axis
I_{zz}	Moment of inertia with respect to missile body z-axis
I_{xy}	Product of inertia with respect to missile body x and y-axis
I_{yz}	Product of inertia with respect to missile body y and z-axis
I_{xz}	Product of inertia with respect to missile body x and z-axis
u	Missile Body velocity (x-axis)
v	Missile Body velocity (y-axis)
w	Missile Body velocity (z-axis)
g	Gravitational Acceleration
$\ddot{\theta}$	Euler Pitch Angular Acceleration
$\ddot{\psi}$	Euler Yaw Angular Acceleration
C_L	Dynamic Rolling Moment Coefficient
C_M	Dynamic Pitching Moment Coefficient
C_N	Dynamic Yawing Moment Coefficient
C_X	Dynamic Axial Force Coefficient
C_Y	Dynamic Side Force Coefficient
C_Z	Dynamic Normal Force Coefficient
C_l	Static Rolling Moment Coefficient
C_m	Static Pitching Moment Coefficient
C_n	Static Yawing Moment Coefficient
C_x	Static Axial Force Coefficient
C_y	Static Side Force Coefficient
C_z	Static Normal Force Coefficient
N	Yawing Moment
w_n	Natural Frequnecy
ζ	Damping Ratio
T	Missile Thrust
$Sref$	Missile Reference Area
$lref$	Missile Reference Length
M	Missile Mach number
$\dot{\phi}$	Euler Roll Rate
$\dot{\theta}$	Euler Pitch Rate
$\dot{\psi}$	Euler Yaw (Heading) Rate
$\dot{p}$	Body Roll Angular Acceleration
$\dot{q}$	Body Pitch Angular Acceleration
$\dot{r}$	Body Yaw Angular Acceleration
C_{mq}	C_m derivative with respect to q
$C_{m\dot{\alpha}}$	C_m derivative with respect to $\dot{\alpha}$
C_{nr}	C_n derivative with respect to r
$C_{n\dot{\beta}}$	C_n derivative with respect to $\dot{\beta}$
C_{yr}	C_y derivative with respective to r
C_{zq}	C_z derivative with respect to q

$\tilde{\theta},\tilde{\psi}$	Model Tracking Error
Upd_θ	Desired Combination of Proportional and Derivative Pitch Tracking Error
U_{ψ}	Desired Yaw Euler Angular Acceleration
U_{θ}	Desired Pitch Euler Angular Acceleration
Upd_ψ	Desired Combination of Proportional and Derivative Yaw Tracking Error
$A1$	Aerodynamic Stability Derivatives with respect to. translational states at trim
$A2$	Aerodynamic Stability Derivatives with respect to rotational states at trim
B	Aerodynamic Control Derivatives with respect to fin deflections at trim
$\overline{X}$	Normalized States Inputs to NN
ε_{ψ}	Inversion Error (yaw)
W, V	Neural Network Weights
μ	Modification term
ε_{θ}	Inversion Error (pitch)
$\hat{A}1$	Estimate of A1 used for Model Inversion
$\hat{A}2$	Estimate of A2 used for Model Inversion
$\hat{B}$	Estimate of B used for Model Inversion
$\Gamma_{learning}$	Learning rates
$\gamma_{learning}$	Learning rates
Bias	Bias Error
Uad_θ	Longitudinal Neural Network Output
Uad_ψ	Directional Neural Network Output

SUBSCRIPTS

x	Along Missile Body x-axis
y	Along Missile Body y-axis
z	Along Missile Body z-axis
com	Command
com_f	Filtered Command
missile	Missile
cg	Center of Gravity
cruise	Cruise
err	Error
e	Elevator
r	Rudder
trim	Trim
CF	Command filter
AF	Missile Airframe

CHAPTER 1

INTRODUCTION

1.1 Literature Survey and Motivation

Cruise missiles are defined as guided weapon system which use aerodynamic lift in order to maintain a certain altitude profile and are continuously powered by an air breathing or solid rocket engine. The major advantages of cruise missiles are their long range, low detectability and high accuracy [1].

The design cost is crucial for this type of munitions. The higher survivability and kill probability of cruise missiles justifies the design cost; providing a challenge for the missile system designers.

In this thesis, a high subsonic cruise missile is used as a case study. Modern high subsonic cruise missiles are required to fly in a wide envelope with regard to speed, altitude and angle of attack (α). This translates into nonlinear dynamics and aerodynamics usually observed as high levels of uncertainty during modeling and design [2].

Accurate system modeling is important in the design phase of a flight controller to maintain stable flight and reasonable control. Parametric uncertainty (changing mass, aerodynamic characteristics, variation in dynamic pressure), un-modeled dynamics, actuator displacement and rate saturation and assumptions made during control design itself may mainly cause errors on system modeling [3]. In addition, in high angle of attack, the dynamics of vehicles are uncertain and control

effectiveness is highly nonlinear. The highly nonlinear aerodynamics is not easily understood and expensive to model, because aerodynamic data for vehicles operating under such conditions is difficult to obtain [4]. The use of nonlinear actuation systems also increases the complexity of the control design and modeling [5]. Applied control laws are usually demonstrated on simulations with nonlinear aerodynamic and engine models, actuator models with position and rate saturations, and wind disturbance in order to approximate the real case as much as possible [6]. However, these approximations are usually not successful.

Several classical and modern control-law methods [7] are applied for controller design. They performed offline at a limited number of Linear Time Invariant (LTI) models. These models represent different conditions in the flight envelope, The time consuming and costly part of most of these linear controller designs is the need to determine the controller parameters (i.e. controller gains) for all possible flight conditions in system modeling. Long and tedious processes for arranging gains in the flight controller software (i.e. extensive gain scheduling computations [7]) are time consuming [2]. If the missile has large flight envelopes, the conventional approaches with gain scheduling might require many flight conditions to be tested. The number of required gains to be scheduled becomes very large [8]. Moreover, proper gain scheduling requires accurate aerodynamic models [9]. Therefore, it requires expensive wind tunnel testing, necessary for comparing and verifying the aerodynamic model of the missile mostly obtained from Computational Fluid Dynamics (CFD) tools or empirical formulae. On the other hand, these “Gain-Scheduling” based control techniques may not compensate these errors on system modeling and give unexpected results for the initial design. Since “Gain-Scheduling” based control techniques do not globally have the desirable properties exhibited locally by its constituent point designs, it is not suitable for highly nonlinear dynamics and different flight conditions in the flight envelope [2].

The modeling error between the mathematical model and the real system may cause performance degradation even if nonlinear controllers are used [10]. Many nonlinear control approaches tend to generate large actuators’ commands or rates and have poor performance when actuators become saturated; this is the result of

actuator nonlinearity. Parametric uncertainty also limits the operational envelope of the vehicle where nonlinear control designs are valid. Un-modeled dynamics and input saturation can also limit the achievable bandwidth of the system [11] and cause robustness problems [10]. Due to these problems, nonlinear control approaches have not been commonly used on missiles [6]; missile control design has still been dominated by classical control techniques [2]. Although gain scheduling has historically been proven to be successful in a variety of applications, future designs will need more advanced methods which explicitly account for the nonlinearities of the system related to flight condition [5].

Nevertheless, substantial advances in nonlinear control have been recorded in the 1990's, due to theoretical achievements and the availability of powerful computer hardware and user-friendly nonlinear simulation software [6]. Dynamic model inversion (a popular feedback linearization method [12]) reduces design cost of a flight control system in comparison to a gain schedule design [13], [14]. Schumacher and Khargonekar [15] analyzed theoretically the stability of the flight control system with the two-timescale separation assumption and Lyapunov theory. It refers to the inversion of the aerodynamic force or moment equations of motion. The primary difficulty associated with the use of this type of feedback linearization is that a detailed and accurate knowledge of the nonlinear plant dynamics is required. Uncertain aerodynamic effects make this difficult, because aerodynamic coefficients are nonlinear functions of several physical variables difficult to identify accurately [10]. The technique assumes exact knowledge of aerodynamic coefficients and aircraft configuration parameters (e.g., reference wing area, mean aerodynamic chord, mass, moment of inertia) in the entire flight envelope. In practice, this assumption is not valid. Since this approach depends on state feedback and dynamic model inversion and is sensitive to modeling error [16], it will raise issues related to robustness. Hence, further study on robustness is necessary. A secondary issue is that accurate full-envelope nonlinear inversion is computationally intensive. A high-fidelity nonlinear force and moment model must be constructed and inverted in real time. Therefore the modeling errors in some aerodynamic coefficients deteriorate the nonlinear controller performance.

One approach to the problem is to implement Adaptive Neural Networks (ANNs) to nonlinear controllers [10], [17], [18]. The method proposed in [19] is that neural networks can be used in conjunction with any nominal controller, like dynamic inversion based controller [20] for linearized systems. Feedback linearization (i.e. dynamic model inversion) [12], [21] is a well known nonlinear control approach that can be combined with a Neural Network (NN). It simplifies the controller design by combining an inverted approximate model designed at a single flight condition with an online learning NN to account for errors due to the approximate inversion [2]. A key property of the ANN based nonlinear controller architecture is their ability to learn on line and to eliminate the need for an extensive design process to provide satisfactory stability. Stability analysis for the control of similar nonlinear systems using nonlinearly parameterized networks first appeared using a discrete time formulation in [22] and using a continuous time formulation in [23]. Extensions to non-similar systems together with applications in flight control can be found in [13], [24], [25], [26]. Research has shown that online learning NNs are well suited to cancel model inversion errors in such controllers [13], [27].

A NN can be thought of as a parameterized class of nonlinear maps [14]. This property applies to simple, or complex dynamic systems, time invariant or time varying systems, noise-free or noise-corrupted systems, and linear or nonlinear systems [2]. Using their universal approximation capability, the adaptive controller based on NNs can be designed without significant prior knowledge of the system dynamics [28]. Unnikrishnan et al [19] explained this type of controller design in two steps: (i) synthesis of a set of NNs that capture the unmodeled dynamics and parametric uncertainties of the plant online (ii) computation of a controller that drives the states of the plant to that of a desired nominal model. It eliminates the limitations in the plant inputs, such as: actuator displacement limits, actuator rate limits, linear input dynamics, and time delay [8]. The method of pseudo-control hedging (PCH) [3], [8], [29] is the one which was used to protect the adaptation process from actuator limits and dynamics (i.e. actuator saturation) by modifying the inner-loop reference model dynamics in a way that allows continued adaptation.

Several authors have used NNs to solve problems optimally associated with control of nonlinear systems [30], [31], [32]. It has been applied to control of a wide variety of nonlinear dynamic systems especially flight controllers [33], [34], [35]. Many successful results in advanced flight control systems have been achieved by Calise and others [13], [24], [36], [37]. In flight control problems, the applications of ANNs can also be found in Refs. [2], [38], [39]and [40]. These control methods have been successfully applied to fighter aircraft [13], [41], unmanned aircraft control [42], [43], helicopter control [44], reusable launch vehicles [8] and in advanced missile guidance techniques [45], [46], [47], [48]. The neural adaptive control system has been successfully demonstrated in high-fidelity simulations of both fixed-wing [49] and rotary-wing aircraft [44]. They are also used in development of fault-tolerant flight control systems for civilian transport aircraft [50]. The ANN approach is used for the F/A-18 aircraft in simulation in [13] and further developed in [26]. Leitner et al [26] designed an online adaptive NN for use in a nonlinear helicopter flight controller. In [26] the network helped the system tracking performance in the face of significant modeling errors. This approach was later modified and used in the Reconfigurable Control for Tailless Fighters (RESTORE) program [42], [43], using a dynamic inversion control law in an explicit model following architecture. The reconfigurable control law on X-36 tailless fighter aircraft was examined for the adaptation of unknown failures and damage in [51]. This same approach also has been applied and flown on the Joint Direct Attack Munition (JDAM) [45], [46], [47], [52] and [53], in which the LQR based flight control system was replaced with a dynamic inversion based scheme augmented with a NN based model reference adaptive control.

In [54] wing rock dynamics of an aircraft at moderate high angles of attack was controlled successfully via NN based adaptive control. The results in [54] show that the Single Hidden Layer Neural Network (SHL NN) adapts much more rapidly than the Linear in the Parameters Sigma-Pi Neural Network (LIP NN) in command tracking, despite having far fewer neurons, and both NN based designs significantly outperform the classical adaptive controller in both regulation and tracking. Narendra and Parthasarathy et al [55] view NNs as highly nonlinear control

elements that offer distinct advantages over conventional linear parameter adaptive controllers in achieving desired system performance.

Lewis et al [56] discussed an online NN that approximates unknown functions and is used in controlling the plant. Krstic et al [57], [58] have developed a technique for the control of feedback linearizable systems with input unmodeled dynamics. A robust adaptive control methodology that uses SHL feed-forward NNs has been presented by McFarland et al [59]. Balakrishnan and Huang [60] developed a Lyapunov equation based theory for robust stability of systems in the presence of uncertainties.

Lian et al [33] proposed an adaptive robust BTT autopilot design to treat the uncertainties efficiently without prior knowledge of the bounds on the uncertainties. Huang and Lin [61] applied sliding mode control to cope with model uncertainty of the BTT missile autopilot design. However, these schemes require a tedious design procedure to perform input/output feedback linearization [28]. Feed-forward NNs with sigmoid hidden units are analyzed in detail for BTT autopilot design [28]. This scheme combines NNs and the sliding-mode control technique.

In [62], the authors applied ANNs to design a pitch-plane autopilot for a medium-range air-to-air missile maneuvering at an unrealistically high angle of attack. An early application of this theory to the missile autopilot design problem is found in [63]. The preliminary study presented in [62] indicated that NNs are capable of attaining sufficiently high learning rates to make adaptation feasible even during the most demanding aerial engagements. McFarland et al [9] used NN implemented controller to enable a single controller to handle multiple versions of guided munitions. In [4] and [27], McFarland et al used feed-forward NNs implemented approximate inversion with one sigmoidal hidden layer for an agile antiair-missile autopilot and demonstrated the effectiveness of the resulting autopilot. Their online learning and functional approximation capabilities and ideal structures for parallel processing [64], [65] make NNs an excellent candidate for uncertain aerodynamic effects and replacement of missile gain tables. Ref. [66] is an example of a missile autopilot using NNs. In [4], McFarland et al claimed that robust, nonlinear, and NN

based control algorithms are suitable for use in agile missile flight control. The success of dynamic model inversion on missile autopilot design was shown in [67].

A short-coming of dynamic model inversion is that it can not be applied directly to non-minimum phase plants. The transfer function from the control surfaces to acceleration (at the center of gravity (Cg)) is mostly non-minimum phase for tail-mounted surfaces [9]. The tail-controlled airframe has a tail normal force opposite to the direction of the desired maneuver acceleration, which causes small initial airframe acceleration in the wrong direction. Analytically, this effect manifests as a right-half-plane zero in the transfer function from the control surface deflections (δ_e and δ_r) to the accelerations normal to the missile's longitudinal axis (a_z and a_y) at the missile Cg, thus tending to limit the speed of response of the guidance system [68]. Different approaches are proposed in [67], [69], [70], [71] and attempted to eliminate the non-minimum phase characteristics of the plants. A method which is described in [67] is called output redefinition and it is an example to solve non-minimum phase problems. "Output redefinition" is originally proposed in [72], the inner loop variable is defined as a linear combination of the state variables. This allows the designer to place the zero of the associated transfer function at a desirable location. Thus, for instance, a combination of both angle-of-attack and pitch rate could be used to define the commanded inner loop variable [9]. Since the non-minimum phase characteristic prevents accurate dynamic inversion, the problem is eliminated by controlling attitude instead of acceleration.

There is a research interests in improving adaptive NNs in the field of advanced flight control system design. Most of the previous NN training laws have NN weight dynamics that are of low rank, nearly unity [12]. There is no known reason why these dynamics could not be full rank [73]. A new adaptive NN control concept is proposed in [14], [73] , and [74] which learns the plant dynamics by an online trained NN and augments the capability of background learning (BL) using previously recorded data to improve system performance. In this concept, online learning architectures are used to compensate for dynamic inversion error caused by system uncertainties and uncertain environment and BL overcomes the low rank of most NN training approaches resulting in faster adaptation to the unknown

dynamics [14]. Furthermore, using combined online and BL methods provides long term learning in the adaptive flight controller, which enhances performance of the controller when it encounters a maneuver that has been performed in the past [14]. The BL law is a projection of the current learning law into the null space of the current learning (LIP NN [14] and SHL NN [74]). The network update law uses both the current (i.e. online) information and stored (i.e. background) information. The resulting adaptive element is able to retain long term memory without compromising its responsiveness to sudden changes in the environment [74].

In this study, the traditional gain scheduling [2] based control architecture is replaced by online learning NN architectures trained with a LIP NN, SHL NN, BL implemented SHL NN. The main advantage of these approaches is that online learning neural controllers will replace gain-scheduling activities, which can potentially be very large [2]. They can account for non-linearity and uncertainty both in the controls as well as in the states. Moreover, it decreases the dependency on accurate aerodynamic models. This control architecture requires only an approximate linear model at a single operating point. Therefore, at the beginning of this study, the aerodynamic coefficients of the cruise missile are obtained by defining the geometry and using empirical tools. Next, a plant model is obtained by substituting the aerodynamic coefficients into the equations of motion. Then they are linearized at a single flight condition (i.e. trim condition). As a result, the linear model is obtained from a priori knowledge of the missile dynamics. Next, dynamic model inversion control is applied. Since the missile is dynamically a nonlinear system, operation at any other flight condition will result in an inversion error. Then, an online ANN is added to compensate for the inversion error [25]. The weights of these NNs are adjusted to eliminate the effect of the modeling error [10]. The network weight update law ensures boundedness of both tracking error and adaptive NN weights [9], [13], [27]. Many of the results for this approach involving ANNs are derived from Lyapunov stability theory [75]. For a complete proof of stability, the reader may refer to [76] and [77]. According to the Lyapunov theory based proof the signals including tracking errors are uniformly bounded and the

weights of the online NNs that are initialized at zero tend to constant values [13], [25], [38], [39].

Modeling errors may mainly come from linearization, approximate inversion, aerodynamic modeling, thrust modeling, un-modeled dynamics (imperfect modeling) and assumptions made during controller design or sudden changes in missile dynamics. First, the missile generally has highly nonlinear dynamics and it is difficult to accurately model. There are inconsistencies between the actual dynamics and its mathematical model. During the simulation, the missile confronts with different flight conditions in the flight envelope, operation at any other flight condition will result in an inversion error. Second, highly nonlinear aerodynamic is difficult to model, aerodynamic coefficients are nonlinear functions of several physical variables, the aerodynamic coefficients obtained from tabulated results and interpolation can not represent actual aerodynamic condition. In addition, the elements of linearized matrices includes these aerodynamic coefficients, thus dynamic model inversion using these matrices can not be exact. This will cause an inversion error. Fourth, thrust force changes nonlinearly with respect to altitude. Changes in missile altitude and variation in dynamic pressure will be an error source potentially. Reduced order modeling can cause errors as well. The inconsistencies between assumptions and actual case can also result in modeling errors. Transformation of aerodynamic parameters, which are the elements of matrices, from inertial frame to body frame can also cause errors. Therefore, online NNs (longitudinal and directional channels) are designed to eliminate these errors. They are used to cancel errors by observing the tracking error between the commanded input to the missile and missile response. These errors are used to update the network. Therefore, the NN adaptation attempts to cancel the inversion error.

The performance of the controller is demonstrated using the nonlinear 5-DOF high subsonic cruise missile simulation code in MATLAB-SIMULINK.

1.2. Scope

"Adaptive Neural Network Applications on Missile Controller Design" is a topic of research interest. In this thesis, various NN based control techniques are studied on generic cruise missile geometry with the physical characteristics shown in Table 1-1.

The missile is a skid-to-turn (STT) missile and it is a cruciform missile with a plus tail configuration. Banking of the missile is not required to maneuver immediately in any plane. Sets of controls at right angles permit the missile to turn immediately in any plane without the necessity of its banking (Φ=0) [68], [78]. The autopilot axes are located in the planes of the control surfaces in this type of missile, so that only two surfaces are deflected by the pitch autopilot (δ_e) and two by the yaw autopilot (δ_r) [68].

Table 1-1 GENERIC CRUISE MISSILE CHARACTERISTICS

Length	4 m
Diameter	0.4 m
Wing/Tail Span	0.9 m
Control	Tail Control
Cruise Speed	0.8 Mach
Launch mass	500 kg
Fuel weight	50 kg
Center of gravity location	2 m (from nose)

The ANN augmented model inversion control is a viable candidate for a missile autopilot implementation. This approach is applied in designing a missile altitude hold autopilot and directional autopilot. Attitude Command Attitude Hold (ACAH) response type is used in the autopilots. Their effectiveness is illustrated using numerical results from nonlinear 5-DOF simulation studies of a high subsonic cruise missile. Simulation results of a linearized model inversion controller without NN, and distinct online learning neural networks (LIP NN, SHL NN and BL implemented SHL NN) applied adaptive controllers for longitudinal (pitch) and directional (yaw) channel are shown, and the effectiveness of on-line learning neural controllers for on-line control laws in a high subsonic cruise missile are compared and evaluated. Simulations are performed, but two of them have been chosen, which are related to the scope of this thesis. The objective is to present results between model inversion controllers proposed in [2] and on-line learning NN implemented model inversion controllers on missile autopilot applications. The capability of adapting to errors caused by the linearized inverted model is demonstrated for different types of pitch and yaw NN. In the first simulation scenario, the aim is comparison of the LIP NN and SHL NN on the improvement of the inversion error compensation. In the second simulation scenario, the scope is to show the improvement of SHL NN implementation with aiding BL algorithm in its weight adaptation process.

It is shown that neural-based control laws for the autopilot functions do not need any gain scheduling in nonlinear 5-DOF simulation.

To deal with problems encountered during the implementation phase of the algorithm some improvements are made such as utilizing learning rate for the computation of NN compensation to nonlinearity. Moreover, some precautions are taken such as checking the trajectory of the missile to validate them.

1.3. Outline

The remaining part of the thesis is organized as follows:

Chapter-2 discusses system modeling that includes missile guidance planning, aerodynamic modeling, thrust modeling and missile nonlinear 5-DOF mathematical model with given assumptions, altitude hold guidance and the directional guidance concepts, and the architecture of 5-DOF simulation.

Chapter-3 describes the main steps for the development and implementation of LIP NN and SHL NN based altitude hold controller, and its architecture.

Chapter-4 demonstrates the main steps for the development and implementation of LIP NN and SHL NN based directional autopilot, and architecture of heading hold controller.

Chapter-5 contains a description of BL implemented NN adaptation method. The improvement related to this new method is also described in this chapter with the comparison of the previous application.

Chapter-6 provides numerical simulation results for a high subsonic cruise missile demonstrating the feasibility of this autopilot design techniques. The performance of each controller are also compared and evaluated in this chapter.

Finally, conclusions about this thesis, the problems of the controller architecture and future research directions are discussed in Chapter-7.

CHAPTER 2

MISSILE FLIGHT DYNAMICS MODELING

Five degrees of freedom (5-DOF) Matlab-Simulink model is created to demonstrate the effectiveness of the controllers. It includes only directional and longitudinal motion of the missile, and the roll motion in the lateral plane is ignored.

5-DOF simulation implies flight model of a missile that can be described by a number of nonlinear first order differential equations. These equations can be analyzed in two groups: kinematics equations and dynamic equations. In this chapter, the sub models of the missile flight model will be described briefly according to assumptions described in Section 2.1 and simulation architecture of 5-DOF equations of motion will be given.

2.1 Assumptions

The simulation uses nonlinear aerodynamics and uses the following assumptions:

1) Missile is a rigid body; therefore aeroelastic effects are not included in the equations. The missile body does not change in size or shape. The forces acting between individual elements of mass are eliminated.

2) I_{xx}, I_{yy}, I_{xz} are constant for a given rigid body due to usual symmetry of missile about the x-y plane.

3) The missile has a vertical plane of symmetry about the x-z plane. The y-axis is the perpendicular to this plane of symmetry and is a principal axis. ($I_{xy}=I_{yz}=0$)

4) The missile is a cruciform missile with rotational symmetry ($I_{yy}=I_{zz}$ and $I_{xz}=0$). (Cruciform missiles commonly have four similar fixed wing panels and four movable tail fins mounted together at a common chord and displaced one from the next by $\pi/2$ radians of arc [68].)

5) The missile is a skid-to-turn (STT) missile and it has a plus tail configuration. Banking of the missile is not required to maneuver immediately in any plane ($\Phi=0$). Roll moments can be neglected ($L\approx 0$).

6) Earth rotation is neglected. Related centrifugal and Coriolis accelerations can be neglected.

7) The earth is treated as flat and stationary in inertial space. This allows us to declare the Earth frame as the inertial frame.

8) The missile has an aerodynamic symmetry in roll. Therefore the aerodynamic forces and moments acting on the missile are assumed to be invariant with the roll position of the missile relative to the free-stream velocity vector.

9) The attitudes (q, r) and altitude (h) of the missile which is measured by gyroscope and altimeter sensors respectively located on the missile that the input axes of the sensors are coincident with the missile body axes. To keep the simulation simple, higher order sensor dynamics and sensor errors are neglected.

10) Thrust is generated along the x-axis of the missile body and thus thrust forces pass through the Cg. ($T_y=T_z=0$)

11) There is no wind during the flight.

With these assumptions, the airframe motion can be described by translations of the center of gravity and by rotations about this point. In addition, they simplify the equations of motion by eliminating the aerodynamic cross-coupling terms between the roll motion and the pitch and yaw motions.

2.2 5-DOF Dynamic Model

This model contains the dynamic equations of missile. These equations are derived by applying Newton's laws of motion that relate the summation of the external forces and moments to the linear and angular accelerations of the body. First, a typical six degrees of freedom (6-DOF) missile model is handled. 6-DOF means longitudinal, lateral, vertical movements and attitude of the motion which are presented at Figure 2.1.

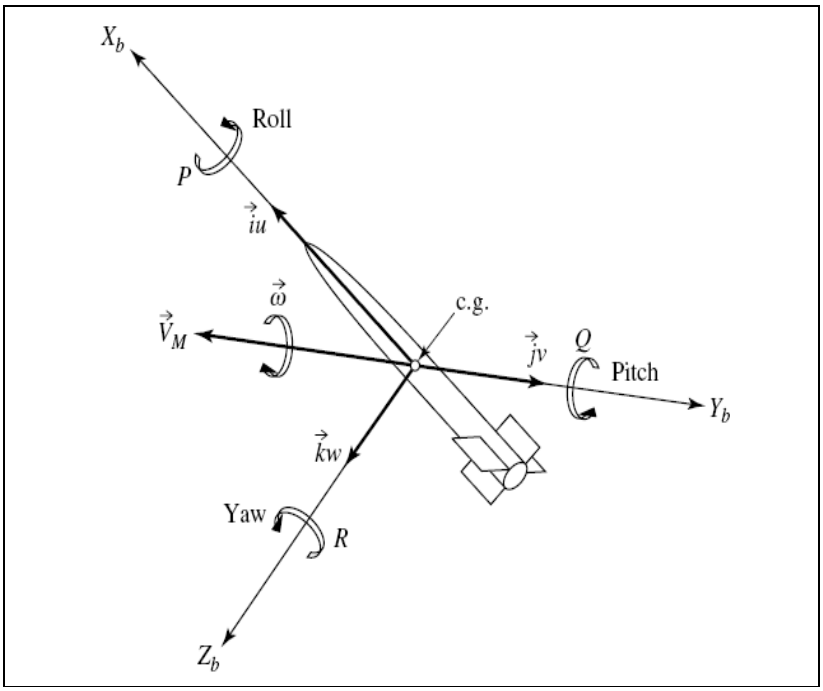

Figure 2.1 Representation of the Missile's Six Degrees of Freedom [68]

The 6-DOF equations of motion consist of three translations, and three rotations, usually along and about the missile body axes. In this study, the roll rate (p) is set to zero in the simulation model. The missile initial roll angle (Φ) and initial roll rate (p) are 0. Since the missile is a cruciform missile and it has a plus tail configuration, sets of controls at right angles permit the missile to turn immediately in any plane without the necessity of banking (skid-to-turn missile Φ=0) [68], [78]. From assumptions 4 and 5, roll moment coefficient is set to zero and I_{yy} equals to I_{zz}, thus roll acceleration equals to zero. This can be realized from equation (2.7) in Section 2.2.2. Therefore roll motion is ignored, and the simulation is called as 5-DOF.

The translational equations of motion are obtained from Newton's second law. Newton states that the summation of all external forces acting on a body is equal to the time rate of the momentum of the body with respect to the inertial frame. The rotational equations of motion are derived from Euler's law, i.e. the time rate of change of angular momentum equals to the summation of externally applied moments. In deriving these equations of motion, the assumptions which were described in Section 2.1 were used.

The relationships between the forces/moments acting on the missile and the kinematics state (position, velocity, attitude, angular velocity) of the missile were established in 5-DOF. Since the missile is subject to the aerodynamic forces, thrust forces and the gravitational acceleration (g) during flight, we can conclude that the external forces and moments acting on a missile are generated by the aerodynamic effects, propulsion and the gravity. As the results of these effects, the components of the inertial position vector were changed on longitudinal, lateral and directional plane. Moreover, yaw and pitch attitudes were also changed.

2.2.1 Translational Motion

Firstly, the translation of a rigid body can be expressed mathematically by the following equations:

$$\sum F = m \cdot \frac{dV_{missile}}{dt}\Bigg]_{inertial} = m \cdot \left[\frac{dV_{missile}}{dt}\right]_{body} + m \cdot \left(w_{body} \times V_{missile}\right) \tag{2.1}$$

$$\begin{aligned} \dot{u} &= \frac{\sum F_x}{m} - qw + rv \\ \dot{v} &= \frac{\sum F_y}{m} - ru + pw \\ \dot{w} &= \frac{\sum F_z}{m} - pv + qu \end{aligned} \tag{2.2}$$

Total external forces consist of aerodynamic forces, thrust and the gravitational forces.

$$\begin{aligned} \sum F_x &= X + \frac{T_x}{m} - g\sin\theta & X &= C_X \frac{Q\ Sref}{m} \\ \sum F_y &= Y + \frac{T_y}{m} + g\sin\phi\cos\theta \quad \Rightarrow & Y &= C_Y \frac{Q\ Sref}{m} \\ \sum F_z &= Z + \frac{T_z}{m} + g\cos\phi\cos\theta & Z &= C_Z \frac{Q\ Sref}{m} \end{aligned} \tag{2.3}$$

In equations (2.3), the first term is related to aerodynamic forces, second term is related to thrust forces, and the third term is related to gravitational forces. Then equations (2.3) are substituted into the first terms of equations (2.2) and the final translational motion equations (2.4) are obtained. Substituting for the aerodynamic forces and using previously stated assumptions, the following is obtained.

$$\begin{aligned} \dot{u} &= C_X \frac{Q\ Sref}{m} + \frac{T_x}{m} - g\ \sin\ \theta - qw + rv \\ \dot{v} &= C_Y \frac{Q\ Sref}{m} + g\ \sin\ \phi\ \cos\ \theta - ru + pw \\ \dot{w} &= C_Z \frac{Q\ Sref}{m} + g\ \cos\ \phi\ \cos\ \theta - pv + qu \end{aligned} \tag{2.4}$$

2.2.2 Rotational Motion

The rotation of a rigid body is expressed mathematically by the following equations:

$$\sum M = \left(\frac{dH}{dt}\right)_{inertial} + w_{body} \times H_{inertial} \tag{2.5}$$

$$\begin{aligned} \dot{p} &= \frac{\sum L}{I_{xx}} + qr\,\frac{(I_{yy} - I_{zz})}{I_{xx}} \\ \dot{q} &= \frac{\sum M}{I_{yy}} + pr\,\frac{(I_{zz} - I_{xx})}{I_{yy}} \\ \dot{r} &= \frac{\sum N}{I_{zz}} + pq\,\frac{(I_{xx} - I_{yy})}{I_{zz}} \end{aligned} \quad \Rightarrow \quad \begin{aligned} \sum L &= C_L\,\frac{Q\ Sref\ \ lref}{I_{xx}} \\ \sum M &= C_M\,\frac{Q\ Sref\ \ lref}{I_{yy}} \\ \sum N &= C_N\,\frac{Q\ Sref\ \ lref}{I_{zz}} \end{aligned} \tag{2.6}$$

Then, moments due to aerodynamic forces on missile are substituted into equations (2.6) and the final rotational motion equations (2.7) are obtained. According to assumption 10, there is no moment arm; therefore moments caused by propulsive forces are zero about the Cg. The details of the Thrust Model will be given in Section 2.5. Since gravitational forces apply at the Cg, it has no moment contribution. Substituting for the aerodynamic moments and using previously stated assumptions, the following is obtained.

$$\begin{aligned} \dot{p} &= C_L\,\frac{Q\ \ Sref\ \ \ lref}{I_{xx}} + qr\,\frac{(I_{yy} - I_{zz})}{I_{xx}} = 0 \\ \dot{q} &= C_M\,\frac{Q\ \ Sref\ \ \ lref}{I_{yy}} + pr\,\frac{(I_{zz} - I_{xx})}{I_{yy}} \\ \dot{r} &= C_N\,\frac{Q\ \ Sref\ \ \ lref}{I_{zz}} + pq\,\frac{(I_{xx} - I_{yy})}{I_{zz}} \end{aligned} \tag{2.7}$$

The reader may refer to [68] and [79] for derivation of the translational and rotational equation in detail.

2.3 Aerodynamic Model

The motion of the air around a missile generates pressure and velocity variations, which produce aerodynamic forces and moments. One of the fundamental problems in flight mechanics is the mathematical modeling of the aerodynamic forces and moments. In the previous section the relationships between aerodynamic forces and translational motion and the relationships between aerodynamic moments and rotational motion were shown. The calculation of aerodynamic forces and moments acting on a missile and aerodynamic coefficients is explained next.

2.3.1 Aerodynamic Parameters

Aerodynamic coefficients are the functions of the flight parameters. These parameters are angle of attack (α), side slip angle (β), the control surface deflections (δ_e and δ_r), the body angular rates (q and r), the time rates of α and β in addition to Missile Mach Number (M) in this study.

Angle of Attack (α): The angle between the x-axis and the projection of the missile velocity vector on the x-z (reference) plane. It is positive when the missile velocity component along the z-axis is negative. Downward direction is taken as positive z-axis.

$$\alpha = \arctan\left(\frac{w}{u}\right) \tag{2.8}$$

Angle of Sideslip (β): The angle with which the missile velocity vector makes with the x-z (reference) plane of the missile. Rightward direction is taken as positive y-axis when the missile is looked from the top.

$$\beta = \arctan\left(\frac{v}{V_{missile}}\right) \tag{2.9}$$

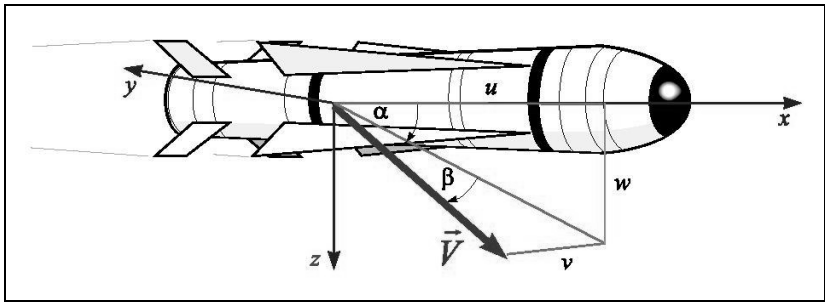

Figure 2.2 Demonstrations of Angle of Attack and Side-Slip Angle [80]

Control Surface Deflections (δ_e and δ_r): These deflections are provided by the control surfaces and are defined in terms of the fin deflections in the following manner:

$$\delta_e = \frac{\delta_2 - \delta_4}{2}$$
$$\delta_r = \frac{\delta_1 - \delta_3}{2} \qquad (2.10)$$

In equations (2.10) δ_i denotes the deflection of control fin i for i=1, 2, 3 and 4. The fin arrangement of the considered cruciform missile is given in Figure 2.3 with the positive rotation senses of the fins. Since the missile is cruciform and has a plus tail configuration, elevator deflection does not have any contribution on missile directional plane and rudder deflection does not have any contribution on missile longitudinal plane. The graphs related to this issue will be given in Appendix A.

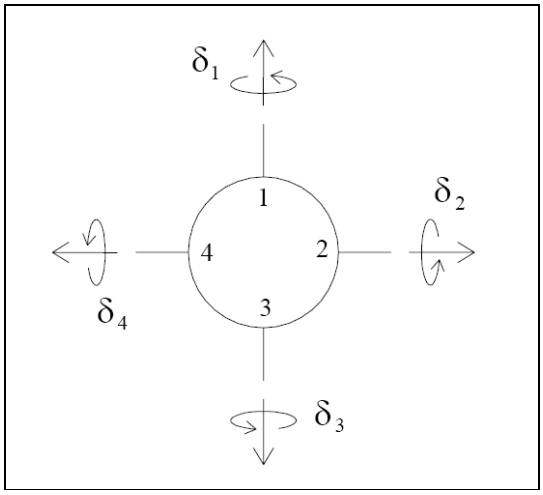

Figure 2.3 Arrangement of Fin from the Rear View of the Missile [80]

Mach number (M): The ratio of the airspeed to the speed of sound. If a is the speed of the sound, then Mach number can be expressed as:

$$\mathrm{M} = \frac{V_{missile}}{a} \tag{2.11}$$

where $V_{missile} = \sqrt{u^2 + v^2 + w^2}$ and a can be stated as $a = \sqrt{\gamma RT}$ where R is the universal air gas constant, T is the ambient temperature which changes with altitude, and γ is the specific heat ratio of the air.

Dynamic Pressure (Q): Pressure on missile created due to dynamic motion.

$$Q = \frac{1}{2}\rho V_{missile}^{\ 2} \tag{2.12}$$

Air Density (ρ): It is the mass of air per unit volume.

It should be noted that air density, ambient pressure, ambient temperature, speed of sound parameters are calculated according to the U.S. Standard Atmosphere 1976 model.

2.3.2 Aerodynamic Force and Moment Coefficients

In order to calculate aerodynamic forces and moments' contribution in equation (2.4) and (2.7), we need the aerodynamic coefficients. These coefficients are found using the Missile DATCOM software [81]. This software finds the combined effects of many different variables on the magnitude of aerodynamic coefficients. These variables are configuration geometry, angle of attack, missile size, free-stream velocity, density of the undisturbed air, Reynolds number (i.e., as it relates to viscous effects), and Mach number (i.e., as it relates to compressibility effects) [68]. Then force and moment coefficients table are created. The relationships between aerodynamic variables and coefficients will be shown on graphs in Appendix A. Force and moment coefficients can be found from these lookup tables that are created from sideslip angle, rudder (yaw fin) deflection, elevator (pitch fin) deflection, Mach number, angle of attack respectively. In addition, dynamic derivatives of aerodynamic coefficients table are also generated according to Mach number. Finally, functional forms of these coefficients are generated.

The derivation of non-dimensional form of aerodynamic forces and moment coefficients can be found as follows:

$$
\begin{aligned}
C_X &= C_x(\beta, \delta_r, \delta_e, \mathrm{M}, \alpha) \\
C_Y &= C_y(\beta, \delta_r, \mathrm{M}) + C_{yr}(\mathrm{M}) \frac{r\, lref}{2V_{missile}} \\
C_Z &= C_z(\delta_e, \mathrm{M}, \alpha) + C_{zq}(\mathrm{M}) \frac{q\, lref}{2V_{missile}} \\
C_M &= C_m(\delta_e, \mathrm{M}, \alpha) + C_{mq}(\mathrm{M}) \frac{q\, lref}{2V_{missile}} + C_{m\dot{\alpha}}(\mathrm{M}) \frac{\dot{\alpha}\, lref}{2V_{missile}} - C_Z \frac{X_{cg}}{lref} \\
C_N &= C_n(\beta, \delta_r, \mathrm{M}) + C_{nr}(\mathrm{M}) \frac{r\, lref}{2V_{missile}} + C_{n\dot{\beta}}(\mathrm{M}) \frac{\dot{\beta}\, lref}{2V_{missile}} + C_Y \frac{X_{cg}}{lref}
\end{aligned}
\tag{2.13}
$$

Aerodynamic coefficients are expressed in (2.13) as linear functions of angle of attack (α), side-slip angle (β), fin deflections (δ_e, δ_r), missile Mach number (M), body angular rates (q and r), the time rates of α and β. Note that the roll motion is neglected. Therefore C_L is set to zero.

2.4 Kinematics Model

Kinematics deals with the motion of bodies without reference to force or mass [82]. Kinematics equations are the results of transformation matrix applications that form a relationship between the reference axis systems using Euler angles [79]. These angles (Φ, θ, ψ) are defined as the missile's attitude with respect to the Earth-fixed frame in Figure 2.4.

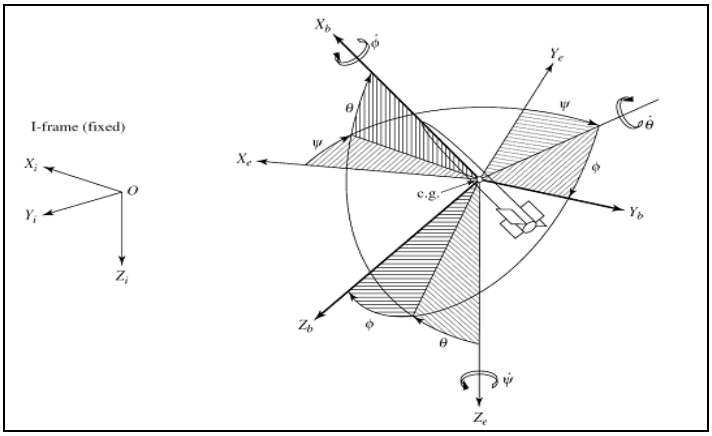

Figure 2.4 Representation of the Kinematics Relation for the Missile [68]

In this study, two orthogonal-axes systems need to be defined to develop the kinematics model of the missile. They are as follows:

1) Earth-fixed frame (i.e. the inertial frame) (Assumptions 6 and 7) is fixed in space, and Newton's Laws of Motion are valid on this frame. In Figure 2.5, we will denote the Earth-fixed frame by (X_e, Y_e, Z_e). In this right-handed coordinate system (NED system), the $X_e - Y_e$ lie in the horizontal plane and the Z_e-axis points down vertically in the direction of gravity. It should be noted that the position of the missile's Cg at any instant of time is given in this coordinate system.

2) The conventional body frame is selected, denoted by (X_b, Y_b, Z_b). It moves with the missile. The center of this frame is at the Cg of the missile, the body coordinate system in this study is a right-handed system with the positive X_b-axis along the missile's longitudinal axis (missile's center line), the Y_b-axis positive to the right in the horizontal plane which is designated as the pitch axis and the Z_b-axis positive down which is the yaw axis. This coordinate frame is similar to the north-east-down frame.

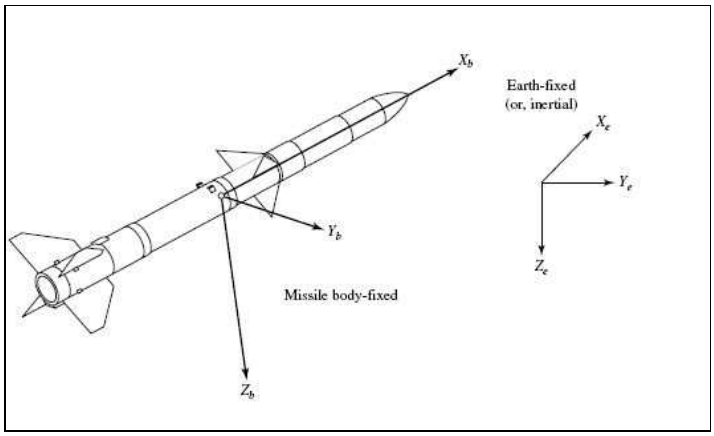

Figure 2.5 Orientation of the missile axes with respect to the Earth-fixed axes [68]

The guided weapons reference axis system is generally centered on the Cg and fixed in the body. Therefore, in this study, the kinematics model is involved in the transformation of the translational and rotational motion equations that described in Section 2.2 from body frame to inertial reference frame. This model calculates body attitudes in the form of Euler angles (Φ, θ, ψ) and the transformation matrices in order to achieve these transformations.

For translational motion:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = T_{I/B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2.14}$$

where $T_{I/B}$ is the transpose of direction cosine matrix that converts body velocities into inertial velocities.

$$T_{I/B} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (2.15)$$

Substituting equation (2.15) into equation (2.14), the following is obtained.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.16)$$

<u>For rotational motion:</u>

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = R_{I/B} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.17)$$

where $R_{I/B}$ is the rate transformation matrix that converts body angular rates into Euler rates.

$$R_{I/B} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \quad (2.18)$$

Substituting equation (2.18) into equation (2.17), the following is obtained.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.19)$$

After these calculations, Euler angles of the missile are obtained by the integration of the Euler rates. For calculating the inertial position, inertial velocities are

integrated. As a result, the missile attitude and location of its Cg are calculated with respect to an Earth (inertial) reference point (i.e. launch point).

2.5 Thrust Model

The thrust model is used during the maneuver starting from the firing phase until the termination phase. It is divided into two successive phases:

1) Boost Phase: It comprises the flight from the firing instant to the end of the booster thrust. In this phase, boost motor provides 40000 N thrust for 4 seconds in order to achieve cruise speed, and then it finishes. During this period, there is no tail fin deflection.

2) Sustain Phase: It comprises the flight from the firing instant of the flight motor to the termination instant. The Thrust Model, which is described in this section, explains this phase.

The Thrust Model is similar to the model described in [1]. Here the model is assumed to resemble a turbojet engine with a 5000 N static thrust. The Turbojet engine shall operate throughout the flight in order to sustain the required missile speed.

This engine will be assumed to have a maximum thrust of 5000 N at sea level. The variation of available maximum thrust with respect to altitude can be represented by equation (2.15).

$$T(h) = T_{MAX} \cdot \left(0.0021 \cdot h^2 - 0.0797 \cdot h + 1\right) \quad for \quad 0 \le h \le 5000\, m \tag{2.20}$$

where

T_{MAX} : Maximum available thrust at sea level (5000 N)

h : Altitude, [km]

Equation (2.20) is derived from [83] which gives information about typical turbojet engine characteristics. In order to maintain the pre-determined cruise speed of 0.8 M, magnitude of the thrust is adjusted by the method described below.

1) Turbojet engine is assumed to be capable of increasing and decreasing its thrust level linearly at a rate of 300 N/s.

2) Although, the engine is assumed to have a 300 N/s thrust increment rate capability, the thrust control command will be calculated according to the measured Mach number. If Mach number is below 0.75 or greater than 0.85, engine is commanded to work at its limits (300 N/s). If Mach number is between 0.75 and 0.85, a quadratic function is used to calculate the thrust control gain, K. The thrust control law is given below:

$$T_{COM\ i+1} = T_{COM\ i} + \Delta t \cdot K \tag{2.21}$$

where

T_{COM} : Thrust control command

Δt : Time step

K : Thrust control gain

And K is defined by the following:

Mach < 0.75 => K = 300 N/s

0.75 < Mach < 0.85 => $K = sign(\Delta M) \cdot \Delta M^2 \cdot 120000$ (2.22)

Mach > 0.85 => K = -300 N/s

The thrust control command, T_{COM} is limited to a minimum value of 1000 N for all altitudes in order to avoid sudden decreases and oscillations in the missile speed.

2.6 Guidance Model

The Guidance Model is used to command the required movement of the cruise missile according to the target trajectories and planned waypoints. In other words, inputs for an autopilot system are generated from this model.

2.6.1 Altitude Hold Guidance

During cruise phase, the difference between commanded altitude and missile altitude is given according to a planned height trajectory. After cruise phase (terminal phase), the difference between the height of the target and missile altitude is the input for the altitude hold autopilot. It is shown in Figure 2.6.

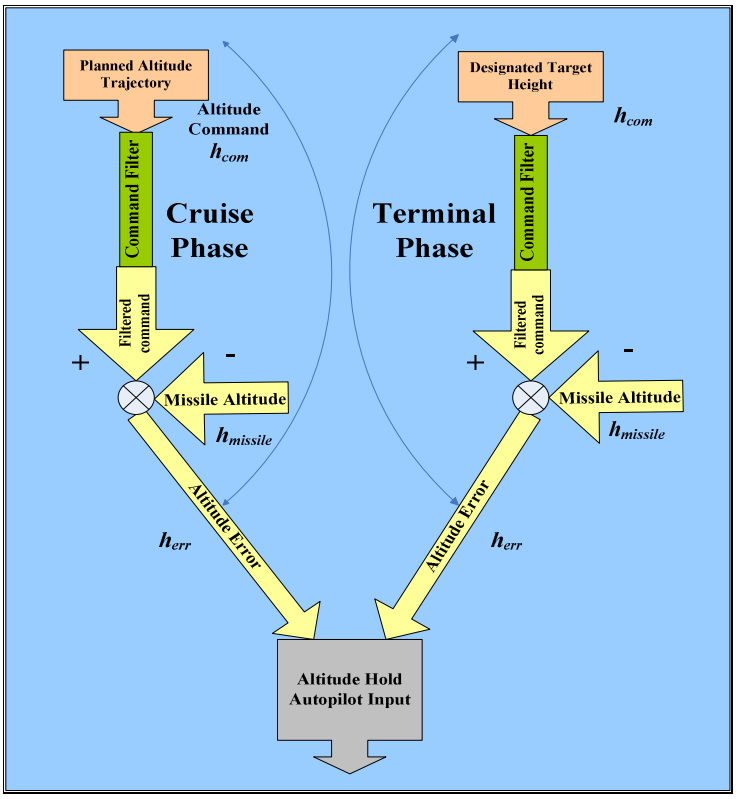

Figure 2.6 Generation of Altitude Hold Autopilot Command

$$h_{err} = \left(h_{com} \cdot CF(s)\right) - h_{missile} \tag{2.23}$$

where $CF(s) = \frac{1}{3s+1}$ the command filter transfer function and h_{err} is the Altitude Hold Autopilot Input.

2.6.2 Directional Guidance

Normally, autopilot inputs are determined by the seeker and guidance systems. However, in this study there is no seeker model. Since the target used in this model is assumed slow, the motion in the longitudinal axis is neglected. Therefore the requirement for the seeker detection is only in the directional plane. Therefore instead of making a seeker model, it was assumed that the way points are tracked ideally via Inertial Navigation Systems (INS) during cruise phase and the LOS rates of the target are provided perfectly in terminal phase of the flight. These LOS rates are used in Guidance Model to obtain directional autopilot command.

The generation of the directional autopilot command is based on the LOS rate following reference [68], [84]:

1) Cruise Phase

Commanded Waypoint Position	$= [X_{wp}, Y_{wp}, 0]$	$= WP_p$
Missile Inertial Position	$= [X_m, Y_m, 0]$	$= M_p$
Commanded Waypoint Velocity	$= [0,0,0]$	$= V_{WP}$
Missile Inertial Velocity	$= [\dot{X}_m, \dot{Y}_m, 0]$	$= V_M$

$R_{I/B}$ is the rate transformation matrix that converts body angular rates into Euler rates which is defined in Section 2.4.

$$\begin{bmatrix} \dot{\lambda}_{roll} \\ \dot{\lambda}_{pitch} \\ \dot{\lambda}_{yaw} \end{bmatrix} = R_{I/B} \frac{(WP_p - M_p) \times (V_{WP} - V_M)}{(WP_p - M_p) \cdot (WP_p - M_p)} \quad \Rightarrow \quad LOS \ \ rates \tag{2.24}$$

Equation (2.24) is used for the cruise phase in order to find the LOS (line-of-sight) rate on the directional plane ($\dot{\lambda}_{yaw}$) which is the third term of the LOS rate vector.

2) Terminal Phase

$$\begin{aligned}
&\text{Target Inertial Position} &&= [\mathrm{X_T}, \mathrm{Y_T}, \mathrm{Z_T}] &&= T_p \\
&\text{Missile Inertial Position} &&= [\mathrm{X_m}, \mathrm{Y_m}, \mathrm{Z_m}] &&= M_p \\
&\text{Target Inertial Velocity} &&= [\dot{\mathrm{X}}_\mathrm{T}, \dot{\mathrm{Y}}_\mathrm{T}, \dot{\mathrm{Z}}_\mathrm{T}] &&= V_T \\
&\text{Missile Inertial Velocity} &&= [\dot{\mathrm{X}}_\mathrm{m}, \dot{\mathrm{Y}}_\mathrm{m}, \dot{\mathrm{Z}}_\mathrm{m}] &&= V_M
\end{aligned}$$

$R_{I/B}$ is the rate transformation matrix that converts body angular rates into Euler rates which is defined in Section 2.4.

$$\begin{bmatrix} \dot{\lambda}_{roll} \\ \dot{\lambda}_{pitch} \\ \dot{\lambda}_{yaw} \end{bmatrix} = R_{I/B} \frac{(T_p - M_p) \times (V_T - V_M)}{(T_p - M_p) \cdot (T_p - M_p)} \quad \Rightarrow \quad LOS\ rates \tag{2.25}$$

Equation (2.25) is used for the terminal phase in order to find the LOS (line-of-sight) rate on the directional plane ($\dot{\lambda}_{yaw}$) which is the third term of the LOS rate vector.

Then from equation (2.26), heading command is calculated by using Proportional Navigation [68], [84]. If the missile is sustaining at a constant velocity and we are considering long times of flight it is reasonable to assume that a given change in body direction will result in the same change in flight path direction [85]. If the motion of the missile on directional plane is considered, missile yaw flight path angle rate can be accepted as its heading rate (equation (2.26)).

$$\begin{aligned}
&\dot{\gamma}_{yaw} = N_{guidance}\, \dot{\lambda}_{yaw} \\
&\dot{\psi}_{com} = \dot{\gamma}_{yaw} \quad \Rightarrow \psi_{com} = \int \dot{\psi}_{com} dt
\end{aligned} \tag{2.26}$$

$N_{guidance}$ is the guidance constant.

The block diagram of directional guidance algorithm is shown in Figure 2.7.

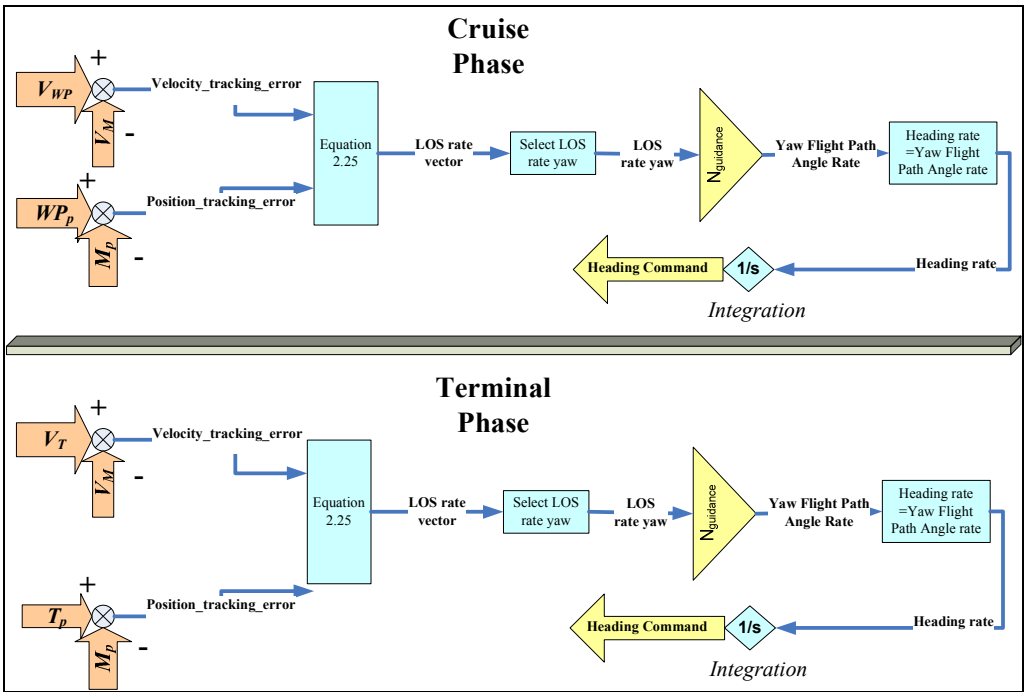

Figure 2.7 Generation of Directional Autopilot Command

2.7 Autopilot

Autopilot Model will be described in detail in Chapter 3, 4 and 5 respectively. This model generates required deflection angles which are the inputs for the missile plant.

2.8 Simulation Architecture

The mathematical architecture of the 5-DOF simulation is generated by the models described above. First, according to the initial conditions, Aerodynamic Parameters are calculated. Next body forces *X*, *Y*, *Z* and moments *M*, *N* are computed by the

help of the Aerodynamic Model. L is set to zero. In parallel, thrust forces are calculated from Thrust Model. Then these forces and moments are substituted into the Dynamic Model, and body translational accelerations and body angular accelerations are obtained. These accelerations are integrated for computing body angular rates and body velocities. Next, body angular rates are transformed into Euler rates and body velocities are converted to inertial velocities using the Kinematics Model. When they are integrated, Euler angles and location of the missile Cg with respect to inertial frame are found. Then transformation matrices that transform from inertial coordinates to body coordinates (DCM) are found by substituting Euler angles into Kinematics Model. According to the target trajectories and planned waypoints, guidance algorithms generate the autopilot commands. Autopilots Model, which will be described in the following sections, uses these commands in order to find fin deflections for desired missile motion. Next, dynamic parameters related to the motion of the missile are computed. Then by using these parameters and fin deflections, aerodynamic coefficients are calculated in the Aerodynamic Model. This cycle repeats at every time step of the simulation.

The flow chart of the simulation architecture is shown in Figure 2.8.

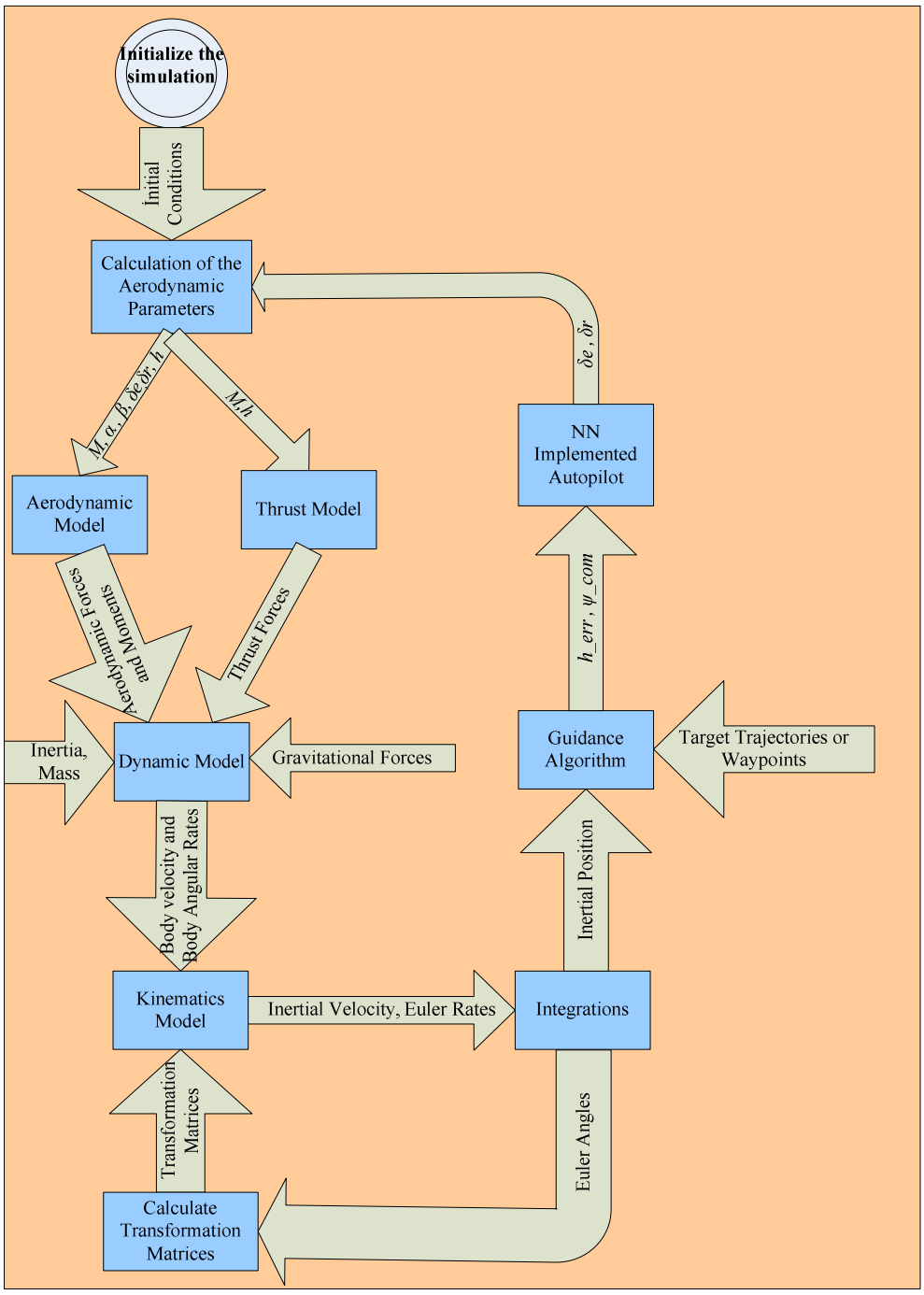

Figure 2.8 Simulation Architecture

CHAPTER 3

ADAPTIVE NEURAL NETWORK BASED ALTITUDE HOLD AUTOPILOT

The altitude hold autopilot allows a cruise missile to hold its altitude, climb or descend according to a planned altitude trajectory. After reaching the desired intermediate cruise or sea-skimming altitude, an altitude autopilot is used to maintain this altitude against atmospheric disturbances and gravity. The mission of the autopilot is to minimize the deviation between the actual altitude and the desired altitude [1].

In this chapter, the design steps of an altitude hold autopilot of a high subsonic cruise missile with NN architecture are explained. In Section 2.2, the missile dynamic model is described in detail. At this instant, the equations derived in this model are linearized at an equilibrium point (i.e. trim condition) under the assumptions in Section 2.1. Next, the linear state equations are obtained to use in the nonlinear model inversion controller. Finally, the ANNs based model inversion applied to the missile altitude hold capability is demonstrated. Note that the actuator dynamics is not modeled in this architecture, but the deflection and deflection rate is limited.

3.1 Linearization

In this section, the equations (2.7) are linearized under designated assumptions. We are interested in the motion on the longitudinal and directional plane, and we

neglect the roll motion. Therefore, the linearization of the equations related to pitch angular acceleration and yaw angular acceleration is performed.

First, by substituting related aerodynamic coefficients from equation (2.13) into equation (2.7), equation (3.1) is obtained.

$$\begin{aligned}\dot{q} &= \left(C_m(\delta_e, \mathrm{M}, \alpha) + C_{mq}(\mathrm{M}) \frac{q\, lref}{2V_{missile}} + C_{m\dot{\alpha}}(\mathrm{M}) \frac{\dot{\alpha}\, lref}{2V_{missile}} - C_Z \frac{X_{cg}}{lref} \right) \frac{Q\, Sref\, lref}{I_{yy}} + pr \frac{(I_{zz} - I_{xx})}{I_{yy}} \\ \dot{r} &= \left(C_n(\beta, \delta_r, \mathrm{M}) + C_{nr}(\mathrm{M}) \frac{r\, lref}{2V_{missile}} + C_{n\dot{\beta}}(\mathrm{M}) \frac{\dot{\beta}\, lref}{2V_{missile}} + C_Y \frac{X_{cg}}{lref} \right) \frac{Q\, Sref\, lref}{I_{zz}} + pq \frac{(I_{xx} - I_{yy})}{I_{zz}}\end{aligned} \tag{3.1}$$

where

$$C_Y = C_y(\beta, \delta_r, \mathrm{M}) + C_{yr}(\mathrm{M}) \frac{r\, lref}{2V_{missile}}$$

$$C_Z = C_z(\delta_e, \mathrm{M}, \alpha) + C_{zq}(\mathrm{M}) \frac{q\, lref}{2V_{missile}}$$

Next, equation (3.1) is derived with respect to aerodynamic variables, linearized at an equilibrium point (i.e. trim condition) and state space forms of the equations (equation (3.2)) are obtained. For a mechanical system, the conditions for equilibrium are defined by the following:

$$\sum \text{Forces} = 0 \text{ and } \sum \text{Moments} = 0$$

In terms of the system state vector, the equilibrium (trim) is defined as the following:

$$0 = \dot{X}_E = f(X_E, U_E), \quad U_E = \text{constant}$$

where X denotes the states, U denotes the input and subscript E denotes the equilibrium.

Trim conditions are determined by flying the missile at a certain altitude and observing the states and inputs where they are constant. This altitude is chosen because of considered sea-skimming cruise altitude.

Table 3-1 Trim Condition Parameters

Altitude (h)	15 m (from sea level)
Density (ρ)	1.23 kg/m3
Cruise Velocity (V_{cruise})	0.8 M
Sideslip Angle (β_{trim})	0.1°
Speed of sound (a)	340 m/sec
Angle of attack (α_{trim})	5°
Elevator Deflection (δe_{trim})	-3°
Rudder Deflection (δr_{trim})	-0.5°

According to the designated trim condition which is given on Table 3-1, aerodynamic coefficients are found from aerodynamic tables, their changes with respect to aerodynamic parameters are calculated and substituted into the matrices in equation (3.2). The elements of $A1$, $A2$ and B matrices are given in Appendix B. The matrices in equation (3.2) are different from the actual matrices which represents the real dynamics. In reality, there are high nonlinearities in the missile motion, thus these matrices do not represent the real motion because of linearization. Here, $\dot{q}$, $\dot{r}$ denotes the pitch and yaw angular accelerations obtained from linearized equations.

$$\begin{bmatrix} \dot{q} \\ \dot{r} \end{bmatrix} = [A1]_{2x3} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + [A2]_{2x2} \begin{bmatrix} q \\ r \end{bmatrix} + [B]_{2x2} \left(\begin{bmatrix} \delta_e \\ \delta_r \end{bmatrix} - \begin{bmatrix} \delta_e \\ \delta_r \end{bmatrix}_{trim} \right) \tag{3.2}$$

In practice $A1$, $A2$ and B are not represented nor exactly known [5], [25], [38], [39], thus the estimate of $A1$, $A2$ and B are used in Model Inversion Controller. $\hat{A}1$, $\hat{A}2$ and $\hat{B}$ matrices denotes estimates of the linearized matrices. In equation (3.3), $\hat{A}1$ and $\hat{A}2$ matrices indicate the estimated aerodynamic stability derivatives, and $\hat{B}$ represents the estimated aerodynamic control derivatives at the trim condition. Body velocities u, v, w are calculated with respect to body fixed frame (slow translational states) and q, r are body angular rates about the body fixed axes (fast rotational states). δ_e and δ_r are the control inputs and $\hat{\delta}_e$, $\hat{\delta}_r$ are their estimates for longitudinal and directional dynamics respectively.

$$\begin{bmatrix} \dot{\hat{q}} \\ \dot{\hat{r}} \end{bmatrix} = \left[\hat{A}1\right]_{2x3} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \left[\hat{A}2\right]_{2x2} \begin{bmatrix} q \\ r \end{bmatrix} + \left[\hat{B}\right]_{2x2} \left(\begin{bmatrix} \hat{\delta}_e \\ \hat{\delta}_r \end{bmatrix}_{required} - \begin{bmatrix} \delta_e \\ \delta_r \end{bmatrix}_{trim} \right) \tag{3.3}$$

$\hat{A}1$, $\hat{A}2$ and $\hat{B}$ matrices are different from actual matrices. Here, $\dot{\hat{q}}$, $\dot{\hat{r}}$ denotes the estimated pitch and yaw angular accelerations respectively, which are obtained from the linearized and estimated rotational equations of motion.

3.2 Neural Network Based Inversion Architecture

In literature, the altitude hold autopilot architecture is generally established with an altitude (h_{com}) commanded outer loop and body pitch rate (q) commanded [86] or normal acceleration (a_n) commanded [1] inner loop. However, if the inner loop controller input is normal acceleration (a_n), the transfer function between elevator deflection (δ_e) and normal acceleration (a_n) at the center of gravity always has a zero on the right half s-plane for tail-controlled missiles. This zero in the right half s-plane causes non-minimum phase characteristics on the missile [9] that is described in Section 1.1. This condition prevents accurate model inversion. In addition, for the pitch rate loop, the transfer function from elevator deflection (δ_e) to body pitch rate (q) has a zero very close to the origin (i.e. near the imaginary axis); therefore it produces a very slow mode when the inversion is not exact [9]. It is

difficult to achieve design criteria such as rise time and settling time. Therefore altitude hold design is carried out by using pitch angle (θ) for the inner loop, and implementing an altitude (h_{com}) commanded outer loop to map from an altitude error to pitch attitude command to the inner loop proposed in [87] and [88]. Missile speed (i.e. Missile Mach number) control is not included in the Altitude Hold Autopilot. On the other hand, Missile Mach number is controlled by a Thrust Model which is described in Section 2.5. The detailed information about this model can be found in [1]. Moreover, any lateral dynamics effect is neglected in the controller design. Therefore the only motion considered is in the vertical plane [89].

This section contains the architecture of the ANN implemented dynamic model inversions on missile altitude hold application. It is based on the neural network applications as described in [13], [25], [36], [38], [39]. In addition, the altitude hold application is taken from [87]. This architecture is shown in Figure 3.1. Since the primary effect of the longitudinal control is a change in pitch attitude, the attitude response to a longitudinal control input is important [2]. As a result, the architecture is based on ACAH for the longitudinal inner loop and a Proportional with an Integral and Derivative Controller for the outer loop. The outer loop converts altitude error into pitch angle command for the inner loop.

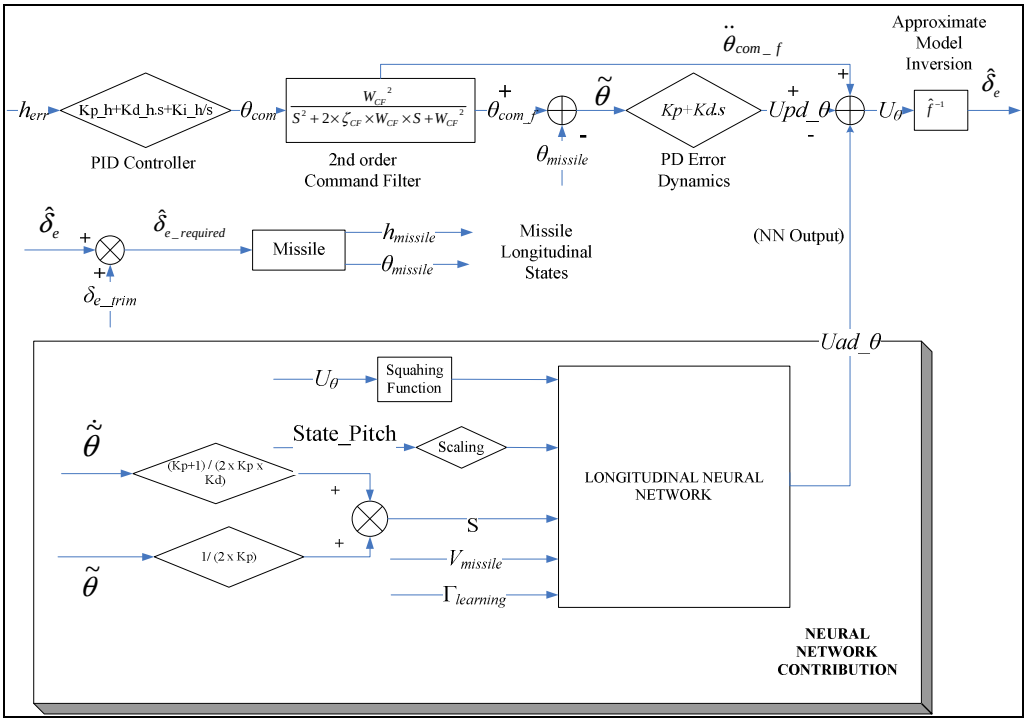

Figure 3.1 Altitude Hold Autopilot

3.2.1 Inner Loop Design

In the preceding section, linearized and estimated equations of rotational motion (equation (3.3)) are obtained. At this stage, these equations are inverted and the approximate model inversion of the missile plant is formed. By inverting these equations, a relationship between desired angular accelerations and estimations of required control deflections is established. The desired angular accelerations are commanded and the corresponding estimated fin deflections ($\hat{\delta}_e, \hat{\delta}_r$) are found from equation (3.4) which represents the approximate inversion.

$$\begin{bmatrix} \hat{\delta}_e \\ \hat{\delta}_r \end{bmatrix}_{required} = \left[\hat{B}\right]_{2x2}^{-1} \left(\begin{bmatrix} \dot{q}_{desired} \\ \dot{r}_{desired} \end{bmatrix} - \left[\hat{A}1\right]_{2x3} \begin{bmatrix} u \\ v \\ w \end{bmatrix} - \left[\hat{A}2\right]_{2x2} \begin{bmatrix} q \\ r \end{bmatrix} \right) + \begin{bmatrix} \delta_e \\ \delta_r \end{bmatrix}_{trim} \qquad (3.4)$$

Here, any cross-coupling between fast rotational states and slow translational states is neglected in the inversion [38], [39]. Since the $\hat{B}$ matrix is a square matrix and non-singular, it can be inverted.

Since it is not straight forward to exactly know, measure or estimate these dynamic equations, an approximation has to be used. Here, an inversion for one flight condition as in equation (3.4) will be an approximation for the rest of the flight conditions [2]. Operations at any other flight condition will cause an inversion error. Therefore Neural Networks (NNs) are designed for cancelling the error between the true model and approximate one. ANNs and inverted model are combined in order to reduce and cancel this error.

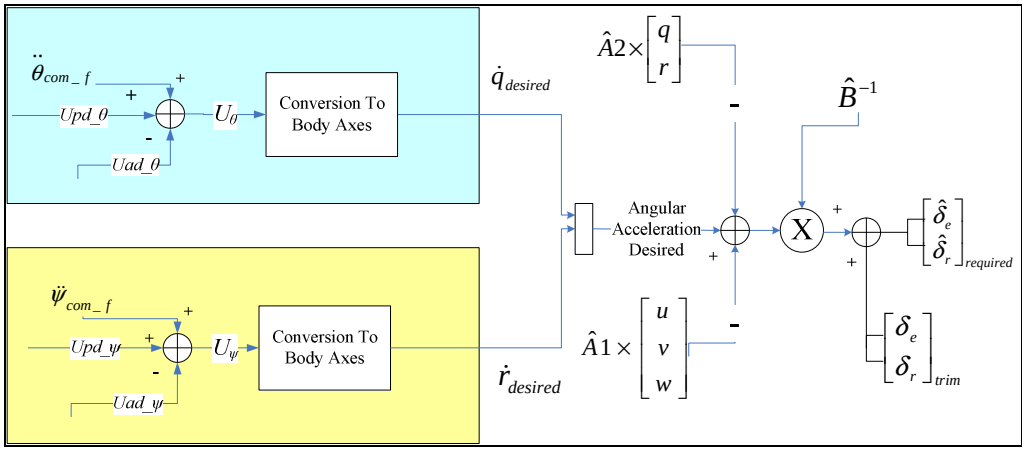

Figure 3.2 Application of Model Inversion on both longitudinal and directional control

Note that equation (3.4) will also be used for the directional autopilot.

From the altitude hold autopilot architecture in Figure 3.1, the pseudo control, U_θ is found by the following equation:

$$U_\theta = Upd\_\theta + \ddot{\theta}_{com\_f} - Uad\_\theta \qquad (3.5)$$

Uad_θ is the adaptive signal that represents the neural network output on the pitch plane. The proportional plus derivative contribution is shown by *Upd_θ* and $\ddot{\theta}_{com\_f}$ is the command filtered Euler angular acceleration for pitch.

According to the ACAH implementation, U_θ is equivalent to desired Euler angular acceleration for pitch. However we need body angular accelerations in equation (3.4), so it will need to be transformed to the body axes reference frame [25]. The transformation in Figure 3.2 is made by the following equations:

$$\begin{aligned} \dot{q}_{desired} &= U_\theta \cos\phi - \dot{\theta}\,\dot{\phi}\sin\phi + U_\psi \sin\phi\cos\theta + \dot{\psi}\,\dot{\phi}\cos\phi\cos\theta - \dot{\psi}\,\dot{\theta}\sin\phi\sin\theta \\ \dot{r}_{desired} &= -U_\theta \sin\phi - \dot{\theta}\,\dot{\phi}\cos\phi + U_\psi \cos\phi\cos\theta - \dot{\psi}\,\dot{\phi}\sin\phi\cos\theta - \dot{\psi}\,\dot{\theta}\cos\phi\sin\theta \end{aligned} \tag{3.6}$$

Here U_ψ denotes desired Euler angular acceleration for yaw. Note that it is obtained from directional autopilot architecture.

The proportional plus derivative dynamics for the pitch plane is obtained by:

$$Upd\_\theta = Kp\left(\theta_{com\_f} - \theta\right) + Kd\left(\dot{\theta}_{com\_f} - \dot{\theta}\right) \tag{3.7}$$

The gains *Kp* and *Kd* are used to define the error dynamics. These dynamics are designed faster than the command filter, and slow enough not to be affected by the actuator dynamics. However, actuator dynamics are not modeled in this study, so we will look for the relationship between command filter and error dynamics. The selection of this parameter will be explained in Section 3.2.2.

The true dynamics are not given by the linear approximated model in equation (3.3). There may be errors due to reasons which described in Section 1.1. The model inversion error can be defined as the difference between the actual values of the rotation and the approximated one (equation (3.8)).

$$\begin{bmatrix} \varepsilon_\theta \\ \varepsilon_\psi \end{bmatrix} = \begin{bmatrix} \dot{q}_{desired} \\ \dot{r}_{desired} \end{bmatrix} - \begin{bmatrix} \hat{\dot{q}} \\ \hat{\dot{r}} \end{bmatrix} \tag{3.8}$$

If equation (3.3) substitutes into equation (3.8), the following equation is obtained:

$$\begin{bmatrix}\varepsilon_\theta\\ \varepsilon_\psi\end{bmatrix}=\begin{bmatrix}\dot{q}_{desired}\\ \dot{r}_{desired}\end{bmatrix}-\left(\begin{bmatrix}\hat{A1}\end{bmatrix}_{2x3}\begin{bmatrix}u\\ v\\ w\end{bmatrix}+\begin{bmatrix}\hat{A2}\end{bmatrix}_{2x2}\begin{bmatrix}q\\ r\end{bmatrix}+\begin{bmatrix}\hat{B}\end{bmatrix}_{2x2}\left(\begin{bmatrix}\hat{\delta}_E\\ \hat{\delta}_R\end{bmatrix}_{required}-\begin{bmatrix}\delta_E\\ \delta_R\end{bmatrix}_{trim}\right)\right) \quad (3.9)$$

From equation (3.9), we can define Euler pitch angular acceleration as:

$$\ddot{\theta}=U_\theta+\varepsilon_\theta \quad (3.10)$$

Combining equation (3.5), (3.7), and (3.10) we obtain:

$$\left(\ddot{\theta}_{com\_f}-\ddot{\theta}\right)+Kp\left(\theta_{com\_f}-\theta\right)+Kd\left(\dot{\theta}_{com\_f}-\dot{\theta}\right)=Uad\_\theta-\varepsilon_\theta \quad (3.11)$$

On the right hand side of equation (3.11), network compensation error is generated. In the ideal case, it is equal to 0. It is expected that ANN output cancels the model inversion errors. If it occurs, consider the desired error dynamics (i.e. NN output cancels the inversion error), the equation (3.11) becomes:

$$\left(\ddot{\theta}_{com\_f}-\ddot{\theta}_{desired}\right)+Kp\left(\theta_{com\_f}-\theta_{desired}\right)+Kd\left(\dot{\theta}_{com\_f}-\dot{\theta}_{desired}\right)=0 \quad (3.12)$$

The details of ANN architecture will be described in Section 3.3.

3.2.2 Command Filter Design and Inner Loop Error Dynamics

Feedback linearization separates the flight dynamics into fast and slow dynamics by using timescale properties [90], [91]. Command filter and PD error dynamics are designed with the consideration of the timescale separation.

The command filter serves both to limit the input rate, and as a model for desired response. Bandwidth separation (time scale separation) of the command filter and the adaptation dynamics are important for getting desired response from the missile. As a result, the command filter must be slower than the missile error dynamics on pitch attitude. Therefore, consider the pitch attitude error dynamics of the missile (equation (3.11)). Define pitch attitude tracking error as $\tilde{\theta}=\theta_{com\_f}-\theta$ and take the Laplace transform.

$$
\begin{aligned}
& s^2\tilde{\theta} + Kd \cdot s\tilde{\theta} + Kp \cdot \tilde{\theta} = Uad\_\theta - \varepsilon_\theta \\
& \qquad \Updownarrow \\
& s^2\tilde{\theta} + \left(2 \cdot \zeta_{err} \cdot w_{nerr}\right) \cdot s\tilde{\theta} + {w_{nerr}}^2 \cdot \tilde{\theta} = Uad\_\theta - \varepsilon_\theta
\end{aligned}
\tag{3.13}
$$

Then the transfer function from inversion error compensation of pitch motion to pitch attitude error becomes:

$$
\frac{\tilde{\theta}(s)}{(Uad\_\theta - \varepsilon_\theta)(s)} = \frac{1}{s^2 + \left(2 \cdot \zeta_{err} \cdot w_{nerr}\right) \cdot s + {w_{nerr}}^2}
\tag{3.14}
$$

where w_{nerr} denotes natural frequency, ζ_{err} is damping ratio of the error dynamics. From equation (3.14), the relationship for error dynamics can be taken as a second-order system. Therefore we have to determine a set of specifications for the desired error dynamics.

Since the missile can be specified as an under-damped system, the design process is performed according to the transient response characteristics of an under-damped second-order system (equation (3.15)) such as the settling time (t_s), rise time (t_r), max % overshoot are specified according to [92].

$$
\begin{aligned}
& for \quad 0 < \zeta < 1 \\
& t_s = \frac{4}{\zeta \times w_n} \quad \text{for } 2\,\% \text{ criterion} \\
& t_r = \frac{\pi - \arctan\left(\dfrac{\sqrt{1-\zeta^2}}{\zeta}\right)}{w_n \cdot \sqrt{1-\zeta^2}} \\
& \max\,\%\ overshoot = (\exp^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}) \cdot 100
\end{aligned}
\tag{3.15}
$$

where w_n denotes natural frequency, ζ is the damping ratio [92]. As can be seen above, the value of ζ is usually determined from the requirement of allowable maximum % overshoot, and thus settling time is determined primarily by the natural frequency [92].

For time scale separation, the slower response is required a command filter compared to error dynamics. In order to find the error dynamics of the missile, the natural frequency and damping ratio of the missile airframe (*AF*) must be found. According to [93], the missile airframe natural frequency (w_{AF}) and damping ratio (ζ_{AF}) for longitudinal motion is found as:

$$w_{AF} = \sqrt{-M_\alpha} = \sqrt{\frac{-\frac{1}{2}\left(C_{m\alpha} - C_{Z\alpha}\frac{X_{cg}}{lref}\right)\rho V^2{}_{missile}\ Sref\ lref}{I_{yy}}} = 5.21\ rad/\sec \tag{3.16}$$

$$\zeta_{AF} = \frac{Z_\alpha w_{AF}}{2M_\alpha} = \frac{\frac{1}{2}C_{Z\alpha}\ \rho V^2{}_{missile}\ Sref\ w_{AF}}{2\cdot\frac{\frac{1}{2}\left(C_{m\alpha} - C_{Z\alpha}\frac{X_{cg}}{lref}\right)\rho V^2{}_{missile}\ Sref\ lref}{I_{yy}}} = 0.06 \tag{3.17}$$

Aerodynamic parameters in the above equations are found using the trim conditions. After finding missile airframe natural frequency from equation (3.16) and damping ratio from equation (3.17), the error dynamics can easily be determined.

According to [92], for a desirable transient response of a second order system, the damping ratio ζ must be between 0.4 and 0.8. Small values of ζ (ζ<0.4) yield excessive overshoot in the transient response, a system with a large value of ζ (ζ>0.8) responds sluggishly. Moreover, under-damped systems ($0 < \zeta < 1$) with ζ between 0.5 and 0.8 gets close to the final value more rapidly than critically damped ($\zeta = 1$) or over-damped systems ($\zeta > 1$) [92]. So the calculated damping ratio can not be selected for missile longitudinal dynamics.

As a result, we consider the 2% design criteria for settling time, its minimum occurs at ζ value around 0.76 [92]. Then desired damping ratio (ζ_{AF}) is selected as 0.75 instead of its current value of 0.06.

Since the actual natural frequency is generally not attainable, the error dynamics is considered slower than the missile dynamics. So they are selected to satisfy the following criterion:

$$w_{n_{err}} < w_{AF} \quad and \quad \zeta_{err} \approx \zeta_{AF} \qquad (*)$$

To sum up, by using relation (*), equation (3.13) and (3.15), the parameters related to error dynamics are tabulated as:

Table 3-2 Missile Parameters Related to Longitudinal Error Dynamics

Airframe Natural Frequency (w_{AF})	5.21 rad/sec
Airframe Damping Ratio (ζ_{AF})	0.75
Error Dynamics Natural Frequency ($w_{n_{err}}$)	4.50 rad/sec
Error Dynamics Damping Ratio (ζ_{err})	0.75
Error Dynamics Settling Time (t_s)	1.19 sec
Error Dynamics Maximum % Overshoot	2.84
Error Dynamics Rise Time (t_r)	0.82 sec
$Kd = 2 \cdot \zeta_{err} \cdot w_{n_{err}}$	6.75
$Kp = w_{n_{err}}^{\ 2}$	20.25

Next, making the command filter response slower compared to error dynamics, command filter transient response characteristics can be determined according to the following relation:

$$w_{CF} < w_{n_{err}} \quad and \quad \zeta_{CF} \approx \zeta_{err} \qquad (**)$$

To sum up, by using relations (**), equation (3.15), the parameters related to command filter are tabulated as:

Table 3-3 Longitudinal Command Filter Parameters

Natural Frequency (w_{CF})	3.0 rad/sec^2
Damping Ratio (ζ_{CF})	0.75
Settling Time (t_s)	1.78 sec
Maximum % Overshoot	2.84
Rise Time (t_r)	1.22 sec

$$\frac{\theta_{com\_f}}{\theta_{com}} = \frac{{w_{CF}}^2}{s^2 + 2\zeta_{CF} w_{CF} s + {w_{CF}}^2} \qquad (3.18)$$

If the values are substituted into equation (3.18), the command filter transfer function is obtained as:

$$\frac{\theta_{com\_f}}{\theta_{com}} = \frac{3^2}{s^2 + (2 \cdot 3 \cdot 0.75)s + 3^2} = \frac{9}{s^2 + 4.5s + 9} \qquad (3.19)$$

The same methodology will also be used for directional autopilot.

3.2.3 Outer Loop Design

When designing control systems, the timescale separation between the inner-loop attitude control and outer-loop altitude control systems is commonly used. It allows

the inner loop and outer loop to be designed separately but requires the outer-loop bandwidth to be lower than that of the inner loop [3].

The outer loop altitude-hold is a simple Proportional Integral Derivative (PID) controller by taking the time scale separation into account. It takes the commanded reference altitude (h_{com}) and the current missile altitude (h) as an input and outputs the pitch angle (θ), required to maintain or reach the commanded reference altitude. Proportional control is used for quick response, integral control is used for eliminating steady state error and derivative control is used for damping oscillations in this autopilot architecture.

The gains Kp_h (Proportional Gain), Ki_h (Integral Gain) and Kd_h (Derivative Gain) are selected and tuned by observing step response characteristics generated through the 5-DOF simulation [89], [92]. Based on the planned altitude trajectory the missile climbs to 400 meters initially, then follows different altitudes and descends to cruise altitude during the simulation studies. Therefore, the step altitude command will be selected as 400 meters for tuning gains. Although this value is an excessive value for the corrections against disturbances, such a value may be required while accomplishing the pre-programmed cruise altitude profile [1]. The graphs related to gain selection and explanations for PID controller design are given in Appendix C.

Table 3-4 PID GAINS

Kp_h	Ki_h	Kd_h
0.005	0.0005	0.01

3.3 Online Learning Neural Network Architecture

Online Learning NNs (i.e. learning-while-controlling [38]) implies that the NN has the capability of changing in real time the values of the numerical components that

make up its architecture [2]. In this section, two types of online neural network are explained. These networks can consist of almost any feed-forward structure that is linear in its parameters or nonlinearly parameterized. Linearly parameterized neural networks are also called Sigma-Pi Networks and nonlinearly parameterized neural networks are defined as Single Hidden Layer Networks. Each approach has been applied in designing the inner loop attitude controller of the missile altitude hold autopilot for compensating the inversion error described in Section 3.2.1. Numerical results from nonlinear 5-DOF simulation studies will be presented and compared for each network in Chapter 6.

3.3.1 Linear In the Parameters Sigma-Pi Neural Networks

The Linear in the Parameters Sigma-Pi Neural Networks (LIP NNs) have one output layer and no hidden layers [94] and are universal approximations [95]. In this architecture, it is assumed that the inversion error is decoupled (i.e. pitch and yaw motions are decoupled) for reducing the size and complexity of Neural Networks. The dimension of the NN increases exponentially with the number of network inputs [13]. Therefore for the ANN in the longitudinal channel, its inputs are only the longitudinal states, the longitudinal pseudo control and bias terms. If there was coupling, the NN theoretically would need feedback from the directional state to be able to approximate the modeling error. In that case, coupling terms have to be used as NN inputs. Figure 3.3 shows a general architecture of LIP NN [25], [39].

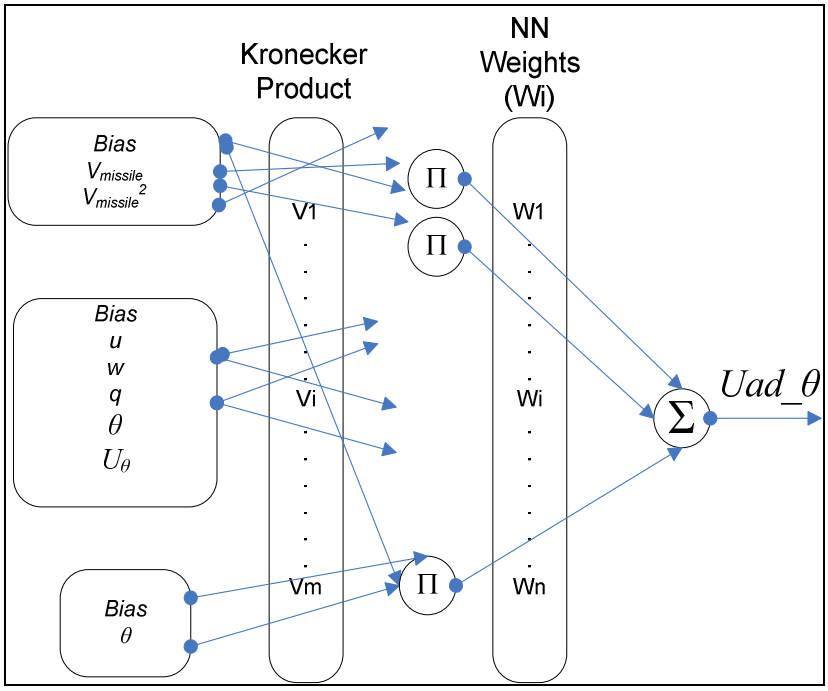

Figure 3.3 LIP NN Structure

NN weights Wi's constitute a vector of neural network weights, shown in Figure 3.3. These variable network weights can be updated by learning' laws derived from Lyapunov stability theorem [25]. Here neuron interactions are provided using Kronecker products [19]. The basis functions Vi are constructed by grouping normalized inputs into three categories similarly described in [5], [13], [25], [36], [38], and [39].

Modeling errors may mainly come from the inconsistencies between the actual highly nonlinear missile dynamics and its mathematical model. Moreover only linearized model is used in the inversion. Controllers are designed for a certain altitude, dynamic pressure etc. They all represent only one flight condition. On the other hand, the equations of motion can be linearized for all possible flight conditions and inverted. In other words, *A*1 *and A*2 matrices are scheduled, but this is a long design process. Artificial NNs, which have the ability to approximate general continuous nonlinear functions, are ideal for canceling these model errors

[3], [5], [13]. NN inputs are selected of parameters needed to approximate the model uncertainty.

The first group of network inputs is used to compansate for the model inversion error due to changes in airspeed, since the stability and control derivatives are strongly dependent on dynamic pressure.

$$C_2 = \begin{bmatrix} Bias & V_{missile} & V_{missile}^{\,2} \end{bmatrix} \tag{3.20}$$

The second group consists of normalized longitudinal state variables, the pseudo control and a bias term. Because the plant is nonlinear and uncertain in the control similarly in the states, the inversion error is a function of both state and pseudo control.

$$C_2 = \begin{bmatrix} Bias & u & v & q & \theta & U_\theta \end{bmatrix} \tag{3.21}$$

Since the input to the LIP NN includes the pseudo control signal U_θ and it is a function of the NN output (Uad_θ), therefore a fixed-point assumption on Uad_θ is needed [5], [25], [38], [39]. U_θ is an input to the NN through a squashing function (Figure 3.1). Squashing functions guarantee the existence of at least one fixed point solution and boundedness of the input to error dynamics (equation (3.28)) [5], [25], [38], [39]. Squashing function for longitudinal NN is:

$$Squashing\ \ Function = \left(\left(\frac{1}{1+e^{-0.5 \times U_\theta}} \right) - 0.5 \right) \cdot 2 \tag{3.22}$$

The third group is used to approximate effects of transformation between the body frame and the inertial frame. This is related to changes in pitch attitude.

$$C_2 = \begin{bmatrix} Bias & \theta \end{bmatrix} \tag{3.23}$$

It should be noted that for all groups bias values are selected according to scaled (normalized) values of the inputs.

Finally, the vectors of basis functions are composed of products of the elements belonging to each group of input (C_1, C_2, and C_3) and are related to each other via the Kronecker product.

$$\beta = kron(kron(C_1, C_2), C_3) \tag{3.24}$$

where

$$kron(x, y) = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \dots & x_m y_n \end{bmatrix}^T \tag{3.25}$$

The relationship between the inputs and outputs of the longitudinal network is established by the following equation:

$$U_{AD\_\theta} = W^T \cdot \beta\left(\overline{X}, U_\theta, bias\right) \tag{3.26}$$

The model inversion error can be accurately produced at the network output using equation (3.26). In this equation, W denotes the vector of current variable network weights, β is a vector of network basis functions (kronecker product of inputs), and $\overline{X}$ represents the normalized states. Adaptation of weights in the neural network is derived using Lyapunov's theory [13], [25], [76]. The weight update rule is found as:

$$\dot{W} = -\Gamma_{learning} \cdot \begin{bmatrix} \tilde{\theta} \\ \dot{\tilde{\theta}} \end{bmatrix}^T \cdot P \cdot b \cdot \beta - \mu \cdot \left\| \begin{bmatrix} \tilde{\theta} \\ \dot{\tilde{\theta}} \end{bmatrix}^T \cdot P \cdot b \right\| \cdot W \tag{3.27}$$

This law guarantees bounded weights and errors. The first term is derived from Lyapunov stability approach, and the second term assures the boundedness of the NN weights. The second term provides additional robustness by introducing damping [25]. P and b parameters are derived from equation (3.28) which represents error dynamics.

$$\begin{bmatrix} \dot{\tilde{\theta}} \\ \ddot{\tilde{\theta}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -Kp & -Kd \end{bmatrix} \cdot \begin{bmatrix} \tilde{\theta} \\ \dot{\tilde{\theta}} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \left(Uad\_\theta - \varepsilon_\theta\right) = A \cdot \begin{bmatrix} \tilde{\theta} \\ \dot{\tilde{\theta}} \end{bmatrix} + b \cdot \left(Uad\_\theta - \varepsilon_\theta\right) \tag{3.28}$$

As a result, *b* is defined as:

$$b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \tag{3.29}$$

P is calculated using the following Lyapunov function [4]. Lyapunov equation is defined as:

$$A^T P + PA = -Q \quad \Rightarrow \quad Q = I_{2\times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad and \quad A = \begin{bmatrix} 0 & 1 \\ -Kp & -Kd \end{bmatrix} \tag{3.30}$$

Q is a positive definite matrix (i.e. identity matrix). From Lyapunov stability analysis of the error dynamics (equation (3.30)), P is derived as:

$$P = \begin{bmatrix} \frac{Kd}{Kp} + \frac{1}{2Kd} & \frac{1}{2Kp} \\ \frac{1}{2Kp} & \frac{1+Kp}{2KpKd} \end{bmatrix} \tag{3.31}$$

The details of the derivation are explained in [5] and [25].

The selection of acceptable values for the fixed parameters in the NN such as $\Gamma_{learning}$ (learning rate) and μ (modification term) required trial-and-error simulation studies because these parameters can affect the missile response. Both of these parameters must be selected as positive. The details of this type of neural network structure are explained in [4], [9] and [48]

Identical construction of the LIP NN applies to the directional autopilot in Chapter 4.

3.3.2 Single Hidden Layer Neural Networks

The Single Hidden Layer Neural Networks (SHL NNs) are an extension of the architecture described in the previous section. The adaptation law in this

architecture is also designed based on a Lyapunov stability analysis of the error signals [96].

SHL NNs are also universal approximators [65], [97]. Hence, given a sufficient number of hidden layer neurons and appropriate inputs, it is possible to train the network online to cancel model error [13], [23], [98]. Figure 3.4 shows the structure of a generic SHL NN.

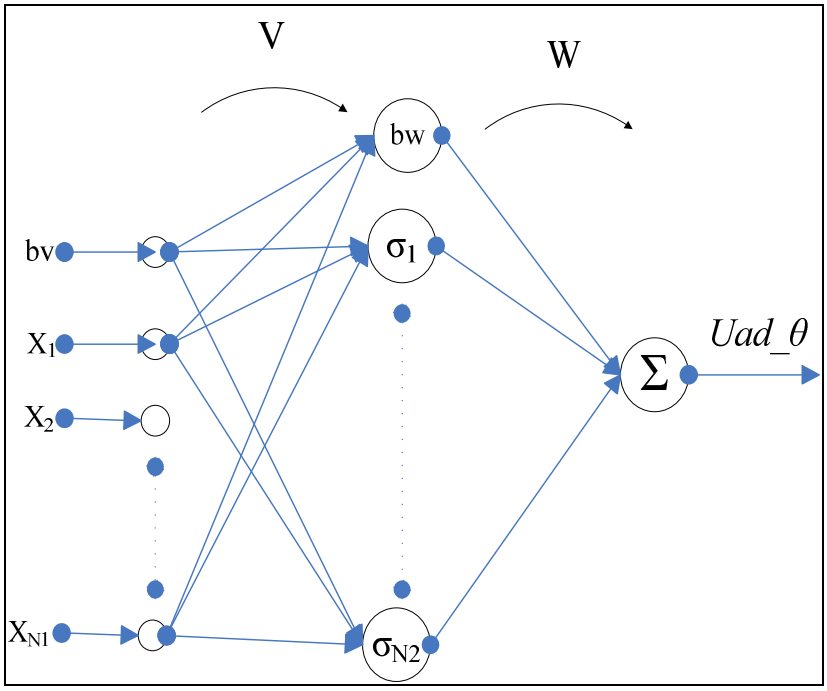

Figure 3.4 SHL NN Structure

It has an output *Uad_θ* which is adapted to cancel the model inversion error. *Uad_θ* can be expressed as:

$$Uad\_\theta = b_w\theta_{w,1} + \sum_{j=1}^{N2} W_{j,1}\sigma_j\left(b_v\theta_{v,j} + \sum_{i}^{N1} V_{i,j}X_i\right) \tag{3.32}$$

Here, $N1$ is the number of inputs; $N2$ is the number of hidden layer neurons and it has one output. $\theta_{w,1}$ is the outer layer threshold, b_w is the outer layer bias, $W_{j,1}$ represents the outer layer weights. $\theta_{v,j}$ is the inner layer threshold, b_v is the inner layer bias, $V_{i,j}$ denotes the inner layer weights. X_i represents the neural network normalized inputs. The scalar function σ_j is hidden layer activation function. The form of the hidden-layer activation function is a design parameter; this function is selected as sigmoidal activation function:

$$\sigma_j\left(z_j\right)=\frac{1}{1+e^{-a_j \cdot z_j}} \tag{3.33}$$

The constant a_j is a distinct value for each hidden-layer neuron (j=1, 2…, N2) which is so called activation potential. z_j is the input to the j^{th} hidden layer neuron.

The relation between input and output of the SHL NN can be established in matrix form as:

$$Uad\_\theta(W,V,X)=W^T\sigma\left(V^T X\right) \tag{3.34}$$

In this architecture, the inputs to the network are selected as:

$$X=\begin{bmatrix} b_v & \tilde{\theta} & \ddot{\tilde{\theta}} & \theta_{com\_f} & \dot{\theta}_{com\_f} & \ddot{\theta}_{com\_f} & Uad\_\theta & \|Z\|_F & u & w & q \end{bmatrix}^T \tag{3.35}$$

The number of inputs are equal to 11 (i.e. $N1$=11) and $b_v \geq 0$. $\|Z\|_F$ is the frobenius norm of the Z matrix that contains all tunable weight parameters.

$$Z=\begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \tag{3.36}$$

In this architecture the following definitions are used which were described in this section:

$$z=V^T X=\begin{bmatrix} z_1 & z_2 & \ldots\ldots & z_{N2} \end{bmatrix}^T \tag{3.37}$$

$$\sigma(z) = \begin{bmatrix} b_w & \sigma(z_1) & \sigma(z_2) & \dots & \dots & \sigma(z_{N2}) \end{bmatrix}^T \tag{3.38}$$

$$V = \begin{bmatrix} \theta_{v,1} & \dots & \dots & \theta_{v,N2} \\ v_{1,1} & \dots & \dots & v_{1,N2} \\ \vdots & \ddots & \ddots & \vdots \\ v_{N1,1} & \dots & \dots & v_{N1,N2} \end{bmatrix} \tag{3.39}$$

where $N2$ is selected as 18 and $b_w \geq 0$. There is one hidden layer which includes 18 neurons.

$$W = \begin{bmatrix} \theta_{w,1} & \dots & \dots & \theta_{w,N3} \\ w_{1,1} & \dots & \dots & w_{1,N3} \\ \vdots & \ddots & \ddots & \vdots \\ w_{N2+1,1} & \dots & \dots & w_{N2+1,N3} \end{bmatrix} \tag{3.40}$$

Since there is only one output, $N3$ is equal to 1.

The adaptation law of the NN weights are found by:

$$\dot{V} = -\Gamma_{learning} \cdot \left[X \cdot \begin{bmatrix} \tilde{\theta} \\ \tilde{\theta} \end{bmatrix}^T \cdot P \cdot b \cdot W^T \cdot \sigma' + \mu \cdot \left\| \begin{bmatrix} \tilde{\theta} \\ \tilde{\theta} \end{bmatrix}^T \cdot P \cdot b \right\| \cdot V \right] \tag{3.41}$$

$$\dot{W} = -\gamma_{learning} \cdot \left[\begin{bmatrix} \tilde{\theta} \\ \tilde{\theta} \end{bmatrix}^T \cdot P \cdot b \cdot \left(\sigma - \sigma' \cdot V^T \cdot X \right) + \mu \cdot \left\| \begin{bmatrix} \tilde{\theta} \\ \tilde{\theta} \end{bmatrix}^T \cdot P \cdot b \right\| \cdot W \right] \tag{3.42}$$

where $\Gamma_{learning}$ and $\gamma_{learning}$ are learning rates of the inner and outer layer NNs respectively. σ' is a matrix that contains derivatives of the sigmoid vector described in equation (3.38).

$$\sigma'(z_j) = \begin{bmatrix} 0 & \dots & 0 \\ \dfrac{\partial \sigma(z_1)}{\partial z_1} & & 0 \\ 0 & \ddots & \\ 0 & & \dfrac{\partial \sigma(z_{N2})}{\partial z_{N2}} \end{bmatrix} \tag{3.43}$$

Details on the NN architecture and derivation of adaptation rule can be found in [97], [99], [100], [101], and [102].

The performance and the acceptability of SHL NN Controller are related to the performances of its training algorithm [3]. Therefore the selection of acceptable values for $\Gamma_{learning}$, $\gamma_{learning}$ (learning rates) and μ (modification term) in adaptation rule are tuned via trial-and-error simulation studies. The performance of missile longitudinal controller with these neural network architectures will be given in Chapter 6.

Similar architecture of this network is also applied to the directional autopilot.

CHAPTER 4

ADAPTIVE NEURAL NETWORK BASED DIRECTIONAL AUTOPILOT

In this chapter, the directional autopilot with NN architecture is explained. The ANN based model inversion is now applied on the missile directional control. It is carried out by using a yaw angle for the control loop. Yaw angle is calculated by integrating yaw rate mapping from a LOS rate on the yaw axis that comes from LOS guidance law to rate command [87].

The target used in this case study can only move in the directional plane (x-y plane). Therefore, the cruise missile is designed to fly on the x-y plane with respect to the target trajectories (terminal phase) or planned waypoints (Mid-Course Phase). This operation can be performed by using a directional autopilot. This autopilot allows the missile to turn to a guided heading and to hold its position at that heading. It uses directional guidance command for the required movement of the cruise missile.

4.1 Neural Network Based Inversion Architecture

This section includes the architecture of the ANN implemented dynamic model inversion on the directional autopilot application. It is based on neural network applications similar to the preceding chapter. This architecture is shown in Figure 4.1. The architecture is based on an ACAH for directional loop which is proposed in [87]. Command filter and PD error dynamics are also designed with the consideration of the timescale separation.

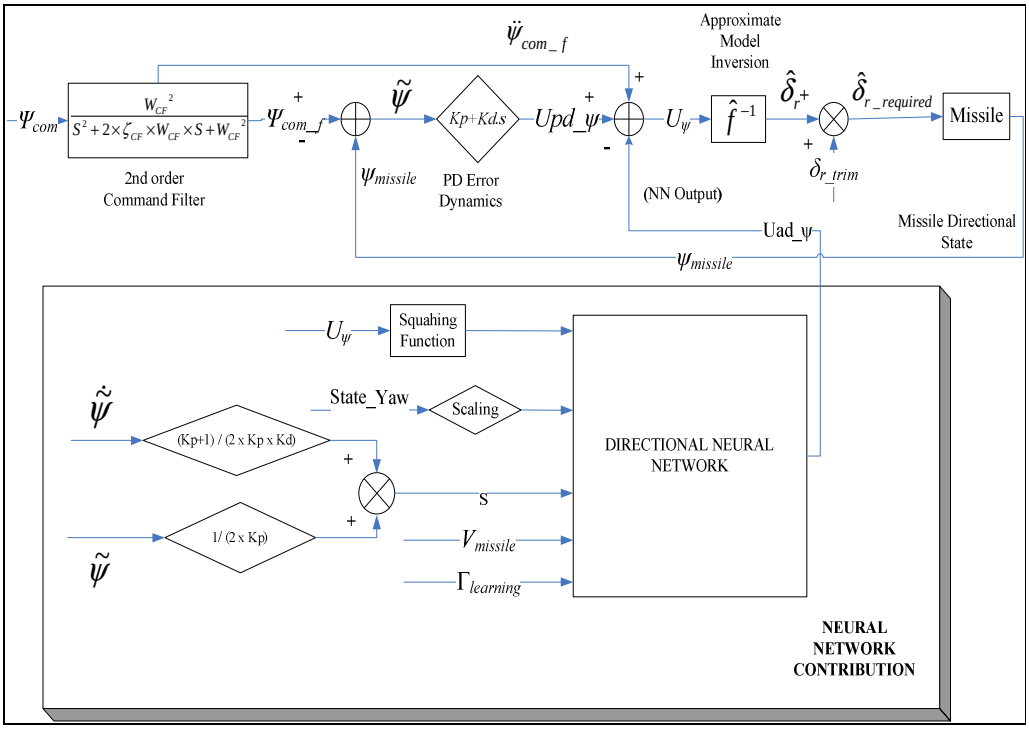

Figure 4.1 Directional Autopilot

Equation (3.4) is also used for the directional autopilot in order to derive the error dynamics. From the directional autopilot architecture in Figure 4.1, the pseudo control, U_ψ is found by the following equation:

$$U_\psi = Upd\_\psi + \ddot{\psi}_{com\_f} - Uad\_\psi \tag{4.1}$$

Uad_ψ is the adaptive signal that represents the neural network output on the yaw plane. The proportional plus derivative contribution is shown by Upd_ψ, and $\ddot{\psi}_{com\_f}$ is the filtered Euler angular acceleration command.

According to attitude command attitude hold implementation, U_ψ is equivalent to the desired Euler angular acceleration for yaw. However we need the body angular accelerations in equation (3.4), so it will need to be transformed to the body axes reference frame [25]. The transformation has been made in equation (3.6).

The proportional plus derivative dynamics for the yaw plane is obtained by:

$$Upd\_\psi = Kp\left(\psi_{com\_f} - \psi\right) + Kd\left(\dot{\psi}_{com\_f} - \dot{\psi}\right) \tag{4.2}$$

The gains *Kp* and *Kd* are used to define the error dynamics. These dynamics are designed similar to those of the longitudinal channel. The selection of this parameter has been explained in Section 4.1.1.

The inversion error ε_{ψ} is defined in equation (3.9). So yaw angular acceleration is defined as:

$$\ddot{\psi} = U_{\psi} + \varepsilon_{\psi} \tag{4.3}$$

Combining equation (4.1), (4.2), and (4.3) we obtain:

$$\left(\ddot{\psi}_{com\_f} - \ddot{\psi}\right) + Kp\left(\psi_{com\_f} - \psi\right) + Kd\left(\dot{\psi}_{com\_f} - \dot{\psi}\right) = Uad\_\psi - \varepsilon_{\psi} \tag{4.4}$$

On the right hand side of equation (4.4), network compensation error is generated. In the ideal case, it is equal to 0. It is expected that ANN output cancels the model inversion error. If it occurs, consider the desired error dynamics (i.e. neural network output cancels the inversion error), equation (4.4) becomes:

$$\left(\ddot{\psi}_{com\_f} - \ddot{\psi}_{desired}\right) + Kp\left(\psi_{com\_f} - \psi_{desired}\right) + Kd\left(\dot{\psi}_{com\_f} - \dot{\psi}_{desired}\right) = 0 \tag{4.5}$$

Details of neural network architecture will be described in Section 4.2.

4.1.1 Command Filter Design and Error Dynamics

The command filter must be slower than the missile error dynamics on the yaw attitude control. Therefore, the design process is similar to those on the longitudinal channel. Firstly, the yaw attitude tracking error is defined as $\tilde{\psi} = \psi_{com\_f} - \psi$ and the Laplace transform is performed.

$$
\begin{aligned}
& s^2\tilde{\psi} + Kd \cdot s\tilde{\psi} + Kp \cdot \tilde{\psi} = Uad\_\psi - \varepsilon_\psi \\
& \Updownarrow \\
& s^2\tilde{\psi} + \left(2 \cdot \zeta_{err} \cdot w_{n_{err}}\right) \cdot s\tilde{\psi} + {w_{n_{err}}}^2 \cdot \tilde{\psi} = Uad\_\psi - \varepsilon_\psi
\end{aligned}
\tag{4.6}
$$

Then the transfer function from the inversion error compensation of yaw motion to yaw attitude error becomes:

$$
\frac{\tilde{\psi}(s)}{\left(Uad\_\psi - \varepsilon_\psi\right)(s)} = \frac{1}{s^2 + \left(2 \cdot \zeta_{err} \cdot w_{n_{err}}\right) \cdot s + {w_{n_{err}}}^2}
\tag{4.7}
$$

where $w_{n_{err}}$ denotes natural frequency, ζ_{err} is damping ratio of the error dynamics. From equation (4.7), the relationship for error dynamics can be taken as an under-damped second-order system similar to methodology described in Section 3.2.2.

Using time scale separation, the command filter must have slower response than those of error dynamics. Finding the error dynamics of the missile, the natural frequency and damping ratio of the missile frame (*AF*) on directional plane must be found. According to [93], since the missile is symmetric in the x-z and x-y planes, the missile airframe natural frequency (w_{AF}) and damping ratio (ζ_{AF}) for directional motion can be found as:

$$
w_{AF} = \sqrt{N_\beta} = \sqrt{\frac{\frac{1}{2}\left(C_{n\beta} + C_{Y\beta}\frac{X_{cg}}{lref}\right)\rho V^2{}_{missile}\ Sref\ lref}{I_{zz}}} = 6.25\ {}^{rad}\!/_{\sec}
\tag{4.8}
$$

$$
\zeta_{AF} = \frac{Y_\beta w_{AF}}{2N_\beta} = \frac{\frac{\frac{1}{2}C_{Y\beta}\ \rho V^2{}_{missile}\ Sref\ \ w_{AF}}{m\ V_{missile}}}{2 \times \frac{\frac{1}{2}\left(C_{n\beta} + C_{Y\beta}\frac{X_{cg}}{lref}\right)\rho V^2{}_{missile}\ Sref\ \ lref}{I_{zz}}} = 0.3
\tag{4.9}
$$

Aerodynamic parameters in the above equations are found using in the trim conditions. After finding missile airframe natural frequency for directional channel from equation (4.8) and damping ratio from equation (4.9), the error dynamics can be determined.

According to reasons similar to those described in Section 3.2.2, the calculated damping ratio can not be selected for missile directional dynamics.

As a result, we consider the 2% design criteria for settling time similar to those for pitch attitude. Then the desired damping ratio (ζ_{AF}) is selected as 0.75 instead of 0.3.

Since actual natural frequency is generally not attainable, the error dynamics is considered slower than the missile dynamics. It should be noted that it has similarity with longitudinal autopilot. So they are selected for obeying the following criterion (*):

$$w_{n\,err} < w_{AF} \quad and \quad \zeta_{err} \approx \zeta_{AF} \qquad (*)$$

To sum up, by using relation (*), equation (4.6) and (3.15), the parameters related to error dynamics are tabulated in Table 4-1.

Table 4-1 Missile Parameters Related to Directional Error Dynamics

Airframe Natural Frequency (w_{AF})	6.25 rad/sec
Airframe Damping Ratio (ζ_{AF})	0.75
Error Dynamics Natural Frequency ($w_{n_{err}}$)	5.0 rad/sec
Error Dynamics Damping Ratio (ζ_{err})	0.75
Error Dynamics Settling Time (t_s)	1.07 sec
Error Dynamics Maximum % Overshoot	2.84
Error Dynamics Rise Time (t_r)	0.73 sec
$Kd = 2 \cdot \zeta_{err} \cdot w_{n_{err}}$	7.50

$Kp = w_{n_{err}}{}^2$	25

Next, making the command filter response slower compared to error dynamics, command filter transient response characteristics can be determined according to the relations (**):

$$w_{CF} < w_{n_{err}} \quad and \quad \zeta_{CF} \approx \zeta_{err} \qquad (**)$$

To sum up, by using relations (**), equation (3.15), the parameters related to command filter are tabulated in Table 4-2.

Table 4-2 Directional Command Filter Parameters

Natural Frequency (w_{CF})	4.35 rad/sec^2
Damping Ratio (ζ_{CF})	0.75
Settling Time (t_s)	1.23 sec
Maximum % Overshoot	2.84
Rise Time (t_r)	0.84 sec

$$\frac{\psi_{com\_f}}{\psi_{com}} = \frac{w_{CF}{}^2}{s^2 + 2\zeta_{CF} w_{CF} s + w_{CF}{}^2} \qquad (4.10)$$

If the determined values are inserted in equation (4.10), the command filter transfer function is obtained as:

$$\frac{\psi_{com\_f}}{\psi_{com}} = \frac{4.35^2}{s^2 + (2 \cdot 4.35 \cdot 0.75)s + 4.35^2} = \frac{18.9225}{s^2 + 6.5250s + 18.9225} \qquad (4.11)$$

4.2 Online Learning Neural Network Architecture

The neural network design process in Section 3.3 is applied to the directional autopilot. Numerical results using the nonlinear 5-DOF simulation will also be presented and compared for each network in Chapter 6.

4.2.1 Linear In the Parameters Sigma-Pi Neural Network

Here neuron interactions are also provided using Kronecker products [19] and Wi's constitute a vector of neural network weights. The basis functions Vi are constructed by grouping normalized inputs into three categories due to similar reasons that were described in Section 3.3.1.

The first group is defined as:

$$C_1 = \begin{bmatrix} Bias & V_{missile} & V_{missile}{}^2 \end{bmatrix} \tag{4.12}$$

The second group consists of normalized directional and lateral state variables, the pseudo control and a bias term.

$$C_2 = \begin{bmatrix} Bias & v & p & r & \phi & \psi & U_\psi \end{bmatrix} \tag{4.13}$$

Similar to longitudinal NN, squashing function for directional NN used in this case study is:

$$Squashing\ \ Function = \left(\left(\frac{1}{1+e^{-0.5 \times U_\psi}} \right) - 0.5 \right) \cdot 2 \tag{4.14}$$

The third group is used to approximate effects of transformation between the body frame and the inertial frame by including changes in heading angle.

$$C_2 = \begin{bmatrix} Bias & \psi \end{bmatrix} \tag{4.15}$$

It should be noted that for all categories, bias values are selected according to scaled (normalized) values of the inputs.

Finally, the vectors of basis functions are composed of all possible products of the elements belonging to each group of input (C_1, C_2, and C_3) and are related to each other via the Kronecker product.

$$\beta = kron(kron(C_1, C_2), C_3) \tag{4.16}$$

Similar to the longitudinal network, the relationship between the inputs and outputs of the directional network is established by the following equation:

$$U_{AD\_\psi} = W^T \cdot \beta\left(\overline{X}, U_\psi, bias\right) \tag{4.17}$$

By equation (4.17), the model inversion error can be accurately produced at the network output. Adaptation of weights in the neural network is derived using Lyapunov theorem similar to that described in Chapter 3. The adaptation rule is formed as:

$$\dot{W} = -\Gamma_{learning} \cdot \begin{bmatrix} \tilde{\psi} \\ \dot{\tilde{\psi}} \end{bmatrix}^T \cdot P \cdot b \cdot \beta - \mu \cdot \left\| \begin{bmatrix} \tilde{\psi} \\ \dot{\tilde{\psi}} \end{bmatrix}^T \cdot P \cdot b \right\| \cdot W \tag{4.18}$$

In Section 3.3.1, b was defined in equation (3.29) and P was defined in equation (3.31). Equation (4.19) represents the error dynamics.

$$\begin{bmatrix} \dot{\tilde{\psi}} \\ \ddot{\tilde{\psi}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -Kp & -Kd \end{bmatrix} \cdot \begin{bmatrix} \tilde{\psi} \\ \dot{\tilde{\psi}} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \left(Uad\_\psi - \varepsilon_\psi\right) = A \cdot \begin{bmatrix} \tilde{\psi} \\ \dot{\tilde{\psi}} \end{bmatrix} + b \cdot \left(Uad\_\psi - \varepsilon_\psi\right) \tag{4.19}$$

The selection process of acceptable values for the fixed parameters in the directional neural network such as $\Gamma_{learning}$ (learning rate) and μ (modification term) are identical to those in Chapter 3. Detailed explanations are given in [4], [9] and [48].

4.2.2 Single Hidden Layer Neural Networks

The architecture of this network is established similar to those described in Section 3.3.2.

It has an output *Uad_ψ* to eliminate model error in the directional channel.

$$Uad\_\psi = b_w \psi_{w,1} + \sum_{j=1}^{N2} W_{j,1} \sigma_j \left(b_v \psi_{v,j} + \sum_{i}^{N1} V_{i,j} X_i \right) \tag{4.20}$$

The scalar function σ_j is hidden layer activation function, selected identical to those in Section 3.3.2.

The relation between input and output of the SHL NN can be established in matrix form as:

$$Uad\_\psi(W, V, X) = W^T \sigma\left(V^T X\right) \tag{4.21}$$

In this architecture, the inputs to the network are selected as:

$$X = \begin{bmatrix} b_v & \tilde{\psi} & \dot{\tilde{\psi}} & \psi_{com\_f} & \dot{\psi}_{com\_f} & \ddot{\psi}_{com\_f} & Uad\_\psi & \|Z\|_F & v & r \end{bmatrix}^T \tag{4.22}$$

The number of inputs are equal to 10 (i.e. *N*1=10) and $b_v \geq 0$. $\| Z \|_F$ is the frobenius norm of the Z matrix that contains all tunable weight parameters. In this architecture, the same definitions are used which were described in Section 3.3.2.

Similar to longitudinal network, the adaptation law of the directional neural network weights are obtained as:

$$\dot{V} = -\Gamma_{learning} \cdot \left[X \cdot \begin{bmatrix} \tilde{\psi} \\ \dot{\tilde{\psi}} \end{bmatrix}^T \cdot P \cdot b \cdot W^T \cdot \sigma' + \mu \cdot \left\| \begin{bmatrix} \tilde{\psi} \\ \dot{\tilde{\psi}} \end{bmatrix}^T \cdot P \cdot b \right\| \cdot V \right] \tag{4.23}$$

$$\dot{W} = -\gamma_{learning} \cdot \left[\begin{bmatrix} \tilde{\psi} \\ \tilde{\dot{\psi}} \end{bmatrix}^T \cdot P \cdot b \cdot \left(\sigma - \sigma' \cdot V^T \cdot X \right) + \mu \cdot \left\| \begin{bmatrix} \tilde{\psi} \\ \tilde{\dot{\psi}} \end{bmatrix}^T \cdot P \cdot b \right\| \cdot W \right] \tag{4.24}$$

where $\Gamma_{learning}$ and $\gamma_{learning}$ are learning rates of the inner and outer layer neural networks respectively. $\Gamma_{learning}$, $\gamma_{learning}$ (learning rates) and μ (modification term) in adaptation rule are tuned via the same methodology described in Section 3.3.2.

The performance of missile directional controller with these neural network architectures will be given in Chapter 6.

CHAPTER 5

BACKGROUND LEARNING IMPLEMENTED NEURAL NETWORK ADAPTATION

The approach defined in the preceding chapters use only the instantaneous states to tune the adaptive gains [14]. Hence, the error parameterization has no real long term memory. That is, the adaptive element rapidly forgets its adaptation profile when the system changes domain of the state space [74]. As a result, they can not show any improvement in performance when performing maneuvers that have been performed previously.

In this chapter a new approach proposed in [14], [73], [74] is explained, known as Background Learning Neural Networks. This application can be separated into two: (1) Online Learning and (2) BL via recorded data. In other words, it combines distinct current online learning algorithms with a BL concept.

5.1 Background Learning Implemented Neural Network Architecture

In this method, recorded data is also used for learning. This increases available information for learning, thus give better results. It helps to overcome the ‘rank-1’ limitation (Section 5.1.1) and shows the properties of semi global learning [73]. Faster adaptation can be obtained for the variations in the flight dynamic regime. It provides the additional DOF to improve the control system performance when subsequently repeated commands are given [14]. Since neural networks weights will indicate quicker convergence to constant values, it can be concluded that the NN is able to adapt to the unknown model error faster when BL is used.

This approach has been applied in the longitudinal channel of the inner loop attitude controller of the altitude hold autopilot and also in the directional NN of the directional autopilot. Numerical results obtained from nonlinear 5-DOF simulation studies will be presented in Chapter 6 with the results compared with SHL NN architectures.

5.1.1 Rank-1 Limitation

Fact 1: A matrix of rank one has the simple form $A=u.v^T$. Where A is (n x m) matrix, u is (n x 1) vector and v is (m x 1) vector.

According to **Fact 1**, the rank of the NN weight dynamic is always at most 1. This is the case when only current data is used for NN training. Thus, Neural Network adaptation law can search its weights only along one direction in the underlying vector space at that instant [74]. It causes no performance improvement on the maneuvers that have been performed previously. Detailed explanation related to this topic can be found in [14], [73] and [74].

5.1.2 Combination of Online & Background Learning Law

Johnson et al [73] explained methods for BL that use both recorded and current data concurrently. However, BL yields undesirable effects on the response of online learning in these methods. Chowdhary et al [74] proved the elimination of this problem by using orthogonal projection method. In this study, the method of orthogonal projection is used for BL law. Online learning part has SHL NN architecture, which was described in Sections 3.3.2 and 4.2.2.

According to [74], the required conditions for no interaction between online and BL response can be expressed as:

$$\dot{W}_B^T \sigma = 0 \qquad (5.1)$$

$$\dot{V}_B^T \Gamma_{learning} X = 0 \tag{5.2}$$

In the above equation, the subscript B denotes the BL law. From equation (5.1) and (5.2), the orthogonal projection of the learning law for the W and the V matrix of the SHL NN can be found as:

$$\dot{W}_B = \left(I - \frac{\sigma \sigma^T}{\sigma^T \sigma} \right) \cdot \dot{W}_O \tag{5.3}$$

$$\dot{V}_B = \left(I - \frac{\Gamma_{learning} \cdot X \cdot X^T \cdot \Gamma_{learning}}{X^T \cdot \Gamma_{learning} \cdot \Gamma_{learning} \cdot X} \right) \cdot \dot{V}_O \tag{5.4}$$

In the above equation, the subscript *O* denotes the SHL NN learning law.

BL trains the SHL NN using both recorded data and current data concurrently to improve global learning behaviour of the NN. It also ensures long term adaptation. In this method, it is provided by current data and the data recorded that satisfies the selection criteria. Total BL can be found by summing the individual contributions of this recorded data point adaptation and then taking the orthogonal projection of total contribution into the null space of the online learning.

The combined online and BL law can be defined as:

$$\dot{W} = \dot{W}_B + \dot{W}_O = \left(\left(I - \frac{\sigma \sigma^T}{\sigma^T \sigma} \right) \cdot \sum_{i=1}^{p} \dot{W}_{Oi} \right) + \dot{W}_O \tag{5.5}$$

$$\dot{V} = \dot{V}_B + \dot{V}_O = \left(\left(I - \frac{\Gamma_{learning} \cdot X \cdot X^T \cdot \Gamma_{learning}}{X^T \cdot \Gamma_{learning} \cdot \Gamma_{learning} \cdot X} \right) \cdot \sum_{i=1}^{p} \dot{V}_{Oi} \right) + \dot{V}_O \tag{5.6}$$

If equation (3.41) and (3.42) are inserted into equation (5.5) and (5.6) respectively, the adaptation of the combined online and BL laws are obtained as follows:

$$\dot{W} = \left(\left(I - \frac{\sigma\sigma^T}{\sigma^T\sigma} \right) \cdot \sum_{i=1}^{p} \left(-\gamma_{learning} \cdot \begin{bmatrix} \left(W^T \cdot \sigma_i - \Delta_i\right) \cdot \left(\sigma_i - \sigma_i' \cdot V^T \cdot X_i\right) \\ +\mu \cdot \left\| \left(W^T \cdot \sigma_i - \Delta_i\right) \right\| \cdot W \end{bmatrix} \right) \right)$$
$$+ \left(-\gamma_{learning} \cdot \left[S \cdot \left(\sigma - \sigma' \cdot V^T \cdot X \right) + \mu \cdot \|S\| \cdot W \right] \right) \tag{5.7}$$

$$\dot{V} = \left(\left(I - \frac{\Gamma_{learning} \cdot X \cdot X^T \cdot \Gamma_{learning}}{X^T \cdot \Gamma_{learning} \cdot \Gamma_{learning} \cdot X} \right) \cdot \sum_{i=1}^{p} \left(-\Gamma_{learning} \cdot \begin{bmatrix} X_i \cdot \left(W^T \cdot \sigma_i - \Delta_i\right) \cdot W^T \cdot \sigma_i' \\ +\mu \cdot \left\| \left(W^T \cdot \sigma_i - \Delta_i\right) \right\| \cdot V \end{bmatrix} \right) \right)$$
$$+ \left(-\Gamma_{learning} \cdot \left[X \cdot S \cdot W^T \cdot \sigma' + \mu \cdot \|S\| \cdot V \right] \right) \tag{5.8}$$

Here,

$S = \begin{bmatrix} \tilde{\theta} \\ \dot{\tilde{\theta}} \end{bmatrix}^T \cdot P \cdot b$ is residual signal for longitudinal channel

$S = \begin{bmatrix} \tilde{\psi} \\ \dot{\tilde{\psi}} \end{bmatrix}^T \cdot P \cdot b$ is residual signal for directional channel

$\Delta_i = \ddot{\theta} - U_\theta$ is stored estimate of the model error for longitudinal channel

$\Delta_i = \ddot{\psi} - U_\psi$ is stored estimate of the model error for directional channel

When BL is on, the difference between the stored estimate of model error and the current estimate of model error reduces with time. The above equations indicate that the NN is concurrently adapting to various data points, exhibiting semi global learning [14].

Detailed description and proof of this NN architecture can be found in [14], [73] and [74].

5.1.3 Selection of Data Points for Background Learning

BL selection points are part of the design of the combined online and BL NN architecture [74]. In this study, the data point selection criterion is chosen as:

$$\frac{\left(X - X_p\right)^T \cdot \left(X - X_p\right)}{X^T X} > \varepsilon_X \tag{5.9}$$

In the above criterion, the subscript p denotes the index of the last data point recorded. It implies that new points are stored whenever the difference between the current input and the last recorded data point is greater than some specified amount [73]. In other words, the data candidates for storage must be sufficiently different from the last data point recorded [74]. Detailed explanation on some methods of selecting data points can be found in [73].

CHAPTER 6

SIMULATION RESULTS

In this chapter, we will examine the performances of the adaptive controllers using the nonlinear 5-DOF missile simulation code. ACAH is desired in both autopilots. The results of NN architectures on missile autopilots are included here. For these purposes, NN algorithms have been tested in two different scenarios in order to evaluate the performance of the implemented NN algorithm. The results are obtained by modeling and inverting the missile about the trim condition, the parameters of which are given in Table 3-1.

6.1 Comparison of LIP NN & SHL NN

In the first simulation scenario, the objective is to present comparisons of the results of the linearized model inversion controller without NN, and online learning neural networks (LIP NN and SHL NN) applied adaptive controllers for longitudinal and directional channel. The capability of adapting to errors caused by the linearized inverted model is demonstrated. The inversion error compensation is evaluated by comparison of the LIP NN and SHL NN results.

The results obtained for three different controllers are shown in Figures from 6.2 to 6.16. The missile is commanded to its first way point on the x-y plane and certain altitude on the –z direction, then required heading command is calculated on x-y plane and altitude command is generated according to the defined altitude profile. Required pitch angle command for the inner loop is calculated from the altitude error that is generated from the outer loop. The graphs given below represent the

comparison of the commanded input and output that the missile performs for three different controllers.

The simulation trajectory is shown in Figure 6.1. It is a scenario where the missile reaches an altitude following the terrain and finally hitting a moving target.

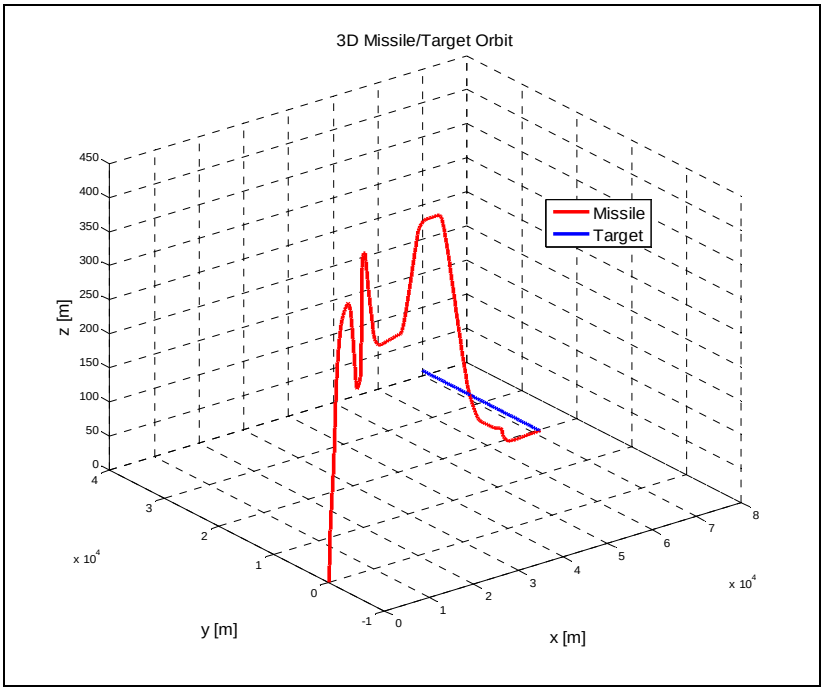

Figure 6.1 Missile and Target Trajectory

6.1.1 Pitch Angle Response

Figure 6.2 shows the missile pitch angle ($\theta_{missile}$) response with the associated command (θ_{com_f}) and Figure 6.3 shows the error (θ_{err}). It is observed that slightly better performance is achieved when the SHL NN based adaptive controller is used. On-line learning NNs improved the performance of the linear inverted controller.

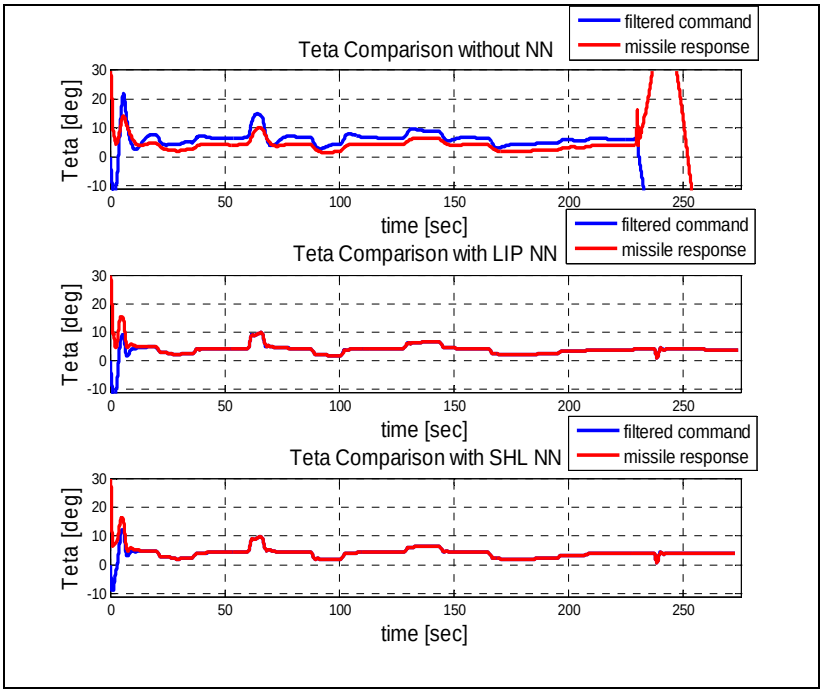

Figure 6.2 Pitch Angle ($\theta_{missile}$) Response Comparison

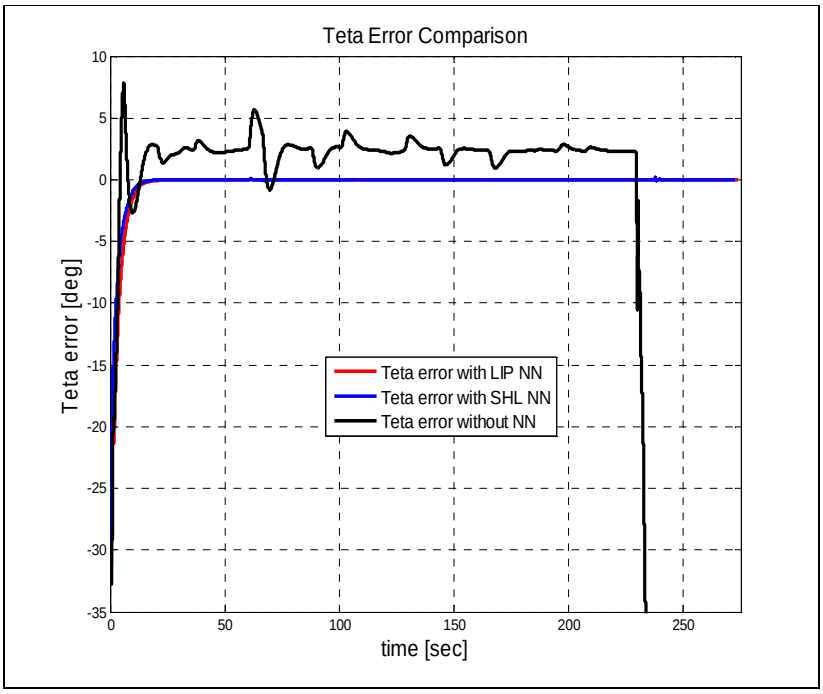

Figure 6.3 Pitch Angle Error (θ_{err}) Comparison

6.1.2 Yaw Angle Response

In Figure 6.4 and Figure 6.5 the performance of the controllers can be observed for the yaw angle response. It is observed that the SHL NN eliminates the model inversion error better than the LIP NN.

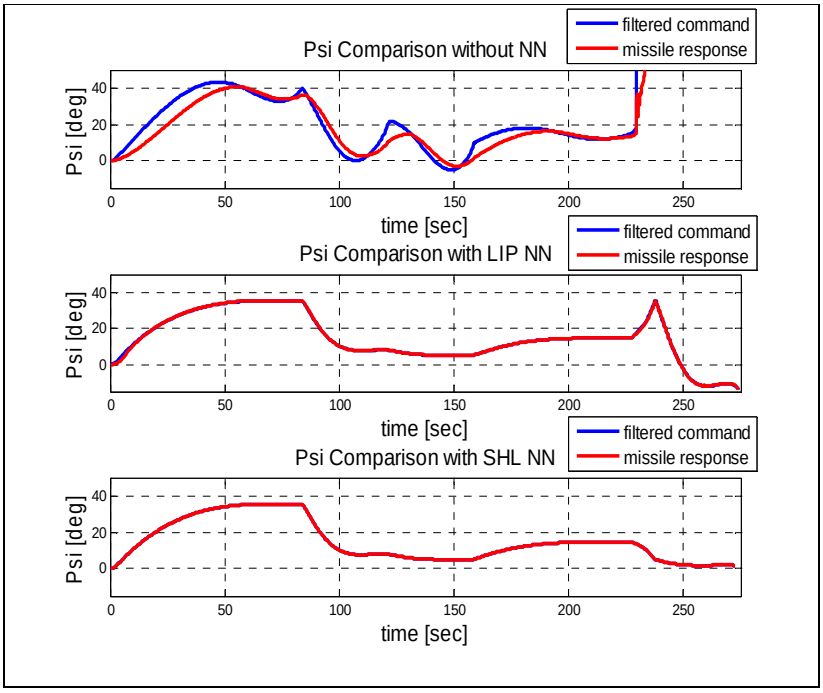

Figure 6.4 Yaw Angle (ψ) Response Comparison

In Figure 6.5, error compensation for yaw channel is demonstrated and both NN performances are acceptable. Since model tracking error is high for yaw channel inversion controller, using NN cancels these errors.

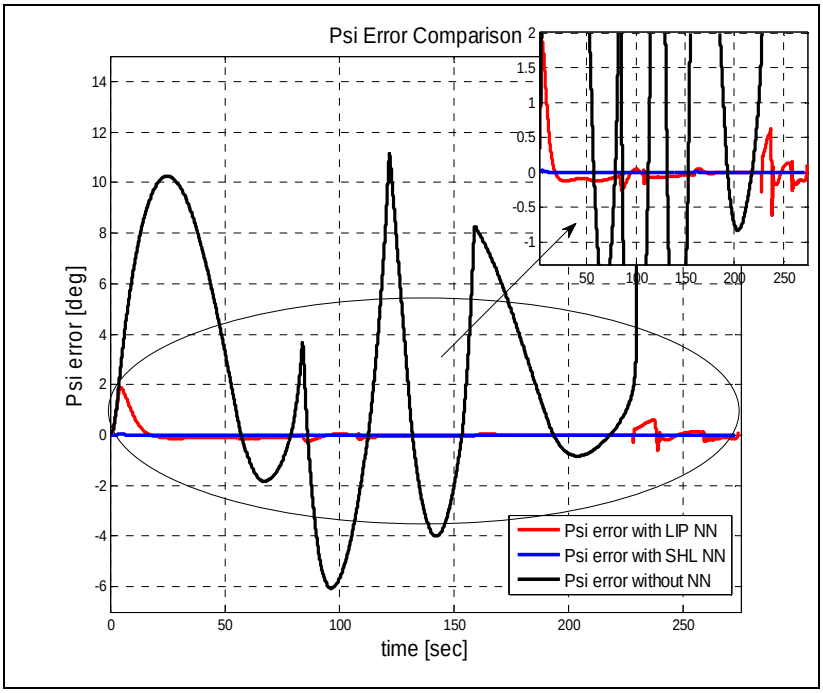

Figure 6.5 Yaw Angle Error (ψ_{err}) Comparison

6.1.3 Altitude Response

In Figure 6.6 and Figure 6.7, the altitude hold controller performance is shown for cases with LIP NN, SHL NN and without a neural network. The performance of the missile with SHL NN is seen to be better. Especially, examining the results of Figure 6.7, the error profile for SHL NN between the filtered command and the missile altitude is preferable. Examining Figure 6.6, the missile could not follow the desired trajectory, since nonlinear inverted controllers without NN commanded high deflection angles which are out of range on both channels and they could not be performed. Another advantage of the NN implementation to nonlinear controller is seen here, NN eliminates input saturations. As a result, the missile could follow the planned altitude and the mission was accomplished.

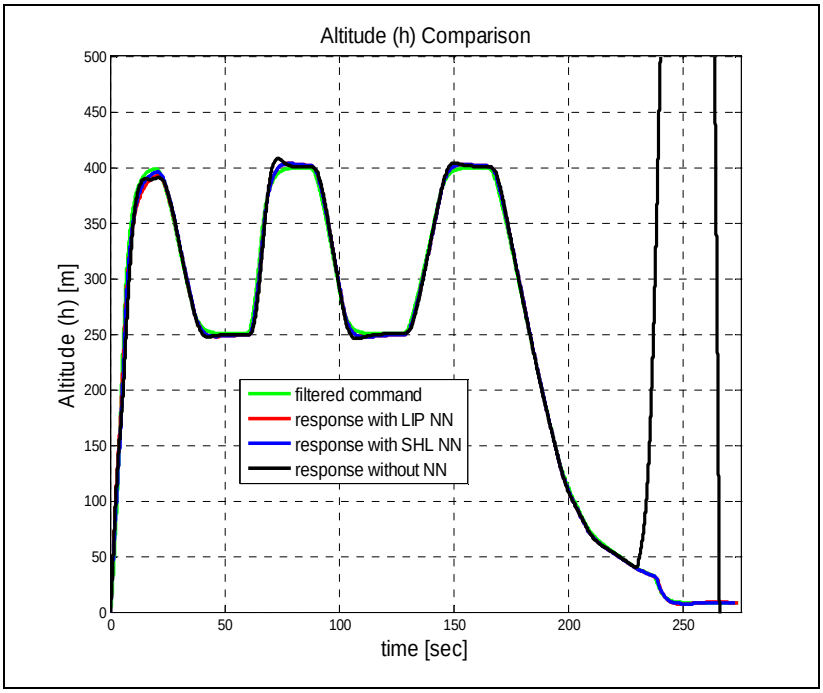

Figure 6.6 Altitudes (*h*) Following Comparison

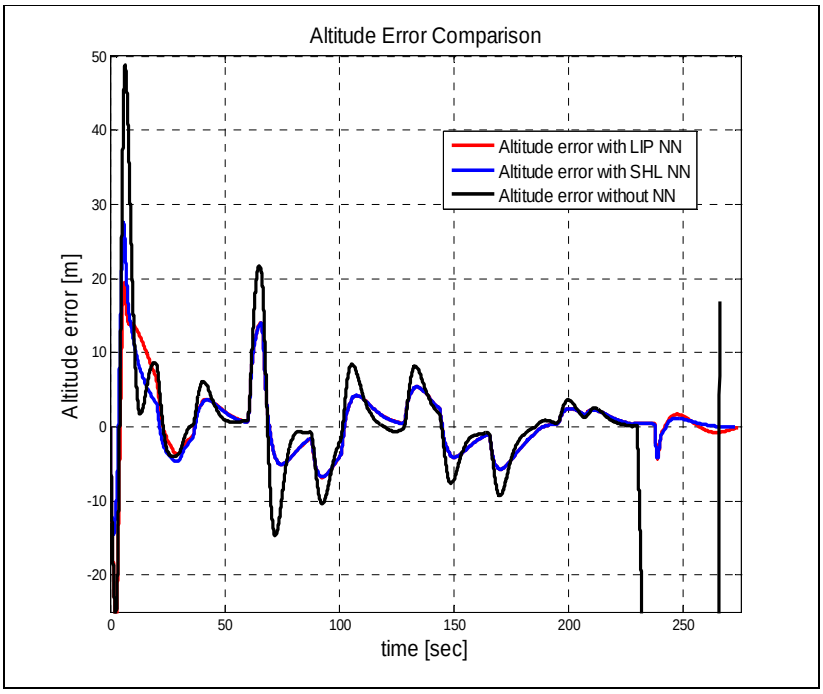

Figure 6.7 Altitude Error (h_{err}) Comparisons

6.1.4 Missile Mach number

Figure 6.8 shows that the Missile Mach number (M) is controlled by the Thrust Model described in Section 2.5 and held constant around the cruise velocity corresponding to M = 0.8. From this figure, it is seen that the system follows the M command better when a SHL NN is used.

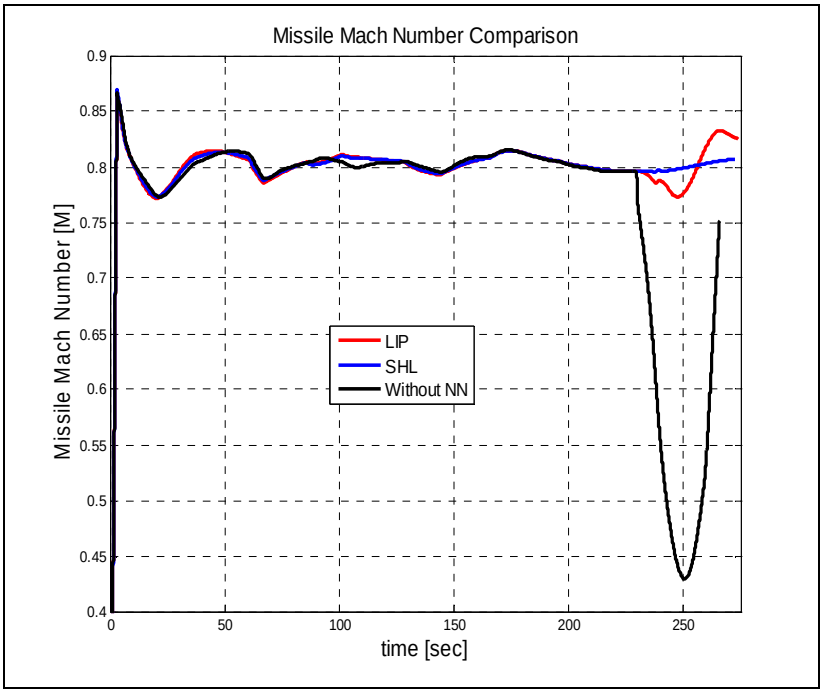

Figure 6.8 Missile Mach number

6.1.5 Adaptation Response

From Figure 6.9 to 6.12, the adaptation performance of the two different ANNs are shown. Adaptation signals generated from the SHL NN have lower bounds than those of the LIP NN. In addition, if Figure 6.11 and Figure 6.12 are compared, it can be observed that the SHL NN weights on the pitch (longitudinal) channel have faster convergence properties, and adaptation for yaw (directional) channel is faster.

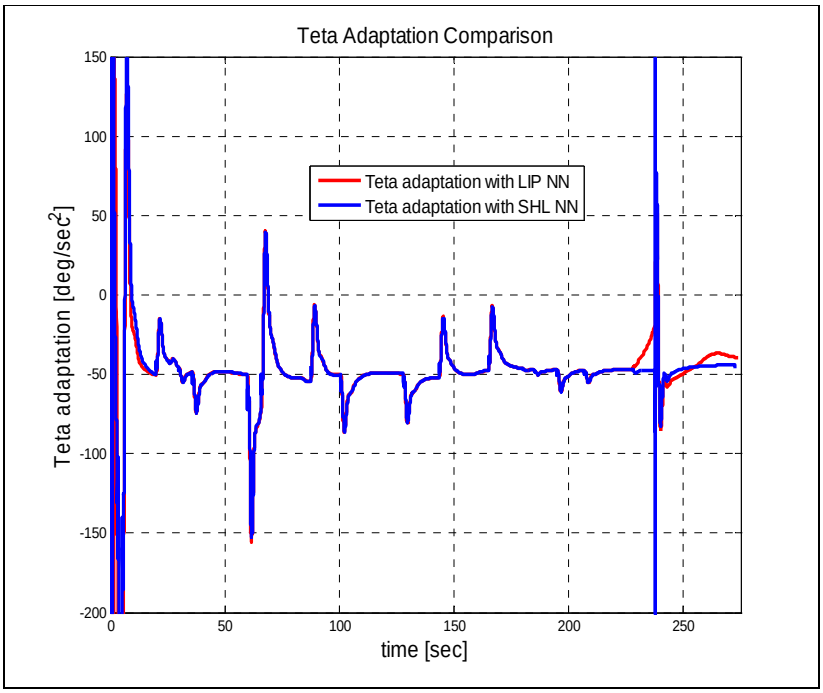

Figure 6.9 Adaptation signal in pitch channel (*Uad_θ*)

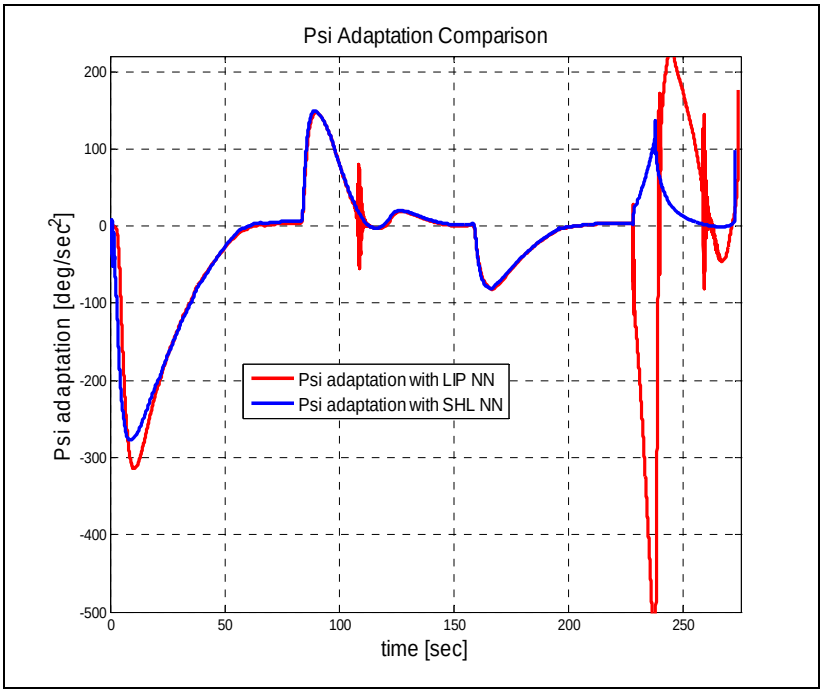

Figure 6.10 Adaptation signal in yaw channel (*Uad_ψ*)

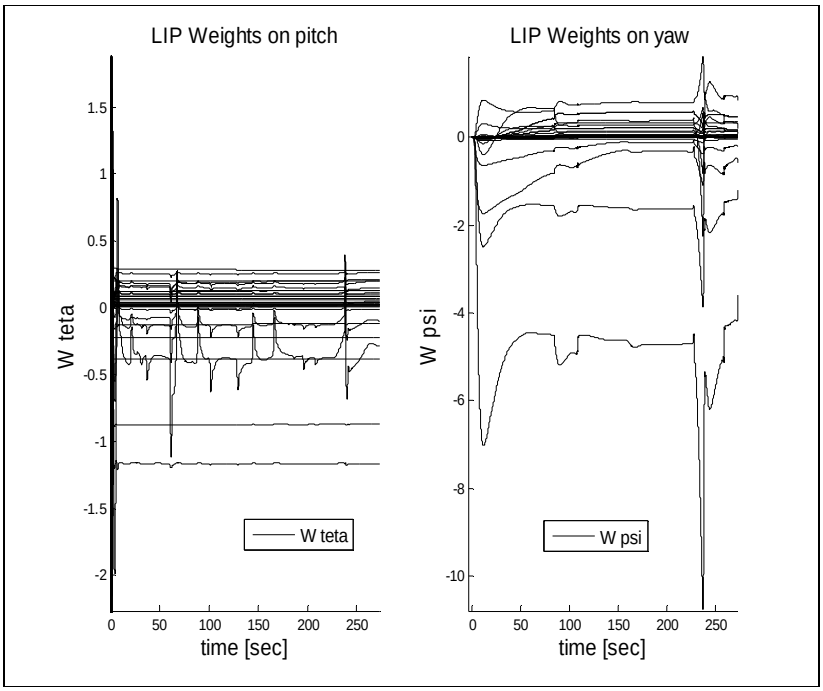

Figure 6.11 LIP NN Weights (W) in pitch and yaw channel

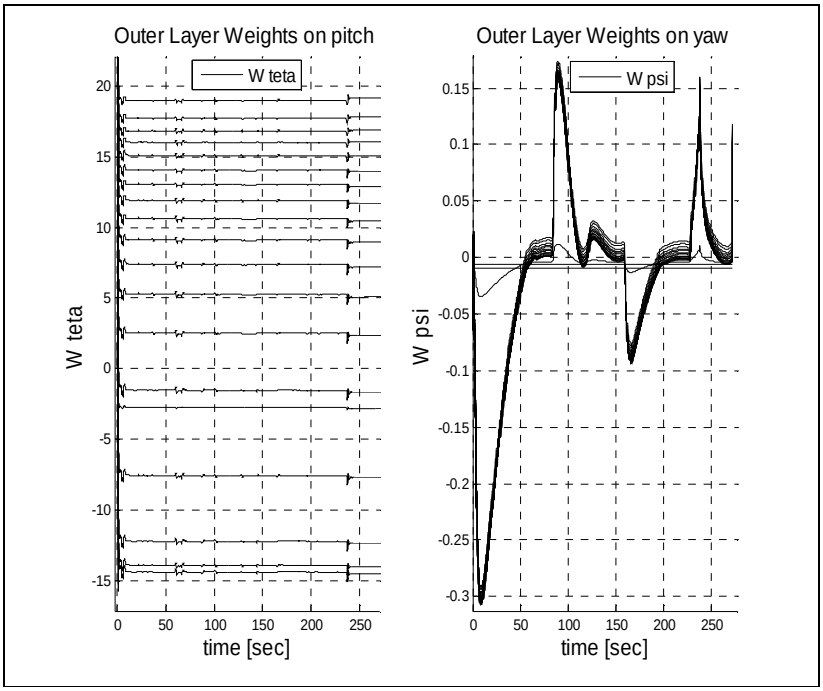

Figure 6.12 SHL NN Outer Layer Weights (W) in pitch and yaw channel

6.1.6 Actuator Deflections

In flight controller design, the measure of the autopilot command and rate is important and they must have limits in order to avoid from actuator saturations and flight instability. Although the actuator dynamics were not modelled, the deflection and deflection rates are limited in the simulation in order to see the relation between the behaviour of the commands and the missile response. From Figure 6.12 to 6.16, it is observed that the actuator deflection angle and rate commands for the required altitude and guided heading angle command are suitable for a typical actuator system when NN applied to nonlinear controller. If Figure 6.13 and 6.14 are observed, it can be seen that rudder and elevator deflection angle was out of range and in Figure 6.15 and 6.16; elevator and rudder deflection rates were too high and unstable when NN was not used, and this condition caused sudden changes in the

motion of the missile and so caused instability. On the other hand, the system gained stable characteristics by using NN.

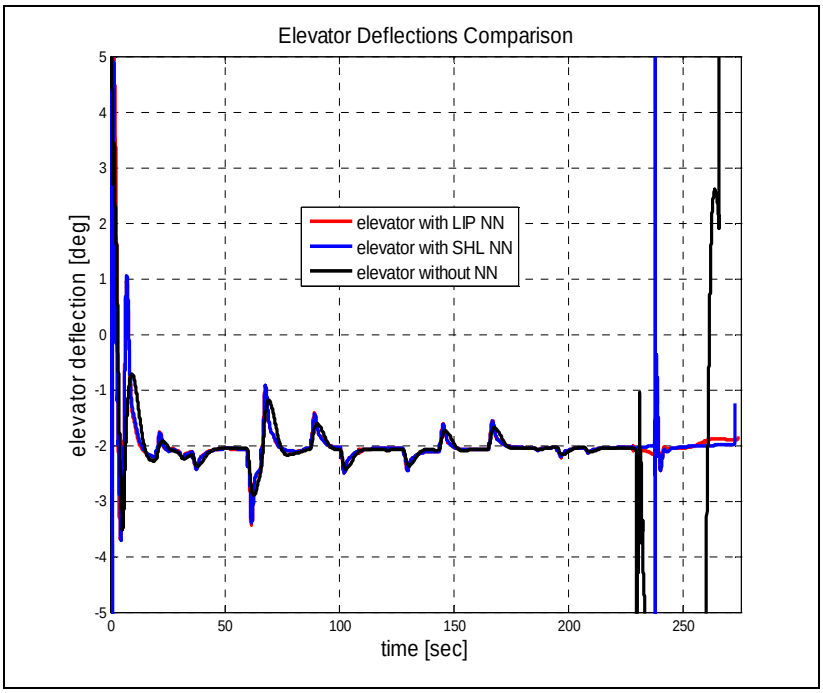

Figure 6.13 Elevator Deflections with Different Controller

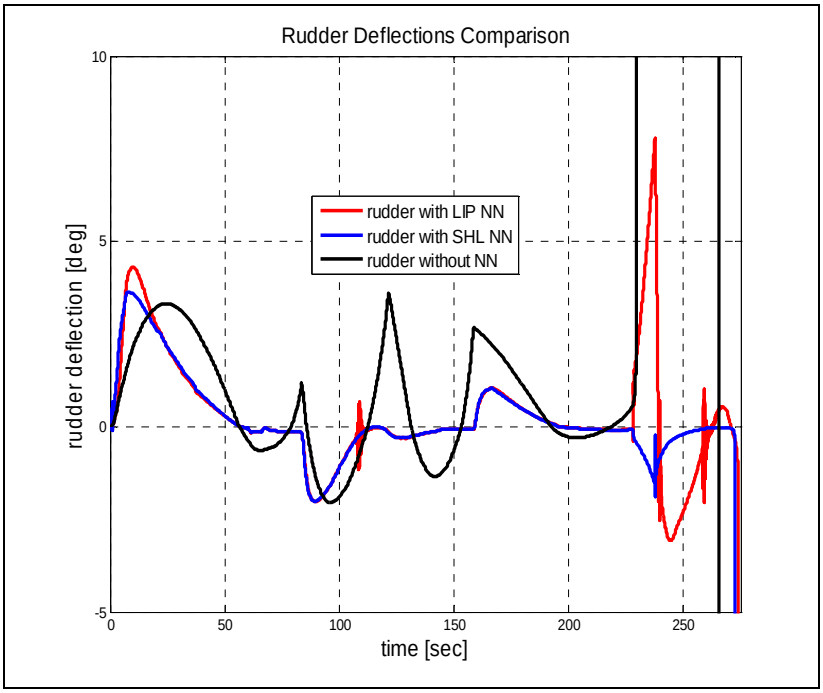

Figure 6.14 Rudder Deflections with Different Controller

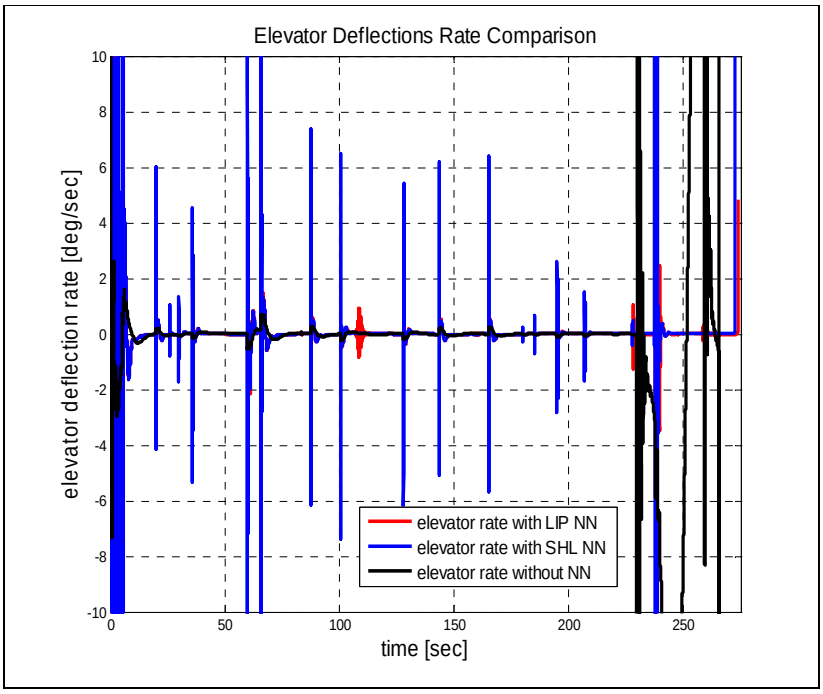

Figure 6.15 Elevator Deflections Rate with Different Controller

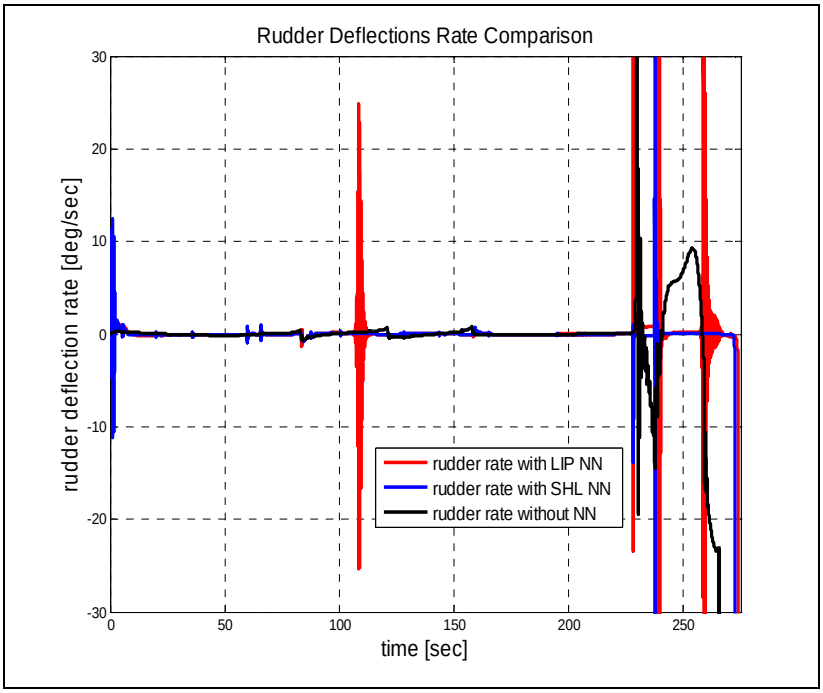

Figure 6.16 Rudder Deflections Rate with Different Controller

6.2 Comparison of SHL NN & BL implemented SHL NN

In the second scenario, the target of interest is to present comparisons of the results of the linearized model inversion controller without NN, on-line learning SHL NN and BL implemented SHL NN applied adaptive controllers for longitudinal (pitch) and directional (yaw) channels. Using NN for the pitch and yaw channels eliminate the inversion errors. In addition, the results show that background learning implementation enhances the performance of the adaptive controller when the missile performs a maneuver that has been recorded in the past. BL implementation to an online network is described in Chapter 5.

The ability of the algorithms is shown from Figure 6.18 to 6.34. The missile is commanded by a sinusoidal heading angle input on the x-y plane whereas its movement is determined by sinusoidal altitude input on the x–z plane. Sinusoidal

inputs are given to the missile autopilots; this type of maneuver is mostly difficult to perform by a missile. On the other hand, the aim of selecting such a scenario is to test the ability and observe the performance of the controllers when it encounters a repeated maneuver. Required pitch angle command for the inner loop is calculated from the altitude error that is generated from the outer loop.

The missile trajectory is illustrated in Figure 6.17. It is a scenario where the missile performs a repeated maneuver on the x-y plane and x-z plane.

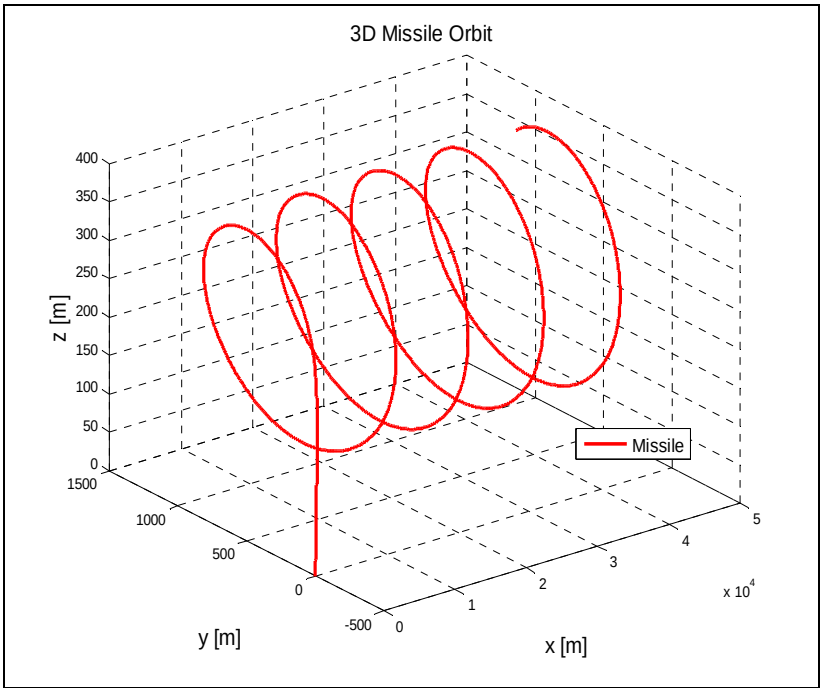

Figure 6.17 Missile Trajectory

6.2.1 Pitch Angle Response

Figure 6.18 shows the missile pitch angle ($\theta_{missile}$) response with the associated command (θ_{com_f}) and Figure 6.19 shows the error (θ_{err}). According to these results, on-line learning NNs improved the performance of linear inverted controller. It is

observed that better performance was achieved when the BL implemented SHL NN based adaptive controller was used. Background learning addition to on-line learning SHL NN improved the error compensation. The ability of the BL implemented SHL NN adaptive controller under a complex longitudinal missile motion showed better results. By recording past data and using those for NNs weight adaptation process, it is seen that the controller performance improves and exhibits long term learning.

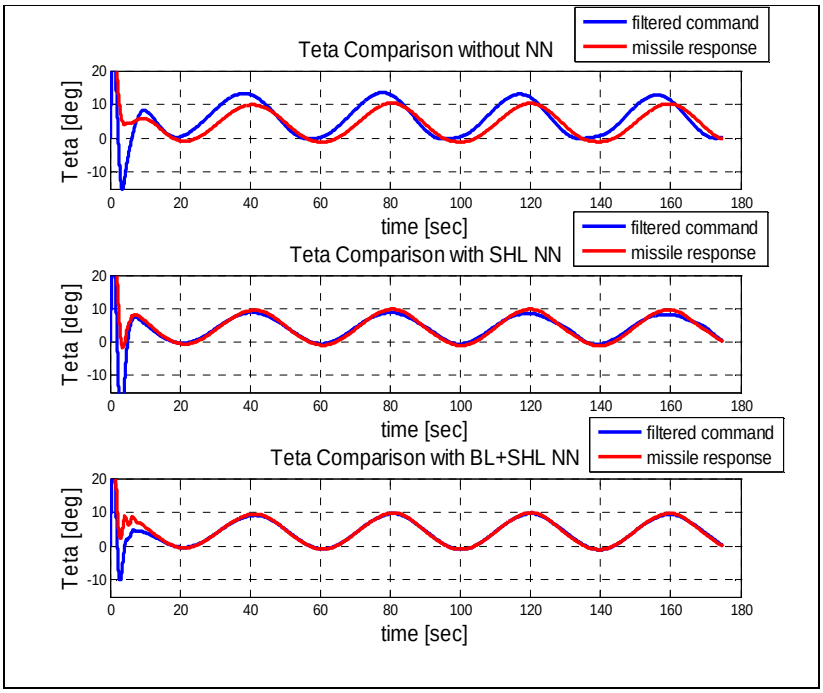

Figure 6.18 Pitch Angle ($\theta_{missile}$) Response Comparison

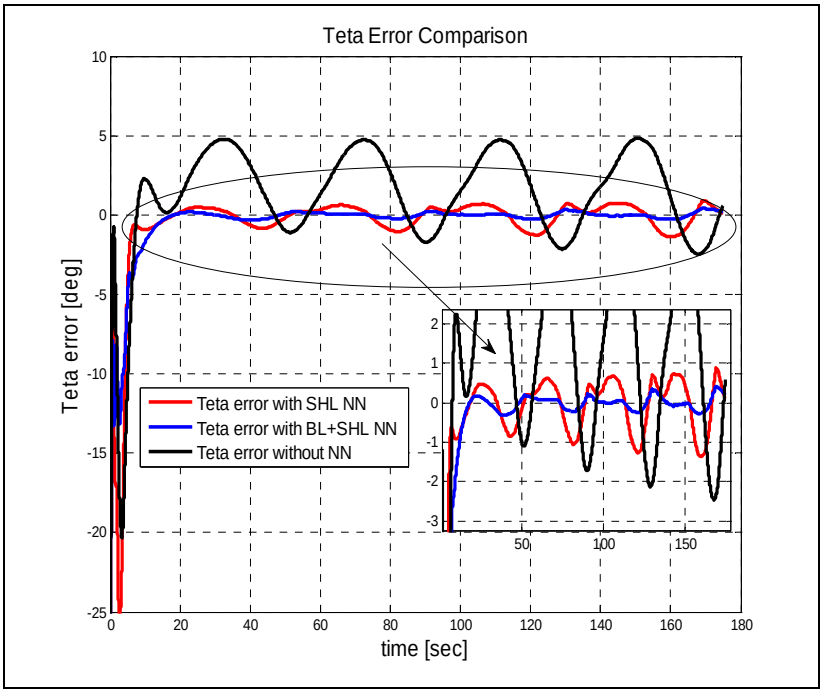

Figure 6.19 Pitch Angle (θ_{err}) Error Comparison

6.2.2 Yaw Angle Response

Plots in Figure 6.20 and Figure 6.21 present the performances of the controllers for yaw angle response. It is concluded that the BL implemented SHL NN eliminates the model inversion error better than the SHL NN for heading control. Hence BL augmentation provides long term memory and shows improvement in performance of the controller when performing maneuvers that have been performed previously.

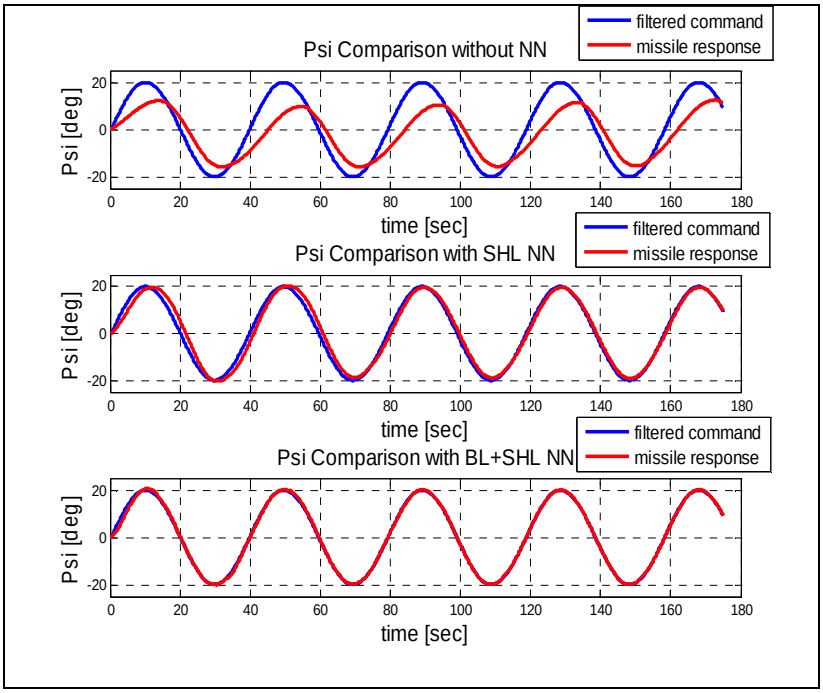

Figure 6.20 Yaw Angle ($\psi_{missile}$) Response Comparison

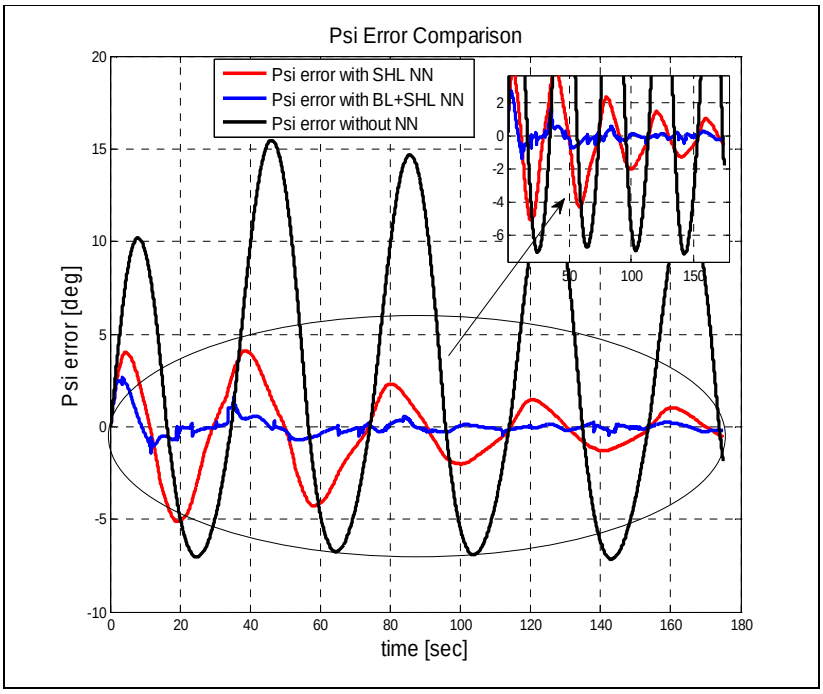

Figure 6.21 Yaw Angle (ψ_{err}) Error Comparison

6.2.3 Altitude Response

In Figure 6.22 and Figure 6.23, the altitude hold controller performance of the missile is shown for cases with SHL NN, BL implemented SHL NN and without a neural network. Especially, upon examining the results on Figure 6.23, the error profile for both NNs between the filtered command and the missile altitude has stable oscillation characteristics when sinusoidal input is commanded for both controls. Since classical gain control is applied to the outer loop of altitude autopilots, the improvement is not observed on error profile. If NN application was applied to the outer loop of the autopilot, the altitude response might improve.

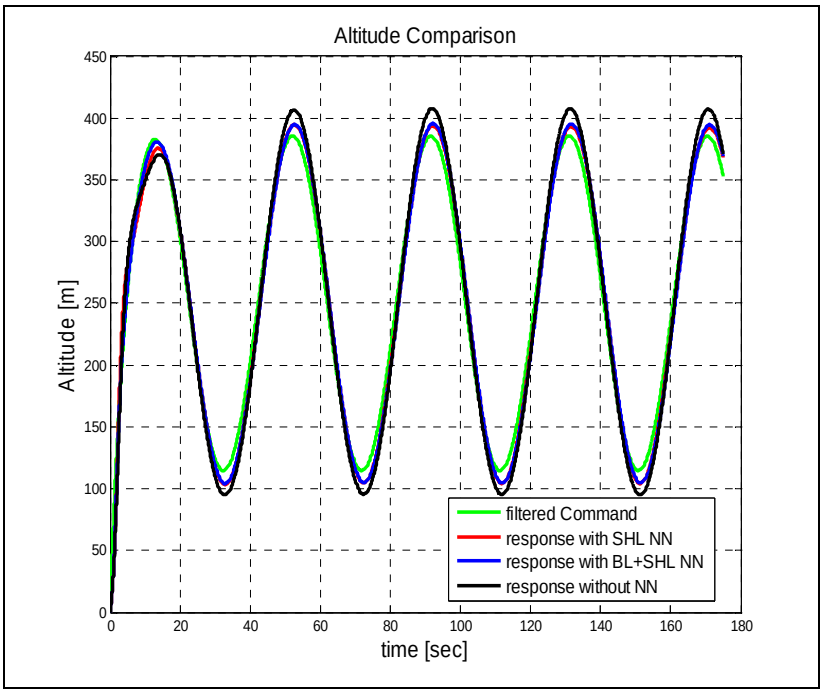

Figure 6.22 Altitudes (*h*) Following Comparison

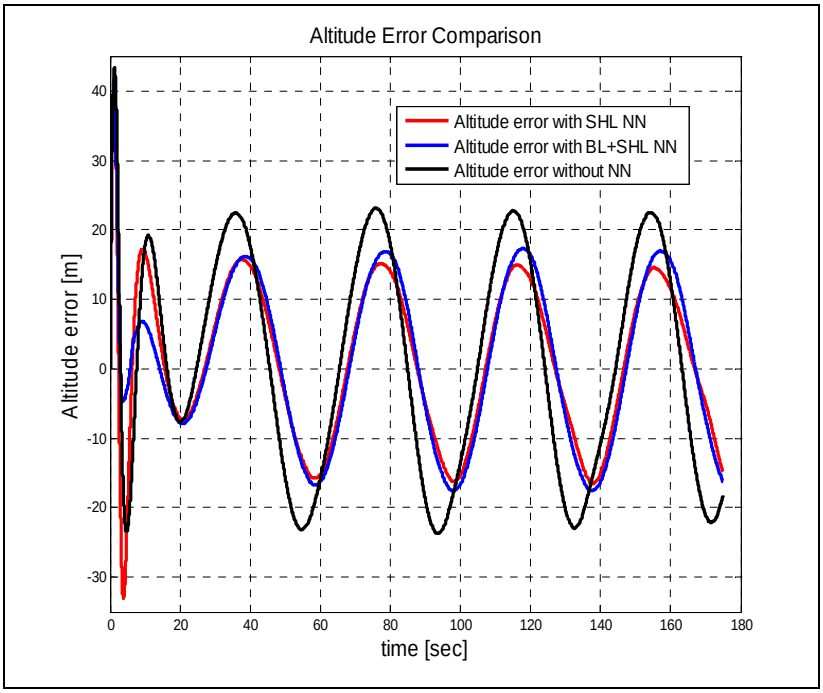

Figure 6.23 Altitude Error (h_{err}) Comparisons

6.2.4 Missile Mach number

Figure 6.24 indicates that as time proceeds, Mach varies within a range between 0.75 and 0.85. Missile Mach number (M) is also controlled to hold cruise velocity around 0.8 M. The control algorithm varies M related to the altitude change. From this figure, it is concluded that M variation is satisfactory and similar to each other as expected for all controllers.

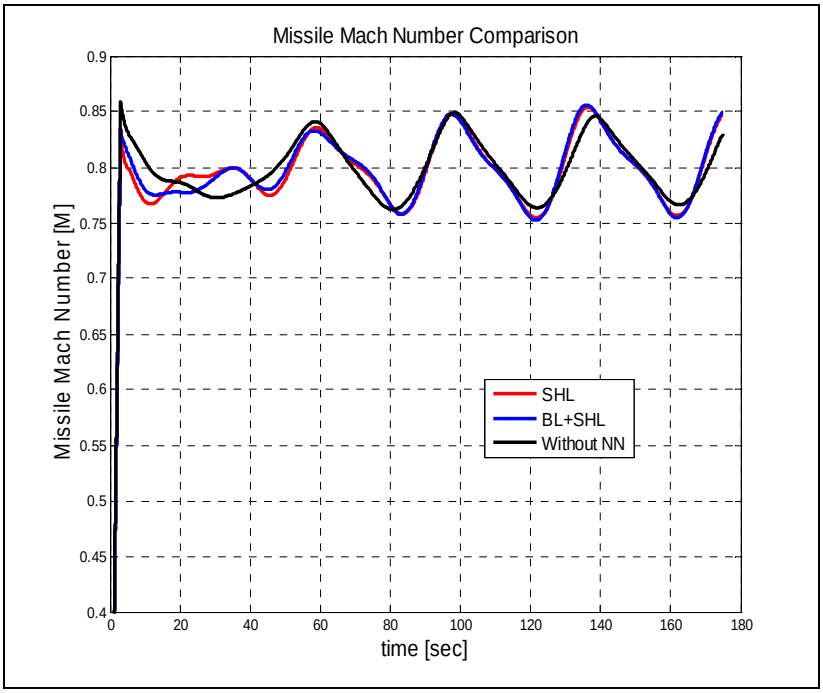

Figure 6.24 Missile Mach number

6.2.5 Adaptation Response

From Figure 6.25 to Figure 6.30, adaptation performance of the two different ANNs are shown. If BL augmented and without hidden (V) and outer (W) layer weights graphs are compared, it can be observed that the BL implementation provides faster convergence properties and long term adaptation of weights on both channels due to retaining the adaptation in the history. As a result, the adaptation signal generated from BL augmentation demonstrates sudden changes for yaw channel in Figure 6.26 to cancel unknown model errors without affecting the online adaptation. In addition, BL algorithms overcome the rank-1 limitation of weights. This allows the learning law to search for the ideal NN weights along any direction in the whole parameter space for various flight conditions which can be seen in Figure 6.28 and 6.30.

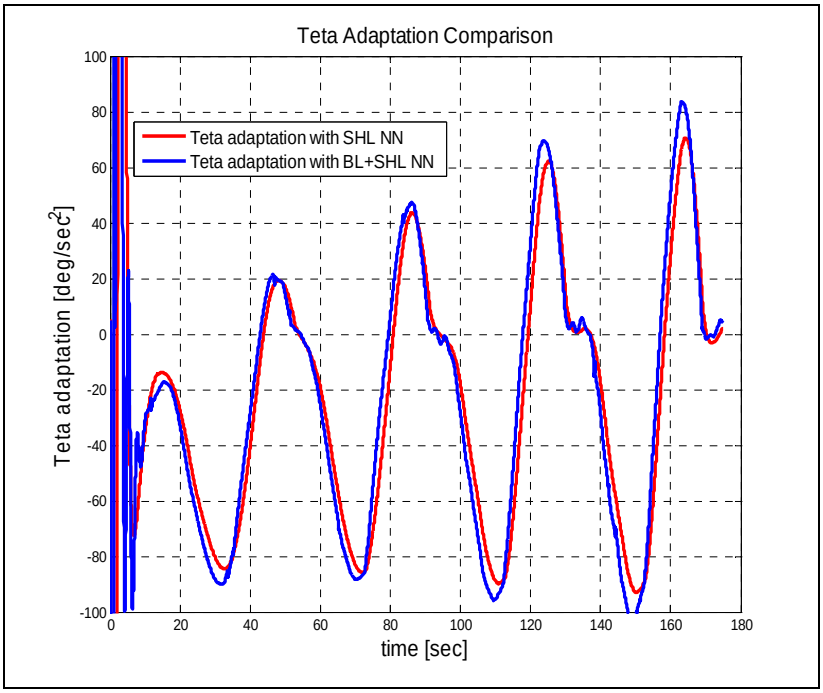

Figure 6.25 Adaptation signal in pitch channel (*Uad_θ*)

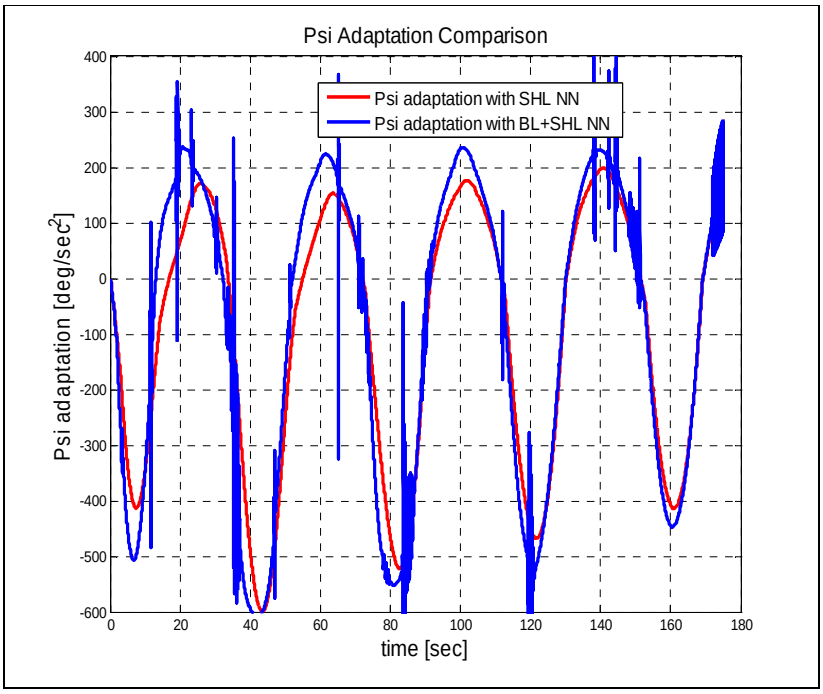

Figure 6.26 Adaptation signal in yaw channel (*Uad_ψ*)

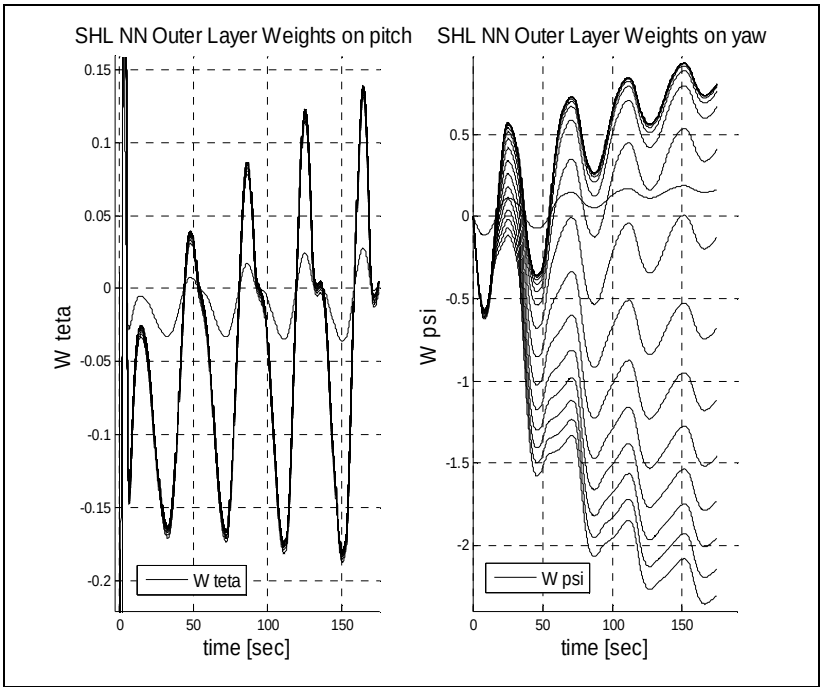

Figure 6.27 SHL NN Weights (W) in longitudinal and directional channel

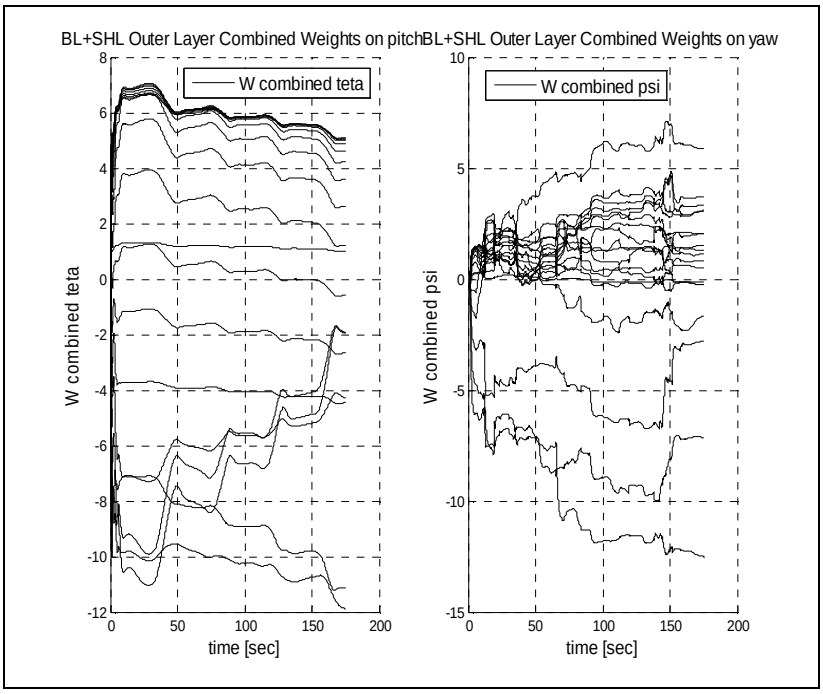

Figure 6.28 BL NN Weights (W) in longitudinal and directional channel

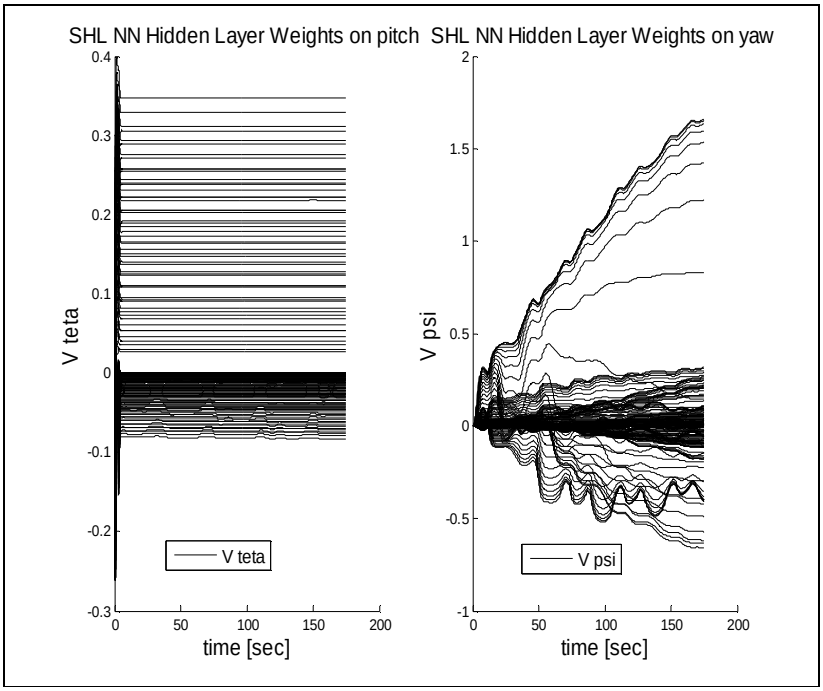

Figure 6.29 SHL NN Weights (V) in longitudinal and directional channel

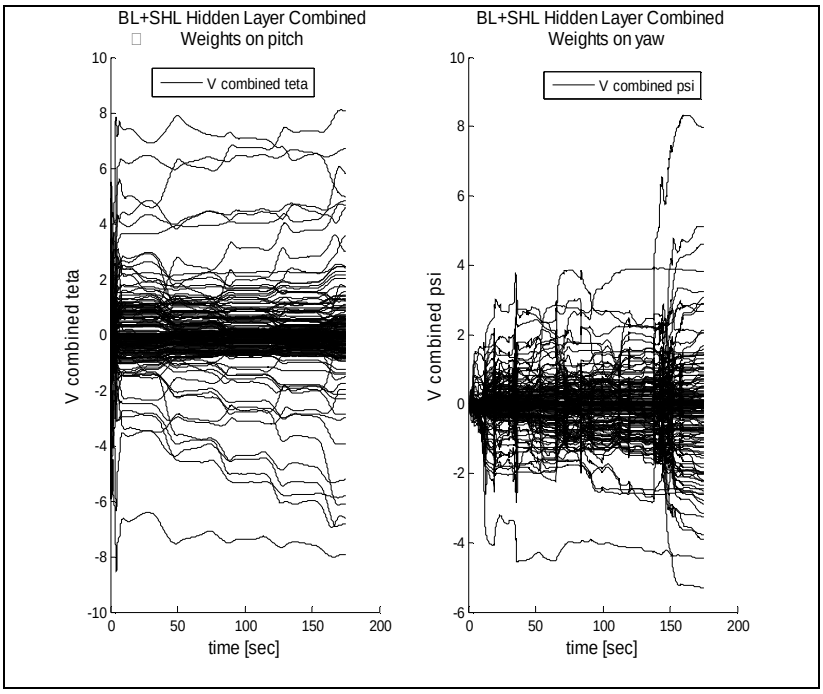

Figure 6.30 BL NN Weights (V) in longitudinal and directional channel

6.2.6 Actuator Deflections

The measure of the autopilot command and rate is important. Although the actuator dynamics were not modelled, by observing the graphs from Figure 6.31 to 6.34, the relation between the behaviour of the commands and the missile response can be realized. In Figures 6.31 and 6.32, it is observed that the actuator deflection responses of both controllers for the sinusoidal altitude and heading angle command are mostly suitable for a typical actuator system. However, looking at the response of BL implemented SHL NN controller, deflection rate values are high. The Pseudo Control Hedging could be used to eliminate this problem of the controller.

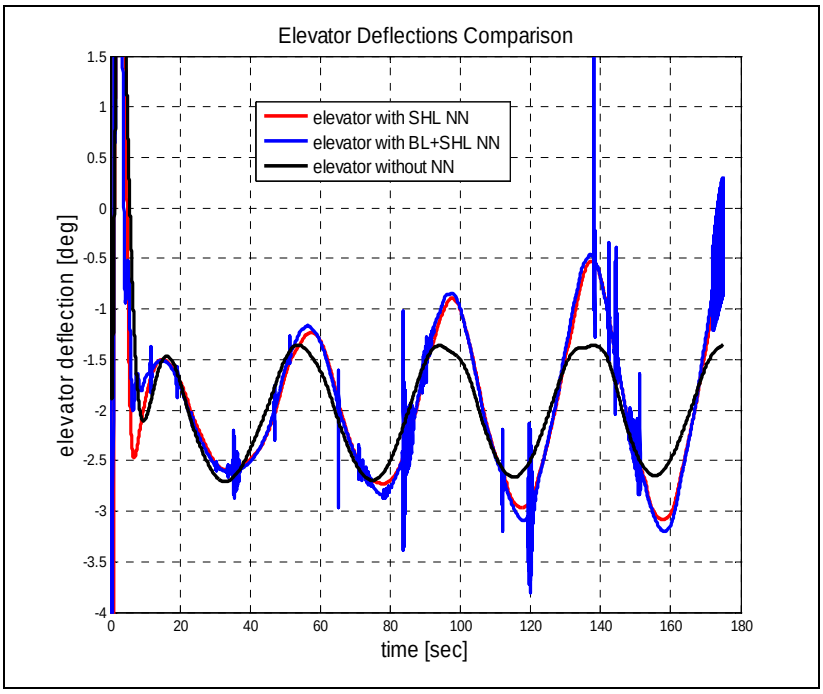

Figure 6.31 Elevator Deflections with Different Controller

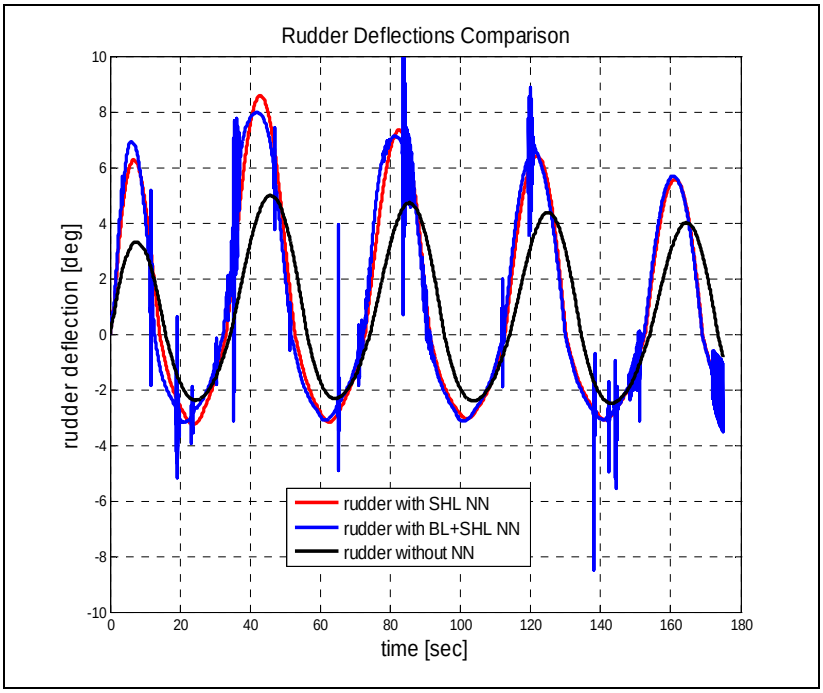

Figure 6.32 Rudder Deflections with Different Controller

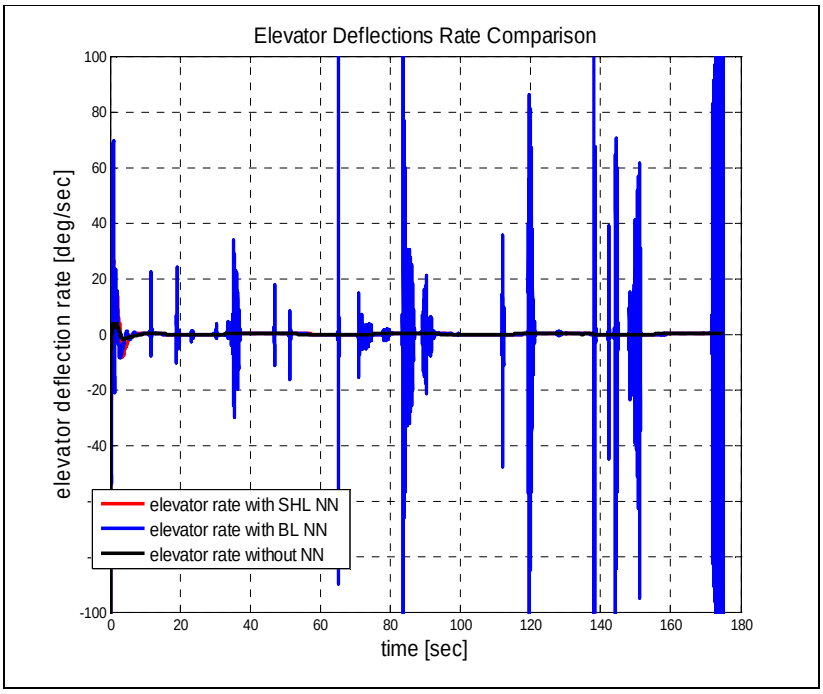

Figure 6.33 Elevator Deflections Rate with Different Controller

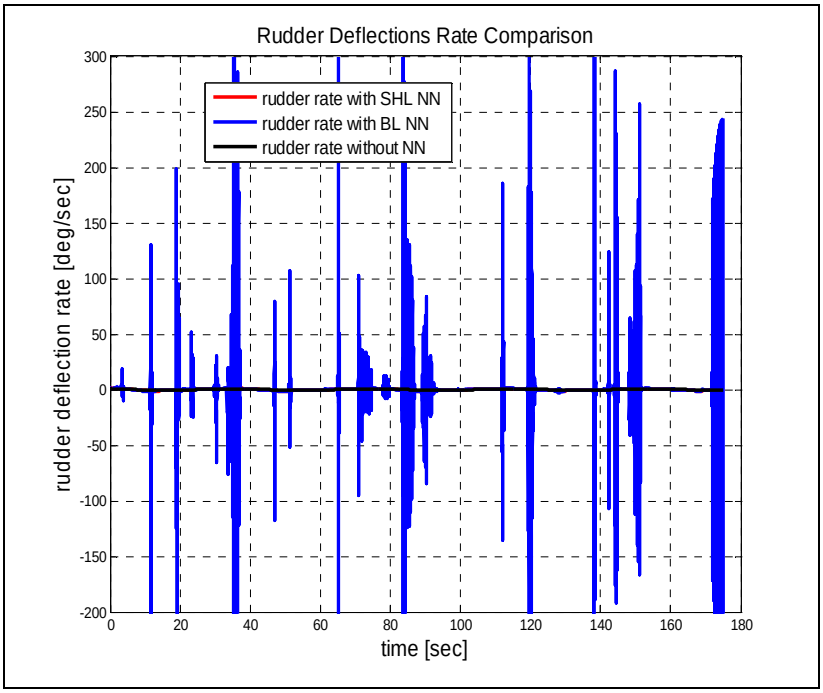

Figure 6.34 Rudder Deflections Rate with Different Controller

CHAPTER 7

CONCLUSION

This thesis presents the design and evaluation of missile autopilots using different neural networks structures and demonstrates the feasibility of applying these approaches to missile controller design. Various NN based adaptive controllers are studied on generic cruise missile autopilots. The architecture of the autopilots is based on feedback linearization (i.e. model inversion). System modeling uncertainties on missile dynamics result in control tracking errors due to approximate model inversion. NN adaptation attempts to cancel the inversion errors.

LIP NN, SHL NN, and BL implemented SHL NN architectures are applied to missile altitude hold and directional autopilots to cancel these errors by observing the tracking error between the commanded input to the missile and missile response. LIP NN and SHL NN uses only the current states and tracking errors to tune the adaptive gains, on the other hand BL implemented SHL NN uses recursive error based NN training by both the instantaneous and recorded states. To examine the NN effectiveness on inversion errors, simulations have been performed, but two of them have been chosen, which are related to the scope of this thesis. According to results obtained from nonlinear 5-DOF simulation studies, implemented NN algorithms have become successful on high subsonic cruise missile autopilots. From the first scenario, it is observed that SHL NN with an arbitrary number of neurons is better to cancel the errors than a LIP NN applied one. These results do not contradict with the results in [54] and [55]. In the second simulation scenario, the improvement of SHL NN implementation with aiding background learning

algorithm in its weight adaptation process has been demonstrated. The resulting BL adaptation retains the important information in the past which increases the rank of the NN training law (i.e searching NN weights more than one direction) and improves the performance of the control system by adding DOF. Moreover, BL algorithm provides concurrent adaptation to various data points; therefore BL implemented SHL NN exhibits semi global learning. Consequently, missile controller performance is improved when the missile performs a maneuver that is encountered in the history of the flight.

It can be seen that ANN controllers can accomplish appropriate compensation of modeling error in missile control. The differences in the responses of controller performances with and without these networks show that using ANN in nonlinear controllers improves performance. The availability of efficient and fast learning algorithms such as BL algorithm improves the NN performance. This design process avoids the need for pre-computation, storing and interpolation between a large numbers of feedback gains of a typical missile autopilot and compensates for nonlinearities and model uncertainties. In other words, these controllers help avoiding the time-consuming and tedious gain scheduling process. Especially, this control method can be beneficial for missiles with large flight envelopes, since the conventional gain scheduling techniques ultimately increases design time and cost.

The following recommendations can be given for future research on this topic:

- The degree of freedom of the flight simulation model can be increased. (i.e roll motion and roll autopilot can be added to the simulation)
- Some error modeling can be performed for missile sensors.
- The seeker can be modeled.
- Earth's curvature and rotation can be modeled. In that case, flight line will be always parallel to the earth surface. Related centrifugal and Coriolis accelerations due to Earth rotation must be taken into account in the equations of motion with respect to inertial reference frame.

- Atmospheric disturbances can be modeled.
- A wider research can be performed for alternative guidance algorithms.
- Although adaptive control can eliminate the need for accurate aerodynamic data, more accurate simulation which is verified with flight test results is needed to validate the design.
- Actuator model can be added to the simulation and Pseudo Control Hedging (PCH) technique can be applied to the autopilot model.

REFERENCES

[1] Avcıoğlu, H. Tolga, “A Tool for Trajectory Planning and Performance Verification of Cruise Missiles”, M.S Thesis Study at METU Aerospace Engineering Department, Sept 2000.

[2] Napolitano Marcello R., Kincheloe M., “On-line Learning Neural Network Controllers for Autopilot Systems”, AIAA /Journal of Guidance, Control, and Dynamics, Vol 33 No 6, Nov/Dec 1995.

[3] Johnson, E. N., Kannan, S. K., “Adaptive Trajectory Control for Autonomous Helicopters”, Georgia Institute of Technology School of Aerospace Engineering Publications, 2005.

[4] McFarland, M. B., Calise, A. J., “Neural Networks and Adaptive Nonlinear Control of Agile Anti air Missiles”, AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 23 No. 3, May/June 2000

[5] Calise, J., Rysdyk, T., “Nonlinear Adaptive Flight Control Using Neural Networks”, Georgia Institute of Technology School of Aerospace Engineering Publications, 1998

[6] Steinberg, M. L., “Comparison of Intelligent, Adaptive, and Nonlinear Flight Control Laws”, AIAA/ Journal of Guidance, Control, and Dynamics Vol. 24, No. 4, July–August 2001.

[7] Stevens, B.L., Lewis, F.L., “Aircraft Control and Simulation”, John Wiley & Sons, Inc., New York, 1992

[8] Johnson, E. N., Calise, A. J., “Limited Authority Adaptive Flight Control for Reusable Launch Vehicles”, AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 26,No. 6, Nov-Dec 2003, pp. 906-913.

[9] Calise, J., Sharma, M., “Adaptive Autopilot Design for Guided Munitions”, AIAA /Journal of Guidance, Control, and Dynamics, Vol 23 No 5, pp. 837-843, Sept/Oct 2000.

[10] Taeyoung, L., Youdan, K., “Nonlinear Adaptive Flight Control Using Backstepping and Neural Networks Controller”, AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 24 No. 4, July/August 2001

[11] Nesline, F.W., P. Zarchan, "Why Modern Controllers Go Unstable In Practice", AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 7, No. 4, 1984, p 495-500.

[12] Isidori, A., “Nonlinear Control Systems”, Third Edition, Springer-Verlag, London, 1995.

[13] Kim, B. S., Calise, A. J., “Nonlinear Flight Control Using Neural Networks”, AIAA /Journal of Guidance, Control, and Dynamics, Vol 20 No 1, pp. 26-33, Jan/Feb 1997.

[14] Chowdhary, G., Johnson, E. N., “Adaptive Neural Network Flight Control Using both Current and Recorded Data”, Proceedings of the 2007 AIAA Guidance Navigation and Control Conference, Hilton Head, South Carolina.

[15] Schumacher, C., and Khargonekar, P. P., “Stability Analysis of a Missile Control System with a Dynamic Inversion Controller,” AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 21, No. 3, 1998, pp. 508–515.

[16] Brinker, J. S., and Wise, K. A., “Stability and Flying Qualities Robustness of a Dynamic Inversion Aircraft Control Law”, AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 19, No. 6, pp. 1270-1277, 1996.

[17] Lewis, F. L., Yesildirek, A., and Liu, K., “Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance,” IEEE Transactions on Neural Networks, Vol. 7, No. 2, March 1996, pp. 388–399.

[18] Farrell, J. A., “Stability and Approximator Convergence in Nonparametric Nonlinear Adaptive Control,” IEEE Transactions on Neural Networks, Vol. 9, No. 5, 1998, pp. 1008–1020.

[19] Unnikrishnan, N., Balakrishnan, S. N., “Missile Longitudinal Autopilot Design Using a New Model-Following Robust Neuro-Adaptive Controller”, Proceedings of the 2004 AIAA Guidance, Navigation, and Control Conference, Providence, Rhode Island.

[20] Padhi, R., Balakrishnan, S. N. “Implementation of Pilot Commands in Aircraft Control: A New Dynamic Inversion Approach”, AIAA Guidance, Navigation, and Control Conference, Austin, Texas, Aug-11-14, 2003.

[21] Khalil, H., “Nonlinear Systems”, Prentice Hall, Inc., Second Edition, New Jersey, 1996

[22] Chen, F. C., Khalil, H. K, “Adaptive control of nonlinear systems using neural networks”, International Journal of Control, Vol. 55, No. 6, pp. 1299-1317, 1992.

[23] Yesildirek, A., Lewis, F. L., “Feedback linearization using neural networks”, Automatica , Vol. 31, No. 11, pp. 1659-1664, 1995.

[24] Calise, A., Lee, S., and Sharma, M., “Development of a reconfigurable flight control law for tailless aircraft,” AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 24, No. 5,2001, pp. 896-902.

[25] Rysdyk, T., Calise, J., “Adaptive Model Inversion Flight Control for Tiltrotor Aircraft”, American Helicopter Society 54th Annual Forum, May. 20-22 1998

[26] Leitner, J., Calise, A. J., and Prasad, J. V. R., "Analysis of Adaptive Neural Networks for Helicopter Flight Controls", Journal of Guidance, Control, and Dynamics, Vol. 20, No. 5, 1997, pp. 972–979.

[27] McFarland, M. B., Calise, A. J., "Neural-Adaptive Nonlinear Autopilot Design for an Agile Anti-Air Missile", AIAA Guidance, Navigation, and Control Conference, San Diego, California, July, 1996.

[28] Fu, Li-Chen, Chang, Wei-Der, Yang, Jung-Hua and Kuo, Te-Son, "Adaptive Robust Bank-to-Turn Missile Autopilot Design Using Neural Networks", AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 20, No. 2, March/April 1997.

[29] Johnson, E. N., "Limited Authority Adaptive Flight Control", Ph.D. Thesis, Georgia Institute of Technology, 2000.

[30] Hunt, K. J., "Neural Networks for Controller Systems, A Survey", Automatica, Vol. 28, No. 6, 1992, pp. 1083–1112.

[31] White, D. A., and Sofge, D., "Handbook of Intelligent Control", Van Nostrand Reinhold, New York, 1992, Chaps. 3, 5, 8, 12, 13.

[32] Balakrishnan, S. N., and Biega, V., "Adaptive Critic Based Neural Networks for Aircraft Optimal Control," AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 19, No. 4, July- August 1996, pp. 893–898

[33] Lian, K.-Y., Fu, L.-C., Chung, D.-M., and Kuo, T.-S., "Adaptive Robust Autopilot Design for Bank-to-Turn Aircraft," Proceedings of American Control Conference, 1993, pp. 1746–1750.

[34] Lane, S. H., and Stengel, R. F., "Flight Control Design Using Non-Linear Inverse Dynamics", *Automatica*, Vol. 24, No. 4, 1988, pp. 478–483.

[35] Romano, J. J., and Singh, S. N., "I-O Map Inversion, Zero Dynamics and Flight Control," IEEE Transactions on Aerospace and Electronic Systems, Vol. 26, No. 6, 1990, pp. 1022–1029.

[36] Calise, A. J., Lee, H., and Kim, N., "High bandwidth adaptive flight control", Proceedings of AIAA Guidance, Navigation, and Control Conference, August, 2000.

[37] Nardi, F. and Calise, A., "Robust Adaptive Nonlinear Control using Single Hidden Layer Neural Networks," Proceedings of the IEEE Conference on Decision and Control, 2000.

[38] Calise, A. J., Rysdyk, R. T., "Nonlinear Adaptive Control of Tiltrotor Aircraft Using Neural Networks", SAE/AIAA World Aviation Congress, Oct. 14-16 1997

[39] Calise, J., Rysdyk, T., "Adaptive Model Inversion Flight Control for Tiltrotor Aircraft", AIAA / Guidance, Navigation and Control Conference, Aug. 1997, Paper No: 97-3758.

[40] Singh, S. N., Yim, W., and Wells, W. R., "Direct Adaptive and Neural Control of Wing-Rock Motion of Slender Delta Wings," AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 18, No. 1, 1995, pp. 25–30.

[41] Wise, K. A., "Reconfigurable Systems for Tailless Fighter Aircraft – RESTORE", Final Report, AFRL-VA–WP-TR-99-3067.

[42] Wise, K., and Brinker, J., "Reconfigurable Flight Control for a Tailless Advanced Fighter Aircraft", Proc. of the AIAA GNC Conference, Boston, MA, August, 1998, pp. 75-87.

[43] Wise, K., Brinker, J., Calise, A., Enns, D., Elgersma, M., and Voulgaris, P., "Direct Adaptive Reconfigurable Flight Control For A Tailless Advanced Fighter Aircraft", Int. Journal of Robust Nonlinear Control, Special Issue on Reconfigurable Flight Control, Vol. 9, pp. 999-1012, 1999.

[44] Leitner, J., Calise, A., and Prasad, J. V. R., "Analysis of Adaptive Neural Networks for Helicopter Flight Controls," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Baltimore, MD, 1995, pp. 871-879

[45] Sharma, M., Calise, A. J. and Corban, J. E. "Application of an Adaptive Autopilot Design to a Family of Guided Munitions", Proceedings. Of the AIAA GNC Conference, Denver, CO, August, 2000, AIAA Paper No, 2000-3969.

[46] Sharma, M., and Calise, A. "Neural Network augmentation of Existing Linear Controllers", Proc. of the AIAA GNC Conf., Montreal, Canada, August, 2001, AIAA Paper No, 2001-4163.

[47] Corban, J., Burkemper, V., Holt, K., J., Calise, A. and Sharma, M., "Flight Test Of An Adaptive Autopilot For Precision Guided Munitions", Proc. of the AIAA Missile Sciences Conference, 2002.

[48] McFarland, M., "Adaptive Nonlinear Control of Missiles Using Neural Networks", Ph.D. thesis, Georgia Institute of technology, Atlanta, GA, July 1997.

[49] Kim, B.S., Calise, A.J., "Nonlinear Flight Control Using Neural Networks," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Scottsdale, AZ, 1994, pp. 930-940.

[50] Idan, M., Johnson, M., Calise, A. J., "A Hierarchical Approach to Adaptive Control for Improved Flight Safety", AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 25, no. 6, 2003, pp. 1012-1020.

[51] Brinker, J. S., Wise, K. A., "Flight Testing of Reconfigurable Control Law on the X-36 Tailless Aircraft", AIAA/ Journal Of Guidance, Control, and Dynamics Vol. 24, No. 5, September–October 2001.

[52] Sharma, M., Wise, K., Lavretsky, E., "Application and Flight Testing of an Adaptive Autopilot on Precision Guided Munitions", Proc. Of the 2006 AIAA Guidance, Navigation, and Control Conference, Keystone CO, August 2006.

[53] Wise, K., Lavretsky, E., Zimmerman, J., Francis-Jr., J., Dixon, D., and Whitehead, B., "Adaptive Flight Control of a Sensor Guided Munition", Proceedings. Of the AIAA Guidance, Navigation and Control Conference, 2005.

[54] Calise, A. J., Shin, Y., Johnson, M. D., "A Comparison Study of Classical and Neural Network Based Adaptive Control of Wing Rock", Proceedings of the 2004 AIAA Guidance, Navigation, and Control Conference, Providence, Rhode Island.

[55] Narendra, K. S., Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transactions on Neural Networks, Vol. 1, No. 1, 1990, pp. 4-27.

[56] Lewis, F.L., Jagannathan, S. and Yesildirek, A., "Neural Network Control of Robot Manipulators and Nonlinear Systems", Taylor and Francis, UK, 1999

[57] Krstic, M., Sun, J., and Kokotovic, P. V., "Control of feedback linearizable systems with input unmodeled dynamics", Proceedings of the 33rd Conference on Decision and Control, Lake Buena Vista, FL, pp 1633-1638, 1994.

[58] Krstic, M., Kanellakopoulos, I., and Kokotovic, P., Nonlinear and Adaptive Control Design, Wiley, New York, 1995, Chap. 2.

[59] McFarland, M. B., Rysdyk, R. T., and Calise, A. J., "Robust Adaptive Control Using Single – Hidden - layer Feedforward Neural Networks", Proceedings of the American Control Conference, pp 4178-4182, June 1999.

[60] Balakrishnan, S. N. and Huang, Z., "Robust Adaptive Critic Based Neurocontrollers for Helicopter with Unmodeled Uncertainties", Proceedings of the 2001 AIAA conference on Guidance, Navigation and Control.

[61] Huang, J., and Lin, C.-F., "Sliding Mode Control of HAVE DASH II Missile Systems", Proceedings of American Control Conference, 1993, pages: 183–187.

[62] McFarland, M. B., Calise, A. J., "Neural Networks for Stable Adaptive Control of Air-to-Air Missiles", Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA, Washington, DC, 1995, pp. 1280–1285.

[63] Tahk, M., Briggs, M., and Menon, P. K. A., "Application of Plant Inversion via State Feedback to Missile Autopilot Design," Proceedings of the 27th Conference on Decision and Control, 1986, pp. 730–735.

[64] Funahashi K., "On the Approximate Realization of Continuous Mappings by Neural Networks", Neural Networks, vol.2, pp.182-192, 1989.

[65] Hornik K., Stinchcombe M., White H., "Multilayer Feedforward Networks Are Universal Approximators", Neural Networks, vol.2, pp.359-366, 1989.

[66] McDowell, D. M., Irwin, G. W., and McConnell, G., "Online Neural Control Applied to a Bank-to-Turn Missile Autopilot," Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA, Washington, DC, 1995, pp. 1286–1294.

[67] Ryu, J.-H., Park, C.-S., and Tahk, M. J., “Plant Inversion Control of Tail Controlled Missiles,” Proceedings of the 1997 AIAA Guidance Navigation and Control Conference, AIAA, Reston, VA, pages: 1691–1696.

[68] Siouris, George M., “Missile Guidance and Control Systems”, Springer-Verlag, Inc., New York, 2004.

[69] Sastry, S., Nonlinear Systems: Analysis, Stability, and Control, Springer–Verlag, New York, 1999, pages: 425–433, 468–498.

[70] Martin, P., Devasia, S., and Paden, B., “A Different Look at Output Tracking: Control of a VTOL Aircraft,” Automatica, Vol. 32, No. 1, 1996, pp. 101–107.

[71] Tahk,M.-J., Briggs, M. M., and Menon, P. K. A., “Applications of Plant Inversion via State Feedback to Missile Autopilot Design,” Proceedings of the 27th IEEE Conference on Decision and Control, IEEE, Piscataway, NJ, 1988, pp. 730–735.

[72] Hedrick, J. K., Gopalswamy, S., “Nonlinear Flight Control Design via Sliding Methods”, AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 13, No. 5, 1990, pp. 850–858.

[73] Johnson, E. N., Ohy, Seung-Min, “Adaptive Control Using Combined Online and Background Learning Neural Network”, AIAA/ Journal of Guidance, Control, and Dynamics, 2004

[74] Chowdhary, G., Johnson, E. N., “Theory and Flight Test Validation of Long Term Learning Adaptive Flight Controller”, Proceedings of the 2008 AIAA Guidance Navigation and Control Conference, Honolulu, Hawaii.

[75] Hadad, W., Chellebaonia V., Nonlinear Dynamical Systems and Control, A Lyapunov Based Approach, Preprint, 2006.

[76] McFarland, M. B., “Adaptive Nonlinear Control of Missiles,” Ph.D. Dissertation, School of Aerospace Engineering, Georgia Inst. of Technology, Atlanta, GA, Sept. 1997

[77] Wise, K. A., “Robust Stability Analysis of Adaptive Missile Autopilots”, Proceedings of the 2008 AIAA Guidance Navigation and Control Conference, Honolulu, Hawaii.

[78] Nielsen, Jack N., “Missile Aerodynamics”, McGraw-Hill Book Company, Inc., New York, 1960.

[79] Kılıç, K. Çağlar, “Autopilot And Guidance Algorithms For Infrared Guided Missiles”, M.S Thesis Study at METU Electrical and Electronics Engineering Department, December 2006.

[80] Özkan, B., “Dynamic Modeling, Guidance, and Control Of Homing Missiles”, PhD. Thesis at METU Mechanical Engineering Department, September 2005.

[81] “Missile Datcom User’s Manual-1997 FORTRAN 90 REVISION”, [On-line] Available: http://www.scribd.com/doc/12516547/Missile-DATCOM-Users-Guide.

[82] Zipfel, P. H., “Modeling and Simulation of Aerospace Vehicle Dynamics”, AIAA, American Institute of Aeronautics and Astronautics, Inc., Second Edition, Virginia, 2007.

[83] Cohen, H., Rogers, G. F. C., Saravanamuttoo, H. I. H., Gas Turbine Theory, Longman, 1972.

[84] Shneydor, N. A., “Missile Guidance and Pursuit Kinematics, Dynamics and Control”, Horwood Publishing Chichester, England, 1998.

[85] Garnell P., East D. J., “Guided Weapon Control Systems”, Pergamon Press Ltd., Oxford, London, England 1977.

[86] Blakelock, J.H., “Automatic Control of Aircraft and Missiles”, Second Edition, John Wiley & Sons Inc., USA, 1991.

[87] Kargın, V., Yavrucuk, İ., “Development of A Flight Control System for a UAV In Autonomous Landing”, AIAC-2007-105.

[88] McLean, D., “Automatic Flight Control Systems”, Prentice-Hall International (UK) Ltd, Hertfordshire, 1990.

[89] Nelson, R.C., “Flight Stability and Automatic Control”, Second Edition, McGraw-Hill Inc., Singapore, 1998.

[90] Menon, P., Badgett, M., and Walker, R., “Nonlinear Flight Test Trajectory Controllers for Aircraft,” Journal of Guidance, Control, and Dynamics, Vol. 10, No. 1, 1987, pp. 67–72.

[91] Snell, S. A., Enns, D. F., and Garrard, W. L., Jr., “Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft”, AIAA/ Journal of Guidance, Control, and Dynamics, Vol. 15, No. 4, 1992, pp. 976–984.

[92] Ogata, K., “Modern Control Engineering”, Third Edition, Prentice-Hall Inc., New Jersey, 1997.

[93] Zarchan, P., “Tactical and Strategic Missile Guidance”, Third Edition, AIAA, American Institute of Aeronautics and Astronautics, Inc., Virginia, 1997.

[94] Hassoun, M. H., “Fundamentals of Artificial Neural Networks”, the MIT Press, Cambridge, Massachusetts, 1995.

[95] Sanner, R. M., and Slotine, J. J. E., “Gaussian Network for Direct Adaptive Control,” IEEE Transactions on Neural Networks, Vol. 3, No. 6, 1992, pp. 837–863.

[96] Rysdyk, T. Rolf, “Adaptive Nonlinear Flight Control”, Ph.D. Thesis, Georgia Institute of Technology, November 1998.

[97] Spooner, J. T., Maggiore, M., Ordonez, R., and Passino, K. M., “Stable Adaptive Control and Estimation for Nonlinear Systems, Neural and Fuzzy Approximator Techniques”, Wiley, 2002.

[98] McFarland, M., and Calise, A., “Multilayer neural networks and adaptive nonlinear control of agile anti-air missiles”, Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA Paper 97-3540, August 1997.

[99] Kim, Y. H., Lewis, F.L., “High Level Feedback Control with Neural Networks”, World Scientific Series in Robotics and Intelligent Systems Vol. 21, World Science Publishing Co. Pte. Ltd. 1998.

[100] Lewis, F. L., “Nonlinear Network Structures for Feedback Control”, Asian Journal of Control, Vol.1, pages: 205-228, December 1999

[101] Haykin Simon, “Neural Networks a Comprehensive Foundation”, Second Edition, Prentice Hall USA, 1998.

[102] Johnson, E. N., Kannan, S. K., “Adaptive Flight Control for an Autonomous Unmanned Helicopter”, Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2002-4439, Monterey, CA, August 2002.

APPENDIX A

AERODYNAMIC CHARACTERISTICS OF THE MISSILE

Longitudinal Plane

The missile in this study has a plus tail configuration. Therefore, rudder deflection (δ_r) does not have any contribution on missile longitudinal plane. In other words, static pitch moment coefficient about the Cg (C_m) remains constant related to change in δ_r. This case is shown on the figure below:

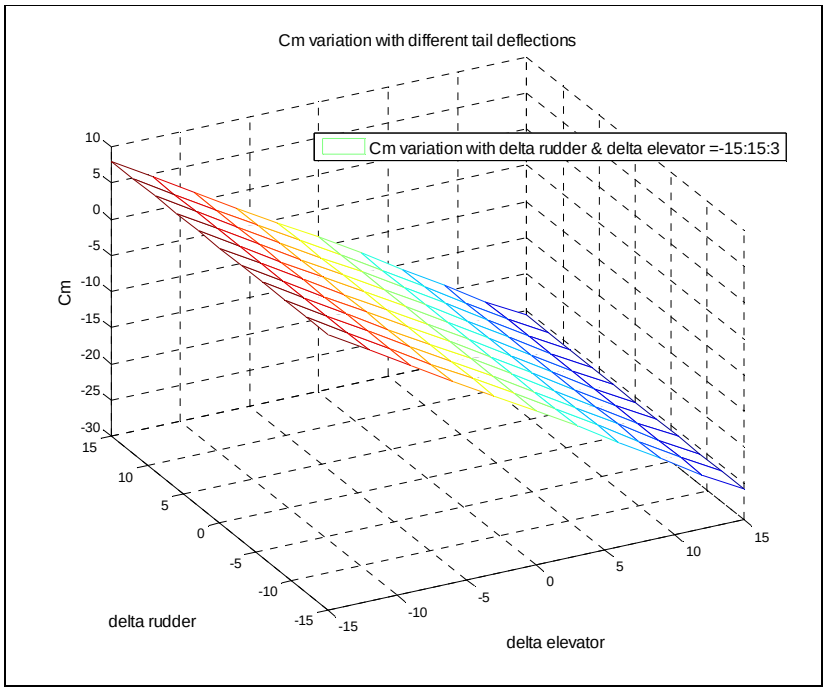

Figure A.1 C_m variation with different tail deflections At M=0.8, α=5º, β=0.1º (trim values)

As can be seen from Figure A.2, C_m variation due to change in side slip angle (β) can be neglected. Thus, it does not depend on β.

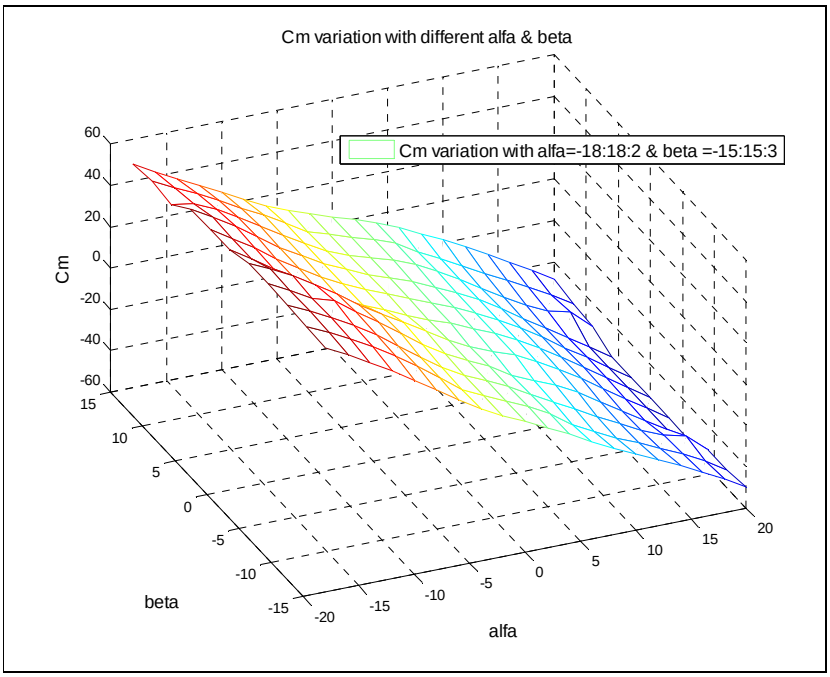

Figure A.2 C_m variation with different alpha (α) and beta (β) At M=0.8, δ_e = -3º, δ_r = -0.5º (trim values)

As a result, rudder deflection (δ_r) and side slip angle (β) does not have any contribution on missile longitudinal motion. Therefore, C_m can be expressed as:

$$C_m = C_m(\delta_e, \mathrm{M}, \alpha) \qquad \text{(A-1)}$$

In addition, according to above figures, C_m variations with alpha (α) are negative ($\frac{\partial C_m}{\partial \alpha} < 0$). So it can be concluded that the missile has positive pitch stiffness properties on the longitudinal plane.

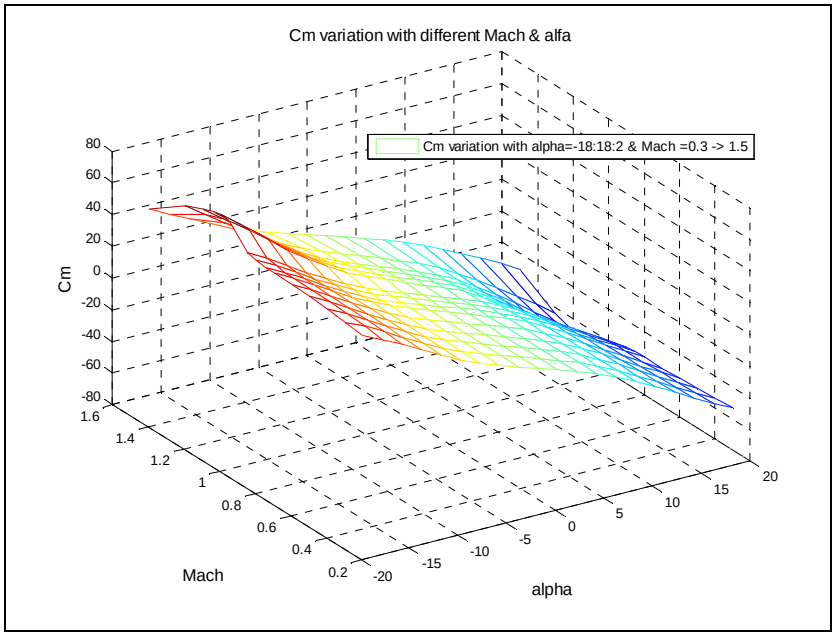

Figure A.3 C_m variation for different alpha (α) and Mach
At β=0.06º, δ_e = -3º, δ_r = -0.5º (trim values)

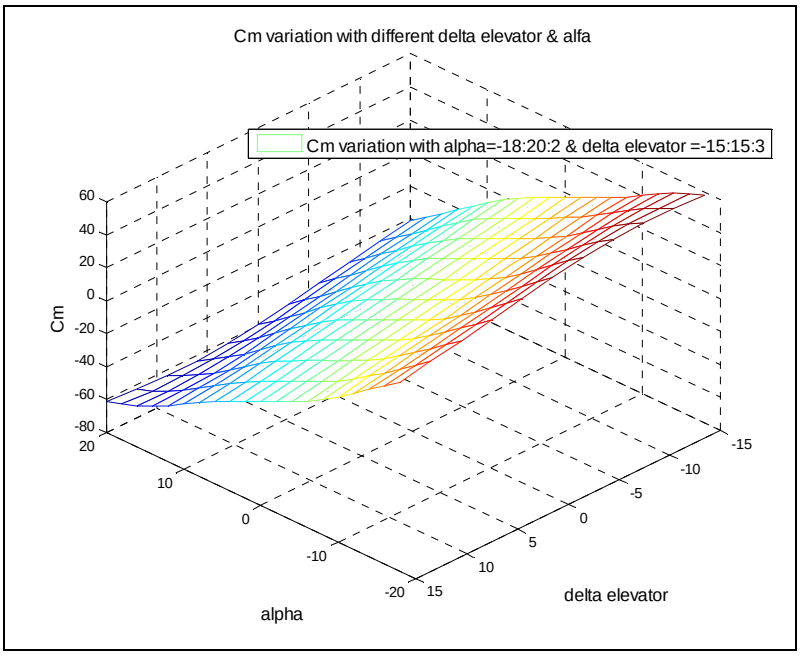

Figure A.4 C_m variation for different alpha (α) and elevator deflections (δ_e)
At β=0.1º, M=0.8, δ_r = -0.5º (trim values)

The variations of C_m at trim values according to Mach Number (M) and angle of attack (α) are shown on Figure A.3. From Figure A.4, it can be observed that C_m variations with elevator deflections (δ_e) are negative ($\frac{\partial C_m}{\partial \delta_e} < 0$). So it can be concluded that the Missile DATCOM notation is true for the longitudinal plane.

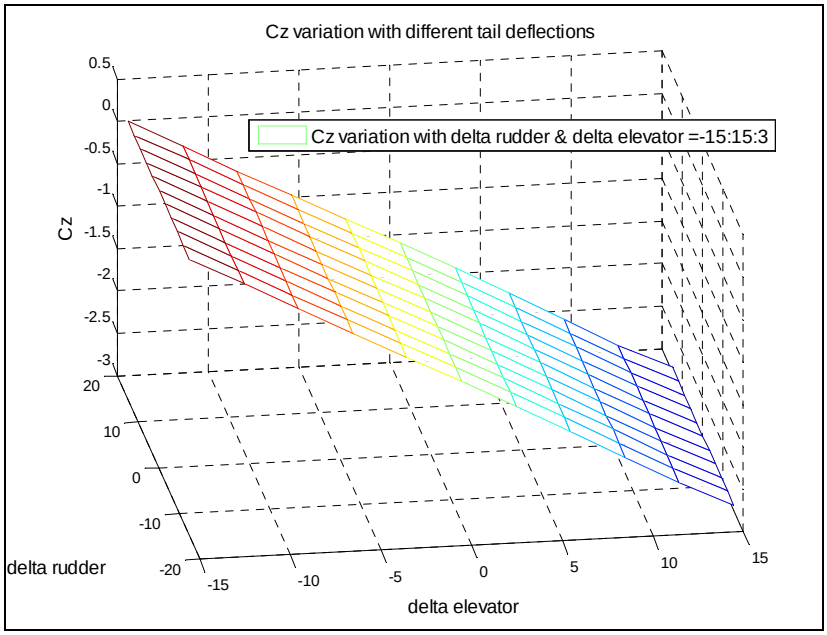

Figure A.5 C_z variation for different deflections
At M=0.8, α=5º, β=0.1º (trim values)

It can be observed from Figure A.5 that static force coefficient along z-direction (C_z) remains constant due to change in rudder deflection (δ_r).

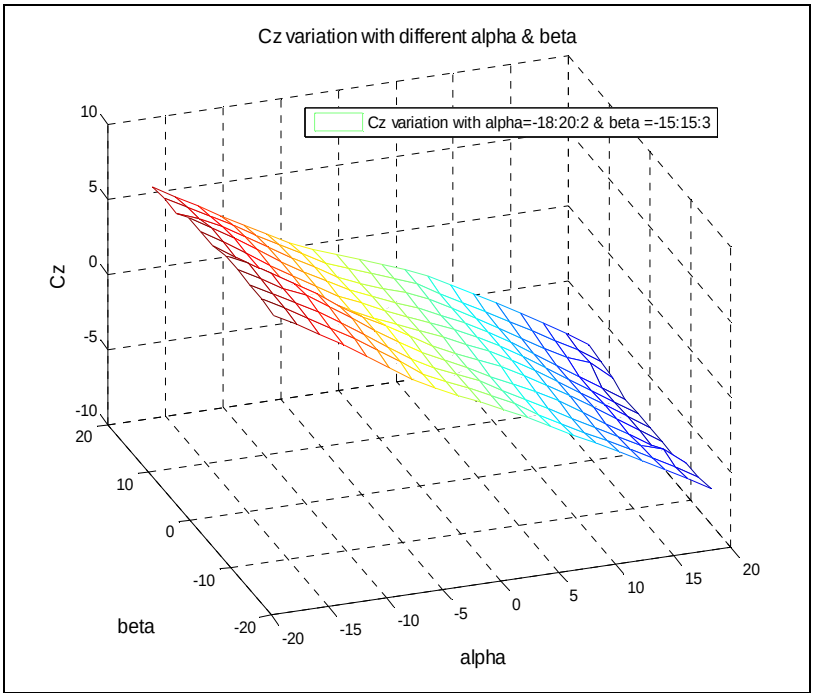

Figure A.6 C_z variation with different alpha (α) and beta (β)
At M=0.8, δ_e = -3º, δ_r = -0.5º (trim values)

From Figure A.6, variation of C_z due to change in side slip angle (β) can be ignored. As a result, C_z can be expressed as:

$$C_z = C_z(\delta_e, \mathrm{M}, \alpha) \qquad \text{(A-2)}$$

Directional Plane

Elevator deflection (δ_e) does not have any contribution on missile directional motion. In other words, static yaw moment coefficient about the Cg (C_n) remains constant related to change in δ_e. This case is shown on the Figure A.7 next page:

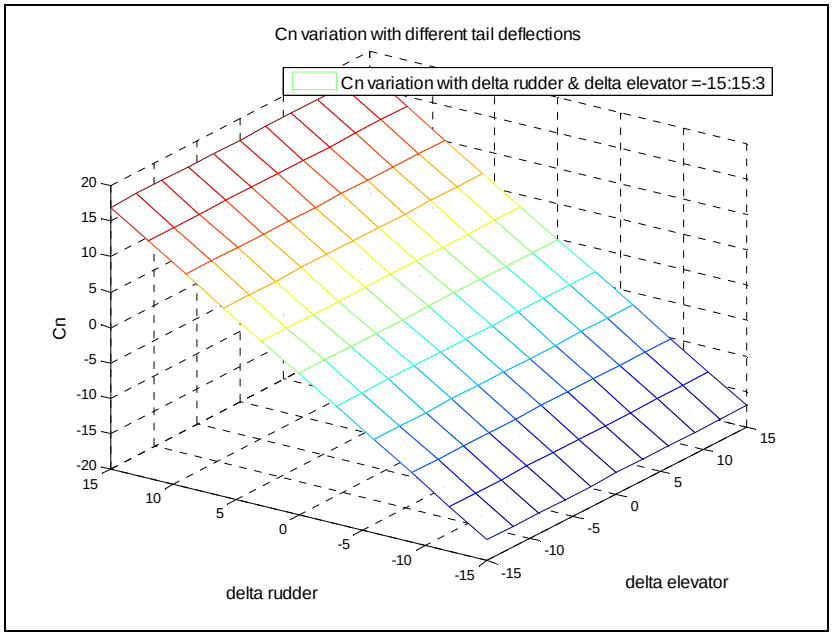

Figure A.7 C_n variation with different deflections
At M=0.8, α=5º, β=0.1º (trim values)

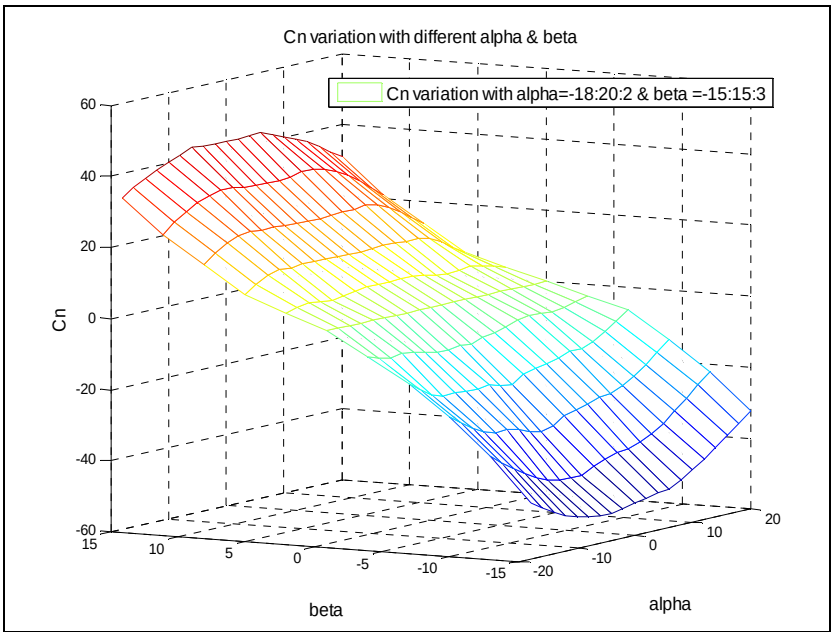

Figure A.8 C_n variation with different alpha (α) and beta (β)
At M=0.8, δ_e = -3º, δ_r = -0.5º

As can be seen from Figure A.8, C_n does not change very much due to change in angle of attack (α). Thus, dependency on α for C_n is negligible. As a result, elevator deflection (δ_e) and angle of attack (α) does not have any contribution on missile directional motion. Therefore, C_n can be expressed as:

$$C_n = C_n(\beta, \delta_r, \mathrm{M}) \qquad \text{(A-3)}$$

In addition, according to above figures, C_n variations with beta (β) are positive ($\frac{\partial C_n}{\partial \beta} > 0$). So it can be concluded that the missile has positive yaw stability properties on the directional plane.

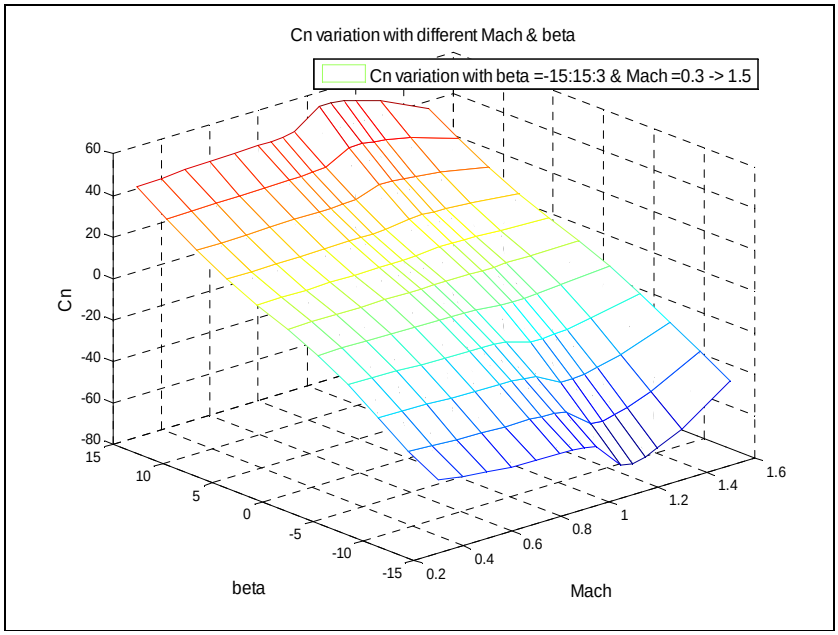

Figure A.9 C_n variation with different Mach (M) and beta (β) At α=5º, δ_e = -3º, δ_r = -0.5º (trim values)

The variations of C_n at trim values according to Mach Number (M) and side slip angle (β) are shown on Figure A.9.

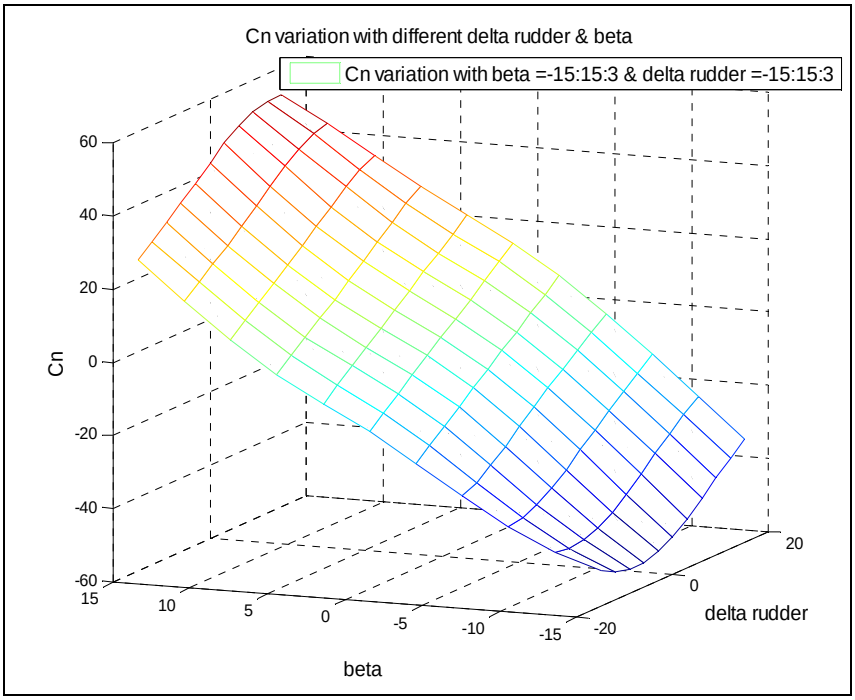

Figure A.10 C_n variation with different beta (β) and rudder deflections (δ_r) At α=5º, M=0.8, δ_e = -3º (trim values)

From Figure A.10, it can be observed that C_n variations with rudder deflections (δ_r) are positive ($\frac{\partial C_n}{\partial \delta_r} > 0$). So it can be concluded that the Missile DATCOM notation is true for the directional plane.

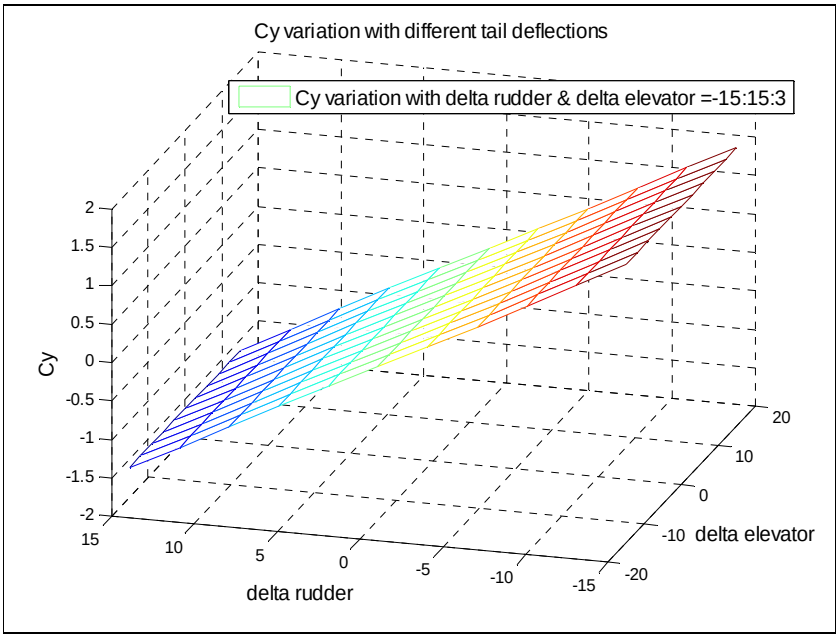

Figure A.11 C_y variation for different deflections
At M=0.8, α=5º, β=0.1º (trim values)

It can be observed from Figure A.11 that static force coefficient along y-direction (C_y) remains constant due to change in elevator deflection (δ_e).

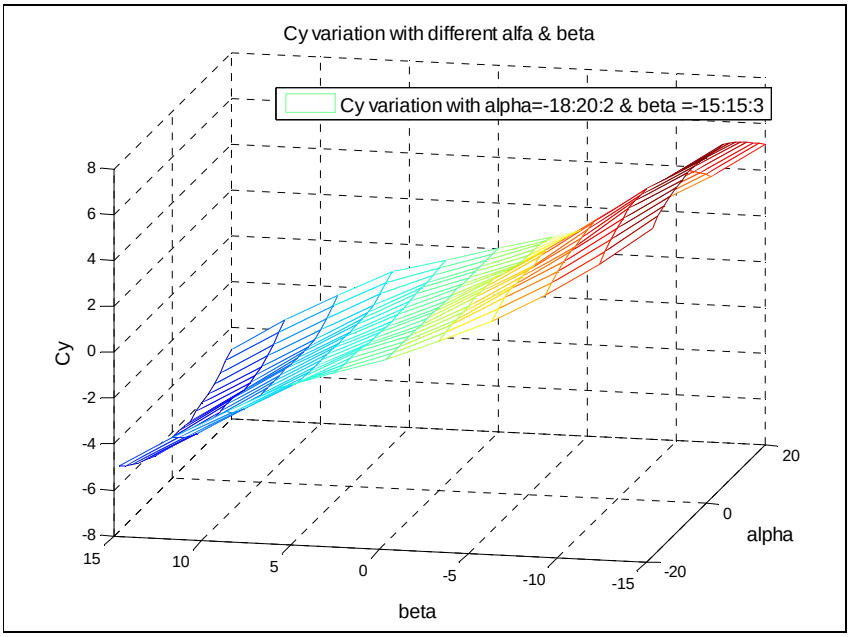

Figure A.12 C_y variation with different alpha (α) and beta (β) At M=0.8, δ_e = -3º, δ_r = -0.5º (trim values)

From Figure A.12, variation of C_y due to change in angle of attack (α) can be ignored. As a result, C_y can be expressed as:

$$C_y = C_y(\beta, \delta_r, \mathrm{M}) \qquad \text{(A-4)}$$

APPENDIX B

LINEARIZATION OF THE ROTATIONAL EQUATIONS OF MOTION

Symbolic Linearization is made by "Jacobian" command in Mapple 10 Software program and the equations below are reached:

Elements of matrix A1

$$A1_{11} = \frac{\rho u\, Sref\, lref}{I_{yy}}\left(C_m + \frac{1}{2}\frac{C_{mq}\, q\, lref}{V_{missile}} + \frac{1}{2}\frac{C_{m\dot{\alpha}}\, \dot{\alpha}\, lref}{V_{missile}} - \left(C_z + C_{zq}\frac{q\, lref}{2V_{missile}}\right)\cdot\frac{X_{cg}}{lref}\right) + \frac{1}{2}\frac{\rho {V_{missile}}^2\, Sref\, lref}{I_{yy}}\left(\begin{array}{l}\left(\frac{\partial C_m}{\partial \mathrm{M}}\frac{u}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{mq}}{\partial \mathrm{M}}\frac{u\, q\, lref}{a V_{missile}^2} - \frac{1}{2}\frac{C_{mq}\, u\, q\, lref}{V_{missile}^3} \\ + \frac{1}{2}\frac{\partial C_{m\dot{\alpha}}}{\partial \mathrm{M}}\frac{u\, \dot{\alpha}\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{m\dot{\alpha}}\, u\, \dot{\alpha}\, lref}{V_{missile}^3} \\ -\left(\left(\frac{\partial C_Z}{\partial \mathrm{M}}\frac{u}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{zq}}{\partial \mathrm{M}}\frac{u\, q\, lref}{a V_{missile}^2} - \frac{1}{2}\frac{C_{zq}\, u\, q\, lref}{V_{missile}^3}\right)\cdot\frac{X_{cg}}{lref}\end{array}\right) \quad \text{(B.1)}$$

$$A1_{12} = \frac{\rho v\, Sref\, lref}{I_{yy}}\left(C_m + \frac{1}{2}\frac{C_{mq}\, q\, lref}{V_{missile}} + \frac{1}{2}\frac{C_{m\dot{\alpha}}\, \dot{\alpha}\, lref}{V_{missile}} - \left(C_z + C_{zq}\frac{q\, lref}{2V_{missile}}\right)\cdot\frac{X_{cg}}{lref}\right) + \frac{1}{2}\frac{\rho {V_{missile}}^2\, Sref\, lref}{I_{yy}}\left(\begin{array}{l}\left(\frac{\partial C_m}{\partial \mathrm{M}}\frac{v}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{mq}}{\partial \mathrm{M}}\frac{v\, q\, lref}{a V_{missile}^2} - \frac{1}{2}\frac{C_{mq}\, v\, q\, lref}{V_{missile}^3} \\ + \frac{1}{2}\frac{\partial C_{m\dot{\alpha}}}{\partial \mathrm{M}}\frac{v\, \dot{\alpha}\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{m\dot{\alpha}}\, v\, \dot{\alpha}\, lref}{V_{missile}^3} \\ -\left(\left(\frac{\partial C_Z}{\partial \mathrm{M}}\frac{v}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{zq}}{\partial \mathrm{M}}\frac{v\, q\, lref}{a V_{missile}^2} - \frac{1}{2}\frac{C_{zq}\, v\, q\, lref}{V_{missile}^3}\right)\cdot\frac{X_{cg}}{lref}\end{array}\right) \quad \text{(B.2)}$$

$$A1_{13} = \frac{\rho w Sref\, lref}{I_{yy}}\left(C_m + \frac{1}{2}\frac{C_{mq}\, q\, lref}{V_{missile}} + \frac{1}{2}\frac{C_{m\dot{\alpha}}\,\dot{\alpha}\, lref}{V_{missile}} - \left(C_z + C_{zq}\frac{q\, lref}{2V_{missile}}\right)\cdot\frac{X_{cg}}{lref}\right) + \frac{1}{2}\frac{\rho V_{missile}^{\ 2}\, Sref\, lref}{I_{yy}}\left(\begin{array}{l}\left(\left(\frac{\partial C_m}{\partial \mathrm{M}}\frac{w}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{mq}}{\partial \mathrm{M}}\frac{w q\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{mq}\, w q\, lref}{V_{missile}^3}\right. \\ + \frac{1}{2}\frac{\partial C_{m\dot{\alpha}}}{\partial \mathrm{M}}\frac{w\dot{\alpha}\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{m\dot{\alpha}}\, w\,\dot{\alpha}\, lref}{V_{missile}^3} \\ \left. - \left(\left(\frac{\partial C_Z}{\partial \mathrm{M}}\frac{w}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{zq}}{\partial \mathrm{M}}\frac{w q\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{zq}\, w q\, lref}{V_{missile}^3}\right)\cdot\frac{X_{cg}}{lref}\right)\end{array}\right) \tag{B.3}$$

$$A1_{21} = \frac{\rho u Sref\, lref}{I_{zz}}\left(C_n + \frac{1}{2}\frac{C_{nr}\, r\, lref}{V_{missile}} + \frac{1}{2}\frac{C_{n\dot{\beta}}\,\dot{\beta}\, lref}{V_{missile}} + \left(C_y + C_{yr}\frac{r\, lref}{2V_{missile}}\right)\cdot\frac{X_{cg}}{lref}\right) + \frac{1}{2}\frac{\rho V_{missile}^{\ 2}\, Sref\, lref}{I_{zz}}\left(\begin{array}{l}\left(\left(\frac{\partial C_n}{\partial \mathrm{M}}\frac{u}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{nr}}{\partial \mathrm{M}}\frac{u r\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{nr}\, u r\, lref}{V_{missile}^3}\right. \\ + \frac{1}{2}\frac{\partial C_{n\dot{\beta}}}{\partial \mathrm{M}}\frac{u\,\dot{\beta}\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{n\dot{\beta}}\, u\,\dot{\beta}\, lref}{V_{missile}^3} \\ \left. + \left(\left(\frac{\partial C_y}{\partial \mathrm{M}}\frac{u}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{yr}}{\partial \mathrm{M}}\frac{u r\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{yr}\, u r\, lref}{V_{missile}^3}\right)\cdot\frac{X_{cg}}{lref}\right)\end{array}\right) \tag{B.4}$$

$$A1_{22} = \frac{\rho v Sref\, lref}{I_{zz}}\left(C_n + \frac{1}{2}\frac{C_{nr}\, r\, lref}{V_{missile}} + \frac{1}{2}\frac{C_{n\dot{\beta}}\,\dot{\beta}\, lref}{V_{missile}} + \left(C_y + C_{yr}\frac{r\, lref}{2V_{missile}}\right)\cdot\frac{X_{cg}}{lref}\right) + \frac{1}{2}\frac{\rho V_{missile}^{\ 2}\, Sref\, lref}{I_{zz}}\left(\begin{array}{l}\left(\left(\frac{\partial C_n}{\partial \mathrm{M}}\frac{v}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{nr}}{\partial \mathrm{M}}\frac{v r\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{nr}\, v r\, lref}{V_{missile}^3}\right. \\ + \frac{1}{2}\frac{\partial C_{n\dot{\beta}}}{\partial \mathrm{M}}\frac{v\,\dot{\beta}\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{n\dot{\beta}}\, v\,\dot{\beta}\, lref}{V_{missile}^3} \\ \left. + \left(\left(\frac{\partial C_y}{\partial \mathrm{M}}\frac{v}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{yr}}{\partial \mathrm{M}}\frac{v r\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{yr}\, v r\, lref}{V_{missile}^3}\right)\cdot\frac{X_{cg}}{lref}\right)\end{array}\right) \tag{B.5}$$

$$A1_{23} = \frac{\rho w Sref\, lref}{I_{zz}}\left(C_n + \frac{1}{2}\frac{C_{nr}\, r\, lref}{V_{missile}} + \frac{1}{2}\frac{C_{n\dot{\beta}}\,\dot{\beta}\, lref}{V_{missile}} + \left(C_y + C_{yr}\frac{r\, lref}{2V_{missile}}\right)\cdot\frac{X_{cg}}{lref}\right) + \frac{1}{2}\frac{\rho V_{missile}^{\ 2}\, Sref\, lref}{I_{zz}}\left(\begin{array}{l}\left(\left(\frac{\partial C_n}{\partial \mathrm{M}}\frac{w}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{nr}}{\partial \mathrm{M}}\frac{w r\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{nr}\, w r\, lref}{V_{missile}^3}\right. \\ + \frac{1}{2}\frac{\partial C_{n\dot{\beta}}}{\partial \mathrm{M}}\frac{w\,\dot{\beta}\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{n\dot{\beta}}\, w\,\dot{\beta}\, lref}{V_{missile}^3} \\ \left. + \left(\left(\frac{\partial C_y}{\partial \mathrm{M}}\frac{w}{V_{missile}\, a}\right) + \frac{1}{2}\frac{\partial C_{yr}}{\partial \mathrm{M}}\frac{w r\, lref}{a\, V_{missile}^2} - \frac{1}{2}\frac{C_{yr}\, w r\, lref}{V_{missile}^3}\right)\cdot\frac{X_{cg}}{lref}\right)\end{array}\right) \tag{B.6}$$

Elements of matrix A2

$$A2_{11} = \frac{1}{2}\frac{\rho {V_{missile}}^2 Sref\, lref}{I_{yy}}\left(\frac{1}{2}\frac{C_{mq}\, lref}{V_{missile}} - \frac{1}{2}\frac{C_{zq}\, X_{cg}}{V_{missile}}\right)$$

$$A2_{12} = \frac{I_{zz} - I_{xx}}{I_{yy}}\, p \qquad \text{(B.7)}$$

$$A2_{21} = \frac{I_{xx} - I_{yy}}{I_{zz}}\, p$$

$$A2_{22} = \frac{1}{2}\frac{\rho {V_{missile}}^2 Sref\, lref}{I_{zz}}\left(\frac{1}{2}\frac{C_{nr}\, lref}{V_{missile}} + \frac{1}{2}\frac{C_{yr}\, X_{cg}}{V_{missile}}\right)$$

Elements of matrix B

$$B_{11} = \frac{1}{2}\frac{\rho {V_{missile}}^2 Sref\, lref}{I_{yy}}\left(\frac{\partial C_m}{\partial \delta e} - \frac{\partial C_z}{\partial \delta e}\frac{X_{cg}}{lref}\right)$$

$$B_{12} = 0 \qquad \text{(B.8)}$$

$$B_{21} = 0$$

$$B_{22} = \frac{1}{2}\frac{\rho {V_{missile}}^2 Sref\, lref}{I_{zz}}\left(\frac{\partial C_n}{\partial \delta r} - \frac{\partial C_y}{\partial \delta r}\frac{X_{cg}}{lref}\right)$$

APPENDIX C

PID CONTROLLER GAIN SELECTION

The gains Kp_h, Ki_h and Kd_h are selected and tuned by giving 400 meters step altitude input to the 5-DOF simulation. The response graphs according to different gains and gain selection procedure are explained in this section.

Firstly, the graphs are obtained for only different Kp_h. Then considering the fast response, oscillation and steady state error criteria, the appropriate Kp_h value is determined. The graphs are shown below.

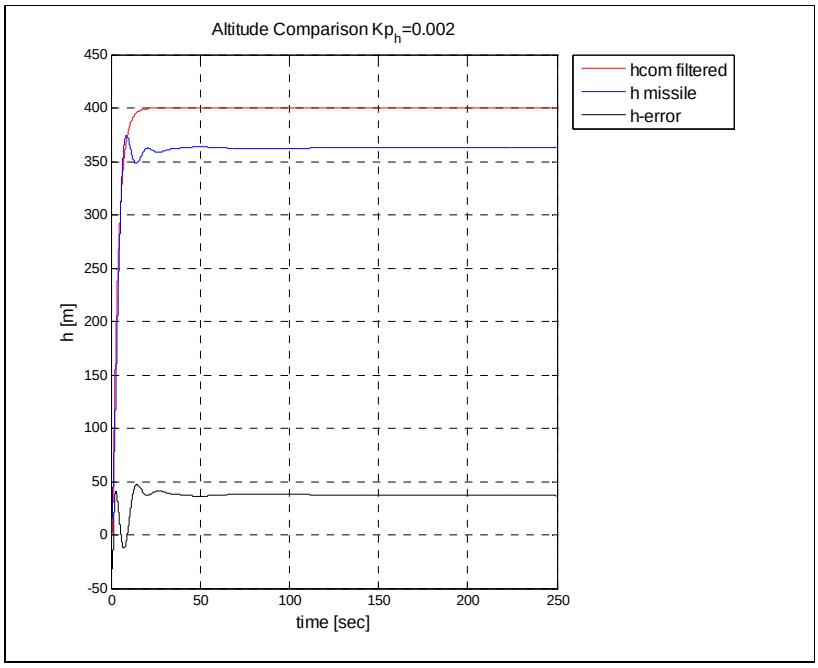

Figure C.1 Altitude Response for Kp_h=0.002 and step input =400 m

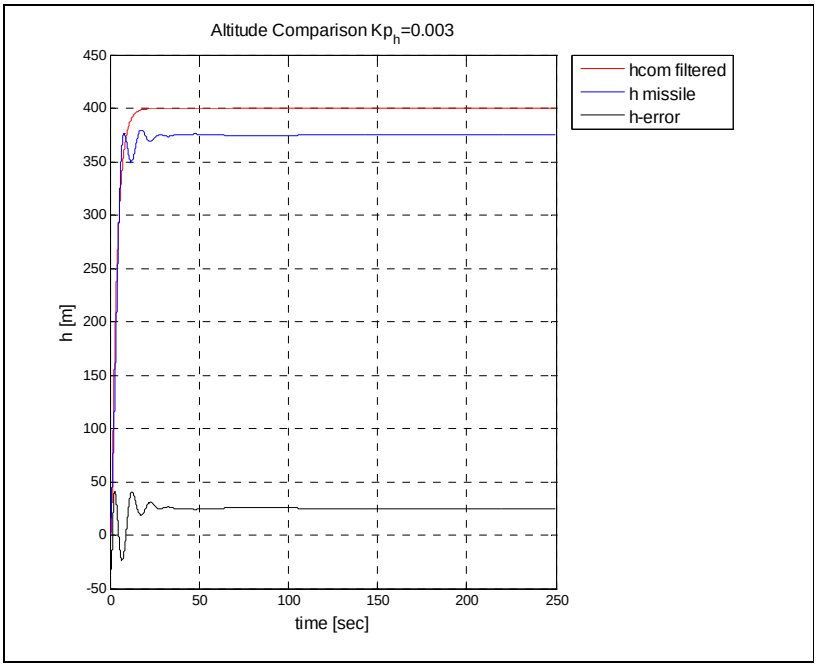

Figure C.2 Altitude Response for Kp_h=0.003 and step input =400 m

If Figure C.2 compared with Figure C.1, it is observed that the increase on Kp_h causes an increase on the oscillation and decrease on the steady state error. The following graphs also support this consideration. In addition, since the system response is converged to a constant altitude value, it can be realized that the system has stability for altitude control.

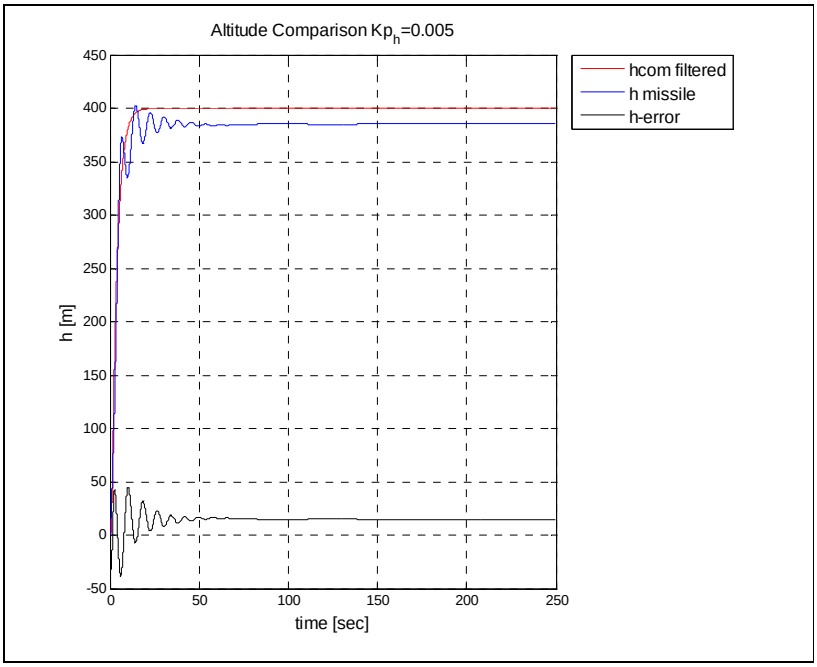

Figure C.3 Altitude Response for Kp_h=0.005 and step input =400 m

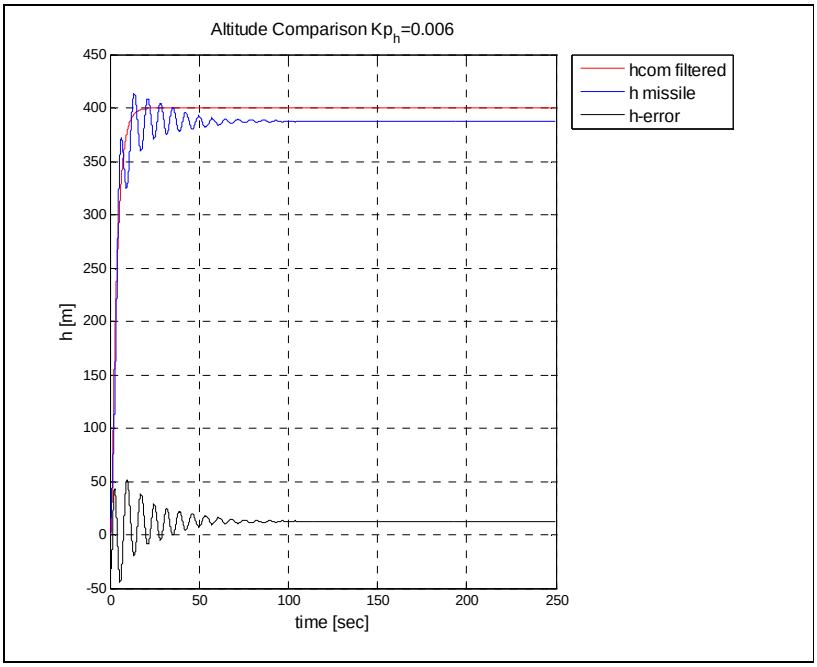

Figure C.4 Altitude Response for Kp_h=0.006 and step input =400 m

From Figure C.3 and C.4, larger Kp_h shows quick response and has much low steady state error, but higher gain results in higher oscillation. Therefore Kp_h must be chosen related to these conditions. The priority is given to fast response characteristics for altitude hold. As a result, Kp_h is selected as 0.005.

Secondly, the steady state error has to be compensated. So we need an integral contribution to the controller. Ki_h is selected considering fast convergence to steady state value and low oscillation.

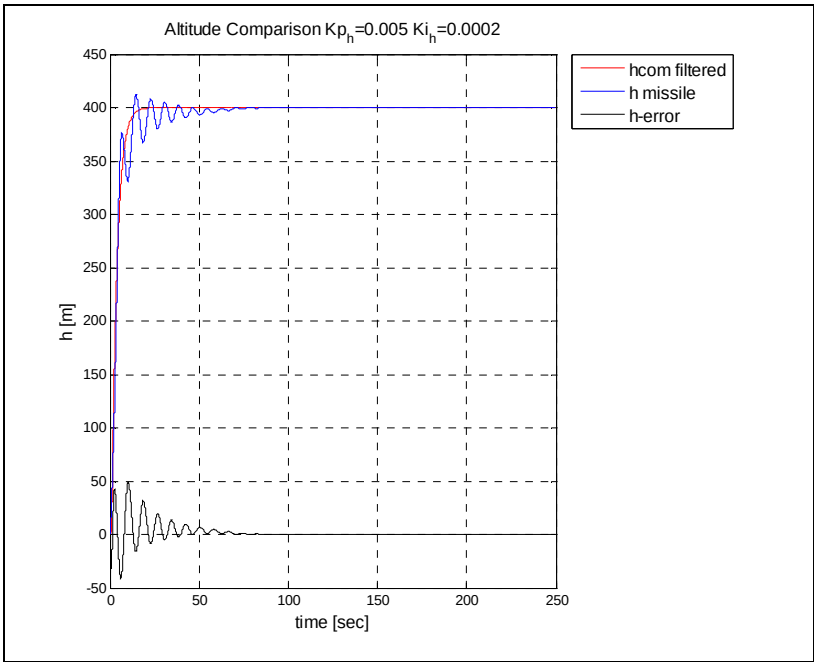

Figure C.5 Altitude Response for Kp_h=0.005 Ki_h=0.0002 Step input =400 m

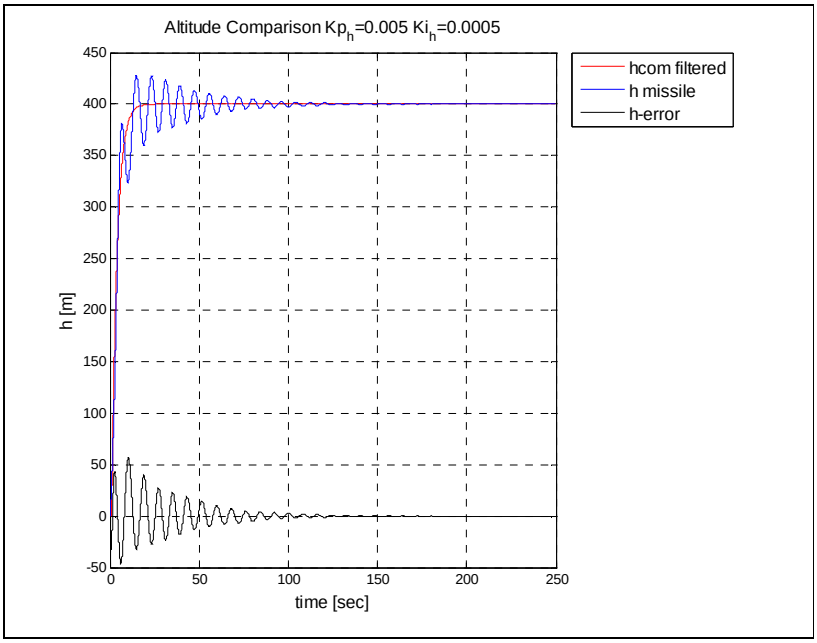

Figure C.6 Altitude Response for Kp_h=0.005 Ki_h=0.0005 Step input =400 m

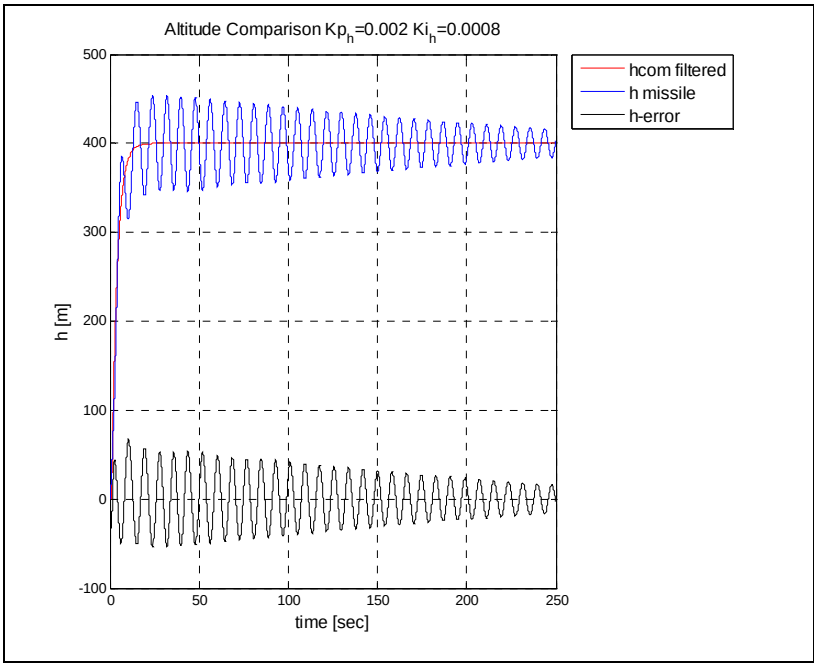

Figure C.7 Altitude Response for Kp_h=0.005 Ki_h=0.0008 Step input =400 m

According to the above figures, lower Ki_h causes slow convergence to steady state value and higher Ki_h results in higher oscillation. Since the response can not avoid from oscillation, Ki_h is chosen as 0.0005 and the derivative contribution is needed for the controller. There must be a reason for this oscillation. If the vertical speed is not included in the autopilot architecture, the missile would tend towards pitching the nose too high and entering a stall or losing all lift. Therefore derivative controller is added for including the climb rate error as input to autopilot. By using climb rate error, the relation can be established between vertical speed of the missile and altitude response.

Finally, derivative gain (Kd_h) is added and final configuration of the PID controller is obtained. The step input response for PID controller is shown on Figure C.8. Kd_h is selected as 0.01. As a result, this controller provides the system with quick response (Proportional gain), zero steady state error (Integral gain) and damped oscillations (Derivative gain).

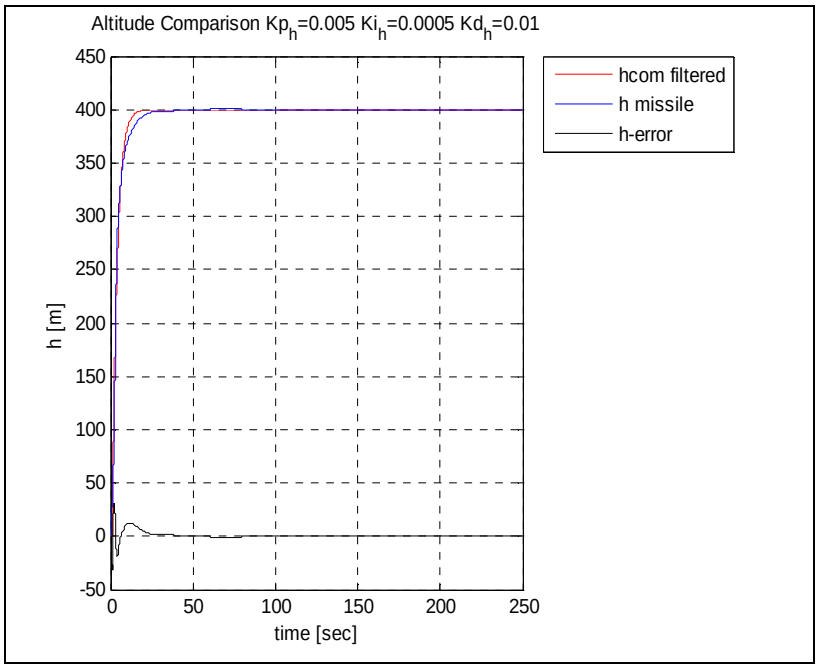

Figure C.8 Altitude Response for Kp_h=0.005 Ki_h=0.0005 Kd_h= 0.01 Step input =400 m