NEURAL NETWORK BASED ONLINE ESTIMATION OF MANEUVERING STEADY STATES AND CONTROL LIMITS

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

BY

GÖNENÇ GÜRSOY

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

JUNE 2010

Approval of the thesis:

NEURAL NETWORK BASED ONLINE ESTIMATION OF MANEUVERING STEADY STATES AND CONTROL LIMITS

submitted by **GÖNENÇ GÜRSOY** 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. Ozan Tekinalp _____
Head of Department, **Aerospace Engineering**

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

Examining Committee Members:

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

Prof. Dr. M. Kemal Özgören
Mechanical Engineering Dept., METU _____

Assoc. Prof. Dr. Tayfun Çimen
Engineering Development Directorate,
ROKETSAN Missile Industries Inc. _____

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

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

Date: _____

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

Name, Last Name: GÖNENÇ GÜRSOY

Signature :

ABSTRACT

NEURAL NETWORK BASED ONLINE ESTIMATION OF MANEUVERING STEADY STATES AND CONTROL LIMITS

Gürsoy, Gönenç

M.Sc., Department of Aerospace Engineering

Supervisor : Assist. Prof. Dr. İlkay Yavrucuk

June 2010, 92 pages

This thesis concerns the design and development of neural network based predictive algorithms to predict approaching aircraft limits. Therefore, approximate dynamics of flight envelope parameters such as angle of attack and load factor are constructed using neural network augmented dynamic models. Then, consructed models are used to predict steady state responses. By inverting the models and solving for critical controls at the known envelope limits, critical control inputs are calculated as well. The performance of the predictor algorithm is then evaluated with a different neural network online adaptation law which uses a stack of recorded data. It is shown that using a stack of recorded data online, constructed models become much more representative of limit parameter dynamics compared to adaptation using instantaneous measured data only. The benefits of recording data online and using it for weight adaptation are presented in the scope of dynamic trim and control limit predictions.

Keywords: Online system identification, limit detection, neural networks, concurrent learning

ÖZ

YAPAY SİNİR AĞI TABANLI MANEVRA DENGE NOKTALARININ VE KONTROL LİMİTLERİNİN ÇEVRİMİÇİ BELİRLENMESİ

Gürsoy, Gönenç
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. İlkay Yavrucuk

Haziran 2010, 92 sayfa

Bu tezde yapay sinir ağları kullanılarak, yaklaşmakta olan hava aracı limitlerinin çevrimiçi tahminini sağlayacak algoritmaların geliştirilmesi ele alınmıştır. Bu amaç ile uçus zarfı parametreleri olan hücum açısı ve yük faktörünün yapay sinir ağı tabanlı dinamik modelleri oluşturulmuştur. Oluşturulan dinamik modeller kullanılarak uçuş zarfı parametrelerinin durağan kararlılık durumları tahmin edilmiştir. Aynı zamanda bu modeller ters çevirilerek, bilinen uçuş zarfı limitlerinde kritik kontrol girdileri de bulunmuştur. Geliştirilen tahmin algoritması, uçuş sırasında uçuş verilerinin kaydedildiği ve kullanıldığı, farklı bir yapay sinir ağı öğrenme yöntemi ile değerlendirilmiştir. Uçuş sırasında kaydedilen verilerin kullanıldığı öğrenme yönteminin, anlık verileri kullanan öğrenme yöntemlerine göre, gerçek hücum açısı ve yük faktörü dinamiğini daha iyi temsil ettiği gösterilmiştir. Uçuş sırasında verilerin kaydedilip kullanılmasının çeşitli faydaları manevra denge noktalarının ve kontrol limitlerinin tahmini açısından gösterilmiştir.

Anahtar Kelimeler: çevrimiçi sistem tanımlama, limitlerin belirlenmesi, yapay sinir ağları, eş zamanlı öğrenme

to my family

ACKNOWLEDGMENTS

First, I would like to emphasize my special thanks to Dr. İlkay Yavrucuk for guiding me throughout the thesis work. All necessary items are provided by Dr. Yavrucuk. He shared his invaluable knowledge with me whenever necessary. Without his guidance and support I would not be able to succeed in completing this work.

I would like to give special thanks to Miray Demir Şahin who were always a close friend and confidante. With her friendship, undergraduate and graduate studies, especially the courses we have taken together became enjoyable and tolerable. I would like to thank for the delicious cakes too. Also, I want to give special thanks to Tuğba Ünlü and Barış Timurkaynak for their close friendship. As being the most enjoyable couple aorund me, they kept my moral and motivation as high as possible during the thesis work.

I would also thank to Flight Sciences Chief Erhan Tarhan for keeping the work environment as comfortable as possible. He always shared his invaluable ideas as a brother. Thanks for the understandings and sincere talks throughout the three years of work life.

I would like to state my sincere gratitude to Senem Atalayer Kırcalı. She was the first to recommend future response prediction problem as a thesis work. During the three years of work life, she was always as a sister for me. Without her encouragement, friendship, support and guidance, accomplishment of this thesis would be nearly impossible.

As well, I would like to express my sincere gratitude to the Flight Sciences people whom I always enjoyed to work with; Erinç Erdoğan, M. Fatih Özakdağ, Berrak Barun, Kıvanç Ülker, Murat Canıbek and Halil Kaya. I was very lucky that I had found a chance to work with these guys. I also want to give special thanks to Onur Tarımcı who is a legendery person of the control lab of aerospace engineering department. He shared his thesis in latek format and everything turned out to be simple.

There were also people who were always with me throught the three years of graduate studies; M. Burak Sayar and Kenan Cengiz. Their close and real friendship always kept my motivation

high. Whenever I felt I was tore down, they always helped and saved me. Honestly I can say that without their existance I would not be succeeded in completing this work. Badbull and rock'n roll will be forever.

Last, but not least, I would like to express my deepest thanks to my parents Sermin and Lütfullah, to my brother Övünç, to my aunt Serpil and to my grandmother Hacer for their endless support, trust, love, understanding and patience. Without their love It would not be possible for me to finish this thesis. This thesis is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

p	Body roll rate
q	Body pitch rate
r	Body yaw rate
u	Velocity in body x-axis
v	Velocity in body y-axis
w	Velocity in body z-axis
ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
X_e	x-coordinate of airplane CG relative to Earth-fixed axis
Y_e	y-coordinate of airplane CG relative to Earth-fixed axis
Z_e	z-coordinate of airplane CG relative to Earth-fixed axis
L_{EB}	Body frame to Earth frame transformation matrix
$\bar{p}$	Nondimensional body roll rate
$\bar{q}$	Nondimensional body pitch rate
$\bar{r}$	Nondimensional body yaw rate
q_{dyn}	Dynamic pressure
δ_e	Elevator deflection
δ_a	Aileron deflection
δ_r	Rudder deflection
δ_{thr}	Percent throttle position
α	Angle of attack
β	Sideslip angle (for Chapter-2)
C_D	Nondimensional aerodynamic drag force coefficient in stability axis
C_Y	Nondimensional aerodynamic side force coefficient in stability axis
C_L	Nondimensional aerodynamic lift force coefficient in stability axis
C_l	Nondimensional aerodynamic roll moment coefficient in stability axis
C_m	Nondimensional aerodynamic pitch moment coefficient in stability axis
C_n	Nondimensional aerodynamic yaw moment coefficient in stability axis
D	Dimensional aerodynamic drag force in stability axis
Y	Dimensional aerodynamic side force in stability axis
L	Dimensional aerodynamic lift force in stability axis
l	Dimensional aerodynamic roll moment in stability axis
m	Dimensional aerodynamic pitch moment in stability axis
n	Dimensional aerodynamic yaw moment in stability axis
C_X	Nondimensional aerodynamic force coefficient in body x-axis
C_Y	Nondimensional aerodynamic force coefficient in body y-axis
C_Z	Nondimensional aerodynamic force coefficient in body z-axis
C_R	Nondimensional aerodynamic moment coefficient in body x-axis
C_M	Nondimensional aerodynamic moment coefficient in body y-axis
C_N	Nondimensional aerodynamic moment coefficient in body z-axis
X_b	Dimensional aerodynamic drag force in body x-axis
Y_b	Dimensional aerodynamic side force in body y-axis
Z_b	Dimensional aerodynamic lift force in body z-axis
R_b	Dimensional aerodynamic roll moment in body x-axis
M_b	Dimensional aerodynamic pitch moment in body y-axis

N_b	Dimensional aerodynamic yaw moment in body z-axis
S	Reference Wing Area
$\bar{c}$	Mean chord length
b	Wing span
$X_{gravity}$	x-component of gravitational force defined in body axis
$Y_{gravity}$	y-component of gravitational force defined in body axis
$Z_{gravity}$	y-component of gravitational force defined in body axis
X_{Thrust}	x-component of thrust force defined in body axis
Y_{Thrust}	y-component of thrust force defined in body axis
Z_{Thrust}	y-component of thrust force defined in body axis
x_f	Fast states of aircraft
$\hat{x}_f$	Instant prediction of fast states
$\hat{x}_{f_{DT}}$	Dynamic trim prediction of fast states
$\hat{x}_{f_{DT_{lim}}}$	Known envelope limits of fast states
x_s	Slow states of aircraft
x	States of aircraft (equivalent to $\vec{x}$)
ξ	Modeling error
$\triangle$	Neural network output
e	Model tracking error
K	Observer gain matrix
β	Neural network activation function vector
r	Residual signal
r_{c_i}	Residual signal containing past information
$\epsilon_{\vec{x}}$	Data selection criteria of transient stack
ϵ_{y_1}	Data selection criteria of steady state stack
ϵ_{y_2}	Data selection criteria of steady state stack
ϵ_{z_1}	Data selection criteria of steady state stack
ϵ_{z_2}	Data selection criteria of steady state stack
L_{R_W}	Instantaneous learning rate of W matrix
L_{R_V}	Instantaneous learning rate of V matrix
$L_{R_{W_{tr}}}$	Transient stack learning rate of W matrix
$L_{R_{V_{tr}}}$	Transient stack learning rate of V matrix
$L_{R_{W_s}}$	Steady state stack learning rate of W matrix
$L_{R_{V_s}}$	Steady state stack learning rate of V matrix

Subscripts and Superscripts

$()_{DT}$	Dynamic trim
$()_{lim}$	Limit value
$\hat{()}$	Instant prediction

Acronyms

EPS	Envelope Protection System
NN	Neural Network

CHAPTER 1

INTRODUCTION

1.1 Focus of the Study

Flight envelope protection is an area of research where the focus is to cue the pilot such that the known envelope limits are not violated during flight. These systems are mainly used for both fixed and rotary wing aircraft in order to improve handling qualities. In envelope protection, aircraft is allowed to use the full flight envelope without possibility of violating known flight envelope limits. In literature, this type of cueing systems are also called as carefree maneuvering systems and they exhibit *limit prediction* and *limit avoidance*. In limit prediction, the algorithms are developed in such a way that the violations of limits are detected in an effective lead time before the actual violation occurs. This is mainly done by predicting the steady-state dynamics of the critical states during flight. Then, this information is used in limit avoidance part in order to apply a preventive action with the aim of pilot cueing.

In this work, we mainly focused on predicting the steady states of the parameters which dominates the flight envelope such as; angle of attack and load factor. Also the calculation of critical controls due to the known envelope limits are aimed. In other words the main focus is the detection of limits during flight. Limit avoidance is out of the scope of this thesis.

1.2 Background

In limit prediction, main purpose is to predict the future response of the limit parameters at the instants the controls are applied. Limit parameters can be taken as any fast states, which will be defined later in detail, of aircraft dynamics such as angle of attack, load factor and angular rates, etc. Note that, once a future response is predicted at the time the controls are applied, it is possible to calculate the distance of limit parameters to the known envelope limits. This distance is also called as limit margin. In limit prediction, critical controls which will cause the violations of upper and lower bounds of flight envelope limits can be calculated as well. The distance of current controls to the critical controls are called control margins. Therefore, once the limit margins and control margins are calculated and sent to EPS at the time the controls are applied, a preventive action can be taken in an effective lead time, in order to avoid approaching limit violations. Envelope protection systems will surely improve the overall confidence and safety of the aircraft. The necessity of envelope protection becomes much more essential, especially, when aggressive maneuvering close to the envelope limits is desired. As a result, the capability of being aware and responding automatically to approaching limits will be an important feature.

In previous works, different approaches have been used in order to develop successful limit prediction algorithms. The most known first application, which is a feasibility study for neural network based limit protection systems, can be found in [1]. In this study, off-line trained neural networks and empirical functions are mainly used to generate limits. In [2] limits are modeled by use of neural-fuzzy logic hybrid systems.

At NASA Ames Research Center, a study was performed for main rotor torque cueing systems [3]. They used polynomial nueral networks (PNN) with *Fixed Time Horizon Prediction Method* to predict near future limit violations. Prediction Horizon refers to a fixed time at which the state to be measured (Torque) is expected to get its maximum value (peak). In the method applied, they showed that a functional relationship can be developed to predict the value of the limit at the prediction horizon by use of current states and control inputs.

In [4-8] and [23-26], limit prediction algorithms are much more developed. Generally, they focused on tilt-rotor aircraft and helicopter models in order to perform limit detection and avoidance algorithms. In these studies, limit parameter response types are categorized into

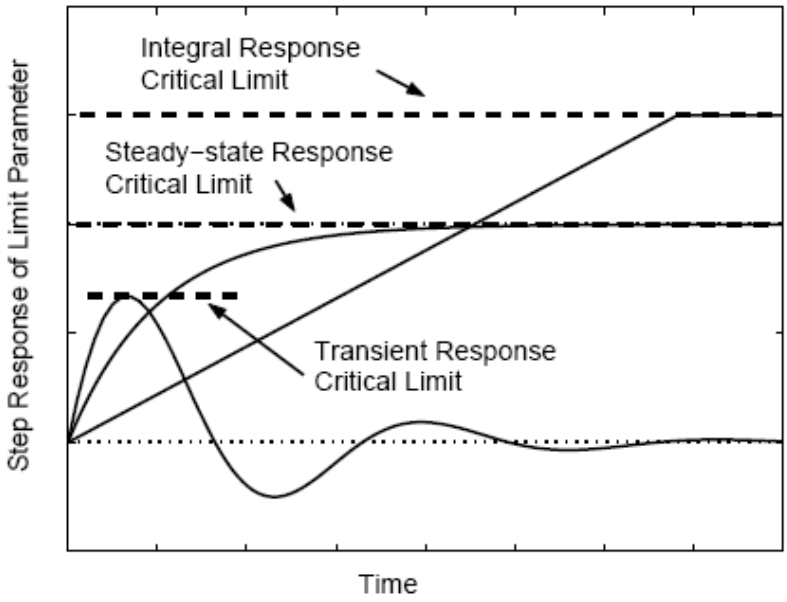

Figure 1.1: Limit Parameter Response Types [12, 5]

three different categories according the responses to a step input: *Transient Peak Response Critical*, *Steady State Response Critical*, and *Integrated Response Critical* (Figure1.1). In other words, for each type of limit response, different types of algorithms are developed. One of those algorithms is *Dynamic Trim Algorithm*, which is mainly applied for steady state response critical type of limit parameters [4, 5, 6, 7, 8]. The algorithm directly relates the steady state response of limit parameters to current state information and control inputs by using an off-line trained neural network. Off-line training can be done with flight test data or simulation data. Other algorithm which is known as *Adaptive Dynamic Trim Algorithm* uses a first order linear model and an adaptive term for the online approximation of real dynamics [27, 28, 29, 30, 4, 5, 6, 7, 8] . Adaptive term compensates for the errors between the real model and the first order linear model. Then the steady states are found by equating the derivatives of approximate model to zero. This algorithm is applicable to steady state response critical as well.

Aircraft states can be categorized into two groups due to their response times: fast and slow states. Generally fast states come to a steady state in a shorter response time compared to slow states. Angular velocities, angle of attack and load factor are some examples of fast states. Whereas, body velocities, altitude and dynamic pressure can be thought of as slow states, since they generally come to a steady state at a larger response time. Fast and slow states are mainly used in the development of *Dynamic Trim* concept. Dynamic trim is a quasi-steady

maneuvering flight condition where the fast states have reached the equilibrium whereas the slow states may still be varying with time [27, 28, 29, 30]. Dynamic trim algorithm uses dynamic trim concept in order to predict steady state response. In the literature, it is shown that adaptive dynamic trim algorithm, which includes an adaptive neural network term, increases the accuracy of prediction over dynamic trim algorithm. Applications of non-real-time and real-time piloted simulations can be found in [4, 6]. By using adaptive dynamic trim algorithm, even if large variations in CG location [28, 30, 8], weight and flight conditions occur, the developed algorithm can still calculate quite successful predictions.

For transient response critical limits, peak response estimation algorithm is developed in [5, 9, 10, 11, 12]. The algorithm is based on estimating the transient peak of the limit parameter that occurs just after the control input. Mainly, predictions for flapping transient limits are studied.

Yavrucuk, Unnikrishnan, Prasad and Calise used adaptive dynamic trim algorithm with linearly and nonlinearly parameterized neural networks [27, 28, 29, 30]. Control margins which correspond to the steady state limits are computed. Especially, load factor, angle of attack and torque predictions are calculated for tilt-rotor simulation.

In [13] and [30], envelope protection system design for autonomous unmanned aerial vehicles is studied. Generated online dynamic models are used to estimate limits on controller commands. Simulation and flight test results are provided for rotor stall limit and load factor protections.

1.3 Neural Networks and System Identification

Neural networks are an evolving field in which the main purpose is to resemble the human brain in order to capture and learn highly nonlinear environments or data. In system identification or automatic flight control problems, those nonlinear environments, where adaptive elements are expected to adapt, are called modeling errors. In the case of limit prediction, the occurences of modeling errors are mainly due to the differences between the real and approximate dynamics. In fig1.2 general structure of a single hidden layer neural network is shown. In most control problems, neural networks are succesfullly used for cancelling out modeling errors [14, 15, 16, 33, 34, 35].

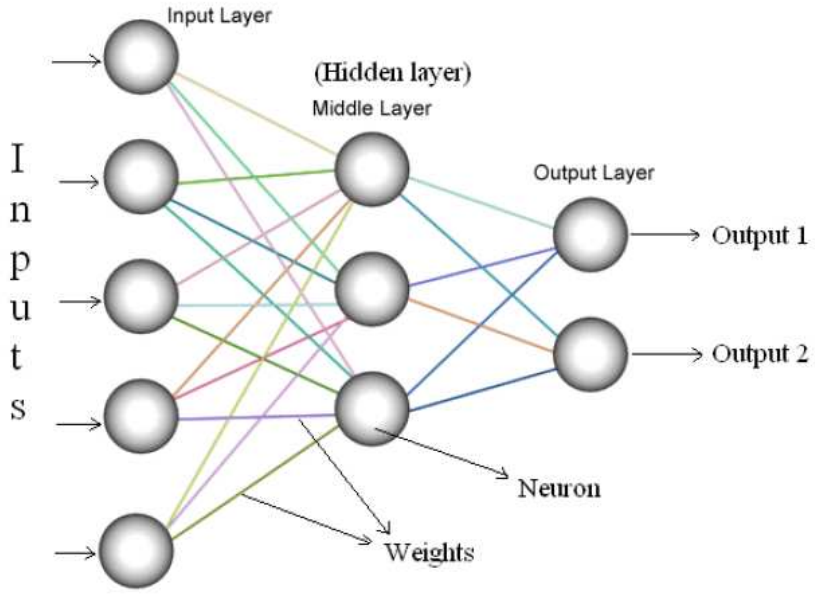

Figure 1.2: Single Hidden Layer Neural Network Structure

1.4 Concurrent Learning

A known problem in adaptive control or online system identification problems is that modeling errors are compansated using instantaneous measured data only. This is referred to as instantaneous learning throughout the thesis. It is commonly a known fact that the use of instantaneous learning causes a re-learning process of the error dynamics over time. On the other hand, intelligent learning algorithms are expected to learn the error dynamics such that the modeling error of a previously encountered maneuver should decrease gradually when that maneuver is repeated in the future.

In [33, 34, 35], a long term learning flight controller is developed such that the expectation mentioned above is satisfied quiet successfully. In those studies, a novel approach to adaptive control, which uses the instantaneous information as well as recorded information for weight adaptations, is proposed. It is shown that by using a stack of recorded information along with the instantaneous or online information, it is possible to guarantee a long term learning in the adaptive flight controller, which increases the performance when the controller encounters a maneuver that has been performed in the past.

In this thesis, the methodology mentioned above is used for neural network weight updates in order to improve the performance of the predictor algorithm over time. This weight update

law is called *concurrent learning* throughout the thesis since the recorded information and the instantaneous information are used concurrently.

1.5 Contribution of the Thesis

This thesis has the following contributions to the area of limit detection:

1. Neural network based adaptive dynamic trim and control limit estimation algorithm, which is developed in [30, 27, 29, 28], is evaluated for short term and long term simulations. In short term simulations, as expected, accurate dynamic trim and control limit predictions are obtained by using instantaneous learning. On the other hand, in long term simulations, where maneuvers are repeated over and over, it is seen that inaccurate control limit predictions are higly probable.

2. Concurrent learning which uses a stack of recorded data online for weight adaptations is implemented to neural network based adaptive dynamic trim and control limit estimation algorithm.

3. For concurrent learning, online data storing is needed. In this thesis data from transient and steady state responses are recorded independently into different history stacks. A data storing criteria which has been used for the design of a long term learning flight controller in [33, 34, 35] is used for recording data from transient response. In order to record data from a steady state, a new data storing criateria is imposed. Comparisons of weight adaptations using transient data only and using transient plus steady state data are presented in the scope of dynamic trim predictions.

4. It is shown that the accuracy of dynamic trim and control limit predictions, using instantaneous measured data only, have a potential to decrease over time, especially in long term simulations. It is demonstrated that by use of concurrent learning better online models as well as dynamic trim and control limit predictions are obtainable for long term simulations.

5. It is shown that ”Learn while flying” capability of neural network based adaptive dynamic trim and control limit estimation algorithm is enhanced when concurrent learning is enabled.

1.6 Structure of the Thesis

In this thesis, fundamental concepts are given in sections 1.2 through 1.4 by introducing the basics of envelope protection, dynamic trim concept and concurrent learning. Chapter-3 and 4 are generally based on the ideas and the contributions given in the first chapter.

In the second chapter, 6DOF modeling of a well known general aviation aircraft Cessna182 is presented. Chapter-2 ends with comparisons of eigen values and simulation results which are used for validating the developed nonlinear model.

In chapter-3, neural network based adaptive dynamic trim algorithm is used for online estimation of approaching aircraft limits of Cessna182. For that purpose, angle of attack, load factor and pitch rate dynamics are constructed online. Then, for known flight envelope limits, constructed models are inverted and solved for critical controls.

In chapter-4, the benefits of concurrent learning over instantaneous learning are presented in the scope of dynamic trim and control limit predictions.

CHAPTER 2

MATHEMATICAL MODEL

2.1 Mathematical Modeling

In this chapter mathematical modeling of a well known general aviation aircraft, Cessna182 is presented. General six degree of freedom nonlinear equations of motion are used for nonlinear modeling. In order to model forces and moments linear aerodynamic coefficients are used [24]. Modeling environment is chosen as Matlab/Simulink environment due to the simplicity of modeling implementation. Since developed model is used for dynamic trim predictions in the following chapters, validation of the model is also required. Although linear models are not used in the following chapters, they are used for validation of the developed nonlinear model. For validation purposes developed nonlinear model is trimmed and linearized in Matlab environment. Then the longitudinal and lateral modes are compared with the ones which are found in [24]. Dynamic responses of the nonlinear model is compared with the responses of transfer functions given in [24] as well. After validations it is concluded that the developed model can be used for dynamic trim and control limit predictions in the following chapters.

2.1.1 General Nonlinear Equations of Motion

It is a well known fact that it is possible to express aircraft dynamics as a set of nonlinear ordinary differential equations. Derivation of the state equations presented here can be found in [17, 18, 20]. Also the equations used are valid for the following assumptions:

1. The airplane is a rigid body in the motion under consideration.
2. The airplane's mass is constant during the simulations.
3. Earth is assumed to be a fixed non-rotating frame.
4. Curvature of the earth is also neglected.
5. x-z plane of the body frame is the plane of mirror symmetry.

Translational equations and rotational equations are expressed in terms of body axes components (u, v, w, p, q, r). The following equations [18] are used for six degree of freedom modeling :

$$\dot{u} = -qw + rv - g\sin(\theta) + \frac{q_{dyn}S}{m}C_X + \frac{T}{m} \tag{2.1}$$

$$\dot{v} = -ru + pw + g\cos(\theta)\sin(\phi) + \frac{q_{dyn}S}{m}C_Y \tag{2.2}$$

$$\dot{w} = -pv + qu + g\cos(\theta)\cos(\phi) + \frac{q_{dyn}S}{m}C_Z \tag{2.3}$$

$$\dot{p} = \frac{1}{I_xI_z - I_{xz}^2}\{q_{dyn}S\,b(I_zC_R + I_{xz}C_N) - qr(I_{xz}^2 + I_z^2 - I_yI_z) + pqI_{xz}(I_x - I_y + I_z)\} \tag{2.4}$$

$$\dot{q} = \frac{1}{I_y}\{q_{dyn}S\,cC_M - (p^2 - r^2)I_{xz} + pr(I_z - I_x)\} \tag{2.5}$$

$$\dot{r} = \frac{1}{I_xI_z - I_{xz}^2}\{q_{dyn}S\,b(I_xC_N + I_{xz}C_R) + pq(I_{xz}^2 + I_x^2 - I_xI_y) - qrI_{xz}(I_x - I_y + I_z)\} \tag{2.6}$$

The position of the aircraft is obtained by transforming $[uvw]^T$ from body frame to earth frame. L_{EB} which is defined as body frame to earth frame transformation matrix is given below [17]:

$$L_{EB} = \begin{Bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{Bmatrix} \tag{2.7}$$

By integrating the following set of equation,

$$\begin{Bmatrix} \dot{X}_e \\ \dot{Y}_e \\ \dot{Z}_e \end{Bmatrix} = L_{EB} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{2.8}$$

the position of the aircraft can be found. Moreover, in order to solve equation 2.8 a kinematic relation between Euler angular rates and body angular rates is required. In equation 2.9 necessary kinematic relation is presented [18].

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \begin{Bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{Bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \tag{2.9}$$

Note that relation 2.9 becomes discontinues when $\theta = \pm\pi/2$. Therefore, all the maneuvers of the following chapters are done within θ limits ($\pm\pi/2$).

2.1.2 Force and Moment Models

As a general approach forces and moments acting upon the aircraft can be thought as a sum of aerodynamic, engine and gravitational forces/moments.

Aerodynamic forces and moments are modeled by use of linear aerodynamic coefficients which are taken from [24]. According to Table 2.2 in [24] nondimensional force and moment coefficients ($C_D, C_Y, C_L, C_l, C_m, C_n$) are written in the following form:

$$C_D = C_{D_0} + C_{D_\alpha}\alpha + C_{D_{\delta_e}}\delta_e \tag{2.10}$$

$$C_Y = C_{Y_\beta}\beta + C_{Y_p}\bar{p} + C_{Y_r}\bar{r} + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \tag{2.11}$$

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_{\dot{\alpha}}}\bar{\dot{\alpha}} + C_{L_q}\bar{q} + C_{L_{\delta_e}}\delta_e \tag{2.12}$$

$$C_l = C_{l_\beta}\beta + C_{l_p}\bar{p} + C_{l_r}\bar{r} + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \tag{2.13}$$

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\dot{\alpha}}}\bar{\dot{\alpha}} + C_{m_q}\bar{q} + C_{m_{\delta_e}}\delta_e \tag{2.14}$$

$$C_n = C_{n_\beta}\beta + C_{n_p}\bar{p} + C_{n_r}\bar{r} + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r \tag{2.15}$$

In equations 2.10 to 2.15, it is assumed that longitudinal aerodynamics are related to longitudinal parameters only. As well, lateral aerodynamics are related to lateral parameters [24].

Note that the above expressions are defined in stability axes system. Since translational and rotational equations of motion (equations 2.1 and 2.6) are written in body frame a transformation from stability axes to body axes is needed. For that purpose $(C_D, C_Y, C_L, C_l, C_m, C_n)$ are first dimensionalized in equations 2.16 and 2.17. Then the transformation matrix defined in equation 2.20 is used for transforming aerodynamic forces and moments to body axes. Equations 2.21 and 2.22 represent nondimensional aerodynamic coefficients.

$$D = C_D\, q_{dyn} S \quad Y = C_Y\, q_{dyn} S \quad L = C_L\, q_{dyn} S \tag{2.16}$$

$$l = C_l\, q_{dyn} S\, b \quad m = C_m\, q_{dyn} S\, \bar{c} \quad n = C_n\, q_{dyn} S\, b \tag{2.17}$$

$$\begin{Bmatrix} X_b \\ Y_b \\ Z_b \end{Bmatrix} = L_{BS} \begin{Bmatrix} D \\ Y \\ L \end{Bmatrix} \tag{2.18}$$

$$\begin{Bmatrix} R_b \\ M_b \\ N_b \end{Bmatrix} = L_{BS} \begin{Bmatrix} l \\ m \\ n \end{Bmatrix} \tag{2.19}$$

$$L_{BS} = \begin{Bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{Bmatrix} \tag{2.20}$$

$$C_X = X_b/(q_{dyn} S) \quad C_Y = Y_b/(q_{dyn} S) \quad C_Z = Z_b/(q_{dyn} S) \tag{2.21}$$

$$C_R = R_b/(q_{dyn} S\, b) \quad C_M = M_b/(q_{dyn} S\, \bar{c}) \quad C_N = N_b/(q_{dyn} S\, b) \tag{2.22}$$

In order to model engine forces a very simple modeling technique is selected. A constant thrust of 5000 N is applied on center of gravity along with the x-axes of the body frame. For

trimming purposes total thrust is found according to the percent throttle position. Dummy thrust model is as follows:

$$T = (5000N)\frac{\delta_{thr}}{100}; \qquad 0 \leq \delta_{thr} \leq 100 \tag{2.23}$$

Equations 2.21,2.22 and 2.23 are used as inputs to six degree of freedom equations 2.1 to 2.6.

Table 2.1: Aircraft data on which aerodynamic model is based, Ref [24]

S	174	ft^2
$\bar{c}$	4.9	ft
b	36	ft
$Altitude$	5000	ft
$TAS,\ u$	220	ft/s
$Dynamic\ pressure,\ q_{dyn}$	49.6	lbs/ft^2
$CG\ location$	26.4	$fraction\ \bar{c}$
$Angle\ of\ attack$	0	deg
$Weight$	2650	lbs
I_{xx}	948	$slugft^2$
I_{yy}	1346	$slugft^2$
I_{zz}	1967	$slugft^2$
I_{xz}	0	$slugft^2$

Table 2.2: Longitudinal and Lateral Stability Derivatives of Cessna182, Ref [24]

C_{D_0}	0.0270	C_{l_β}	−0.0923
C_{D_u}	0	C_{l_p}	−0.484
C_{D_α}	0.121	C_{l_r}	0.0798
C_{L_0}	0.307	C_{y_β}	−0.393
C_{L_u}	0	C_{y_p}	−0.075
C_{L_α}	4.41	C_{y_r}	0.214
$C_{L_{\dot{\alpha}}}$	1.7	C_{n_β}	0.0587
C_{L_q}	3.9	C_{n_p}	−0.0278
C_{m_0}	0.04	C_{n_r}	−0.0937
C_{m_u}	0	$C_{l_{\delta_a}}$	0.229
C_{m_α}	−0.613	$C_{l_{\delta_r}}$	0.0147
$C_{m_{\dot{\alpha}}}$	−7.27	$C_{y_{\delta_a}}$	0
C_{m_q}	−12.4	$C_{y_{\delta_r}}$	0.187
$C_{D_{\delta_e}}$	0	$C_{n_{\delta_a}}$	−0.0216
$C_{L_{\delta_e}}$	0.43)	$C_{n_{\delta_r}}$	−0.0645
$C_{m_{\delta_e}}$	−1.122		

2.1.3 Alfadot Estimation

Since $\dot{\alpha}$ derivatives are used for modeling aerodynamic forces and moments its estimation is required. For estimating $\dot{\alpha}$ we start with the following expression:

$$\alpha = \arctan(\frac{w}{u}) \tag{2.24}$$

by taking the time derivative of both sides $\dot{\alpha}$ is obtained as:

$$\dot{\alpha} = \frac{\dot{w}u - \dot{u}w}{u^2 + w^2} \tag{2.25}$$

During simulations $\dot{\alpha}$ is followed by a memory block so that the expected algebraic loops are avoided. Also, $\dot{\alpha}$ becomes zero during trimming since $\dot{u}$ and $\dot{w}$ are made zero.

2.1.4 Atmosphere Model

It is a known fact that a multiplication with $\frac{1}{2}\rho V^2 S$ is needed to calculate actual forces and moments. As a result the air density ρ is required for solving equations of motion. For that purpose 'International Standart Atmosphere' model is used. Necessary equations can be found in refs [25] and [26] along with the calculation of calibrated and equivalent airspeeds.

2.1.5 Calculation of Accelerations and Specific Forces

In this section acceleration expressions in the vehicle's centre of gravity are presented. Details are given in Ref [21]. The body axes components of acceleration vector **a** can be written as:

$$a_x = \frac{1}{m}(X_{gravity} + X_{Thrust} + X_b) \tag{2.26}$$

$$a_y = \frac{1}{m}(Y_{gravity} + Y_{Thrust} + Y_b) \tag{2.27}$$

$$a_z = \frac{1}{m}(Z_{gravity} + Z_{Thrust} + Z_b) \tag{2.28}$$

Specific forces (unitless parameters) are calculated by eliminating gravity terms and dividing thrust and aerodynamic forces by the weight:

$$A_x = \frac{1}{W}(X_{Thrust} + X_b) \tag{2.29}$$

$$A_y = \frac{1}{W}(Y_{Thrust} + Y_b) \tag{2.30}$$

$$A_z = \frac{1}{W}(Z_{Thrust} + Z_b) \tag{2.31}$$

2.2 Matlab/Simulink Environment

Basic components of the developed nonlinear model is presented in fig2.1 along with the definitions of input vector, output vector and state vector. It is sure that the sequence of states of the state vector must be known exactly for trimming and linearization purposes. Generally for all simulink models the sequence of continuous states, in other words the state vector itself, can easily be reached by using the related simulink object: 'Simulink.BlockDiagram.getInitialState'. By using this command the sequence states defined in fig2.1 is obtained.

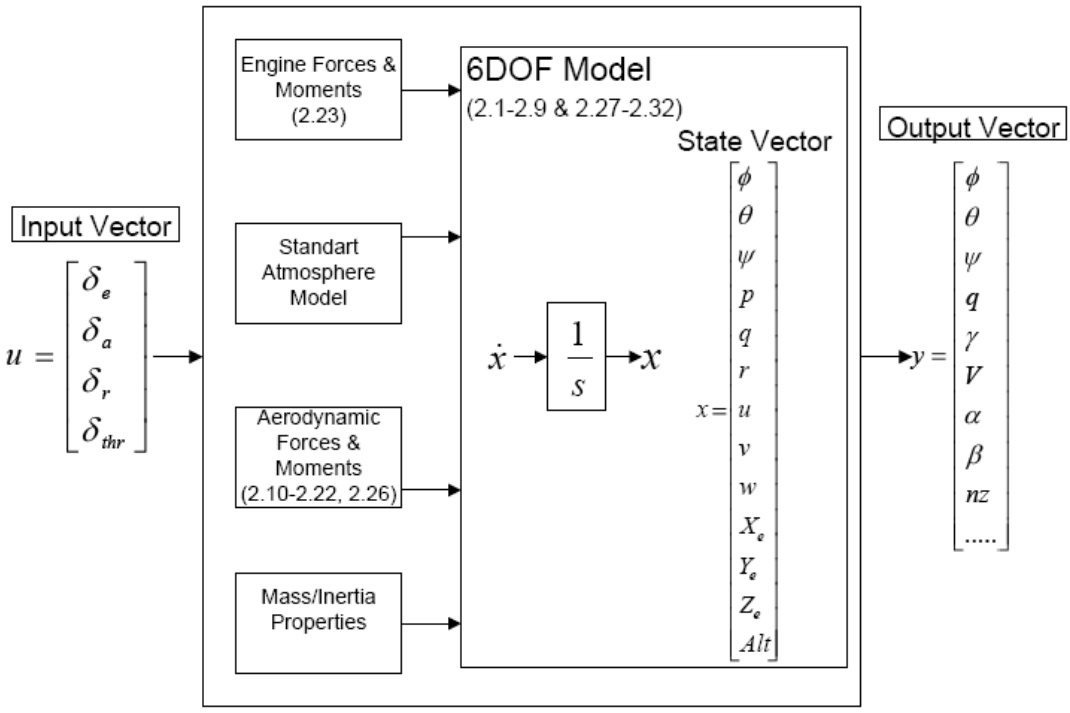

Figure 2.1: Basic Components of Developed Simulink Model

2.3 Validation of Developed Model

Validation is done in two respects. Firstly, developed nonlinear model is trimmed and linearized around the trim point given in [24]. Then the eigen values of the linear model is compared with the known eigen values of the aircraft which is given in the same reference. Secondly, starting from the same trim point, the responses of the available transfer functions of [24] are compared with the responses of the developed nonlinear model.

2.3.1 Trimming

In Matlab environment, there exist simulink tools or routines which can be used to find equilibrium points of a nonlinear dynamical system. Using trim routines, it is possible to find steady values for the states while satisfying the mathematical relations between outputs, inputs and states of the system. Trim points are needed since they will be the initial conditions of the simulations. In this thesis, an easy method is used for trimming the aircraft. In order to find a trim point, the following procedure is followed:

1. A state vector is chosen manually at the desired airspeed and altitude.

2. Input vector is also chosen such that all surface deflections made zero and throttle input is set to 40 percent.

3. Simulation is started from that point. After simulation starts, PID controllers are used for controlling the airspeed and the altitude at the desired conditions. Therefore, a steady condition is found after a time later. A speed controller is used for keeping the aircraft at the desired trim speed by changing the throttle. As well, an altitude controller is used for keeping the altitude at the desired trim altitude with using elevator. If there were lateral effects, lateral inputs could have been used for keeping roll and yaw rates as zero for a level flight trim.

4. The resulting steady condition of the simulation is recorded as a trim point and used as an initial condition.

2.3.2 Linearization

Simulink has some linearization routines as well. A nonlinear simulink model can easily be linearized around a specified trim point by using *linmod* command. The command *linmod* extracts a linear state space model of a nonlinear continuous system at a given operating point. For our case, *linmod* is used for linearizing the system at the trim point which is found previously. A and B matrices presented in appendix A.2 are obtained.

$$A = \begin{Bmatrix} -0.0453 & 0.0797 & 0.2637 & -9.8099 & 0 & 0 & 0 & 0 \\ -0.2976 & -2.1089 & -65.2924 & 0.0392 & 0 & 0 & 0 & 0 \\ 0.0101 & -0.2081 & -6.8265 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.1877 & -0.4683 & -66.7113 & 9.8099 \\ 0 & 0 & 0 & 0 & -0.4512 & -13.0178 & 2.1357 & 0 \\ 0 & 0 & 0 & 0 & 0.1395 & -0.3349 & -1.2185 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.0040 & 0 \end{Bmatrix} \tag{2.32}$$

2.3.3 Comparison of Longitudinal and Lateral Modes with [24]

Eigen values of matrix A can be found in the table 2.3. It is seen that the eigen values obtained are very close to the values presented in [24].

Table 2.3: Comparison of Modes with Ref[24]

	Matlab	ROSKAM [24]
Short Period	-4.4689 ± 2.8326	-4.4497 ± 2.8240
Phugoid	-0.0214 ± 0.1695	-0.0220 ± 0.1697
Dutch Roll	-0.6700 ± 3.1822	-0.6702 ± 3.1748
Roll	-13.06574	-13.0127
Spiral	-0.0184	-0.0180

2.3.4 Comparison of Model Responses with [24]

For validation of the developed model longitudinal and lateral responses are compared with the ones given in [24]. Pitch attitude and angle of attack responses are chosen for comparing longitudinal dynamics. Elevator input is given to both models (linear model of [24] and nonlinear model developed) as a doublet input. Results can be seen in Figure 2.2.

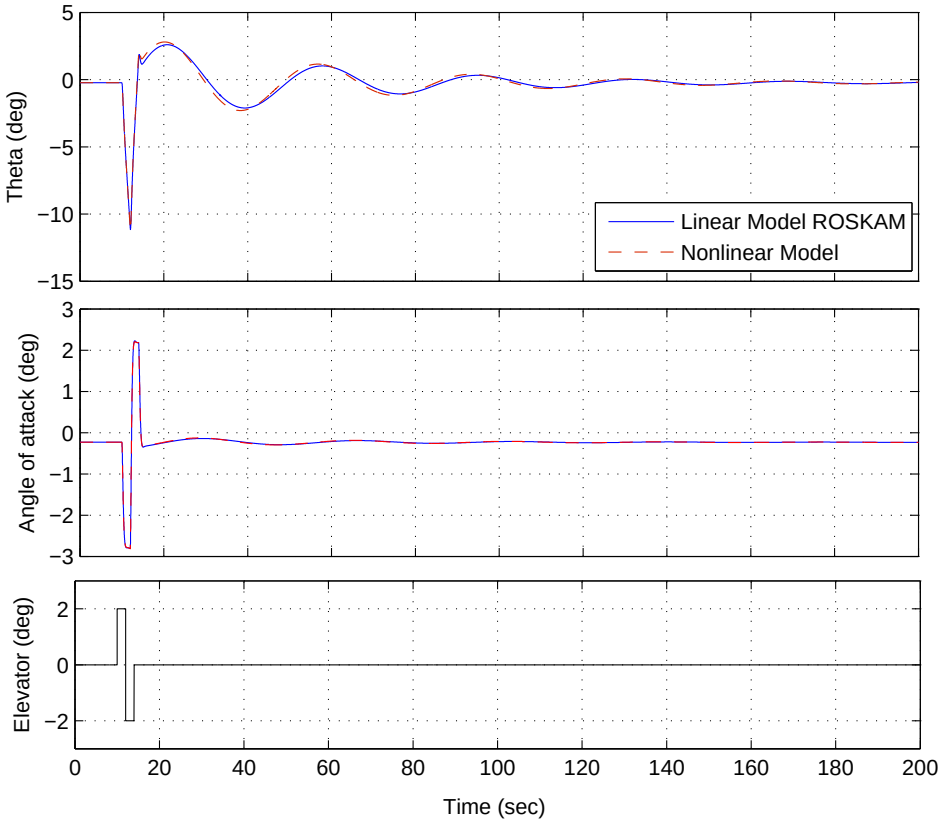

Figure 2.2: Comparison of Linear Model (Ref [24]) and Nonlinear Model Responses for a doublet Elevator input

Roll attitude, sideslip angle and heading angle responses are chosen for comparing lateral dynamics. Aileron input is applied as a doublet input to both systems while other controls are held zero. Results are presented in Figure 2.3. For comparing lateral dynamics due to rudder input only, one more simulation is presented too. Responses due to rudder input only can be seen in Figure 2.4.

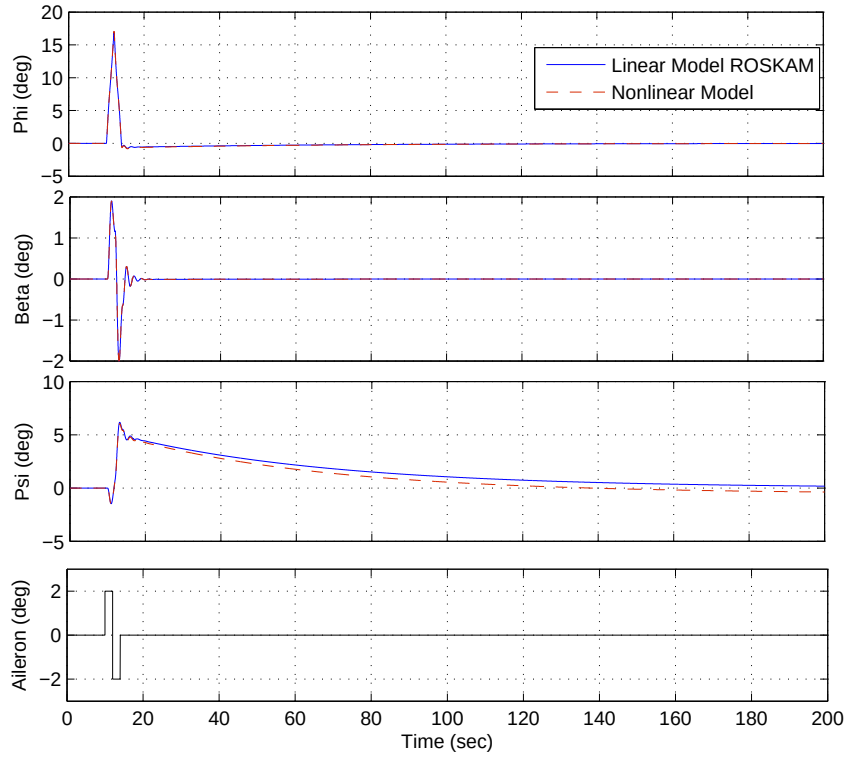

Figure 2.3: Comparison of Linear Model [24] and Nonlinear Model Responses for a doublet Aileron input

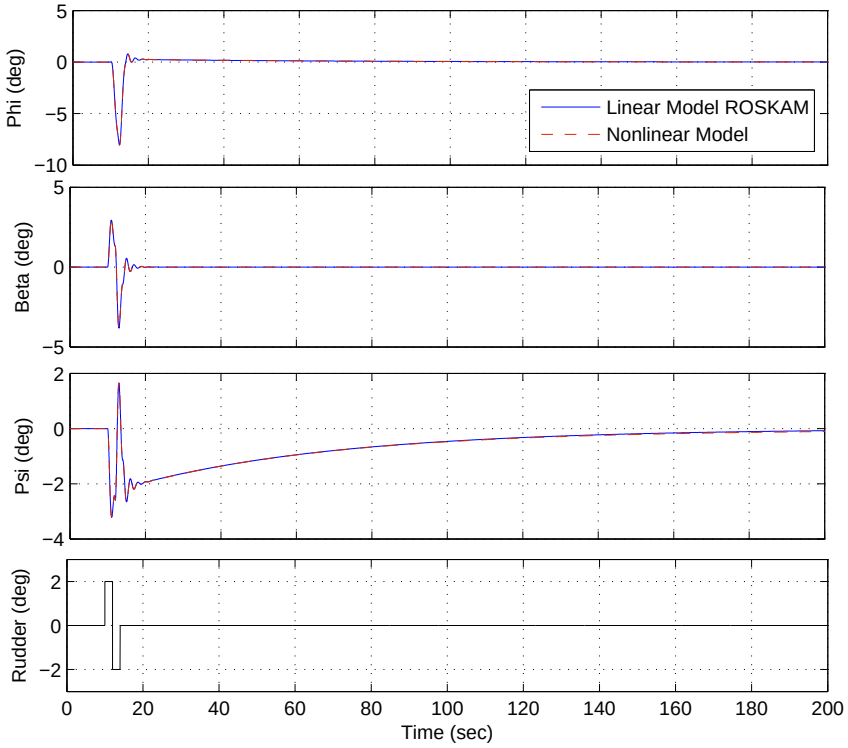

Figure 2.4: Comparison of Linear Model [24] and Nonlinear Model Responses for a doublet Rudder input

2.4 Conclusion

In this chapter 6DOF modeling of a general aviation aircraft is presented. Validation of the model is also performed. The model developed in this chapter is used for dynamic trim and control limit predictions of the following chapters.

CHAPTER 3

ONLINE ESTIMATION OF MANEUVERING STEADY-STATES AND CONTROL LIMITS

3.1 Introduction

This chapter covers a known approach for online estimation of maneuvering steady states and control limits of a general aviation aircraft. Maneuvering steady state predictions of angle of attack, load factor and pitch rate parameters are presented. Especially, angle of attack and load factor predictions are needed by envelope protection systems since they dominate the flight envelope limits. For that purpose, linear models representing angle of attack, load factor and pitch rate dynamics are constructed. In order to represent the real dynamics, these linear models are augmented with adaptive terms such as adaptive neural networks. Once the neural network augmented linear models are constructed and adaptation is maintained it is possible to calculate the steady states of $\dot{\alpha}$, $\dot{n}_z$ and $\dot{q}$ dynamics. Also for known α, n_z and q limits, it is shown that it may possible to obtain the critical control vector by inverting the constructed models.

Neural network augmentation of linear models are done with the help of single hidden layer neural networks which are known to be universal approximators in the literature. Gaussian and complementary Gaussian activation functions are used in the hidden layer. For the neural network weight updates, a classical weight update law, which is also used in recent studies [14, 15, 16], is applied.

As an application, a general aviation aircarft, Cessna182, is exposed to various pull-up, push-over maneuvers. During these maneuvers approaching α, n_z and q limits along with the prediction of control limits are presented. Moreover, maneuvers are done with various aggrassive

control inputs too. This is done for demonstrating the violation of known limits and to show the available lead time just before the actual violation occurs. With the analysis performed, it is shown that the proposed algorithm can predict limit violations in an effective lead time. Results are also harmonious with recent studies [27, 28, 29, 30] for predicting rotorcraft limits. Simulations are also done starting from a different trim point. It is seen that the adaptive term can represent real dynamics for different flight conditions, resulting with accurate predictions.

As a motivaton for the next chapter, aircraft is exposed to pull-up and push-over maneuvers over and over again with the same input scenario. Here, the aim was to see the long term response of predictions. It is seen that there exist a re-learning process of aircraft dynamics over time due to the oscillatory behavior of weight updates. In other words the models generated online are generally local models, they are very far from representing a global model. This is also demonstrated by freezing the weights after a long time of simulation. It is seen that even if the weights come to a steady-state, local oscillations may continue and generated models may still be local models. Increasing the globality of neural network based linear models and its effect on dynamic trim predictions along with prediction of control limits are treated in the next chapter, by using a different learning algorithm.

3.2 Theoretical Development

Let us represent the equations of motion of an aircraft with the following nonlinear state equations:

$$\dot{x} = f(x, u); \qquad x(t_0) = x_0, \qquad x \in \Re^n, u \in \Re^p, \tag{3.1}$$

where x is the state vector with known initial condition, x_0, and u is a known control vector. The vector field f: $\Re^n \times \Re^p \longrightarrow \Re^n$ is continuous and satisfies a global Lipschitz condition with respect to x so that the solution $x(t)$ to the differential equation 3.1 is unique for any finite initial condition x_0 and any $u(t)$ in the control space [30].

The state x can be divided into fast and slow states, such that

$$\dot{x}_f = f_1(x_f, x_s, u), \tag{3.2}$$

$$\dot{x}_s = f_2(x_f, x_s, u), \tag{3.3}$$

and,

$$x = [x_f \quad x_s]^T \quad , x_f \in \Re^l, x_s \in \Re^{n-l}. \tag{3.4}$$

Compared to the slow states the fast states reach a steady state much quicker during a maneuver, hence include states such as angular rates, angle of attack and sideslip. The slow states include flight parameters such as forward speed and Euler angles. Note that slow states could also include changes in parameters like weight, altitude, CG location, etc.

The dynamic trim condition is defined as

$$\dot{x}_f = 0. \tag{3.5}$$

Let $\hat{f}_1$ represent an approximation of $f_1 = f_1(x_f, x_s, u)$. Then the actual fast state dynamics can be written as

$$\dot{x}_f = \hat{f}_1 + \xi(x_f, x_s, u), \tag{3.6}$$

where the modeling error in the fast state dynamics is represented by $\xi = \xi(x_f, x_s, u)$, such that

$$\xi = f_1 - \hat{f}_1. \tag{3.7}$$

Consider the following model approximation for the estimation of the fast state dynamics:

$$\dot{\hat{x}}_f = \hat{f}_1 + \Delta(x_s, \hat{x}_f, u, ...) + K(x_f - \hat{x}_f), \tag{3.8}$$

where 'hat's denote the estimated variables and $\Delta = \Delta(x_s, \hat{x}_f, u, ...)$ contains a set of known flight parameters or signals. K is called the observer gain matrix. In order to establish the error dynamics let us define the error between the true and the approximate fast states as

$$e = x_f - \hat{x}_f. \tag{3.9}$$

When both f and $\hat{f}$ are fed the same signals, by subtracting equation 3.8 from equation 3.6, the error dynamics can be written as

$$\dot{e} = -Ke + \xi - \Delta. \tag{3.10}$$

Thus, when the modeling error, ξ, can be cancelled through Δ, with a positive definite matrix K, any estimation error of the actual plant dynamics will decay asymptotically to zero. Otherwise, the term $(\xi - \Delta)$ will act as a forcing term to the error dynamics. Neural networks can be used to generate the signal Δ.

If the relationship in equation 3.8 is used, an estimate of the dynamic trim condition can be found, when $\dot{\hat{x}}_f = 0$:

$$\hat{f}_1(\hat{x}_{f_{DT}}, x_s, u) + \Delta(x_s, \hat{x}_{f_{DT}}, u, ...) + K(x_f - \hat{x}_{f_{DT}}) = 0. \tag{3.11}$$

where subscript 'DT' denotes dynamic trim. The term $K(x_f - \hat{x}_{f_{DT}})$ is the *error bias* and is commonly estimated using $K(x_f - \hat{x}_f)$. Therefore an estimate of the dynamic trim values of the fast states can be obtained by solving the following equation for $\hat{x}_{f_{DT}}$:

$$\hat{f}_1(\hat{x}_{f_{DT}}, x_s, u) + \Delta(x_s, \hat{x}_{f_{DT}}, u, ...) + Ke = 0. \tag{3.12}$$

Similarly, for a given (limiting) dynamic trim state, $x_{f_{DT_{lim}}}$, a prediction of the control vector, $\hat{u}_{DT}$, (the control vector that results in that particular dynamic trim state) can be calculated by solving the following set of algebraic equations for $\hat{u}_{DT}$:

$$\hat{f}_1(x_{f_{DT_{lim}}}, x_s, \hat{u}_{DT}) + \Delta(x_s, x_{f_{DT_{lim}}}, \hat{u}_{DT}, ...) + Ke = 0, \tag{3.13}$$

In this thesis, in order to construct f_1 of equation 3.8 linear models consist of $\hat{x}_f$, x_s and u are used. equation 3.8 takes the following form:

$$\dot{\hat{x}}_f = A[\hat{x}_f \; x_s]^T + Bu + \Delta(x_s, \hat{x}_f, u, ...) + K(e), \tag{3.14}$$

We can also rewrite equations 3.12 and 3.13 in the following forms:

$$A[\hat{x}_{f_{DT}} \; x_s]^T + Bu + \Delta(x_s, \hat{x}_{f_{DT}}, u, ...) + Ke = 0. \tag{3.15}$$

$$A[x_{f_{DT_{lim}}} \; x_s]^T + B\hat{u}_{DT} + \Delta(x_s, x_{f_{DT_{lim}}}, \hat{u}_{DT}, ...) + Ke = 0, \tag{3.16}$$

Iterative solutions can be expressed as follows:

$$\hat{x}_{f_{DT_{i+1}}} = -C_1^{-1}(C_2 x_s + B_1 \delta_e + \Delta_1(\hat{x}_{f_{DT_i}}, x_s, \delta_e, bias) + K_1 e_1) \tag{3.17}$$

$$\hat{u}_{DT_{i+1}} = -B_1^{-1}(C_1 \hat{x}_{f_{DT_{lim}}} + C_2 x_s + \Delta_1(\hat{x}_{f_{DT_{lim}}}, x_s, \hat{u}_{DT_i}, bias) + K_1 e_1) \tag{3.18}$$

where, C_1 and C_2 are the elements of A matrix.

Block diagram representation of equation 3.14 can be seen in Fig(3.1).

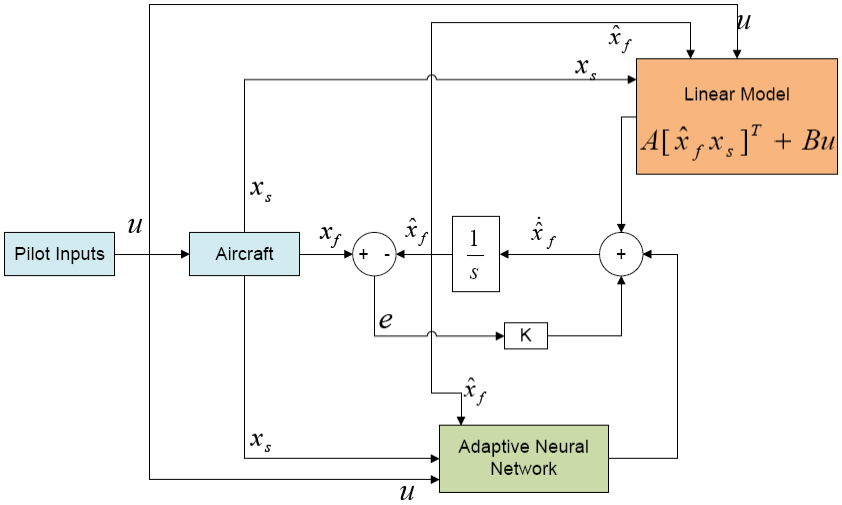

Figure 3.1: Observer Loop for Dynamic Trim Predictions

3.3 Single Hidden Layer Neural Network Augmentation

Single hidden layer neural networks (SHLNN) are used to estimate the function ξ mentioned above. In other words, the difference between real dynamics and approximate dynamics are compansated by these adaptive elements. In general, SHLNNs are universal approximators and consist of an input layer, a hidden layer and an output layer. The ideal and estimator dynamics can be written in these forms [27, 29]:

$$\dot{x}_f = A_1[x_f \;\; x_s]^T + B_1 u + W^{*^T}\beta(V^{*^T}\bar{x}) + \epsilon(t) \tag{3.19}$$

$$\dot{\hat{x}}_f = A_1[\hat{x}_f \;\; x_s]^T + B_1 u + \hat{W}^T\beta(\hat{V}^T\bar{x}) + K(x_f - \hat{x}_f) \tag{3.20}$$

where $\hat{W}$ and $\hat{V}$ are the estimates of the ideal weights W^* and V^* respectively. We can also define the weight errors as:

$$\tilde{W} = \hat{W} - W^* \tag{3.21}$$

$$\tilde{V} = \hat{V} - V^* \tag{3.22}$$

The error dynamics can be found by subtracting equation 3.20 from equation 3.19 :

$$\dot{e} = (A_1 - K)e - \hat{W}^T\beta(\hat{V}^T\bar{x}) + W^{*^T}\beta(V^{*^T}\bar{x}) + \epsilon(t) \tag{3.23}$$

Assume that P is the solution of the following Lyapunov equation:

$$(A_1 - K)^T P + P(A_1 - K) = -Q \tag{3.24}$$

where $Q > 0$.

The classic update law for ultimately bounded error and weight signals is obtained [27, 29]:

$$\dot{\hat{W}} = -(\beta - \beta^{'}\hat{V}^T\bar{x})(e^T P)L_{R_W} \tag{3.25}$$

$$\dot{\hat{V}} = -L_{R_V}\bar{x}(e^T P)\hat{W}^T\beta^{'}(V^T\bar{x}) \tag{3.26}$$

where L_{R_V} and L_{R_W} are the corresponding learning rates.

Single hidden layer neural network structure and related mathematical expressions are given in Fig 3.2.

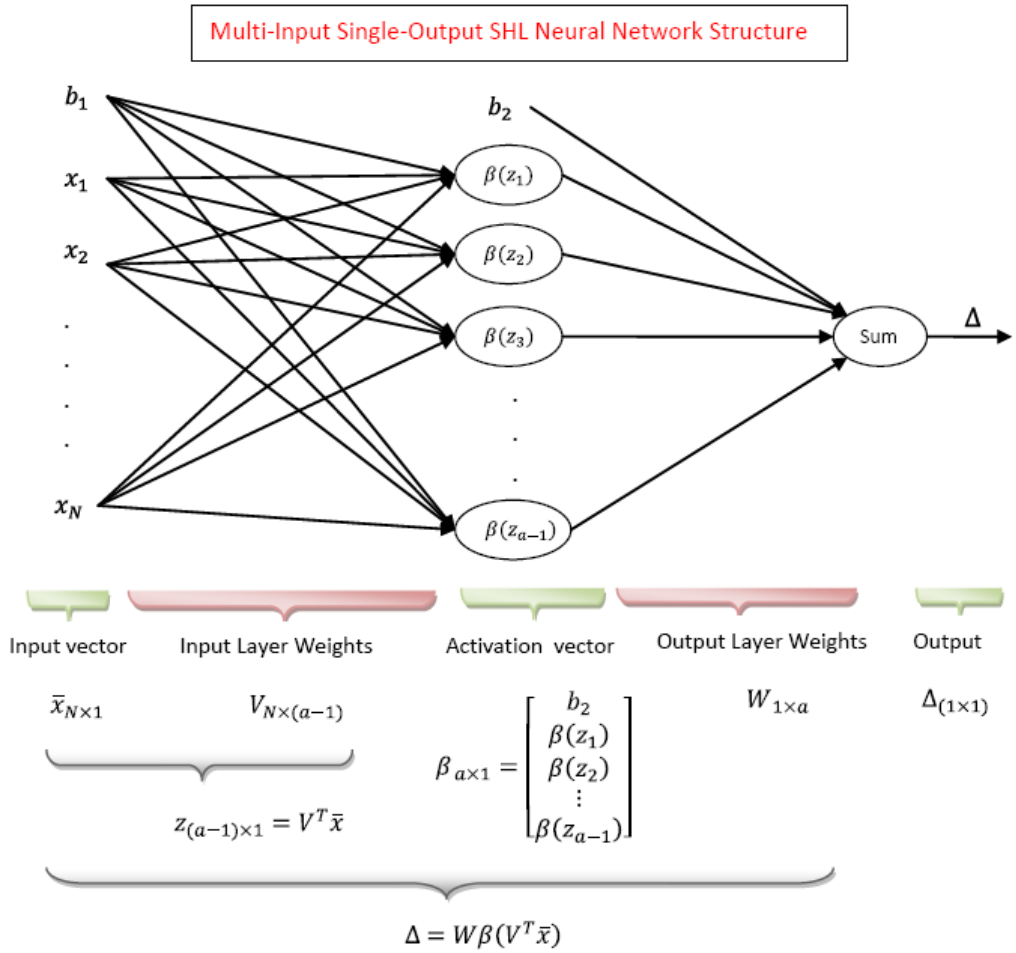

Figure 3.2: Multi-input Single-output Single Hidden Layer Neural Network Structure and Mathematical Expressions

3.4 Construction of Neural Network Augmented Linear Models

In this section we try to build up approximate models of $\dot{\alpha}$, $\dot{n}_z$ and $\dot{q}$ dynamics. Since there exist multiple limits equation 3.14 may written in the following form:

$$\begin{bmatrix} \dot{\hat{\alpha}} \\ \dot{\hat{n}}_z \\ \dot{\hat{q}} \end{bmatrix} = A \begin{bmatrix} \hat{\alpha} \\ \hat{n}_z \\ \hat{q} \\ V \end{bmatrix} + B[\delta_e] + K \begin{bmatrix} \alpha - \hat{\alpha} \\ n_z - \hat{n}_z \\ q - \hat{q} \end{bmatrix} + \begin{bmatrix} \triangle_1(\hat{\alpha}, V, \delta_e, 1) \\ \triangle_2(\hat{n}_z, V, \delta_e, 1) \\ \triangle_3(\hat{q}, V, \delta_e, 1) \end{bmatrix} \tag{3.27}$$

As a general approach for detecting multiple limits simultaneously [30], each fast state in $\hat{x}_f = [\hat{\alpha}\ \hat{n}_z\ \hat{q}]^T$ needs an observer loop as in fig 3.1. In this case we have 3 fast states and 3 observer loops. Note that the slow state V is directly measured and used in equation 3.27. Here, the first 3 by 3 part of A matrix is taken as a diagonal matrix assuming that there exist no functional dependence between fast states. Moreover, the fourth column of A is taken differently than zero implying that a functional dependence between each fast state and the slow state V can be established. Just to remind, x_f could be written as functions of other slow sates such as; pressure altitude, dynamic pressure, attitudes, body axes velocities and etc. In deed, if x_f is required to be as functions of all aircraft states, it would be more convenient to use a reduced order linearized model for the selection of A and B matrices. Whereas in this chapter and in the following chapters, A and B are selected with large modeling errors, in other words, adaptive neural networks are used significantly to represent existing and unknown modeling errors. Selection of K matrix is another issue. Since observer loops are independent, K matrix can be taken as a digonal matrix too. The acceptances stated here are mainly design choices and highly dependent on the user.

With the assumptions and acceptances above, approximate models of $\dot{\alpha}$, $\dot{n}_z$ and $\dot{q}$ can be written independently in the following forms:

$$\dot{\hat{\alpha}} = [A_{11}\ A_{14}] \begin{bmatrix} \hat{\alpha} \\ V \end{bmatrix} + B_1\delta_e + K_{11}(\alpha - \hat{\alpha}) + \Delta_1(\hat{\alpha}, V, \delta_e, bias) \tag{3.28}$$

$$\dot{\hat{n}}_z = [A_{22}\ A_{24}] \begin{bmatrix} \hat{n}_z \\ V \end{bmatrix} + B_2\delta_e + K_{22}(n_z - \hat{n}_z) + \Delta_2(\hat{n}_z, V, \delta_e, bias) \tag{3.29}$$

$$\dot{\hat{q}} = [A_{33}\ A_{34}] \begin{bmatrix} \hat{q} \\ V \end{bmatrix} + B_3\delta_e + K_{33}(q - \hat{q}) + \Delta_3(\hat{q}, V, \delta_e, bias) \tag{3.30}$$

For the above expressions, A and B constants are selected with large modeling errors which are expected to be compansated by adaptive elements. To make a proper initial guess, one can think of available lead time after a step control input as a constraint for selecting A_{11}, A_{22} and A_{33} constants. Or, they can be thought as time constants associated with related dynamics. For demonstration, different linear model responses and angle of attack response after a step control input are compared in fig 3.3. It is obvious that the linear model with $A_{11} = -5$ and $B_1 = -3$ has a lead time which is closer to the available lead time of model response. A similar approach can be used for n_z and q dynamics too.

Owing to the fact that the slow state V is directly measured and not observed, the multiplication of V with A_{14}, A_{24} and A_{34} will act as bias terms in equations 3.28, 3.30 and 3.29. As a result, the selection of A_{14}, A_{24} and A_{34} would not be as critical as the selection of A_{11}, A_{22} and A_{33}.

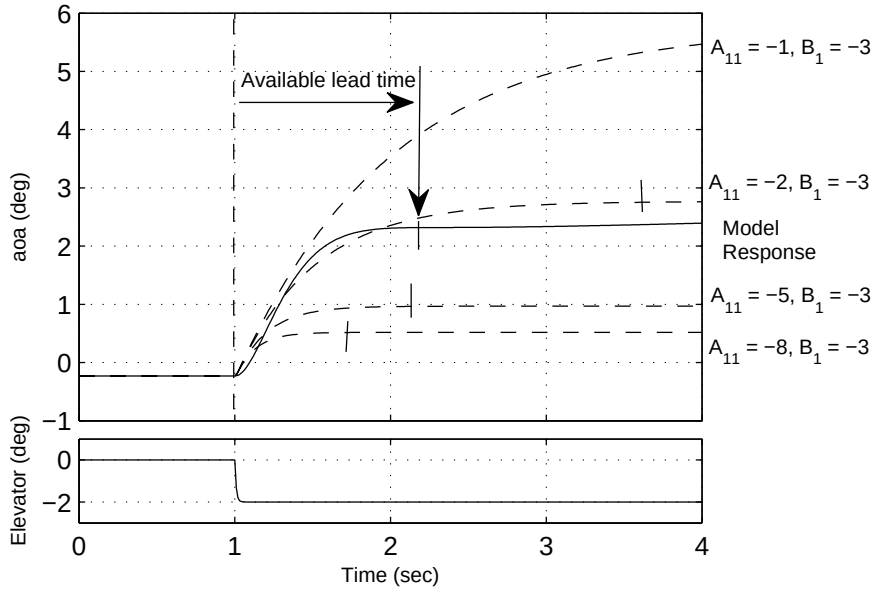

Figure 3.3: Different linear models and angle of attack response, Elevator input

A proper initial guess for selecting B constant can also be done. Let's rewrite equation 3.16 without neural network and Ke terms and solve for $\hat{u}_{DT}$:

$$\hat{u}_{DT} = -\frac{Ax_{f_{DT_{lim}}}}{B} \tag{3.31}$$

With equation 3.31 it is possible to find limit controls $\hat{u}_{DT}$ at known limit boundaries $x_{f_{DT_{lim}}}$. For angle of attack response limit boundaries are taken as 13 and -6 degrees. It is clear that an improper selection of B constants might cause too wide or too narrow control limits. This is demonstrated in fig 3.4. In that figure linear model with B=-1 has an enormous control margin which is also not practical for use due to the control surface deflection limits. As a result, those disscussions might be kept in mind during the selection of A and B constants. Besides selected A and B's are shown in table 3.1.

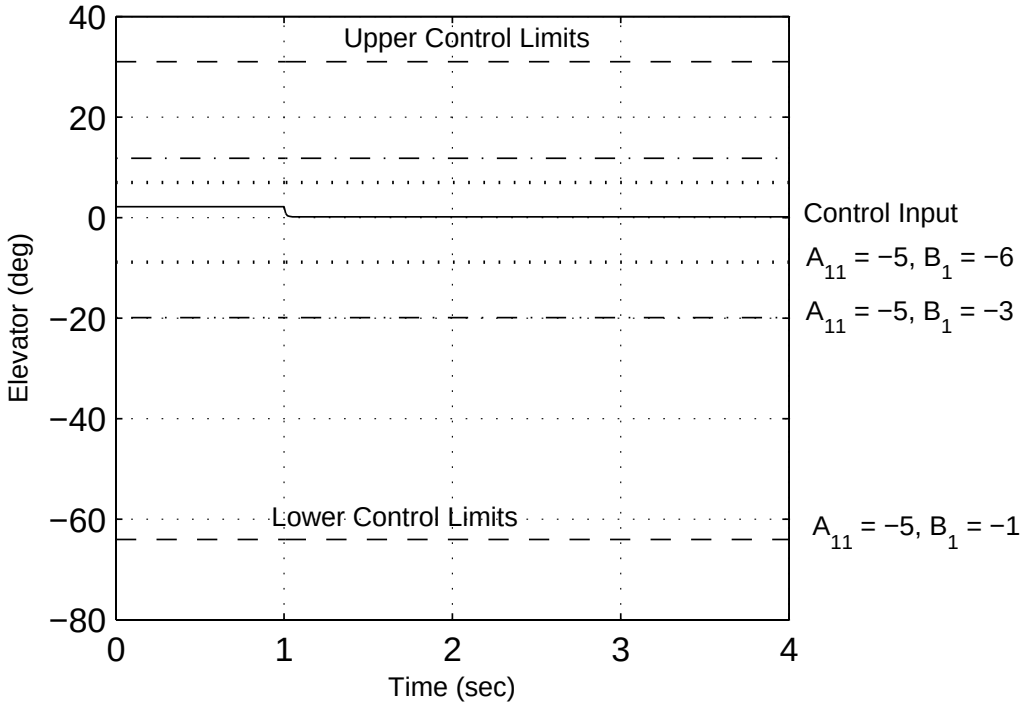

Figure 3.4: Control limits at known angle of attack limits, solution of equation 3.31

We also select proper observer gains K as a requirement of the observer loop. They are introduced to reduce the effect of the modeling error which may probably be nonzero at the initial condition. They are also used to make certain appropriate convergence of network weights.

Neural network structure selection is also another important point. Single hidden layer neural networks (SHLNN) are generally used in the literature due to their universal approximation capabilities. In the literature, sigmoidal functions in the hidden layer of SHLNN are used generally. In this thesis SHLNN which consist of 11 Gaussian and 11 complematary Gaussian activation functions in the hidden layer is used.

$$Gaussian\ activation\ function: \quad f(x) = e^{-(1+x/a)^2} \tag{3.32}$$

$$Complementary\ Gaussian\ activation\ function: \quad f(x) = 1 - e^{-(1+x/a)^2} \tag{3.33}$$

In equations 3.32 and 3.33 Gaussian and complementary Gaussian activation functions are presented. The activation potential a can be used as a scaling factor if selected adequately large.

To calculate the number of weights of Δ_1 network, we can use fig 3.2. Since we have 23 activation functions it makes 23 output layer weights (W). Also, Δ_1 has 4 inputs, that makes 4 by 22 input layer weight matrix (V). In total 111 weights are adjusted by the network. Δ_2 and Δ_3 networks have the same number of weights too.

Table 3.1: Selected A and B constants

$\dot{\hat{\alpha}}$	$A_{11} = -5$	$A_{14} = 0.1$	$B_1 = -3$
$\dot{\hat{n}}_z$	$A_{22} = -5$	$A_{24} = 0.5$	$B_2 = -3$
$\dot{\hat{q}}$	$A_{33} = -11$	$A_{34} = 0.8$	$B_3 = -10$

3.4.1 Dynamic Trim Predictions

Dynamic trim equation of $\dot{\hat{\alpha}}$ can be written in the following form of equation 3.28:

$$[A_{11}\ A_{14}]\begin{bmatrix}\hat{\alpha}_{DT}\\ V\end{bmatrix} + B_1\delta_e + \Delta_1(\hat{\alpha}_{DT}, V, \delta_e, bias) + K_1 e_1 = 0. \tag{3.34}$$

An approximate iterative solution for $\hat{\alpha}_{DT}$ may be obtained using:

$$\hat{\alpha}_{DT_{i+1}} = -A_{11}^{-1}(A_{14}V + B_1\delta_e + \Delta_1(\hat{\alpha}_{DT_i}, V, \delta_e, bias) + K_1 e_1) \tag{3.35}$$

Similar iterative solutions may be written for $\dot{\hat{n}}_z$ and $\dot{q}$ too:

$$\hat{n}_{z_{DT_{i+1}}} = -A_{22}^{-1}(A_{24}V + B_2\delta_e + \Delta_2(\hat{n}_{z_{DT_i}}, V, \delta_e, bias) + K_2 e_2) \tag{3.36}$$

$$\hat{q}_{DT_{i+1}} = -A_{33}^{-1}(A_{34}V + B_3\delta_e + \Delta_3(\hat{q}_{DT_i}, V, \delta_e, bias) + K_3 e_3) \tag{3.37}$$

Note that equations 3.35, 3.36 and 3.37 contain highly nonlinear adaptive terms which includes the dynamic trim variables. In general, dynamic trim equations will have at least one fixed point solution in as much as the bounded activation functions are used. The point which is not guaranteed is that there may occur more than one iterative solutions which might not be correct due to the high nonlinearty of the adaptive elements.

Generally, the initial value of $\hat{x}_{f_{DT_i}}$ should be taken as $\hat{x}_f$ in the first iteration. This way, it is highly probable to obtain a closer solution to the current neural network input vector. Besides, it is a commonly known experience [30, 13, 28] that even when the iteration number per simulation time step is taken as one ($i = 1$), the convergence error ($\hat{x}_{f_{DT_{i+1}}} - \hat{x}_{f_{DT_i}}$) may stay within acceptable ranges. And, it is demonstrated in references [30, 13, 28] that it is probable to obtain sufficiently correct dynamic trim solutions in one or two iterations. The dynamic trim solutions presented in this thesis are found with using one iteration per simulation time step. Also, simulations with two and four iterations are performed and convergences are compared.

Therefore, solutions can be found with acceptable convergence errors. The exact solutions are not afforded since there exist a possibility of converging to other fixed point solutions by increasing the number of iterations even more.

At a maneuvering steady state $\hat{x}_{f_{DT}}$ and $\hat{x}_f$ are equivalent. Therefore, a fixed point solution around the current neural network input vector is guranteed.

3.4.2 Control Limit Predictions

For known limit boundaries of α , equation 3.34 takes the following form:

$$[A_{11}\ A_{14}]\begin{bmatrix}\hat{\alpha}_{DT_{lim}}\\ V\end{bmatrix} + B_1\delta_{e_{DT}} + \Delta_1(\hat{\alpha}_{DT_{lim}}, V, \delta_{e_{DT}}, bias) + K_1e_1 = 0. \tag{3.38}$$

An iterative solution is given by:

$$\delta_{e_{DT_{i+1}}} = -B_1^{-1}(A_{11}\hat{\alpha}_{DT_{lim}} + A_{14}V + \Delta_1(\hat{\alpha}_{DT_{lim}}, V, \delta_{e_{DT_i}}, bias) + K_1e_1) \tag{3.39}$$

Similar iterative solutions may be written for limit controls due to known n_z and q limits too:

$$\delta_{e_{DT_{i+1}}} = -B_2^{-1}(A_{22}\hat{n}_{z_{DT_{lim}}} + A_{24}V + \Delta_2(\hat{n}_{z_{DT_{lim}}}, V, \delta_{e_{DT_i}}, bias) + K_2e_2) \tag{3.40}$$

$$\delta_{e_{DT_{i+1}}} = -B_2^{-1}(-A_{33}\hat{q}_{DT_{lim}} + A_{34}V + \Delta_1(\hat{q}_{DT_{lim}}, V, \delta_{e_{DT_i}}, bias) + K_3e_3) \tag{3.41}$$

Solution of control limits is much more different than dynamic trim solution. In equations 3.39, 3.40 and 3.41 neural network is evaluated at the known envelope limits. A probable problem may happen especially when the aircraft is flying far from limits. This is because, adaptive neural networks are expected to compansate for the modeling errors whatever the flight regime is, and at the same time, they are desired to represent the envelope limits as well. Therefore, it may not always be possible to result with accurate limit control predictions especially when limit dynamics are forgotten over time. This problem is treated in detail in the next chapter. In this chapter, satisfactory limit control predictions are obtained in short simulation times. And, in the next chapter it is shown that control limit predictions may get worse when simulations are evaluated at long term since networks may get highly nonlinear and forget about the previous limit violations in time.

Table 3.2: Selected Learning rates and Observer gains

	Δ_1 network	Δ_2 network	Δ_3 network
L_{R_W}	2000	2000	2000
L_{R_V}	3500	2000	3000

K_1	K_2	K_3
12	30	30

3.5 Simulations and Results

In this section dynamic trim predictions and control limit predictions during different pull-up push-over maneuvers are presented. Dynamic trims and control limits are iteratively found by solving equations 3.35 to 3.37 and equations 3.39 to 3.41 at each simulation time step. In all maneuvers here same configuration of neural network learning rates and observer gains are applied. They are presented in table 3.2.

For the first maneuver aircraft is trimmed at 121.5 KIAS and exposed to elevator inputs of different magnitudes. Results can be seen in fig 3.5.

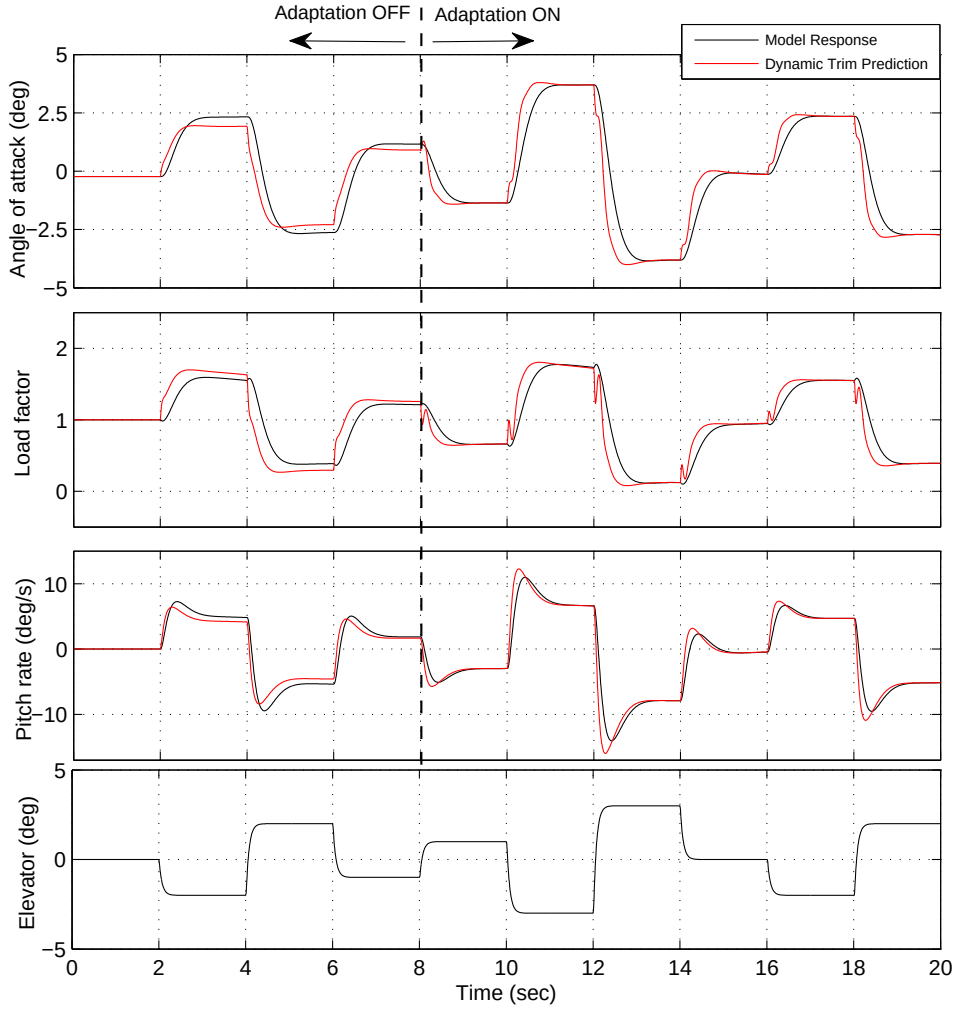

Figure 3.5: Dynamic trim predictions of α, n_z and q (Maneuver-1)

During the first eight seconds of the maneuver neural network adaptation is not used. In that time interval observer follows its own approximate dynamics and results with inaccurate predictions. After eight seconds, adaptation begins and dynamic trim predictions become accurate.

Modeling error compensation is also shown in fig 3.6. As expected prior to adaptation modeling error is compensated by observer gain. Whereas after the adaptation, the contribution of the neural network to the modeling error significantly increases and Ke converges to zero.

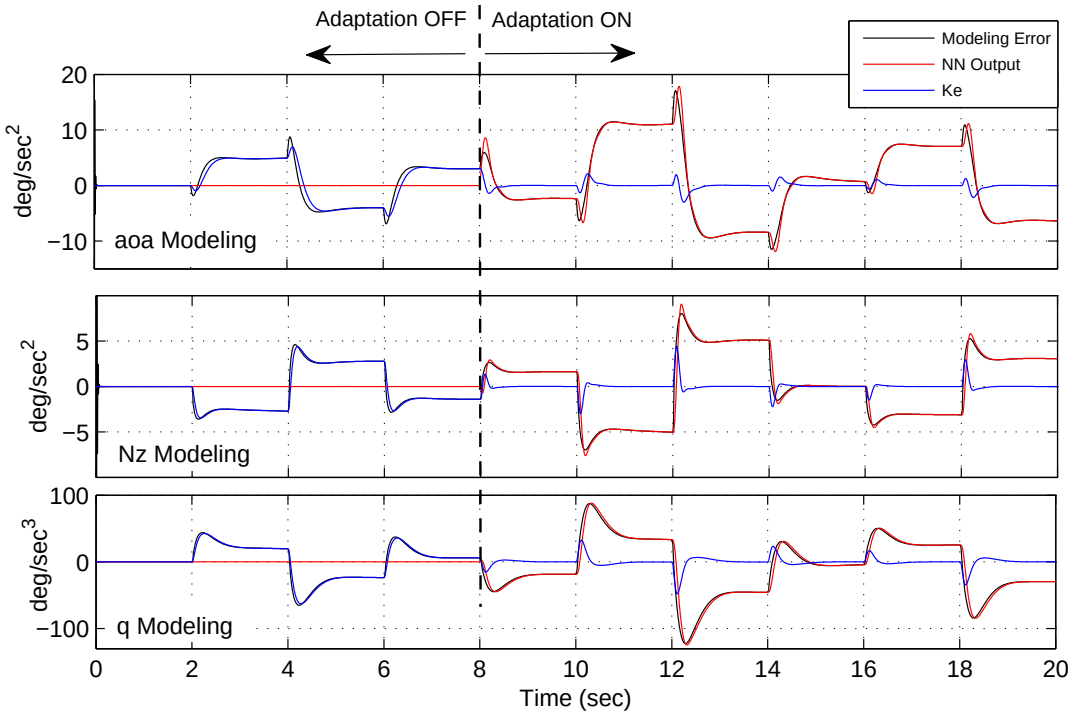

Figure 3.6: Modeling Error compensation for $\dot{\hat{\alpha}}$,$\dot{\hat{n}}_z$ and $\dot{\hat{q}}$ (Maneuver-1)

Since proximity to known envelope limits can be calculated by use of the dynamic trim condition for a given control input, it is desired to estimate the allowable control travel and cue the pilot at the time of control input, in other words before the limiting state is reached. To do that control limits at known envelope limits are calculated by inverting the dynamic trim equations and solving for controls. In this study, angle of attack and g limits are assumed to be as -6 to 13 degrees and -0.5 to 2.5 g's. Maximum and minimum pitch rate limits are assumed to be as -16 to 30 deg/sec. Control limits of the first maneuver is shown in fig 3.7.

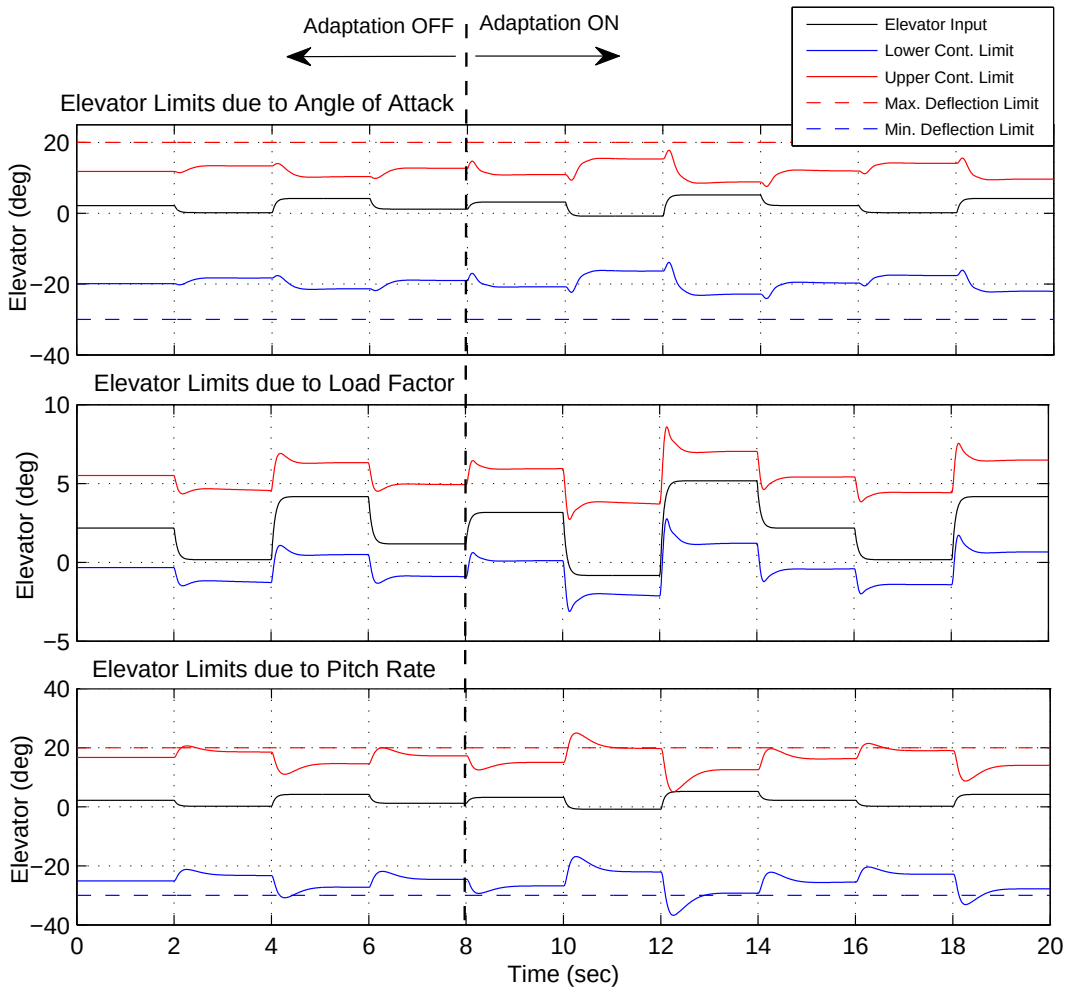

Figure 3.7: Limit controls due to α, n_z and q (Maneuver-1)

Note that control margins due the load factor limits are narrower than the ones obtained from α and q limits. According to the discussions of section 3.4, this should be mainly due to the selection of the B_2. Results may improve by a proper B constant.

As a second maneuver, aircraft is exposed to aggressive elevator inputs. Here, the aim is to demonstrate the violation of envelope limits as well as control limits. For that purpose after the twenty seconds of the first maneuver, simulation is contiuned and aggrassive inputs are given to the system. Figures 3.8, 3.9 and 3.10 include dynamic trim prediction along with control limit predictions. It is obvious that angle of attack, g and pitch rate limits are predicted in effective lead times. In addition to that whenever the dynamic trim predictions violate the known envelope limits, control limits are violated too. These information are mainly required by envelope protection systems in order to cue the pilot before the actual limiting state is reached.

Dynamic trim convergence $y_{p_{DT_{i+1}}} - y_{p_{DT_i}}$ and control limit convergences $u_{lim_{i+1}} - u_{lim_i}$ of Maneuver1-2 are demonstrated in figs3.11, 3.12 and 3.13 with different number of iterations per simulation time step. According to the results one iteration per simulation time step is adequately sufficient for dynamic trim predictions. As well, one iteration per simulation time step is sufficient for lower and upper control limit predictions in order to achieve a fixed point solution.

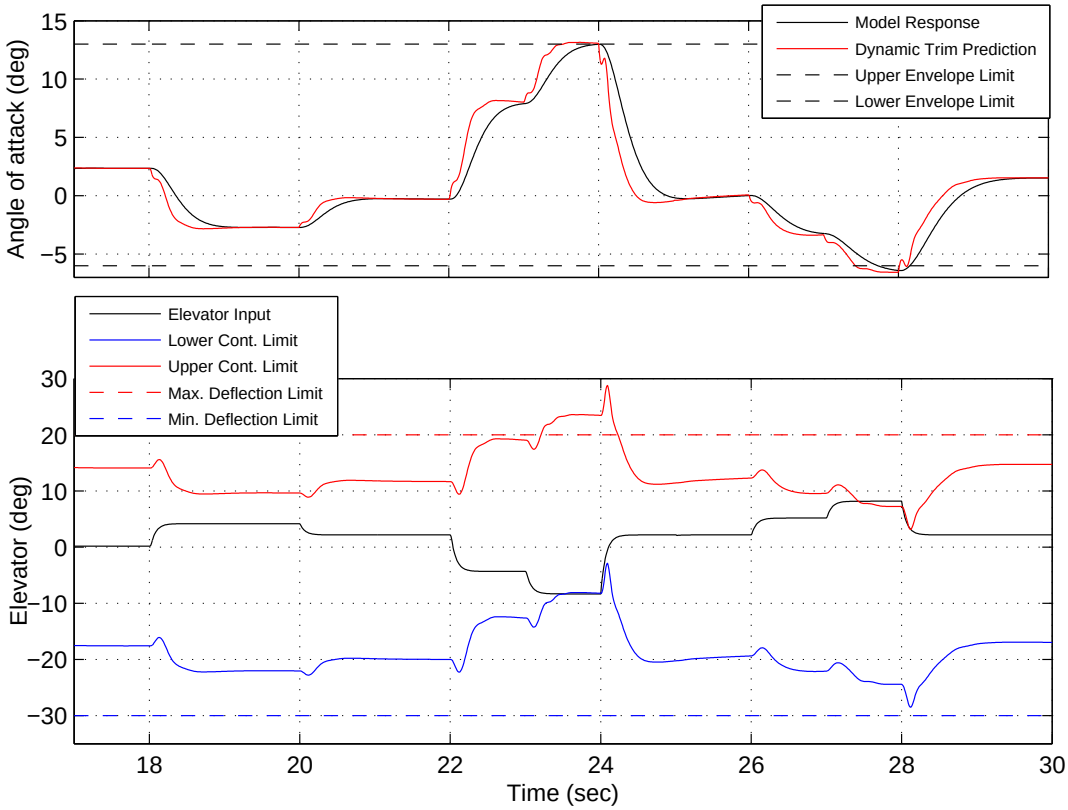

Figure 3.8: Dynamic trim predictions of α and limit controls due to α limits (Maneuver-2)

One more simulation is performed at a different trim point to show the adaptation capability of the adaptive scheme. Model is trimmed at 145 KIAS and the elevator scenario of the first maneuver is doubled. This is the third maneuver of this section. Dynamic trim predictions and control limit predictions are given in figs 3.14 and 3.15. Again predictions are made in effective lead times.

Aircraft states during first and second maneuvers can be seen in figures 3.16, 3.17 and 3.18.

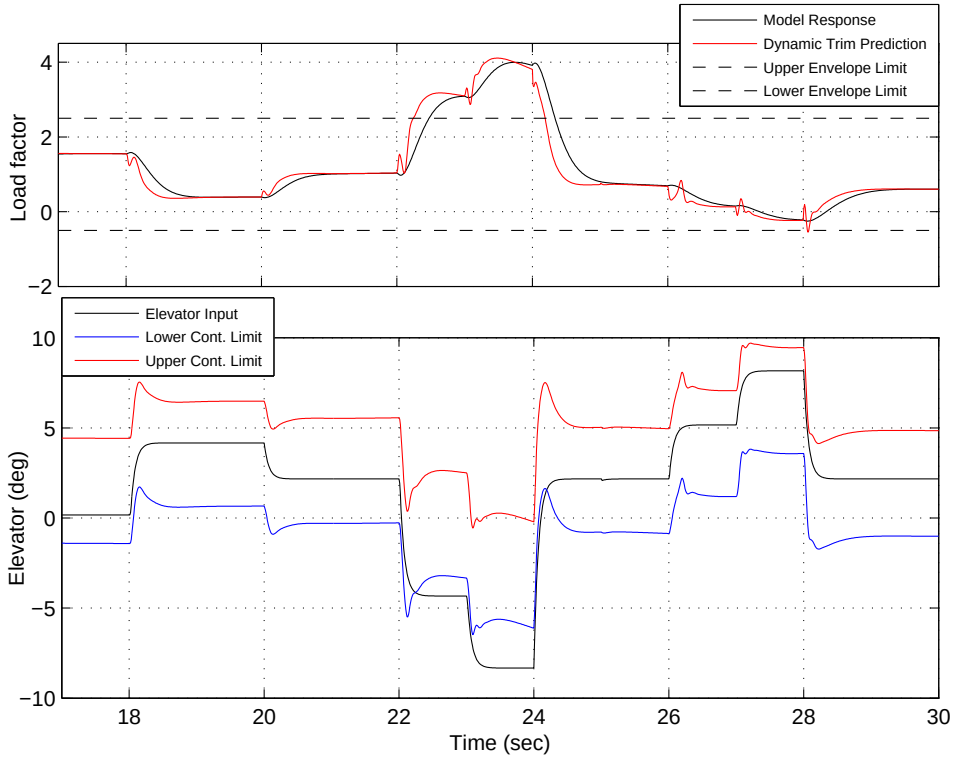

Figure 3.9: Dynamic trim predictions of n_z and limit controls due to n_z limits (Maneuver-2)

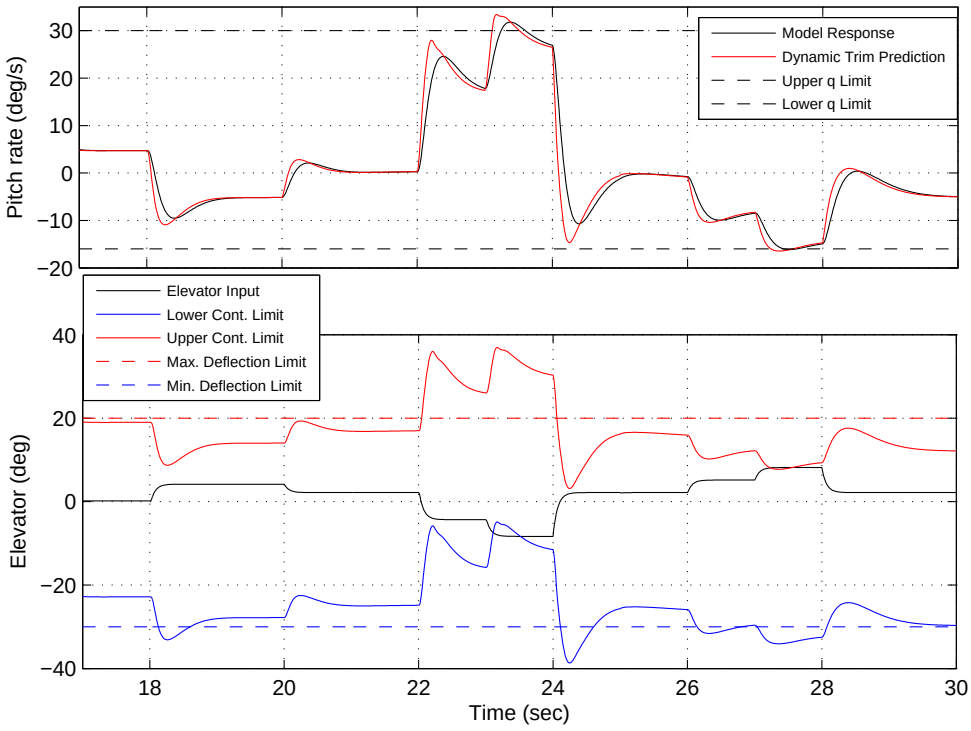

Figure 3.10: Dynamic trim predictions of q and limit controls due to q limits (Maneuver-2)

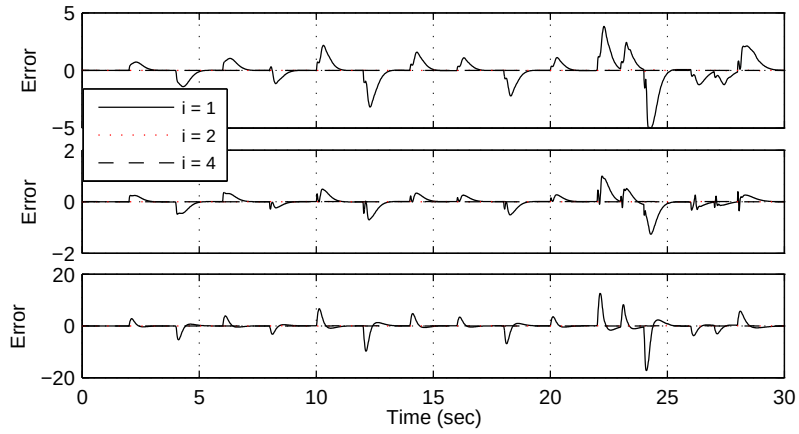

Figure 3.11: Top to bottom: Dynamic trim convergence errors of α, n_z and q predictions (Man1-2)

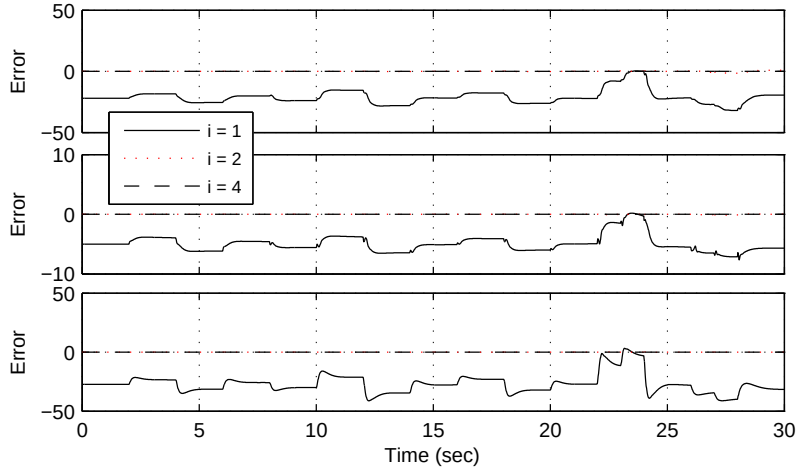

Figure 3.12: Top to bottom: Lower limit control convergence errors of α, n_z and q predictions (Man1-2)

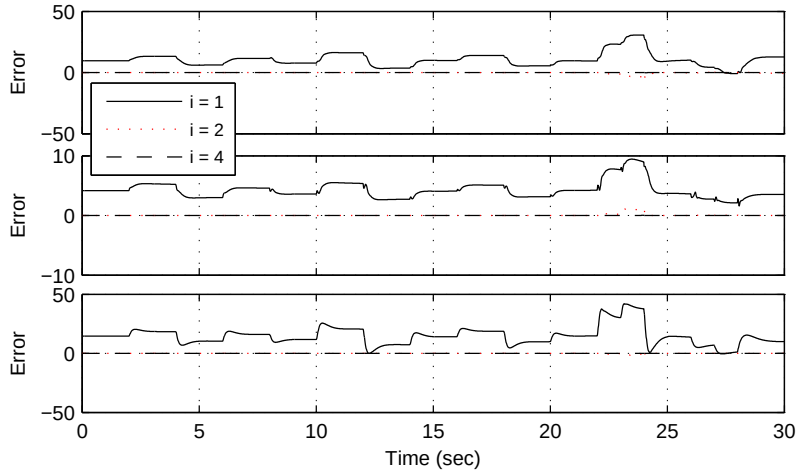

Figure 3.13: Top to bottom: Upper limit control convergence errors of α, n_z and q predictions (Man1-2)

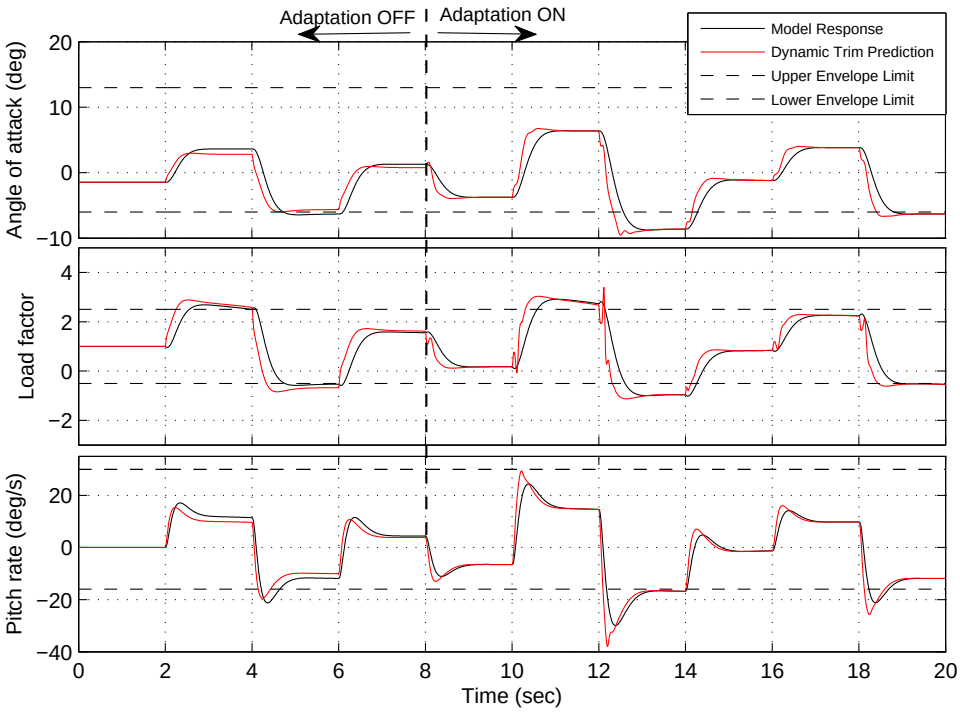

Figure 3.14: Dynamic trim predictions of α, n_z and q (Maneuver-3)

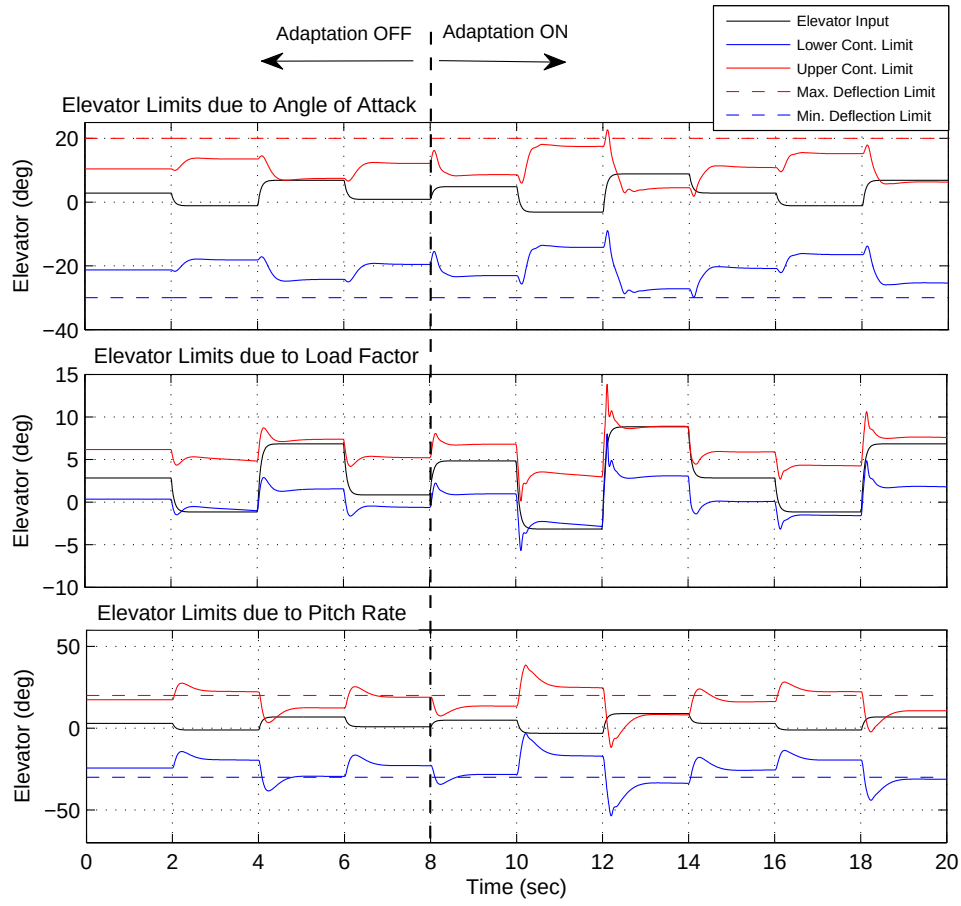

Figure 3.15: Limit controls due to α, n_z and q (Maneuver-3)

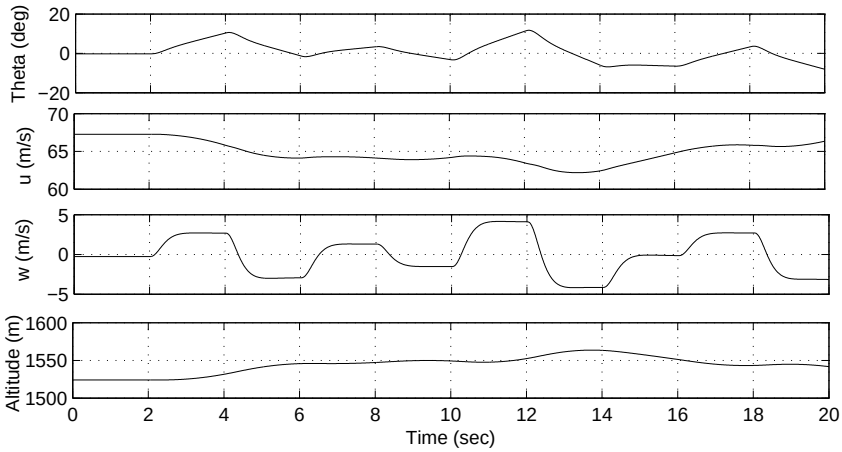

Figure 3.16: Aircraft States during Maneuver-1

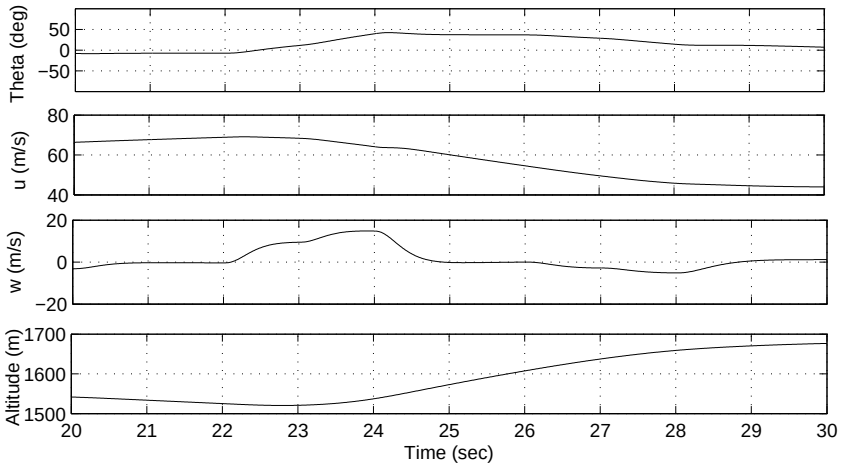

Figure 3.17: Aircraft States during Maneuver-2

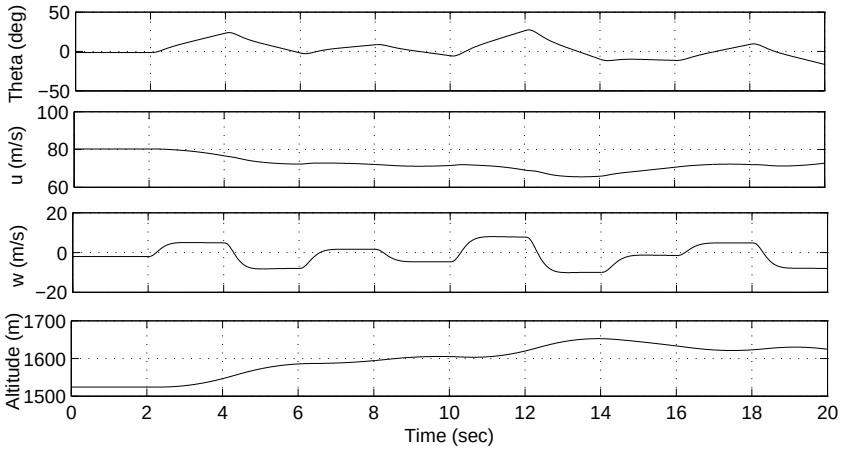

Figure 3.18: Aircraft States during Maneuver-3

3.6 Motivation for the Next Chapter

As it is mentioned before neural networks used here are universal approximators and should have been able to approximate the modeling error perfectly. The task is to train the networks in a short time such that the modeling error is adequately represented. Here the main problem is that network weights rarely converged to steady values. This implies that the aircraft dynamics is relearned during flight even if the performed maneuver has been encountered before. To demonstrate the re-learning process, weight updates of maneuvers 1,2 and 3 are presented in figs 3.19 and 3.20. It can be seen that the weight updates are oscillatory and not resulting steady values in that short time. As a result the generated models are accurate only locally.

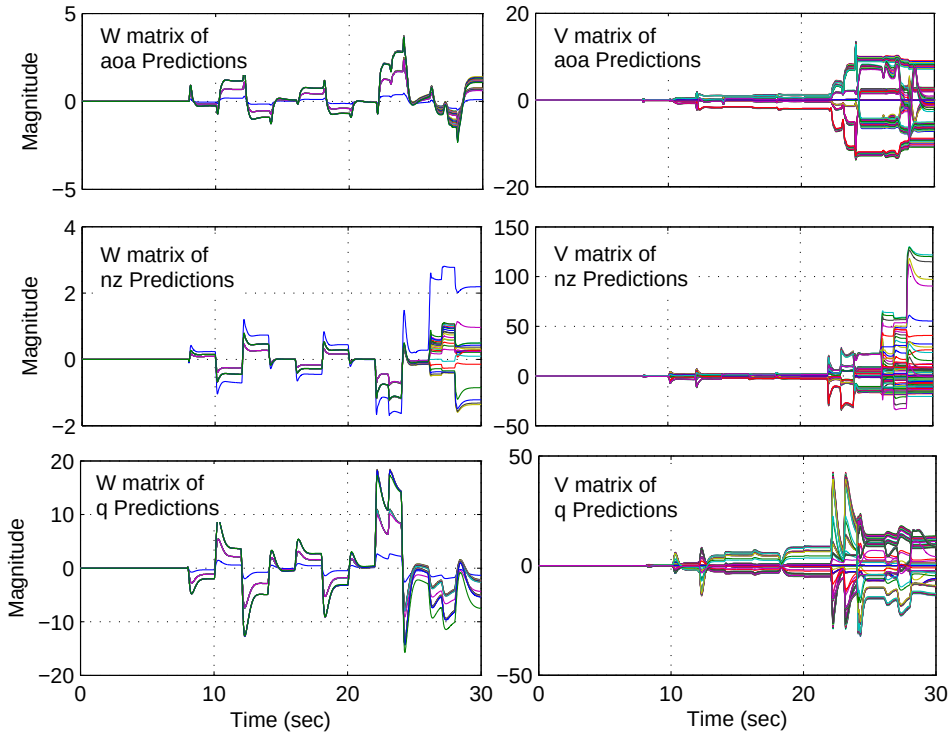

Figure 3.19: Neural Network Weight Updates (Maneuver-1 and 2)

Further, in order to support the ideas above a long term simulation is also performed. Maneuver-1 is run over 300 seconds. Indeed, this is not a practical scenario, just performed intentionally to have an understanding of the long term response of the weight updates. It is a commonly known fact that although the neural network weights can be located at nearly steady values, the local oscillations may still continue and the generated models may still be local models

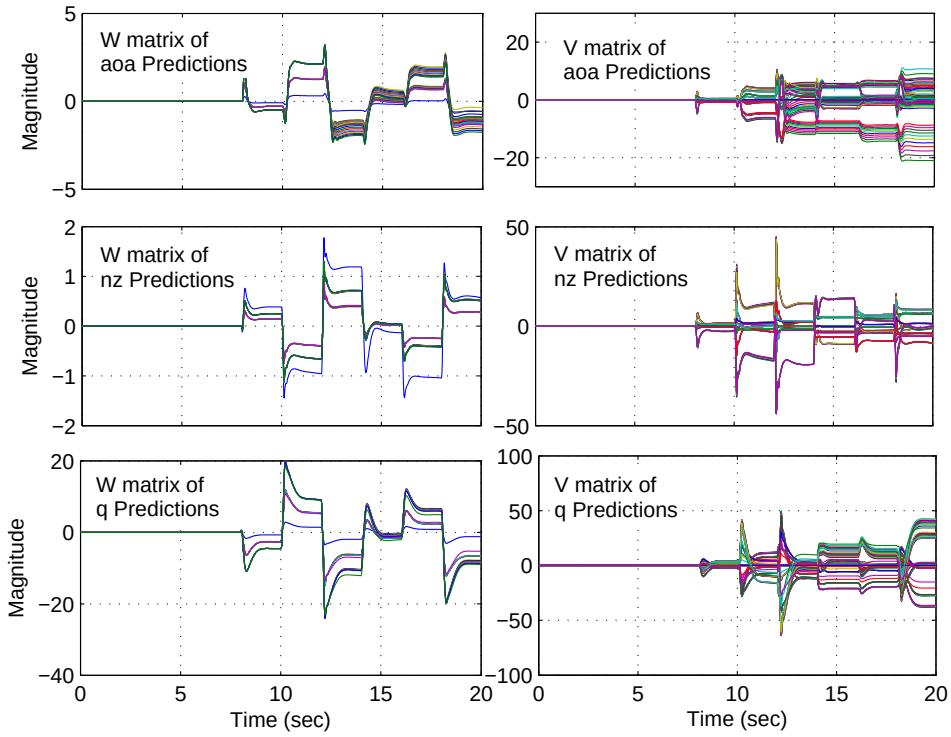

Figure 3.20: Neural Network Weight Updates (Maneuver-3)

which are far from representing a global model. In figs 3.21 and 3.22, it is shown that the neural network activation functions are well shaped after about 150 seconds. As well after about 150 seconds weights of Δ_1 and Δ_2 networks achieve nearly steady values. In order to have an insight about the globality/locality of the solutions weights are frozen at 300th second. Modeling error compensations before and after that time is presented in fig 3.23. As expected after weights are frozen neural network is not capable of approximating modeling errors anymore. This also indicates that the weights are not located in the proper locations in that long time of simulation. Learning law used in this chapter is called *instantaneous learning law* throught the thesis.

As it is mentioned in section 3.4.2 adaptive neural networks are expected to compansate for the modeling errors whatever the flight regime is, and at the same time, they are expected to represent the envelope limits as well. Therefore, in order to have accurate control limit solutions neural network is expected to learn the limit dynamics during limit violations and never forget those limits. This motivation is also treated in detail in the next chapter by performing long term simulations.

In the next chapter, a recently known neural network weight update law is implemented for dynamic trim predictions. A long term learning algorithm is implemented such that the encountered modeling errors are recorded and the information gained in the past is used for further adaptations. And, it is shown that generation of global models are more probable when the update law of next chapter is used.

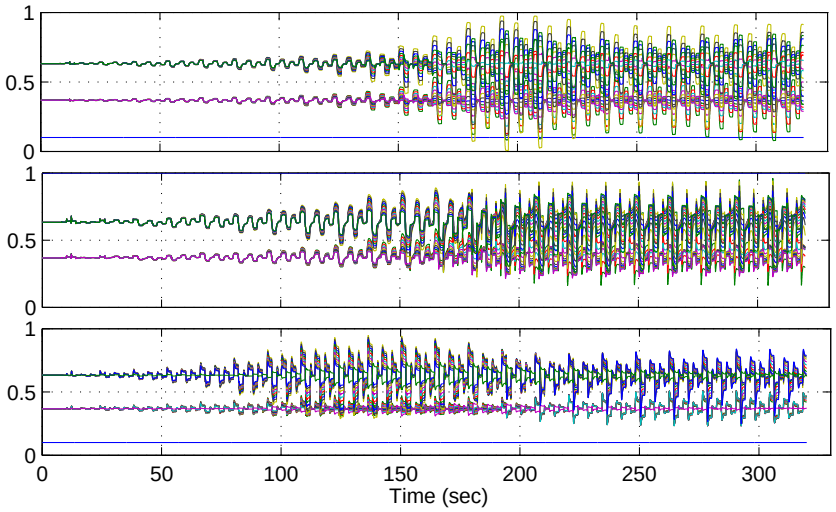

Figure 3.21: Activation function outputs of Δ_1, Δ_2 and Δ_3 networks (Maneuver-4)

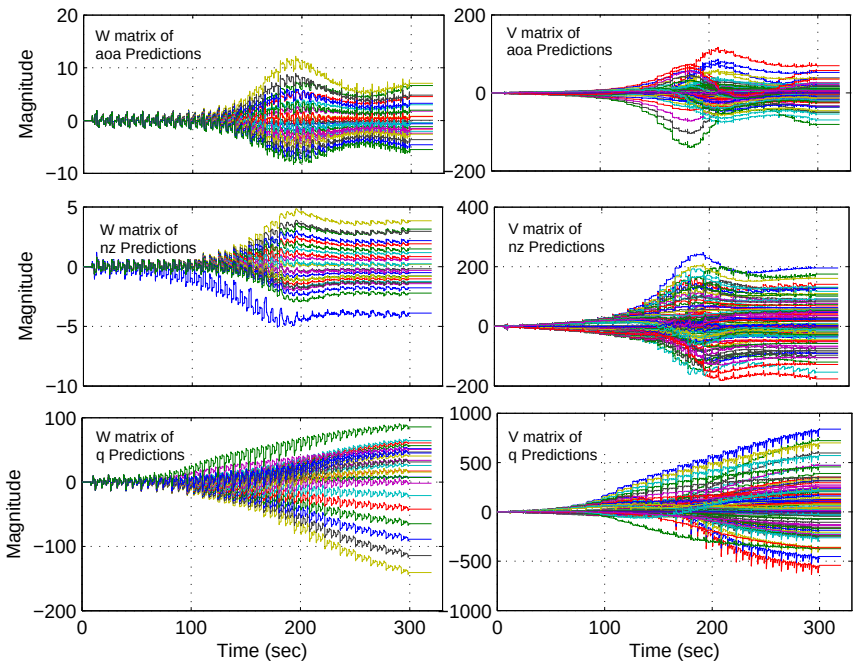

Figure 3.22: Neural Network Weight Updates (Maneuver-4)

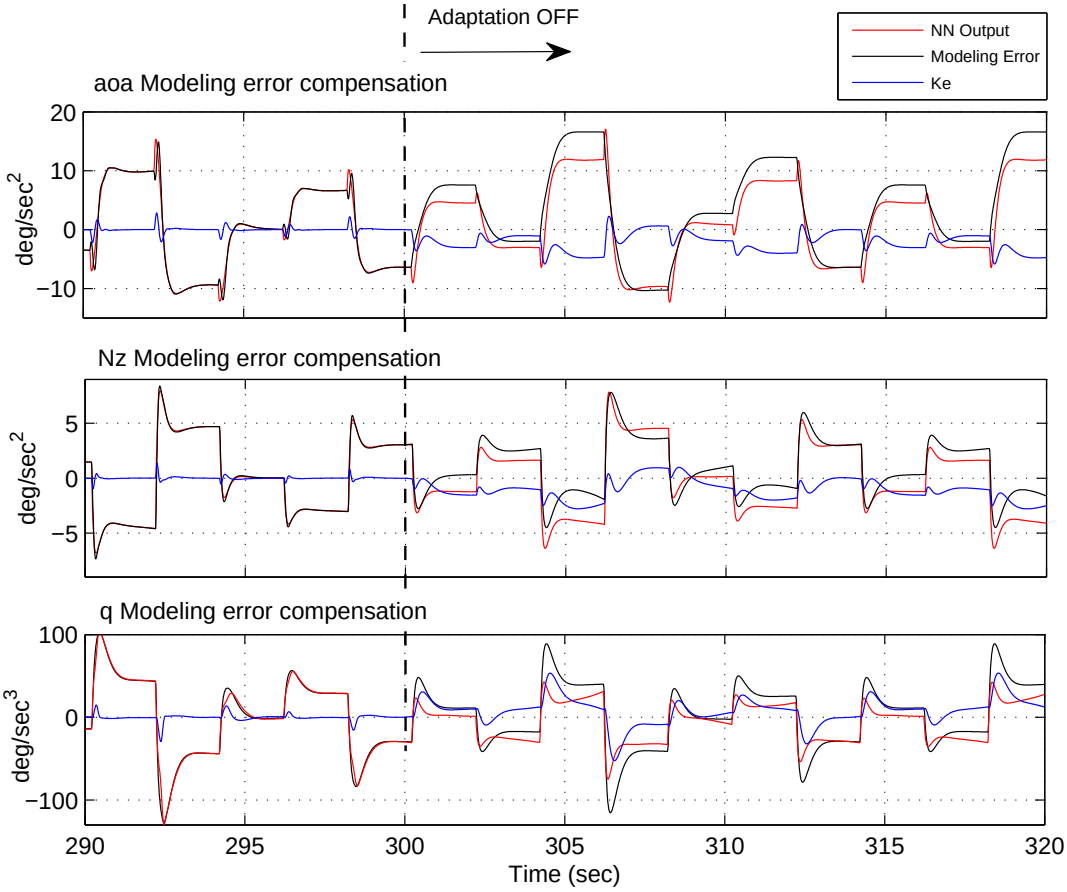

Figure 3.23: Modeling Error Compensations of α, n_z and q dynamics

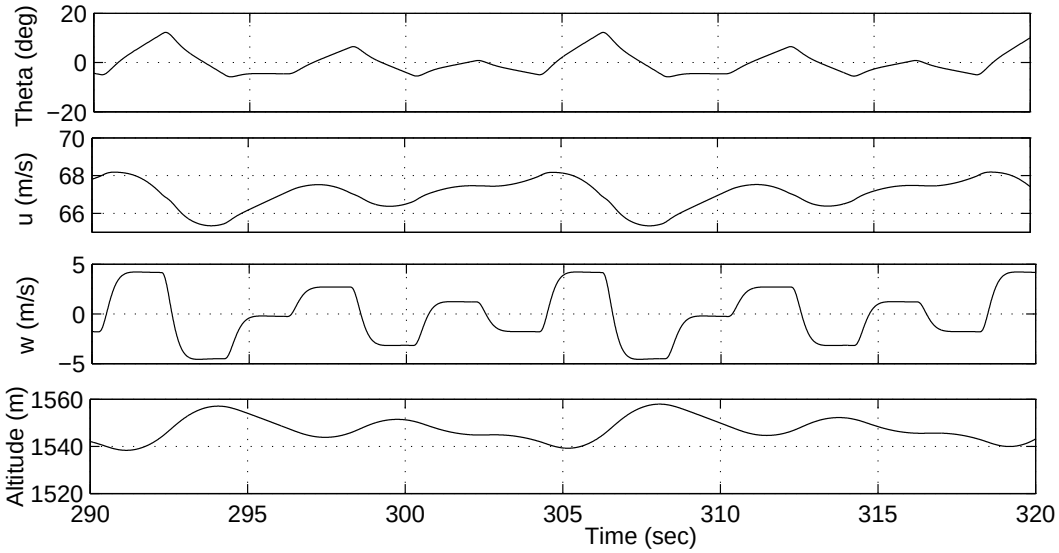

Figure 3.24: Aircraft States during Maneuver-4

3.7 Conclusion

In this chapter, neural network based adaptive dynamic trim and control limit prediction algorithm is applied on a fixed wing aircraft model. It is seen that with the proposed algorithm limit violations and approaching aircraft limits are detected in effective lead times. Results obtained in this chapter are also harmonious with the limit predictions of previous works.

It is seen that when the classical weight update law is used for limit predictions, a re-learning process of aircraft dynamics is highly probable. In the next chapter a different learning algorithm is used and its benefits over the classical weight update law (instantaneous learning) are presented with various simulations in the scope of dynamic trim and control limit predictions.

CHAPTER 4

CONCURRENT LEARNING IMPLEMENTATION

4.1 Introduction

In this chapter, a recently known neural network learning algorithm, which is commonly known as *concurrent learning*, is studied in the scope of dynamic trim predictions and estimation of control limits. The key point of concurrent learning is that a stack of data collected during flight is used to update the neural network weights online. Data is collected from transient and steady state responses seperately. The collection of data representing transient and steady state conditions is used along with instantaneous measured data for weight adaptations. It is seen that colllecting steady state conditions and using them for weight updates results in faster and accurate dynamic trim predictions compared to using instantaneous data only.

In view of the fact that past information is used in weight adaptations, much better modeling error compensations are expected from the proposed algorithm. The first expectation is that the modeling error of an encountered maneuver should decrease gradually when that maneuver is repeated again. This also indicates the long term learning capabilities of such learning algorithms. Secondly, a faster generation of accurate online models are expected for newly introduced modeling errors which are mainly due to the varying aircraft configurations and flight conditions as well. For instance, a shift in the center of gravity location or a decrement in aircraft weight or flying at different places of flight envelop including the limits are just reasons for occurences of new modeling errors. In the case of new modeling errors algorithm is still expected to give correct dynamic trim predictions. Moreover, since the control limits are found by evaluating the network at the known envelope limits, the network is expected to represent the limit dynamics and the current flight regimes simultaneously. Therefore, as

a third expectation, much more global adaptations are needed for accurate control limit solutions. The simulations performed in this chapter mainly focused on the expectations stated here.

The weight update law applied in this chapter is also used for the design of a long term flight controller presented in [33, 34, 35]. In those studies data points ”sufficiently different” from each other are recorded in a stack. Since sufficiently different data points are mainly found in transient response of a fast state, in this chapter the same criteria are used for recording data points into the transient stack. Besides, the main purpose is to predict maneuvering steady states. According to this, we improved the conditions to record data to fit the main purpose. In addition to record data that is sufficiently different, we put importance on the states that represent a maneuvering steady state. As a result a different storing criteria is used for recording steady state data, and the related stack is called the steady state stack.

It is a commonly known fact that the instantaneous learning law has the rank-1 limitation in the weight update [33]. In other words the rank of NN weight dynamics is always at most one when only instantaneous data is used for training. Readers may refer to [35] for details of rank-1 limitation. The oscillatory behavior of weight updates are mainly due to the rank-1 limitation. Further, by using the recorded information for weight updates, alleviation of that limitation can be maintained [35]. Hence, a convergence of the network weights to steady values is more probable when concurrent learning is used.

Analyses are done at high and low learning rate configurations seperately. For the first case, a considerably low learning rate is chosen such that the contribution of instantaneous learning to the modeling eror compensation is made almost zero. It is seen that even if the contribution of instantaneous learning was almost zero, the use of history stacks located the weights in the proper locations and accurate dynamic trim predictions are maintained in a short time. The comparisons of predictions with transient and steady state data are presented as well. For the second type of analysis high learning rates are chosen. This time there exist a significant contribution of instantaneous learning for modeling error compensation. Simulations are done in such a way that after a time later new modeling errors are introduced by changing the B constants of linear models. And, it is shown that by use of the recorded data nonlinearity of new modeling errors are better learned and resulted with accurate dynamic trim predictions compared to using instantaneous measured data only. In both analysis the use of steady state

data points incresed the speed of adaptation and the accuracy of predictions at the approaching limit boundaries. For the third type of analysis, aircraft is exposed to limit violations intentionally at the first few maneuvers of a long term simulation. Here, high learning rates are used as well. It is demonstrated that adaptation using concurrent learning results with accurate control limit predictions at long term since limit violations are recorded into long term memories. Comparisons of instantaneous learning and concurrent learning in the scope of control limit predictions are presented.

4.2 Theoretical Development

Concurrent Learning Through Recorded Data

Let's first remember the model tracking error dynamics of the previous chapter:

$$\dot{e} = -Ke + \xi - W^T\beta(V^T\bar{x}) \tag{4.1}$$

$$r = \xi - W^T\beta(V^T\bar{x}) \tag{4.2}$$

In equation 4.1, the difference between current modeling error (ξ) and the adaptive neural network output $W^T\beta(V^T\bar{x})$ is called the residual signal r and is used for online learning (instantaneous learning). Residual signal in the form of equation 4.2 doesn't contain any past information. The residual signal can be written in a more general form:

$$r_{c_i} = \xi_i - W^T\beta(V^T\bar{x}_i) \tag{4.3}$$

Using the residual signal of equation 4.3, the concurrent learning law, which will be defined later, attempts to reduce the difference between the stored estimate of model error (ξ_i) and the neural network output $W^T\beta(V^T\bar{x}_i)$ which is also based on stored state information.

Selection of Data Points for Concurrent Learning

A key part in the design is to establish the correct data recording criteria. For control problems [33, 34, 35], the criteria is to record data that are sufficiently different:

$$\frac{(\vec{x} - \vec{x}_p)^T(\vec{x} - \vec{x}_p)}{\vec{x}^T\vec{x}} > \epsilon_{\vec{x}} \tag{4.4}$$

where p denotes the last point stored in the history stack.

States and controls which are sufficiently different from the previously recorded ones, are mainly found in transient response. We call the history stack of this criteria as the transient stack as it will mostly effect the modeling error compensation during transient response of our predictions.

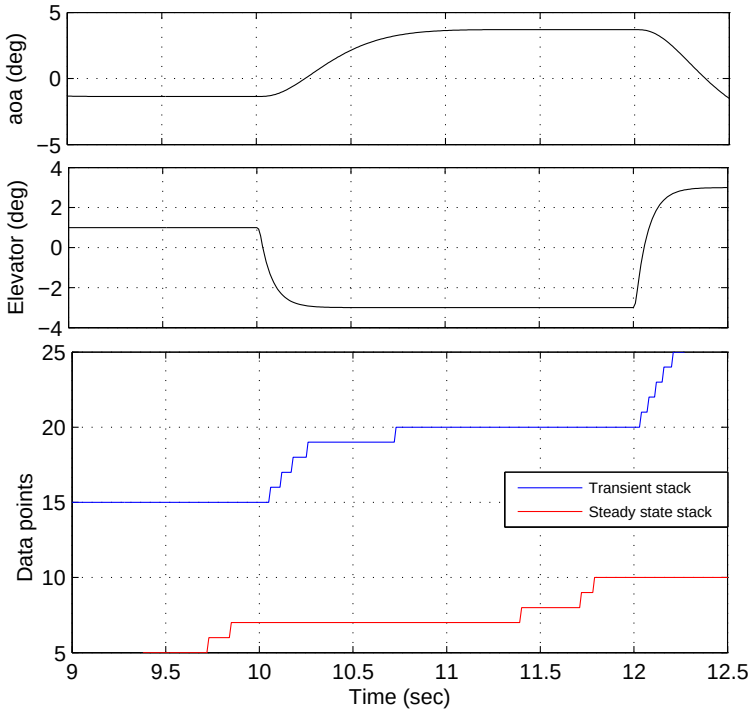

Figure 4.1: Comparison of data storing criterias for α response

However, on top of the classic data collection algorithm we impose the following criteria:

$$\epsilon_{y_1} < \sqrt{[x_l(k) - x_l(k-1)]^2 + [x_l(k) - x_l(k-2)]^2 + ... + [x_l(k) - x_L(k-N)]^2} < \epsilon_{y_2} \quad (4.5)$$

and

$$\epsilon_{z_1} < \sqrt{[\delta_l(k) - \delta_l(k-1)]^2 + [\delta_l(k) - \delta_l(k-2)]^2 + ... + [\delta_l(k) - \delta_l(k-N)]^2} < \epsilon_{z_2} \quad (4.6)$$

Here x is a one dimensional limiting state in consideration, $x(k)$ is the current state and $x(k-N)$

is the state information of N time steps before. Similarly, δ represents the corresponding controls. Here, the algorithm checks for a steady state condition (steady states and controls) and if so, it adds it into the recorded stack to be used in future concurrent learning. We call this criteria the statedy state stack. An example of data storing activity can be seen in fig4.1.

Note that these are only conditions for data recording. Once it is decided to be a relevant data to be stored all related data used in the network update such as state vector, control vector and modeling error are all stored. If the stack becomes large the oldest data in the stacks is replaced with the new ones.

Concurrent Learning Weight Update Law

For the simulations presented in this chapter the folllowing NN weight adaptation law is used [35]:

$$\dot{W} = -(\beta - \beta^{'} V^T \bar{x}) r^T L_{R_W} - W_c \sum_{i=1}^{p} (\beta_i - \beta_i^{'} V^T \bar{x}_i) r_{c_i} L_{R_W} \tag{4.7}$$

$$\dot{V} = -L_{R_V} \bar{x} r^T W^T \beta^{'} (V^T \bar{x}) - V_c \sum_{i=1}^{p} L_{R_V} \bar{x}_i r_{c_i}^T W^T \beta^{'} (V^T \bar{x}_i) \tag{4.8}$$

where W_c and V_c are orthogonal projection operators and are defined as:

$$W_c = (I - \frac{\beta \beta^T}{\beta^T \beta}) \tag{4.9}$$

$$V_c = (I - \frac{L_{R_V} \bar{x} \bar{x}^T L_{R_V}}{\bar{x}^T L_{R_V} L_{R_V} \bar{x}}) \tag{4.10}$$

By using orthogonal projections, it made possible to constrain background learning to the nullspace of online adaptation [35].

Since there are two history stacks used in this study, equations (4.7) and (4.8) can be written in the following form:

$$\dot{W} = -(\beta - \beta^{'} V^T \bar{x}) r^T L_{R_W} - W_c \sum_{i=1}^{p1} (\beta_{t_i} - \beta_{t_i}^{'} V^T \bar{x}_{t_i}) r_{t_{c_i}} L_{R_{W_{tr}}} - W_c \sum_{i=1}^{p2} (\beta_{ss_i} - \beta_{ss_i}^{'} V^T \bar{x}_{ss_i}) r_{ss_{c_i}} L_{R_{W_s}} \tag{4.11}$$

$$\dot{V} = -L_{R_V} \bar{x} r^T W^T \beta^{'} (V^T \bar{x}) - V_c \sum_{i=1}^{p1} L_{R_{V_{tr}}} \bar{x}_{t_i} r_{t_{c_i}}^T W^T \beta^{'} (V^T \bar{x}_{t_i}) - V_c \sum_{i=1}^{p2} L_{R_{V_s}} \bar{x}_{ss_i} r_{ss_{c_i}}^T W^T \beta^{'} (V^T \bar{x}_{ss_i}) \tag{4.12}$$

Here subscript t and ss denote the variables which are calculated with the recorded information of transient and steady-state stacks. Also, $p1$ and $p2$ are the sizes or the maximum number of data points of the related stacks.

Note that learning rates of transient stack and steady state stack are denoted by $L_{R_{W_{tr}}}$,$L_{R_{V_{tr}}}$,$L_{R_{W_s}}$ and $L_{R_{V_s}}$. In theory [35], learning rates are equivalent such that: $L_{R_W} = L_{R_{W_{tr}}} = L_{R_{W_s}}$ and $L_{R_V} = L_{R_{V_{tr}}} = L_{R_{V_s}}$. For the simulations done in this chapter, learning rates of the history stacks are chosen lower values compared to instantaneous learning rates L_{R_W}, L_{R_V}. In this way, weights may converge to steady values smoothly. So that by keeping the learning rates of the previous chapter as the instantaneous learning rates of that chapter, we show the benefits of concurrent learning over the instantaneous adaptations of chapter-3.

Detailed observer loop for dynamic trim predictions can be seen in fig 4.2.

Table 4.1: Design Param. of Low Learning Rate Simulations
Angle of Attack Predictions

$A_{11} = -5$	$A_{14} = 0.1$	$B_1 = -3$	$K_1 = 12$	$L_{R_W} = 10$
$L_{R_V} = 10$	$L_{R_{W_{tr}}} = 5$	$L_{R_{V_{tr}}} = 10$	$L_{R_{W_s}} = 5$	$L_{R_{V_s}} = 10$
$\epsilon_{y_1} = 0.0008$	$\epsilon_{y_2} = 0.01$	$\epsilon_{z_1} = 0.0008$	$\epsilon_{z_2} = 0.01$	$\epsilon_{\vec{x}} = 0.002$

Table 4.2: Design Param. of Low Learning Rate Simulations
Load Factor Predictions

$A_{22} = -5$	$A_{24} = 0.5$	$B_2 = -3$	$K_2 = 15$	$L_{R_W} = 20$
$L_{R_V} = 200$	$L_{R_{W_{tr}}} = 1.3$	$L_{R_{V_{tr}}} = 1.4$	$L_{R_{W_s}} = 10$	$L_{R_{V_s}} = 10$
$\epsilon_{y_1} = 0.0005$	$\epsilon_{y_2} = 0.0008$	$\epsilon_{z_1} = 0.0005$	$\epsilon_{z_2} = 0.0008$	$\epsilon_{\vec{x}} = 0.003$

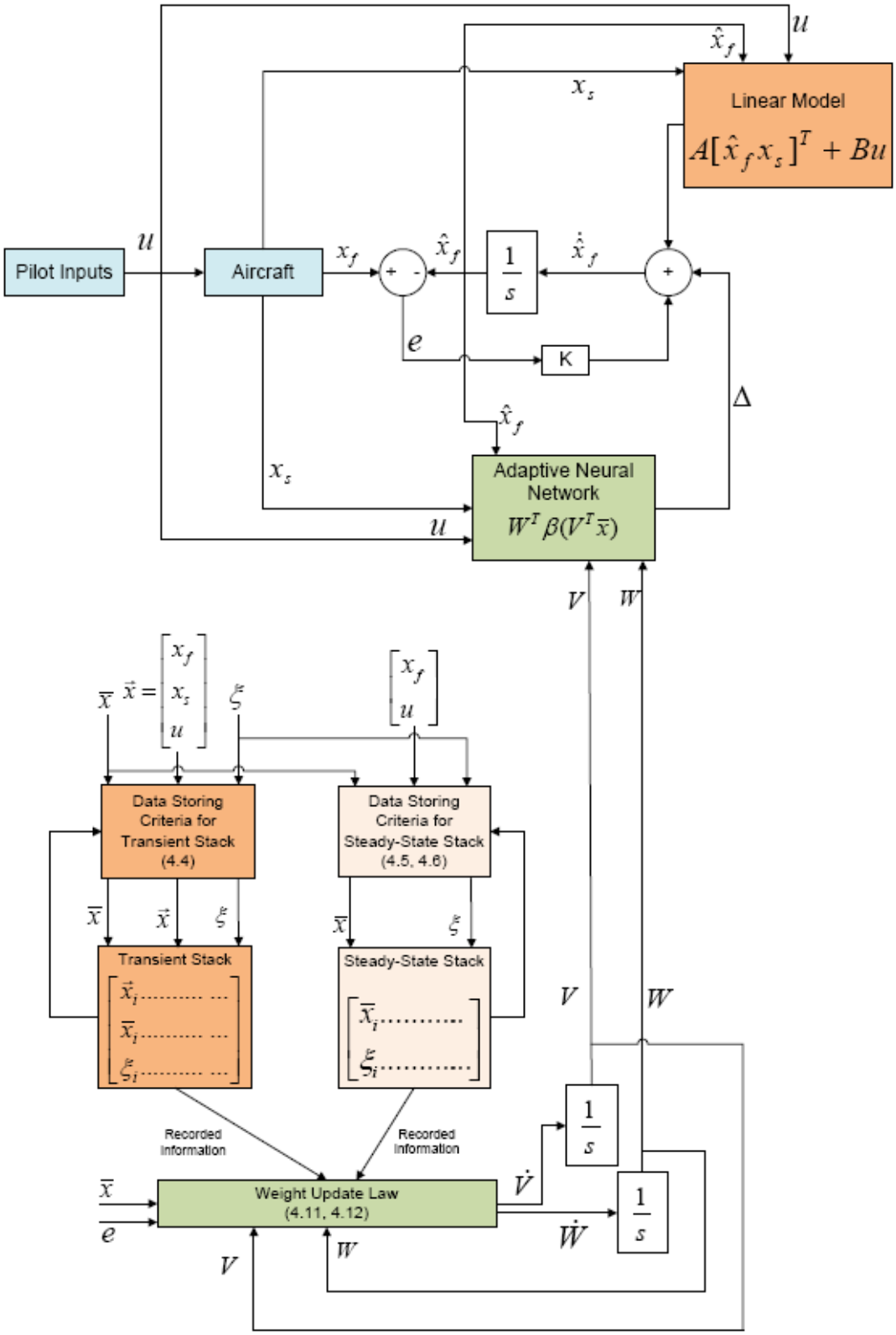

Figure 4.2: Concurrent Learning Enabled Observer Loop for Dynamic Trim Predictions

4.3 Simulation Results

4.3.1 Dynamic Trim Predictions

In this section, concurrent learning implemented dynamic trim predictions are presented. Aircraft is exposed to pull-up and push-over maneuvers as it is done in the maneuver-1 of the previous chapter. Simulations started from 121.5 KIAS level flight trim point. The algorithm applied in this chapter is the same algorithm with the one applied in previous chapter except the weight update law.

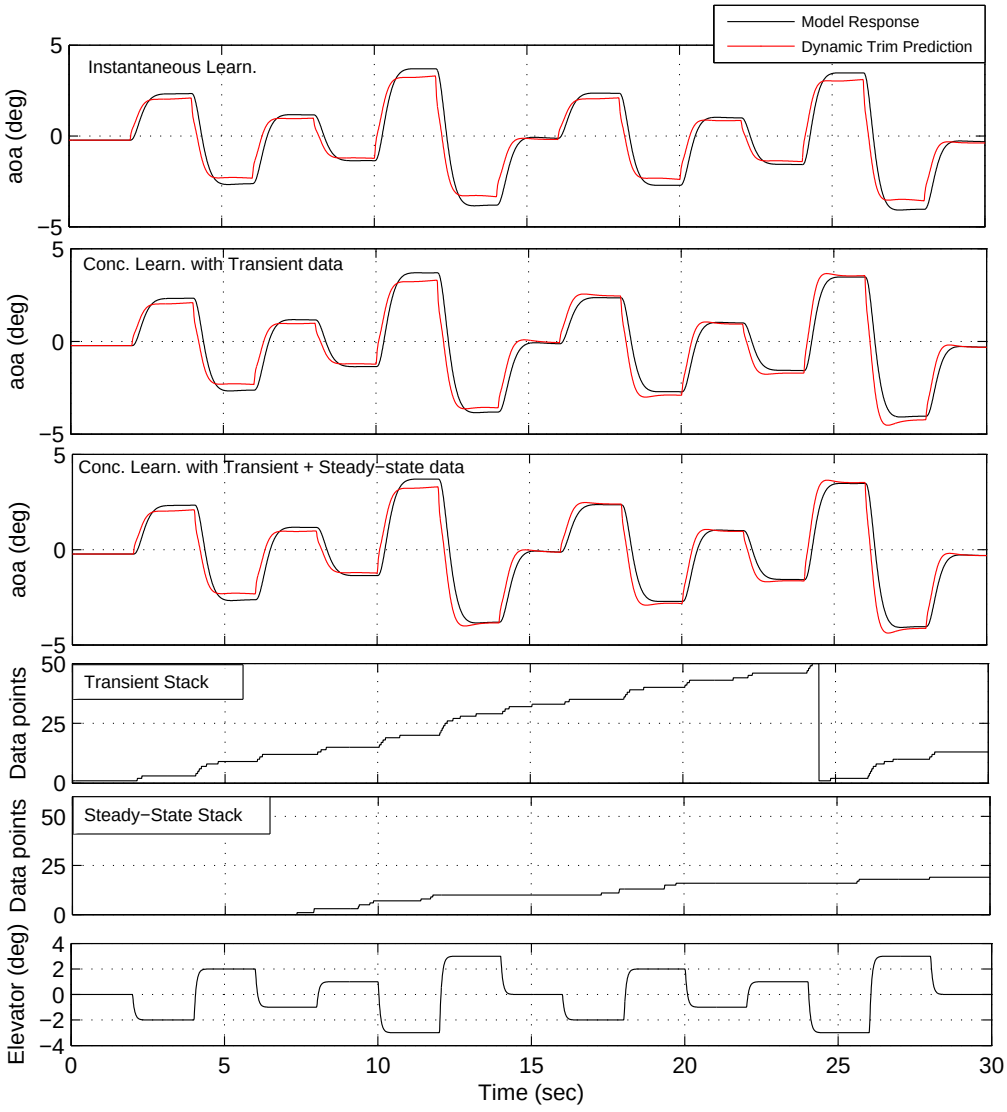

Figure 4.3: Short Term Comparison of Dynamic Trim Predictions at a Low L_{R_W} and L_{R_V} for $\hat{\alpha}$ dynamics

In order to obtain the benefits of concurrent learning, analyses are performed at high and low learning rate configurations.

For the first analysis, instantaneous learning rates L_{R_W} and L_{R_V} are chosen intentionally low values such that the contribution of instantaneous learning to modeling error cancellation is almost made zero. With this approach it is possible to see the effect of recorded data on weight adaptations. In fig4.3 dynamic trim predictions for angle of attack dynamics is presented. It is obvious that concurrent learning using transient data have a much more adaptation capability compared to learning with instantaneous data only, even if the learning rates are very low. Moreover, using steady state data on top of the transient stack increases the speed of adaptations and more accurate dynamic trim predictions are obtained at the limit boundaries in a short time. Design parameters can also be seen in tables 4.1 and 4.2

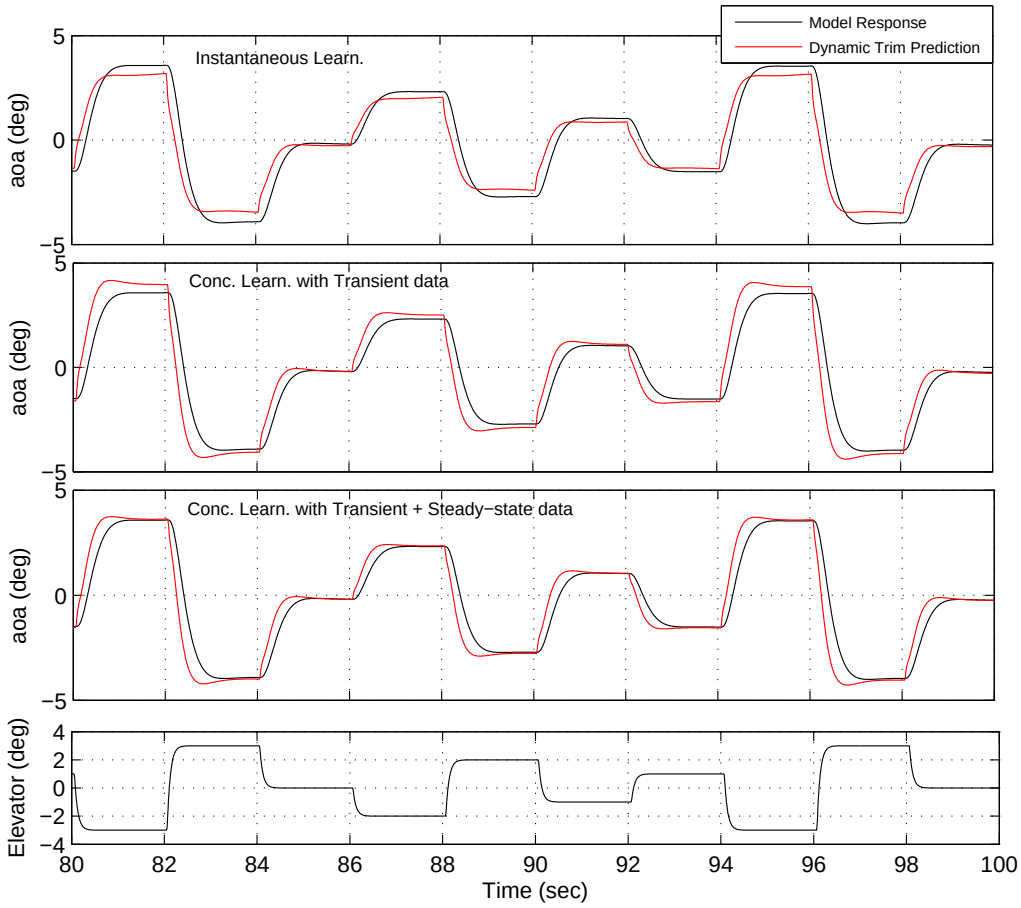

Figure 4.4: Long Term Comparison of Dynamic Trim Predictions at a Low L_{R_W} and L_{R_V} for $\hat{\alpha}$ dynamics

Data recording activity of both stacks can also be seen in fig4.3. Here the size of both stacks are chosen to be 50 data points. Whenever the stacks are full, the oldest data is replaced with the new ones.

Since concurrent learning is expected to have a better long term learning capability, same simulation is contiuned up to 100 seconds. Dynamic trim predictions of last 20 seconds are presented in fig4.4. As it is expected, concurrent learning using both transient and steady state data results in accurate dynamic trim predictions in long term. Since recorded transient data are mainly responsible for compensating modeling errors during transient response, training with transient data only cannot locate the weights in the proper locations and results with inaccurate dynamic trim predictions at approaching limits. This fact can also be seen in fig4.5. In that figure, it is shown that transient only adaptation has no contribution to modeling error cancellation during a maneuvering steady state. Hence, each stack has a different job for modeling error cancellations.

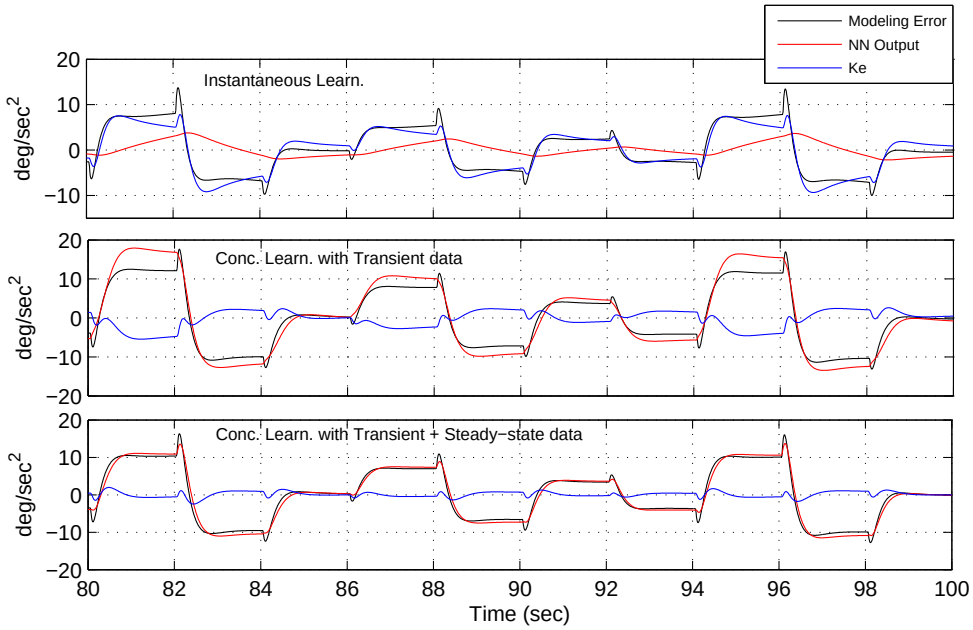

Figure 4.5: Long Term Comparison of Modelling Error Compensations at a Low L_{R_W} and L_{R_V} for $\hat{\alpha}$ dynamics

Comparison of weight updates of angle of attack predictions are presented in fig4.6. Note that the oscillations found in the weight dynamics of chapter 3 do not exist in the weight updates of fig4.6. Here weights are being located such that the modelling errors of multiple data points are taken into account instead of instantaneous modeling error only.

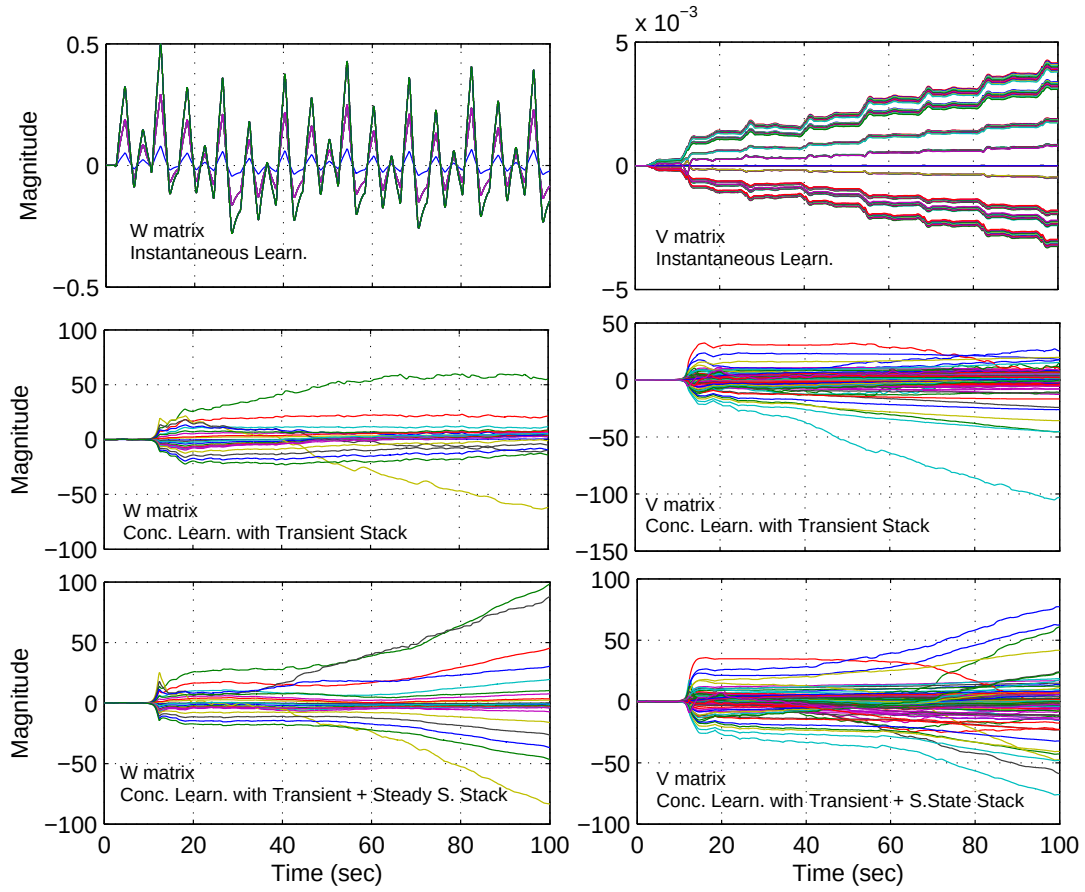

Figure 4.6: Top to bottom: Weight update comparisons of instantaneous learning and concurrent learning at short term for $\hat{\alpha}$ dynamics

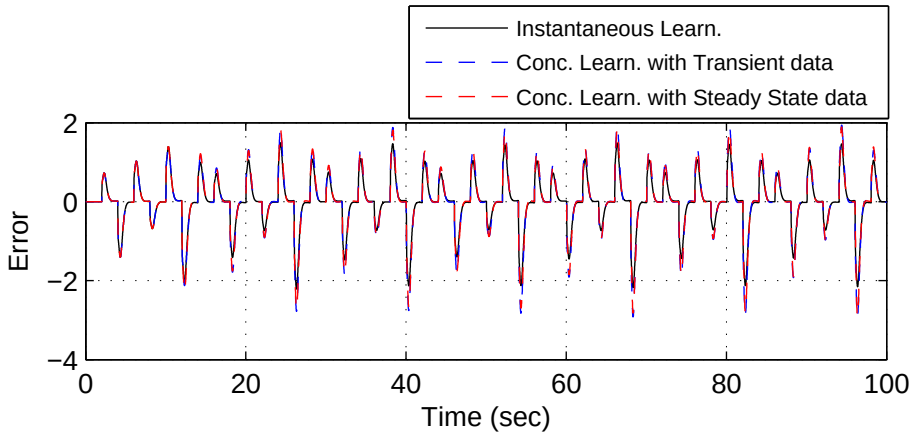

Figure 4.7: Dynamic Trim Convergences of α Predictions (i=1)

The results obtained in this chapter may be improved by selecting much better data storing criteria.

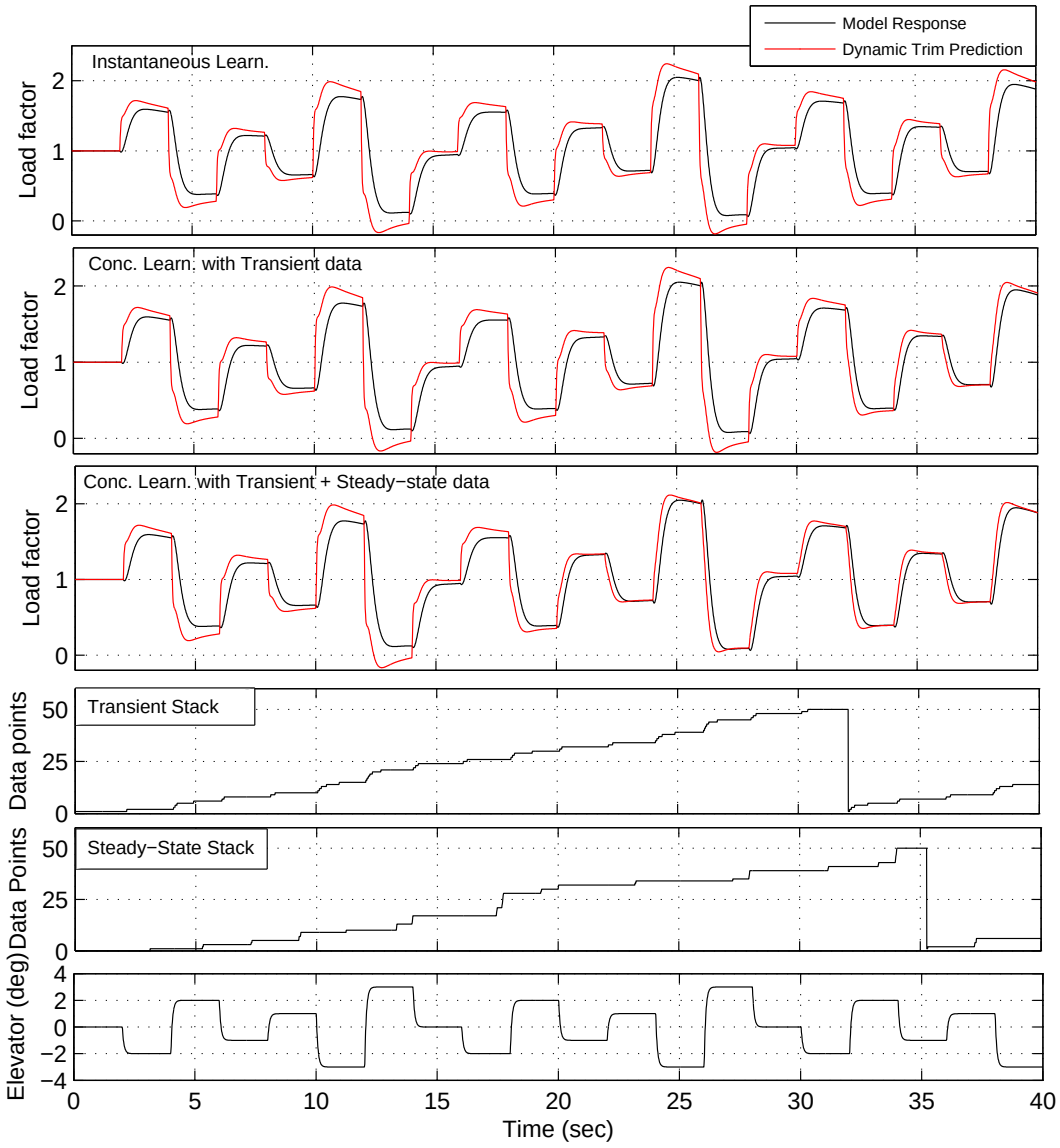

Figure 4.8: Short Term Comparison of Dynamic Trim Predictions at a Low L_{R_W} and L_{R_V} for $\hat{n}_z$ dynamics

Dynamic trim predictions of load factor dynamics are presented as well. In fig4.8 short term prediction responses can be seen. For load factor predictions, the importance of steady state stack is increased by selecting a relatively high learning gain while the instantaneous learning rate and the learning rate of transient stack are kept low. It can be seen in fig4.8 that the adaptation using steady state stack predicts the dynamic trims 10 seconds before, compared

to learning using transient data only. This fact is also obvious from the weight updates given in fig4.9. Besides, neural network weights have converged to steady values this time. Long term responses and modeling error compensations are given in figures 4.10 and 4.12.

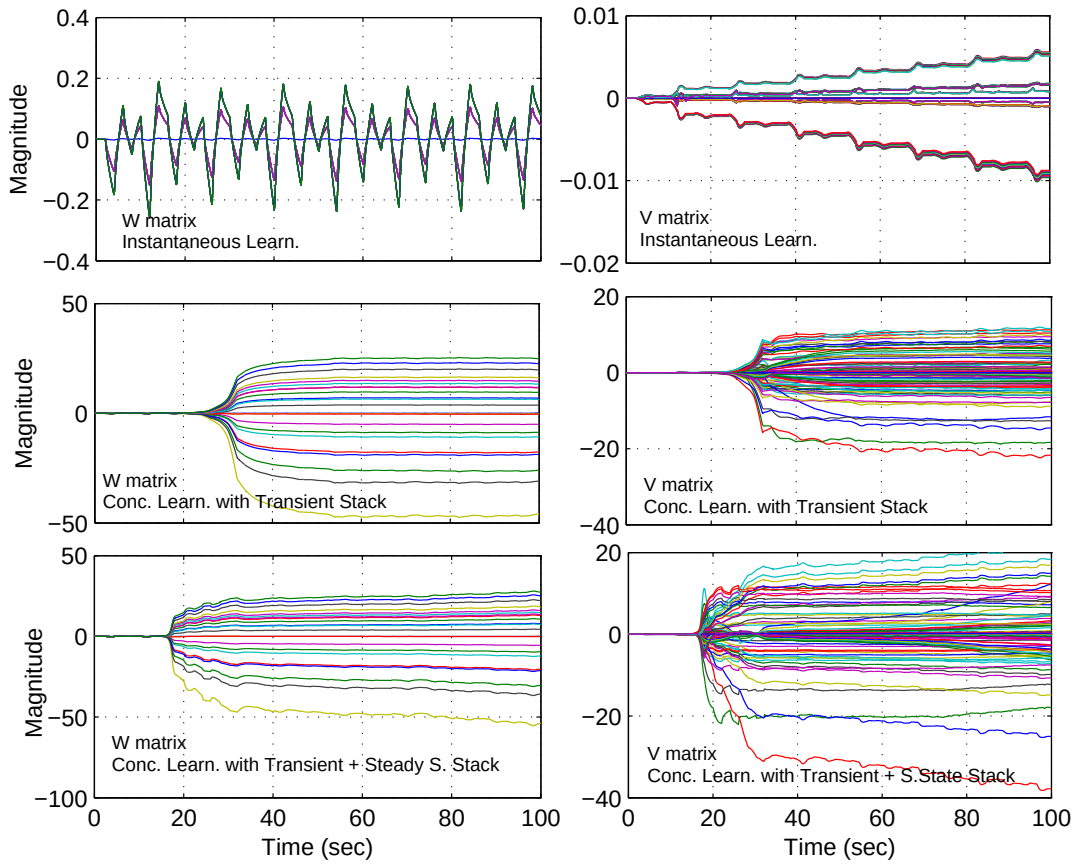

Figure 4.9: Top to bottom: Weight update comparisons of instantaneous learning and concurrent learning at short term for n_z dynamics

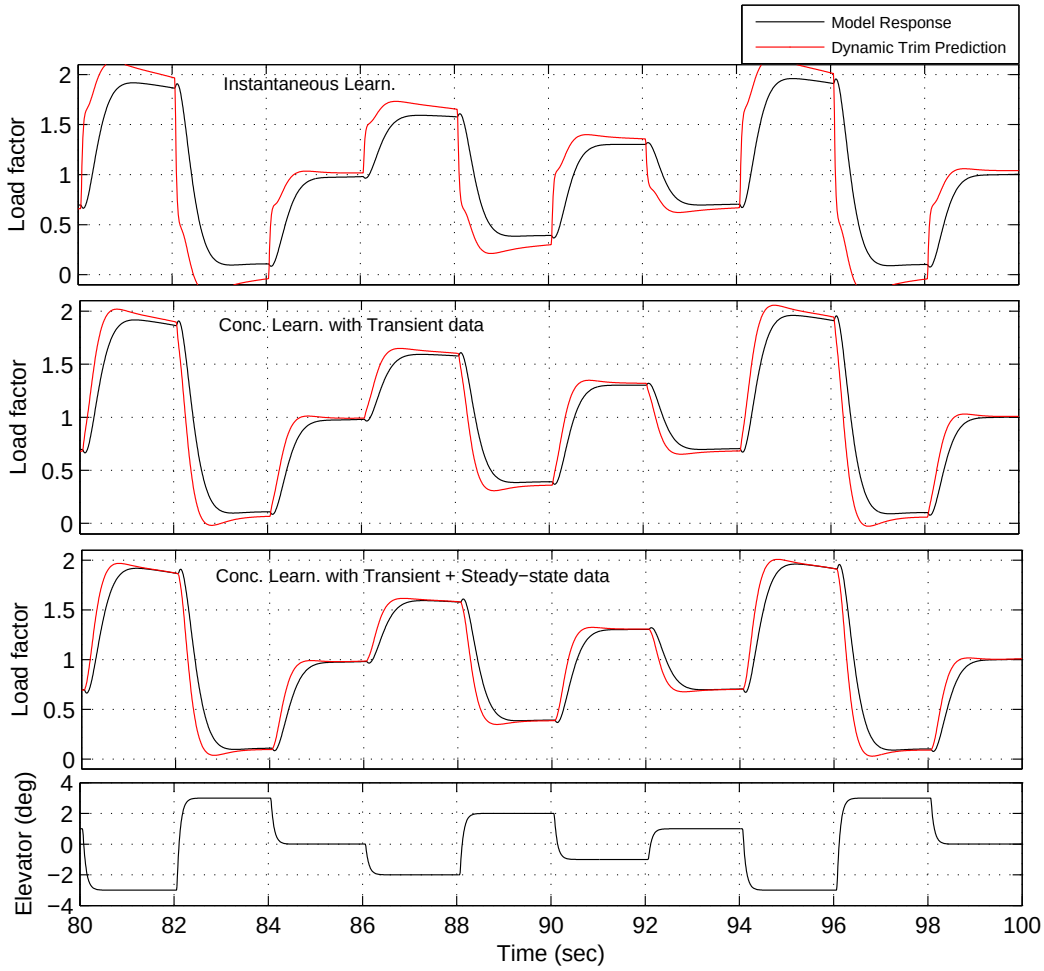

Figure 4.10: Long Term Comparison of Dynamic Trim Predictions at a Low L_{R_W} and L_{R_V} for $\hat{n}_z$ dynamics

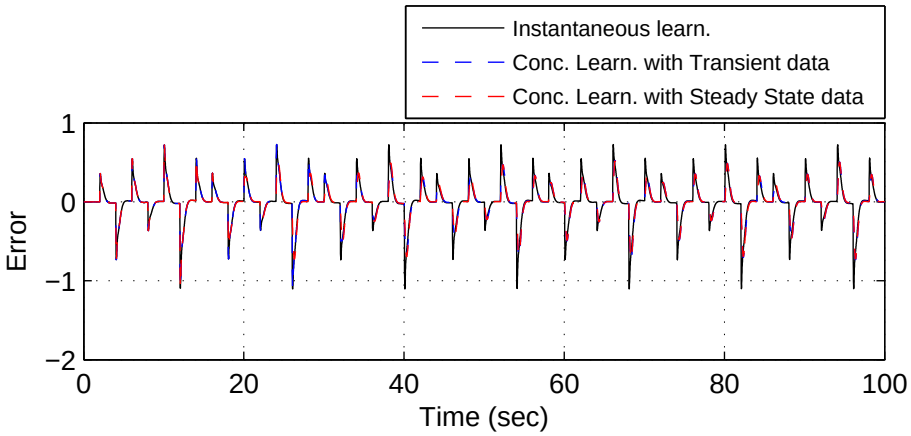

Figure 4.11: Dynamic Trim Convergences of n_z Predictions (i=1)

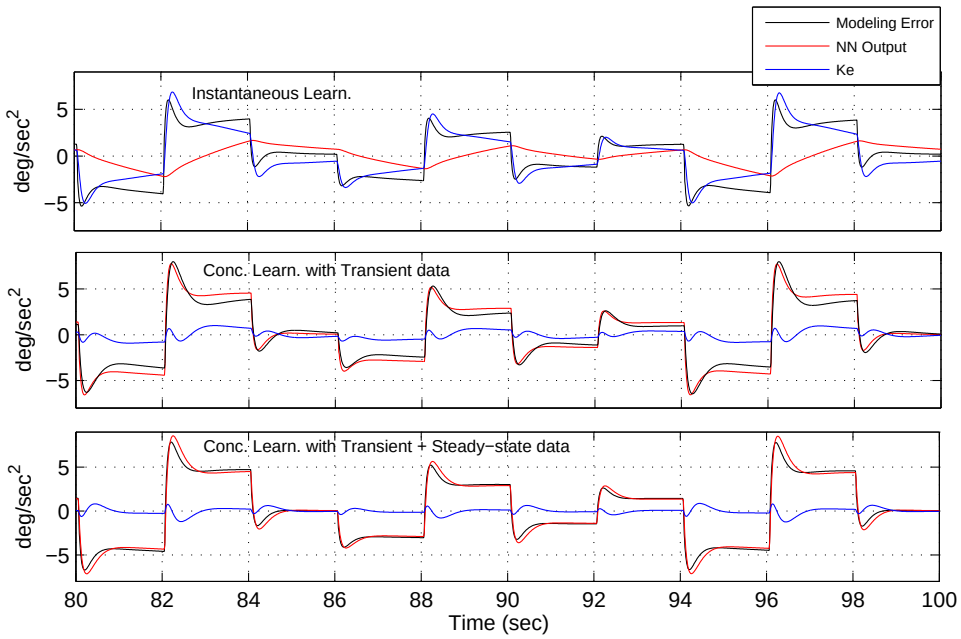

Figure 4.12: Long Term Comparison of Modelling Error Compensations at a L_{R_W} and L_{R_V} Rate for $\hat{n}_z$ dynamics

For the second type of analysis, instantaneous learning rates L_{R_W} and L_{R_V} are chosen high values such that the contribution of instantaneous learning to modeling error cancellations is almost made significant. It is sure that instantaneous learning at a high learning rate will compansate for modeling errors for both transient and steady state responses. Therefore, the benefit of using recorded data may not always be visible in dynamic trim prediction point of view. In order to see the benefit of concurrent learning in the existence of a high instantaneous learning rate, new modeling errors are introduced during simulations and the adaptations to those errors are compared. Design parameters can also be seen in tables 4.3, 4.4 and 4.5.

The scenario which is done for low learning rate case is repeated for high learning rate case this time. Dynamic trim predictions for angle of attack dynamics can be seen in fig4.13. After 50 seconds of simulation new modeling errors are introduced intentionally by changing the B_1 constant of approximate linear model.

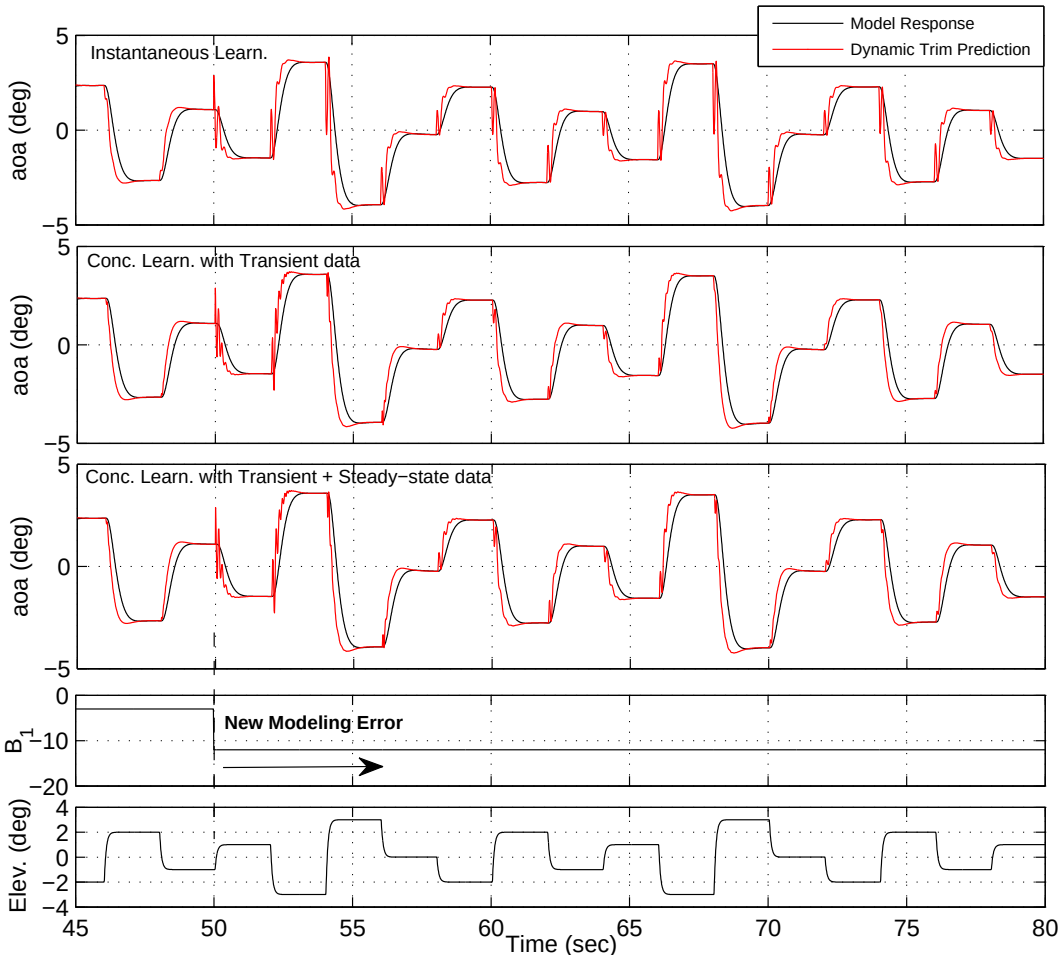

Figure 4.13: Long Term Comparison of Dynamic Trim Predictions at a High L_{R_W} and L_{R_V} for $\hat{\alpha}$ Dynamics (New Modeling Error is Introduced)

It is clear that after introducing the new B_1, weight updates using instantaneous data only give inaccurate and oscillatory dynamic trim predictions at the time controls are applied (fig4.13). From the EPS (Envelop Protection System) point of view, it is desired to estimate the allowable control travel and cue the pilot at the time the controls are applied, before the limiting state is reached. Hence, in the case of adaptations using instantaneous data only, after the new modeling errors are introduced, EPS may receive inaccurate and oscillatory dynamic trim and limit control predictions. Whereas the adaptations using recorded data have a capability of learning the new modelig errors in a short time. Therefore, the inaccurate and oscillatiory predictions at the transient response no longer exist (fig4.13).

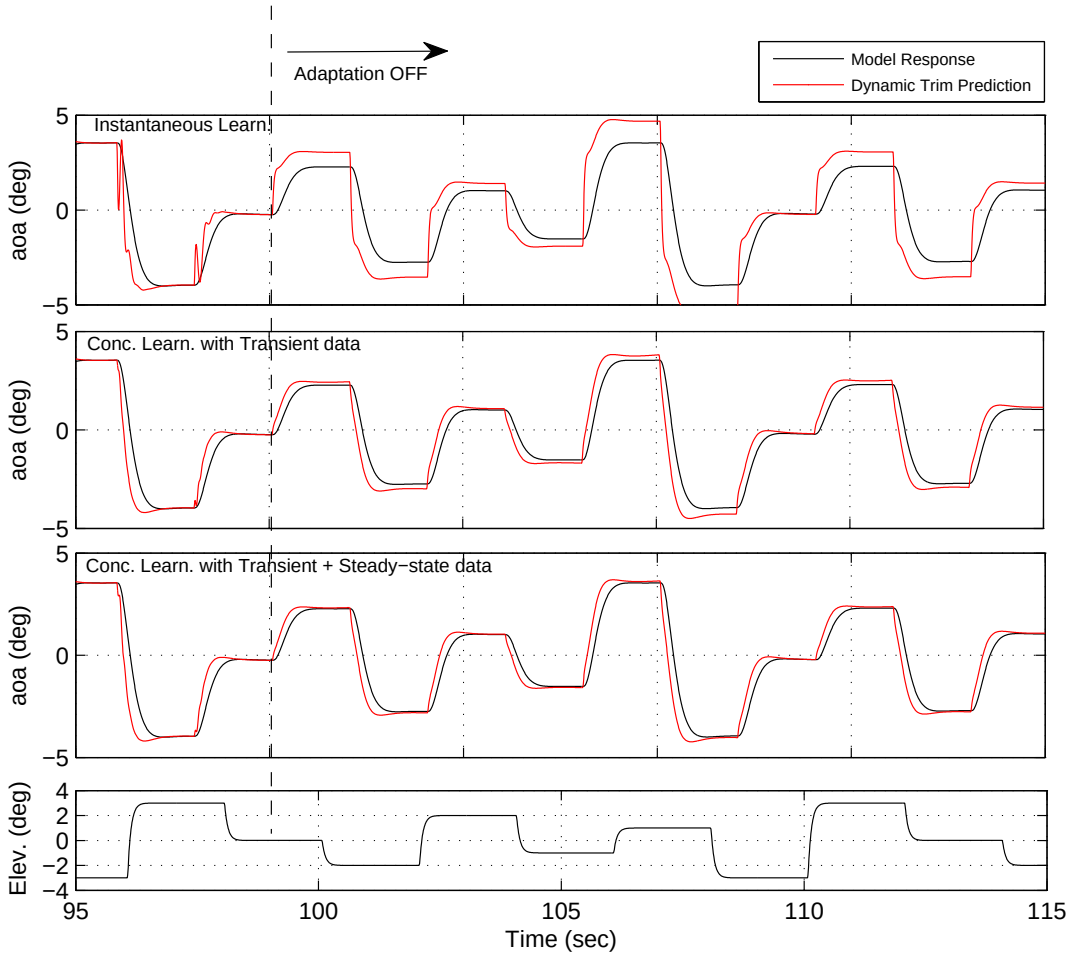

Figure 4.14: Long Term Comparison of Dynamic Trim Predictions at a High L_{R_W} and L_{R_V} for $\hat{\alpha}$ Dynamics (Weights are Frozen)

After the new modeling errors are introduced, adaptations using steady state stack may locate the weights such that the generated online models may represent global behaviors. To demonstrate this, simulations are continued up to 100 seconds and after that time weights are

frozen intentionally. Simulation results are shown in fig4.14. It is clear that the online models which are generated using steady state stack are much more representative of the $\dot{\alpha}$ dynamics. Although the weights are frozen, dynamic trim prediction is still accurate.

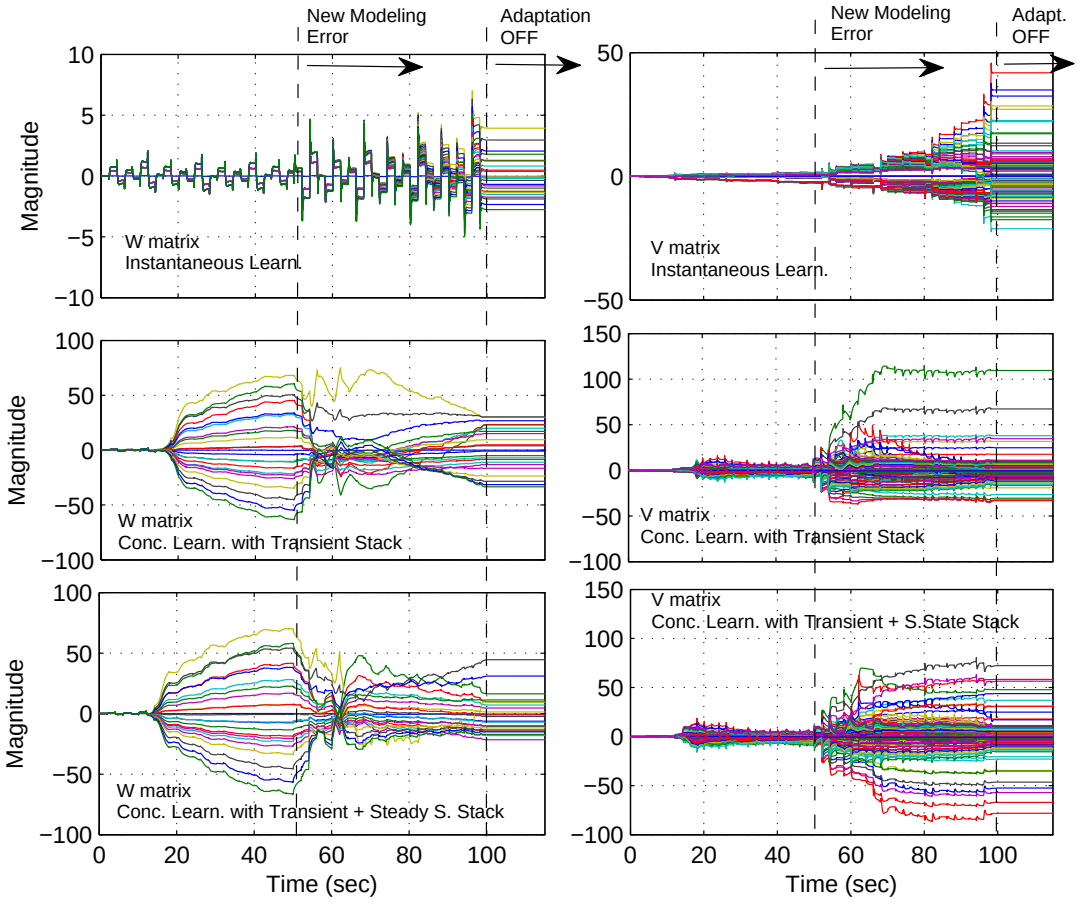

Figure 4.15: Overall Weight update Comparisons of Instantaneous Learning and Concurrent Learning at High a L_{R_W} and L_{R_V} for $\hat{\alpha}$ Dynamics

In addition, the whole weight update time history is given in fig4.15. The convergence of neural network weights to steady values after the application of new modeling errors can also be seen. Note that weight updates using instantaneous measured data only give an oscillatory response.

Load factor dynamic trim predictions of the same scenario is presented in fig4.17. This time modeling error is increased in such a way that the controls are reversed by altering the sign and the magnitude of B_2 constant. This way, much more inaccurate dynamic trim predictions are obtained at the time the controls are applied. Also by use of the recorded data, those

erroneous predictions are avoided in a short time. Predictions after weights are frozen and the overall weight updates can be seen in figs4.19 and 4.21.

Results of pitch rate predictions for the same scenario are presented in figs4.18 and 4.20. Weight updates are given in fig4.23 too. Results similar to angle of attack and load factor predictions are obtained. Comparisons of dynamic trim convergences are presented in figures 4.16, 4.22 and 4.24.

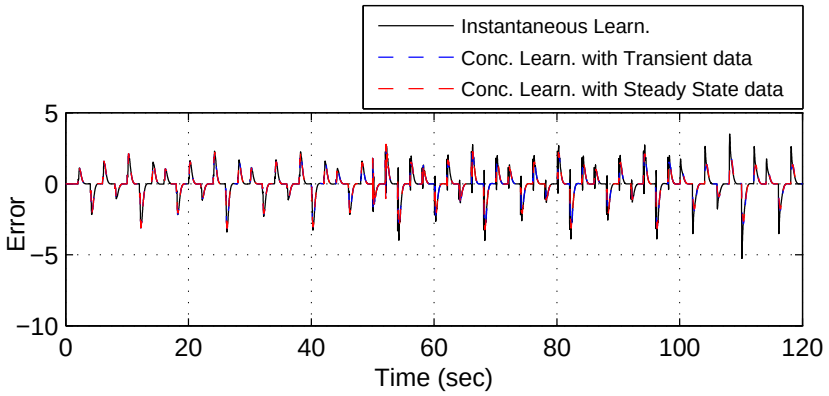

Figure 4.16: Dynamic Trim Convergences of α Predictions (i=1)

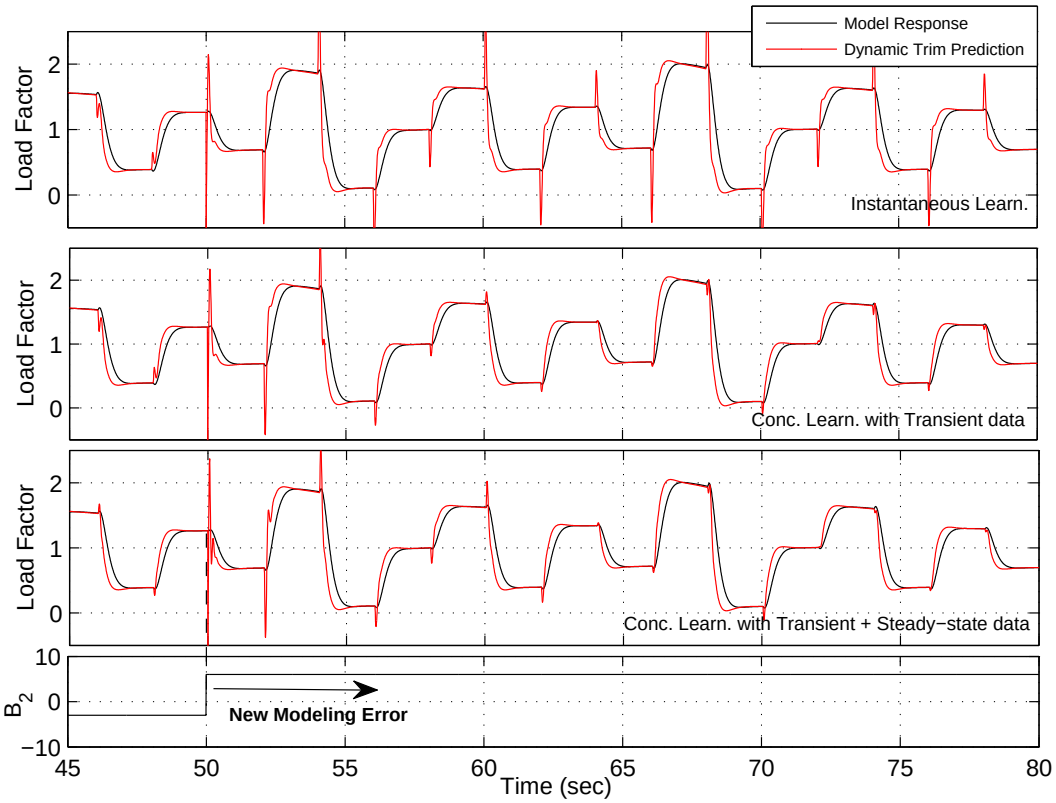

Figure 4.17: Long Term Comparison of Dynamic Trim Predictions at a High L_{R_W} and L_{R_V} for $\hat{n}_z$ Dynamics (New Modeling Error is Introduced)

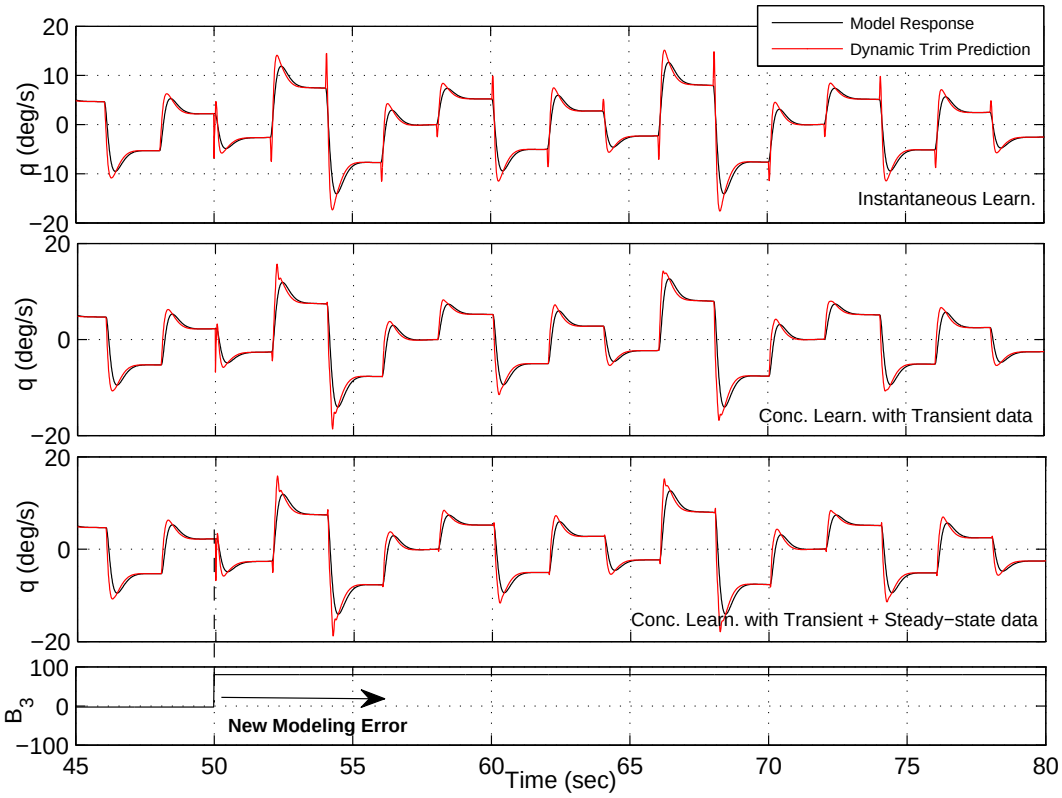

Figure 4.18: Long Term Comparison of Dynamic Trim Predictions at a High L_{R_W} and L_{R_V} for $\hat{q}$ Dynamics (New Modeling Error is Introduced)

Table 4.3: Design Param. of High Learning Rate Simulations
Angle of Attack Predictions

$A_{11} = -5$	$A_{14} = 0.1$	$B_1 = -3$	$K_1 = 12$	$L_{R_W} = 2000$
$L_{R_V} = 3500$	$L_{R_{W_{tr}}} = 1$	$L_{R_{V_{tr}}} = 1$	$L_{R_{W_s}} = 1$	$L_{R_{V_s}} = 1$
$\epsilon_{y_1} = 0.0008$	$\epsilon_{y_2} = 0.01$	$\epsilon_{z_1} = 0.0008$	$\epsilon_{z_2} = 0.01$	$\epsilon_{\vec{x}} = 0.002$

Table 4.4: Design Param. of High Learning Rate Simulations
Load Factor Predictions

$A_{22} = -5$	$A_{24} = 0.5$	$B_2 = -3$	$K_2 = 30$	$L_{R_W} = 2000$
$L_{R_V} = 2000$	$L_{R_{W_{tr}}} = 0.5$	$L_{R_{V_{tr}}} = 0.5$	$L_{R_{W_s}} = 0.5$	$L_{R_{V_s}} = 0.5$
$\epsilon_{y_1} = 0.0005$	$\epsilon_{y_2} = 0.0008$	$\epsilon_{z_1} = 0.0005$	$\epsilon_{z_2} = 0.0008$	$\epsilon_{\vec{x}} = 0.003$

Table 4.5: Design Param. of High Learning Rate Simulations
Pitch Rate Predictions

$A_{33} = -10$	$A_{34} = 0.8$	$B_3 = -11$	$K_3 = 30$	$L_{R_W} = 2000$
$L_{R_V} = 3000$	$L_{R_{W_{tr}}} = 0.5$	$L_{R_{V_{tr}}} = 0.5$	$L_{R_{W_s}} = 0.5$	$L_{R_{V_s}} = 0.5$
$\epsilon_{y_1} = 0.003$	$\epsilon_{y_2} = 0.004$	$\epsilon_{z_1} = 0.003$	$\epsilon_{z_2} = 0.004$	$\epsilon_{\vec{x}} = 0.0018$

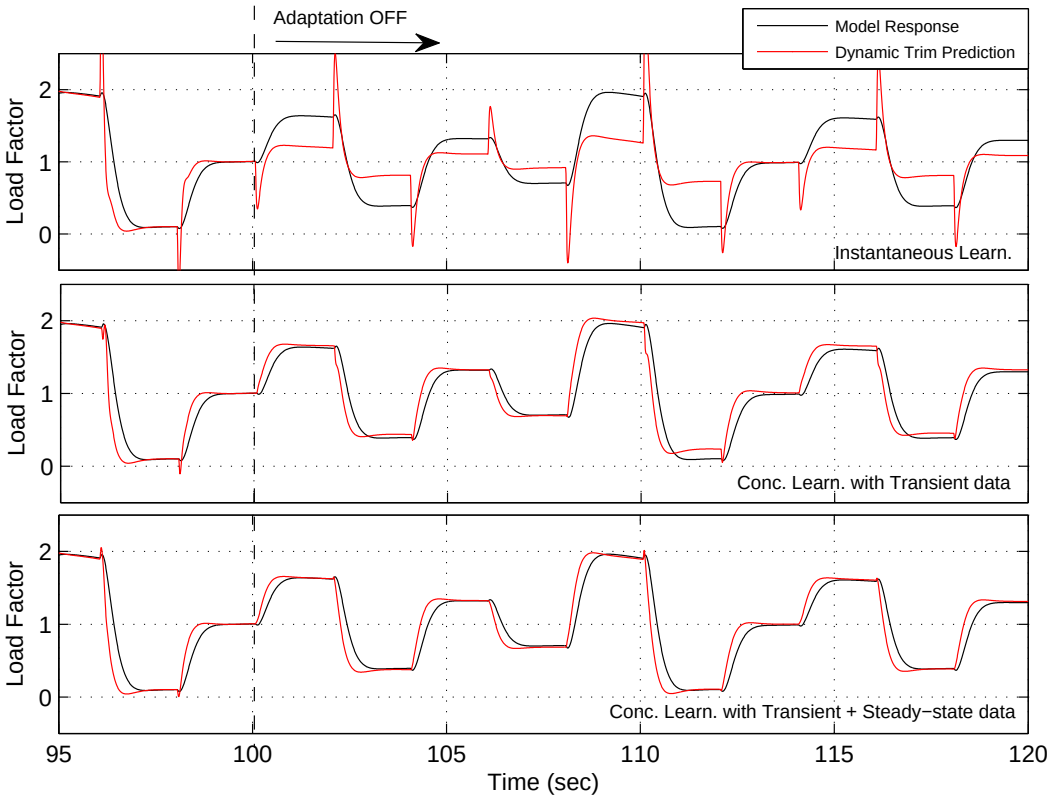

Figure 4.19: Long Term Comparison of Dynamic Trim Predictions at a High L_{R_W} and L_{R_V} for $\hat{n}_z$ Dynamics (Weights are Frozen)

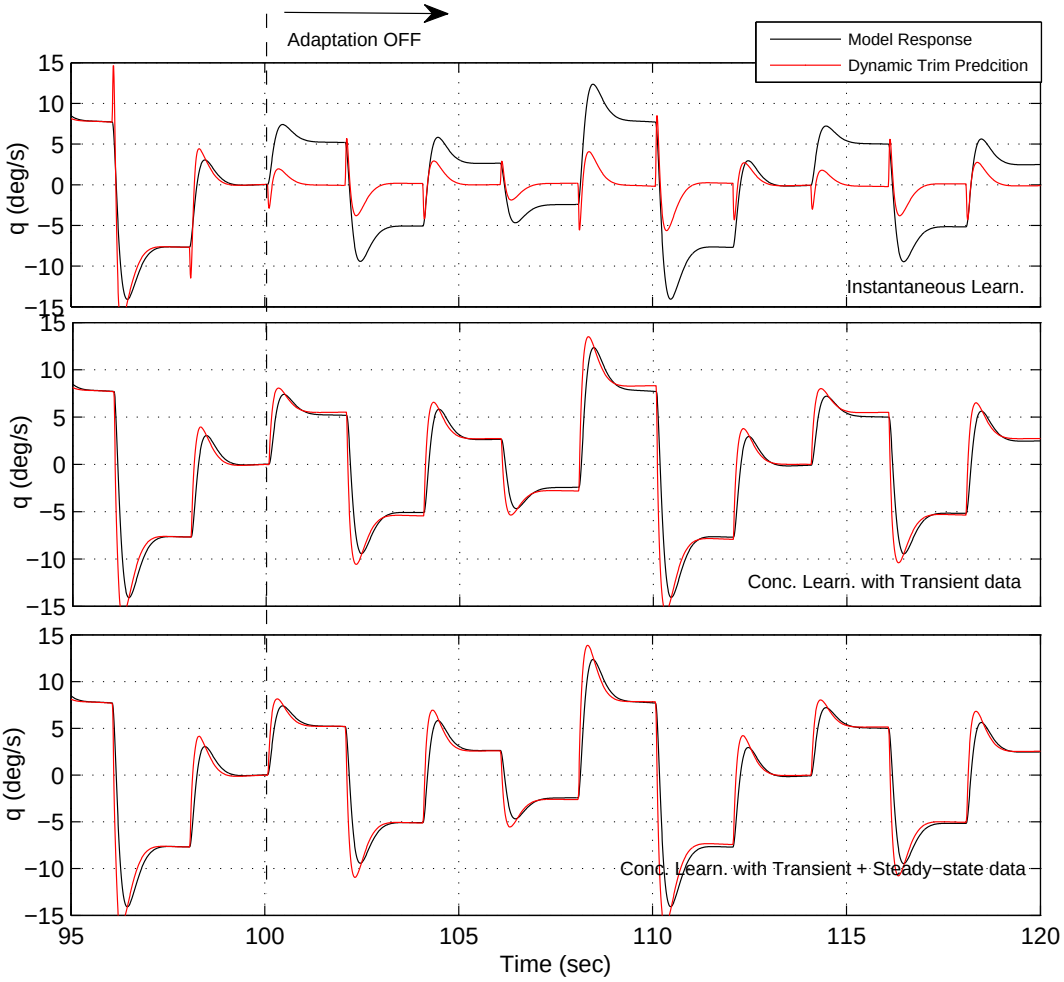

Figure 4.20: Long Term Comparison of Dynamic Trim Predictions at a High L_{R_W} and L_{R_V} for $\hat{q}$ Dynamics (Weights are Frozen)

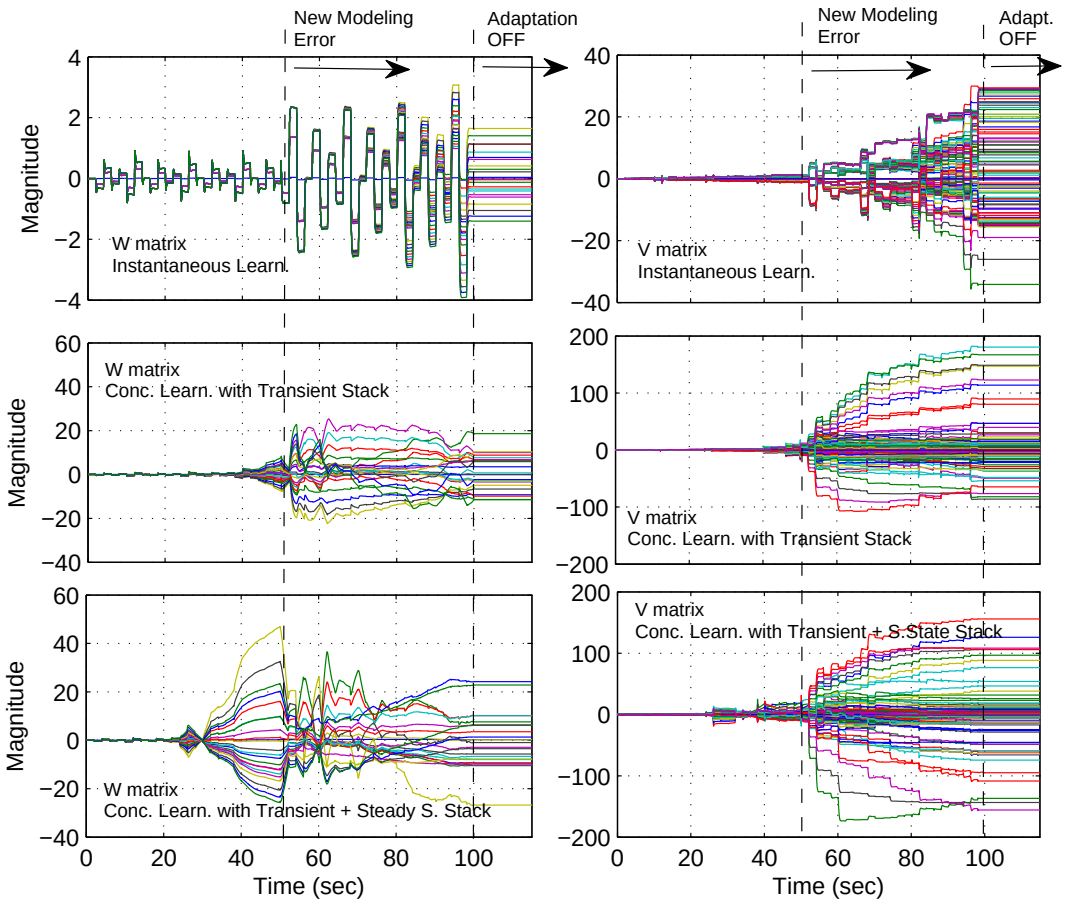

Figure 4.21: Overall Weight update Comparisons of Instantaneous Learning and Concurrent Learning at a High L_{R_W} and L_{R_V} for $\hat{n}_z$ Dynamics

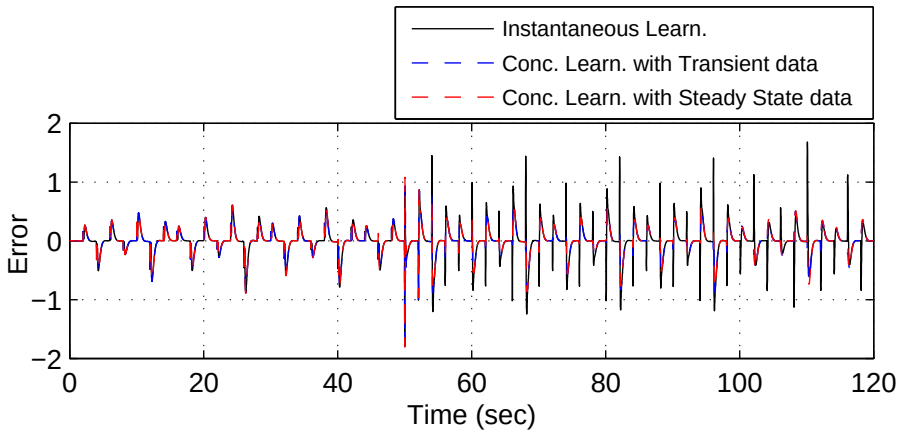

Figure 4.22: Dynamic Trim Convergences of n_z Predictions (i=1)

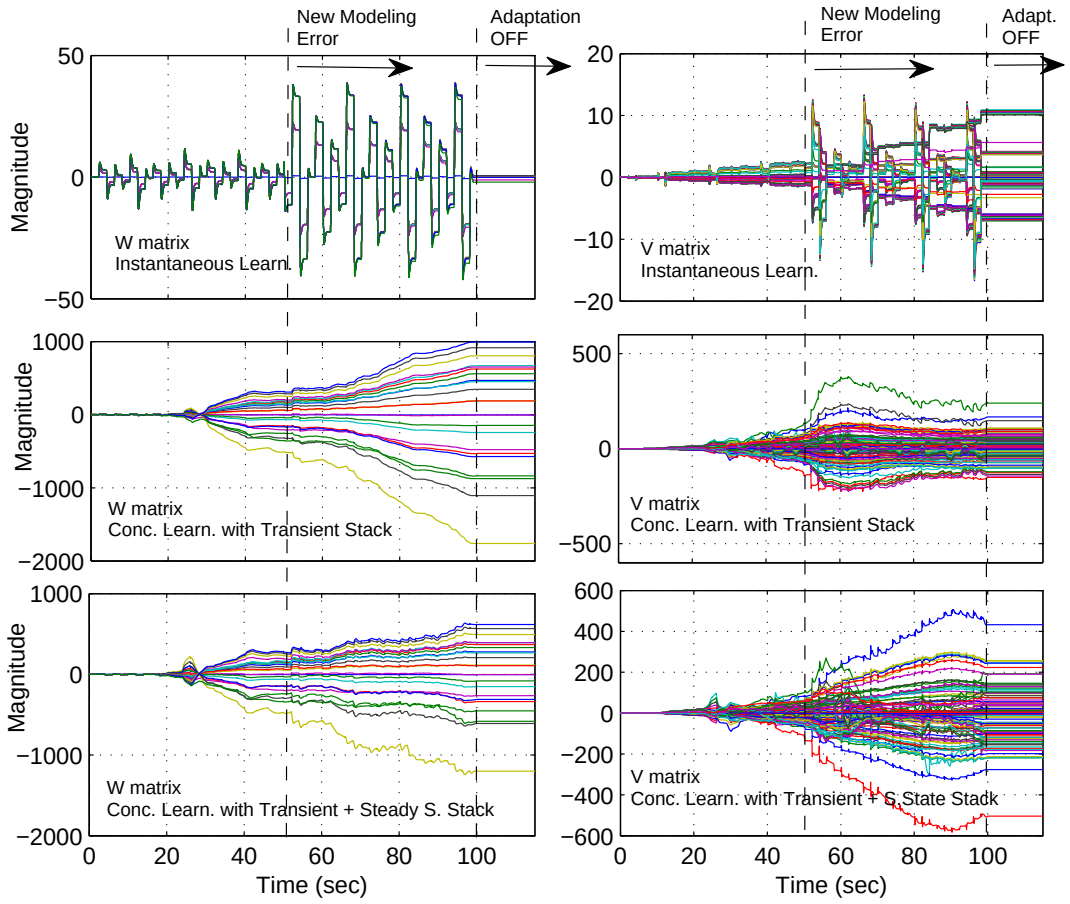

Figure 4.23: Overall Weight update Comparisons of Instantaneous Learning and Concurrent Learning at a High L_{R_W} and L_{R_V} for $\hat{q}$ Dynamics

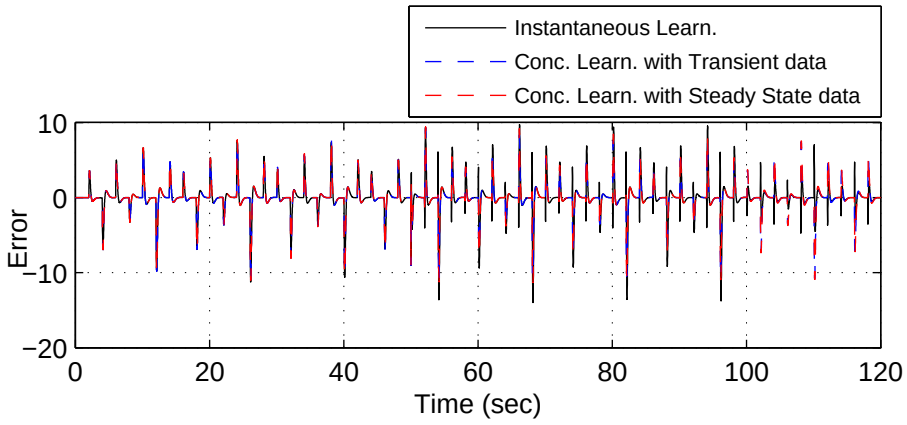

Figure 4.24: Dynamic Trim Convergences of q Predictions (i=1)

4.3.2 Control Limit Predictions

Generally dynamic trim solutions are easily obtained with acceptable error convergences. Whereas, the situation may not be so easy for predicting control limits. Note that in equations 3.41, 3.40 and 3.39 neural network is required to be calculated at the known envelope boundaries. A probable difficulty here is that the network may not be a well representative of the envelope limits if it has not been trained at the limits before. In other words, neural networks may give incorrect outputs when they are evaluated at the flight conditions where they have never been passed before. So, in order to represent the envelope limits as well as the normal operating conditions correctly, online limit violations or offline trained neural networks at the limits may be needed. It is also known that although there may exist limit violations online, adaptations using instantaneous data only will have a tendency to forget about the nature of the violated limits. Therefore, a global neural network adaptation will be appropriate in order to solve for the control limits online.

In chapter-3, satisfactory results are obtained at short term simulations. Results are harmonious with previous works too. It is known that neural network have a tendency to increase its nonlinearity for perfect modeling error compensations. When this happens, iterative solutions may become difficult. This fact may be seen in long term simulations even better, especially when maneuvers are repeated over and over again.

The following results compare control limit predictions using instantaneous learning and concurrent learning for a long simulation time. Although it is not practical, different elevator inputs are repeated over time. At the first 30 seconds of the simulation, aggrassive inputs are given such that the limits are violated. After the 30th second, smaller inputs are given to the system so that the flight around the normal operating conditions are maintained. Then the magnitudes of the inputs are changed again. Design parameters can also be seen in tables 4.6, 4.7 and 4.8.

In fig 4.25 load factor response of the above scenario is presented. Violation of limits can be seen in the first 30 seconds of the simulation. Here, neural network is expected to learn the limit dynamics by recording the transient and steady state data of the first 30 seconds. Data recording activity can be seen in fig 4.26. Control limit predictions using instantaneous learning and concurrent learning up to that time are compared in fig 4.26. It is seen that

whenever the envelope limits are violated (fig 4.25) in the first 30 seconds, control limits are violated too.

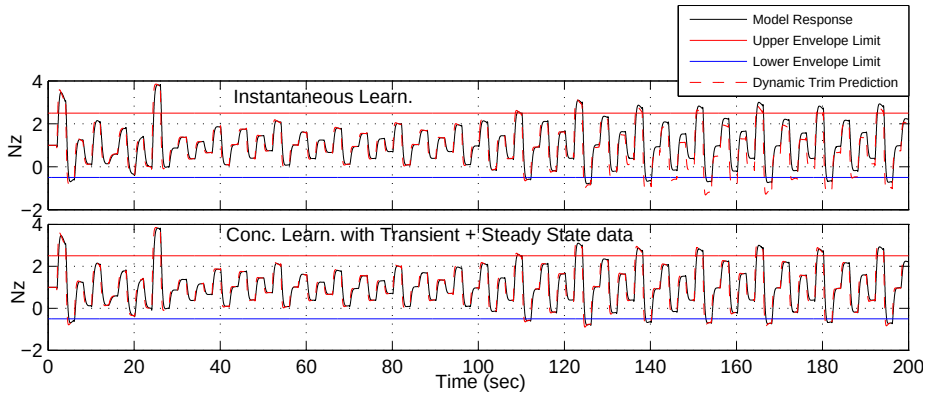

Figure 4.25: $\hat{n}_z$ Response at Long Term

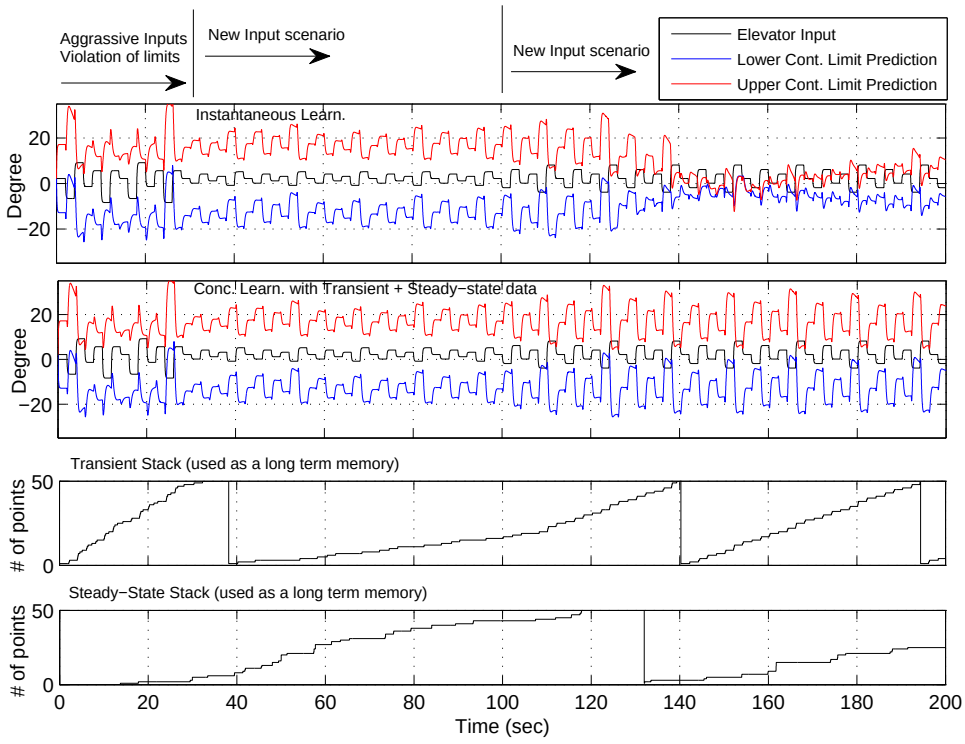

Figure 4.26: Long Term Comparison of Limit Control Predictions at a High L_{R_W} and L_{R_V} for $\hat{n}_z$ Dynamics

Between 30 and 100 seconds, aggrassiveness of the elevator input is decreased such that a flight regime far from limits are maintained (fig 4.25). In that time period, control limit predictions using instantaneous data and recorded data gives similar results (fig 4.26).

In the scenario, after the 100th second, aggrassiveness of the inputs are increased intention-

ally and limits are violated again. It is seen that adaptation using instantaneous data forgets about the limit dynamics it has learned in the first 30 seconds. Due to this fact neural network tries to relearn the limits, and during this relearning process, inaccurate control limit predictions are obtained. Whereas, by using concurrent learning accurate control limit predictions are achieved at long term, especially when the limits are violated again. Weight updates and activation function outputs are presented in the fig 4.27. Note that weight updates of instantaneous learning have a rapid increase at the 100th second and the nonlinearity of the network changes dramatically at that time.

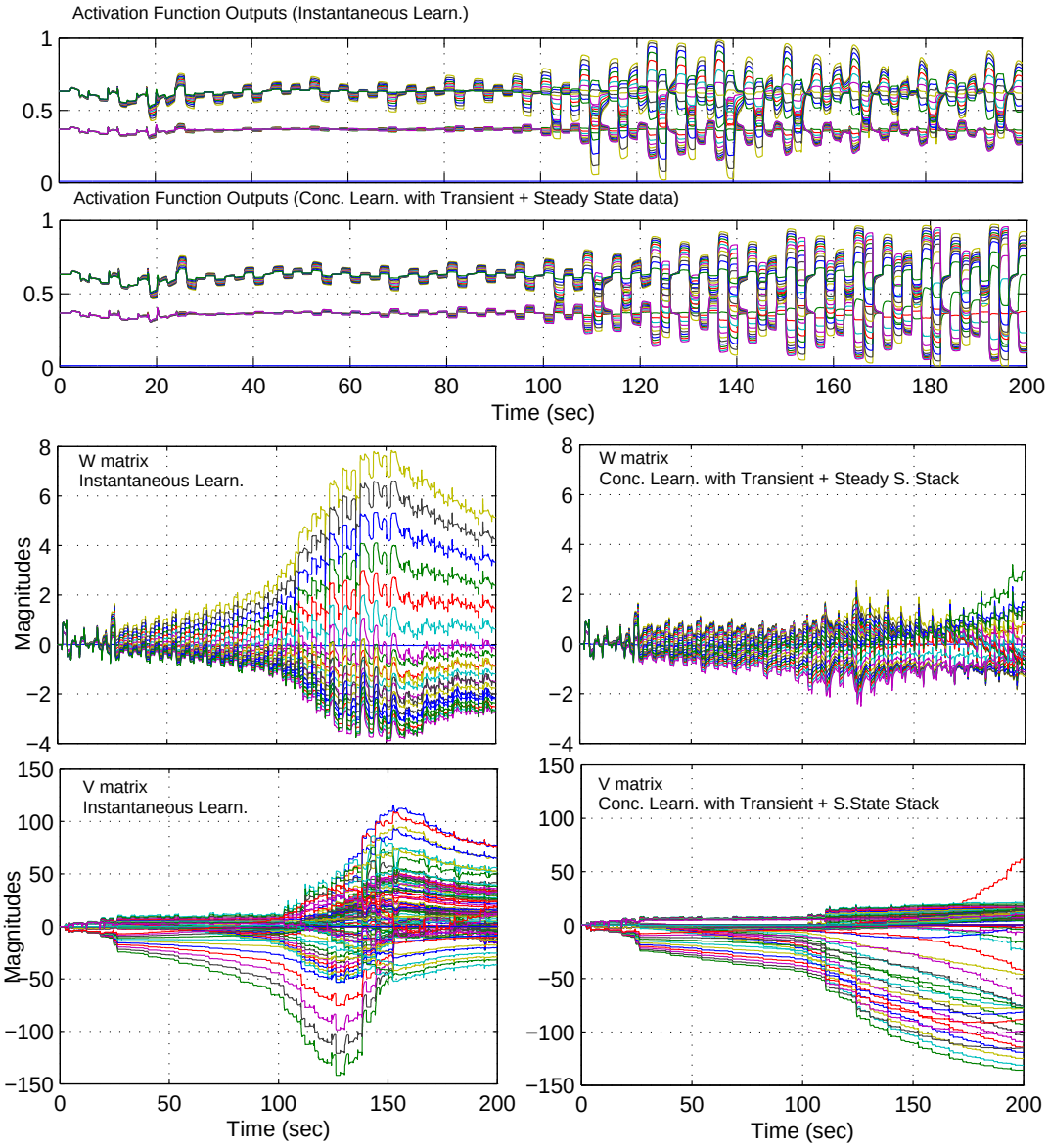

Figure 4.27: Long Term Comparison Activation Function Outputs and Weight Updates at a High L_{R_W} and L_{R_V} for $\hat{n}_z$ Dynamics

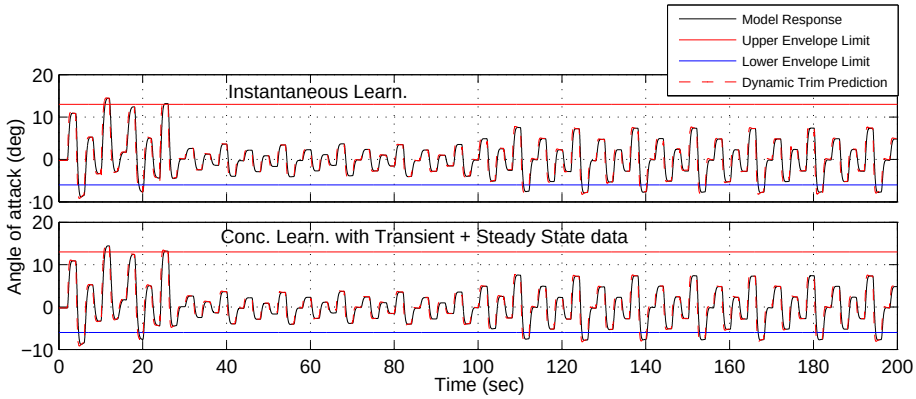

Figure 4.28: $\hat{\alpha}$ Response at Long Term

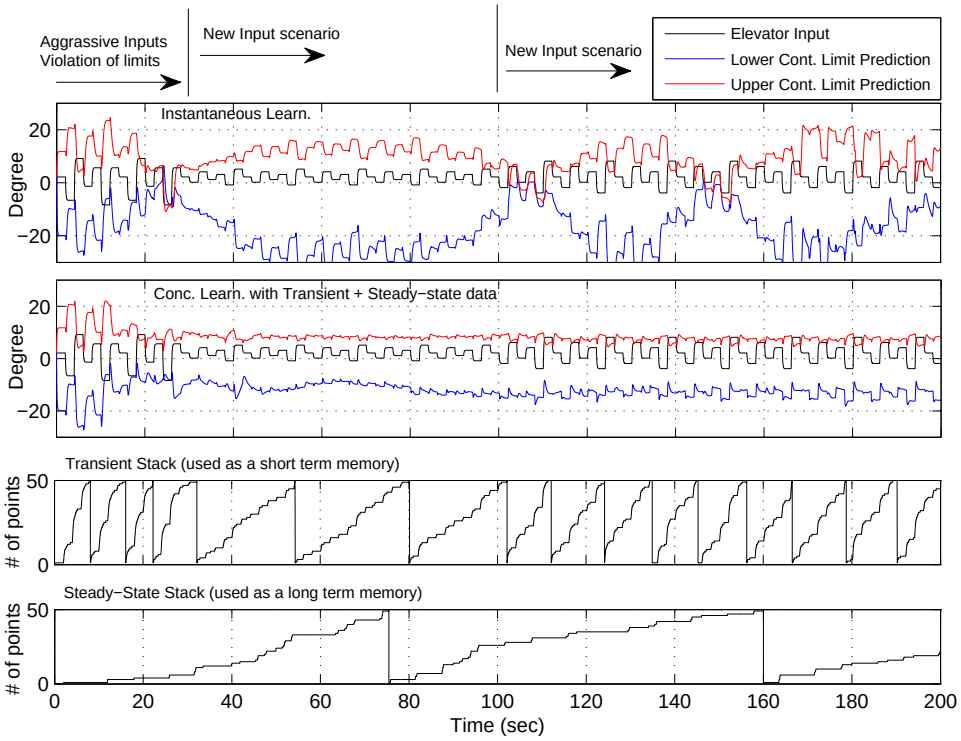

Figure 4.29: Long Term Comparison of Limit Control Predictions at a High L_{R_W} and L_{R_V} for $\hat{\alpha}$ Dynamics

Angle of attack response and control limit predictions of the same simulation scenario are presented in fig 4.28. This time aircraft violates only the negative angle of attack limits after the 100th second. It can be seen in fig 4.29 that although the aircraft is far from the positive angle of attack limits, negative control limits are violated due to the inaccurate control limit predictions of instantaneous learning. Besides, reliable and non-oscillating control limit predictions are calculated by use of concurrent learning at long term. Whenever the negative

envelope limits are violated positive control limits are violated as well. Also, observe that transient stack is used as a short term memory since recorded data are renewed in a shorter time compared to steady state stack. This is a user dependet selection and the results obtained may get better by selecting different storing criteria. Weight updates and activation function outputs are also presented in fig 4.30. It is seen that V matrix of concurrent learning is much more steady compared to the V matrix of instantaneous learning.

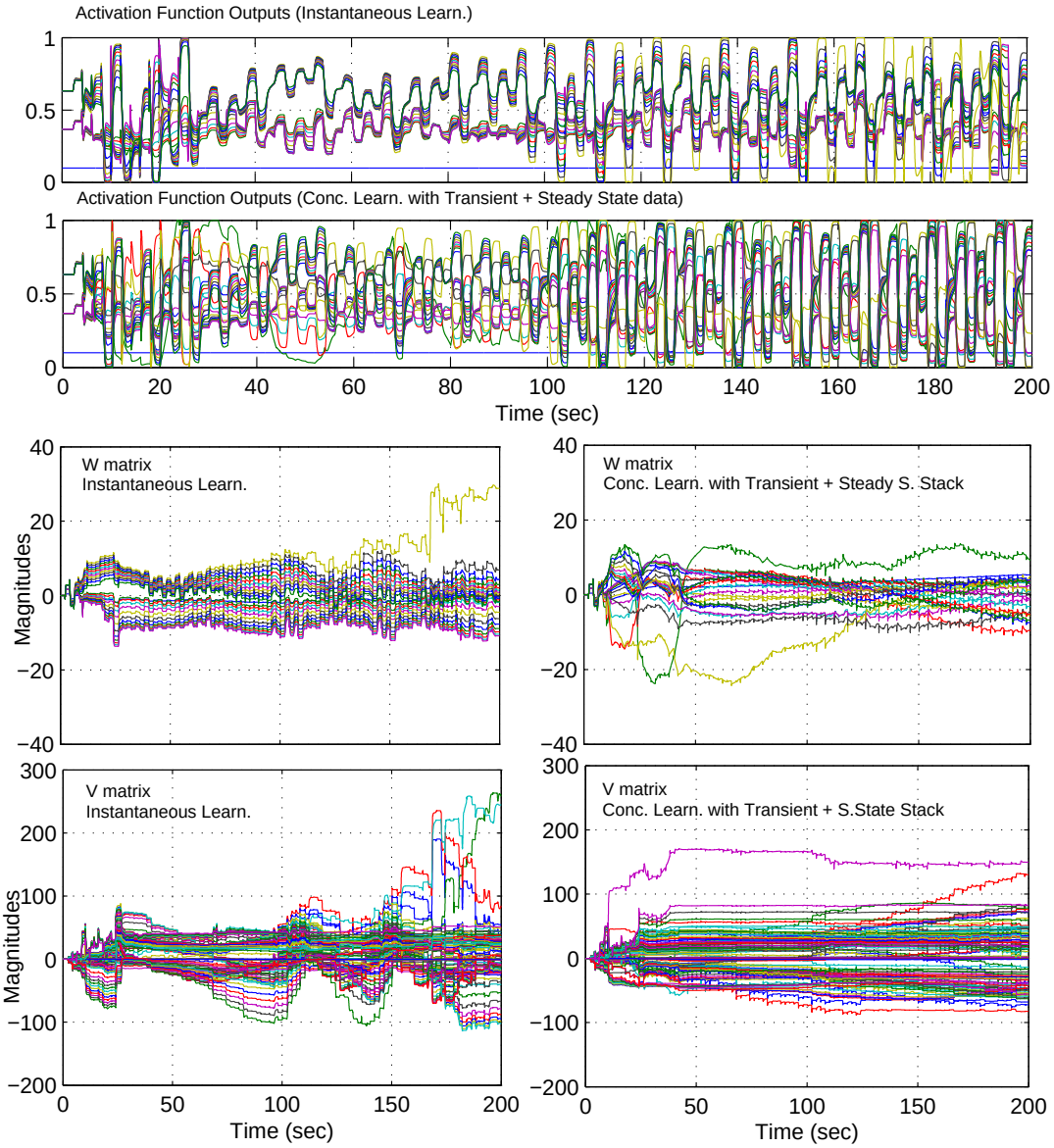

Figure 4.30: Long Term Comparison Activation Function Outputs and Weight Updates at a High L_{R_W} and L_{R_V} for $\hat{\alpha}$ Dynamics

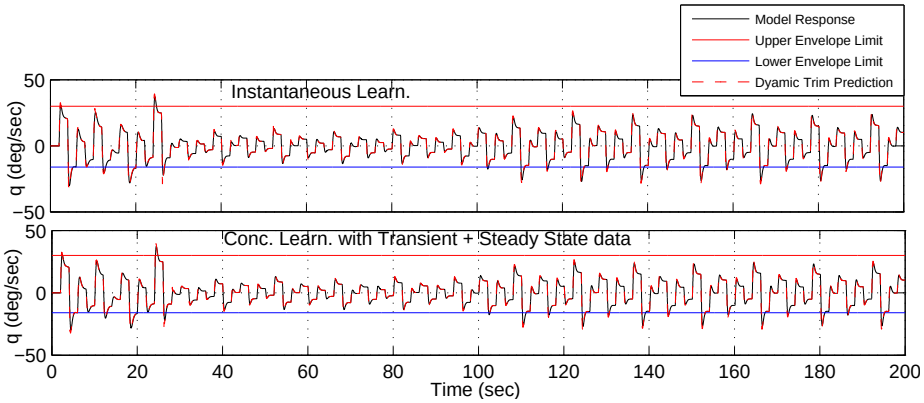

Figure 4.31: $\hat{\alpha}$ Response at Long Term

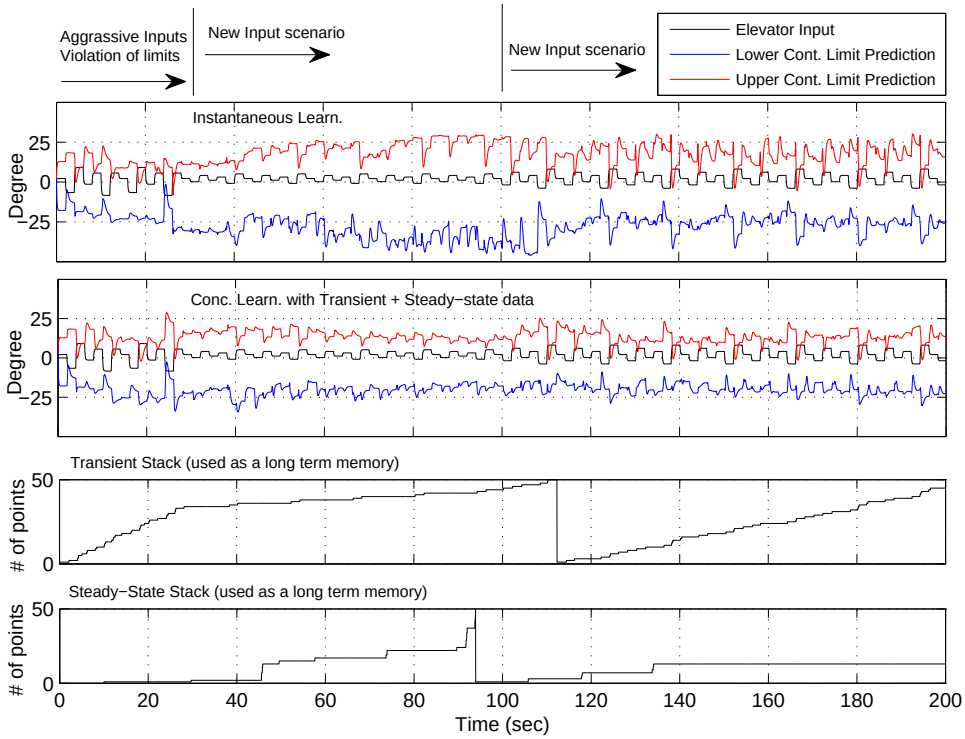

Figure 4.32: Long Term Comparison of Limit Control Predictions at a High L_{R_W} and L_{R_V} for $\hat{q}$ Dynamics

Similar results are achieved for pitch rate limits too. Pitch rate response, control limits due to pitch rate limits and weight updates can be seen in figures 4.31, 4.32 and 4.33. When concurrent learning is used better control margins are obtained at long term compared to using instantaneous data only (fig 4.32)

Aircraft states of the long term simulation of this section can be seen in figure 4.34

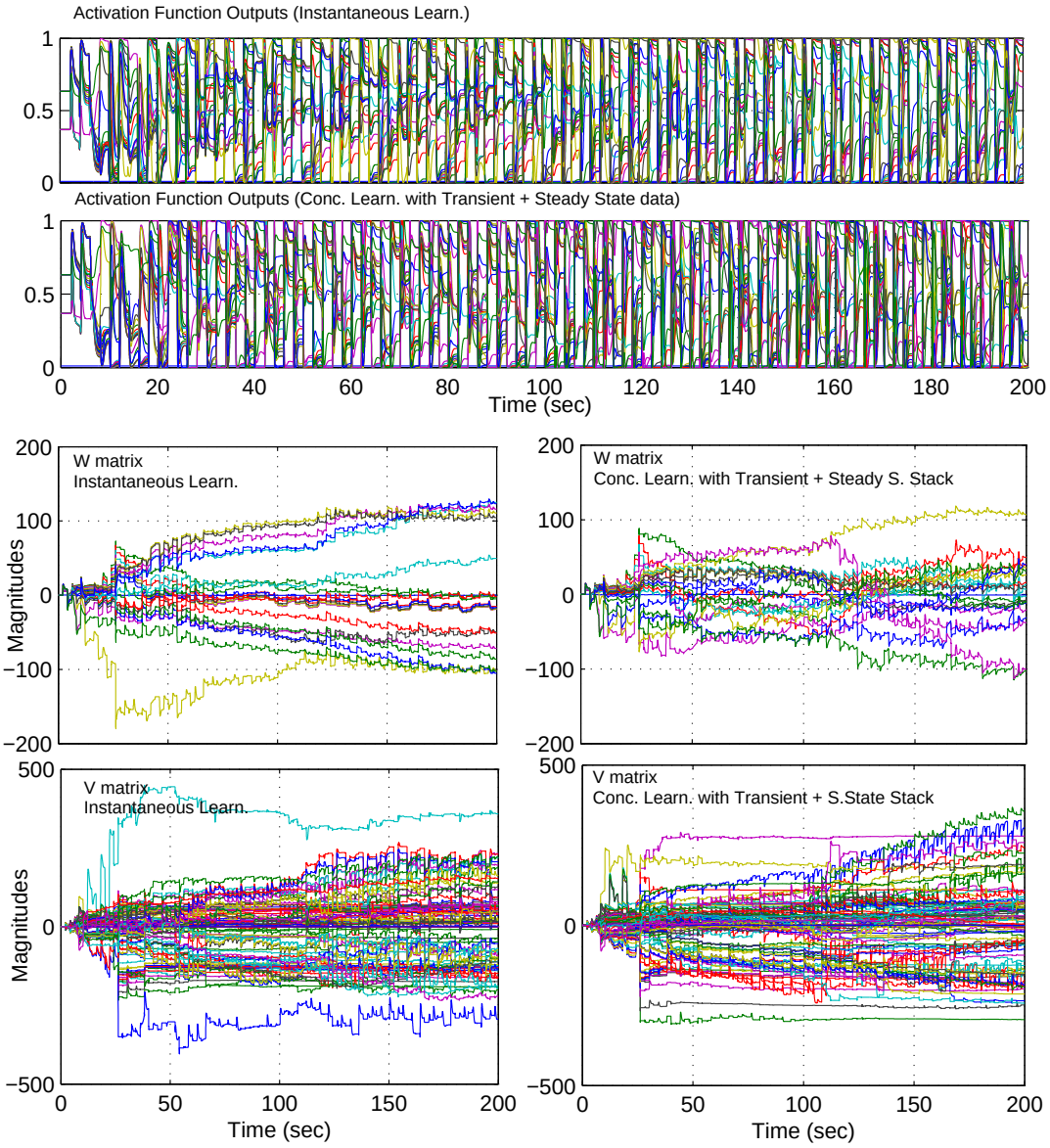

Figure 4.33: Long Term Comparison Activation Function Outputs and Weight Updates at a High L_{R_W} and L_{R_V} for $\hat{q}$ Dynamics

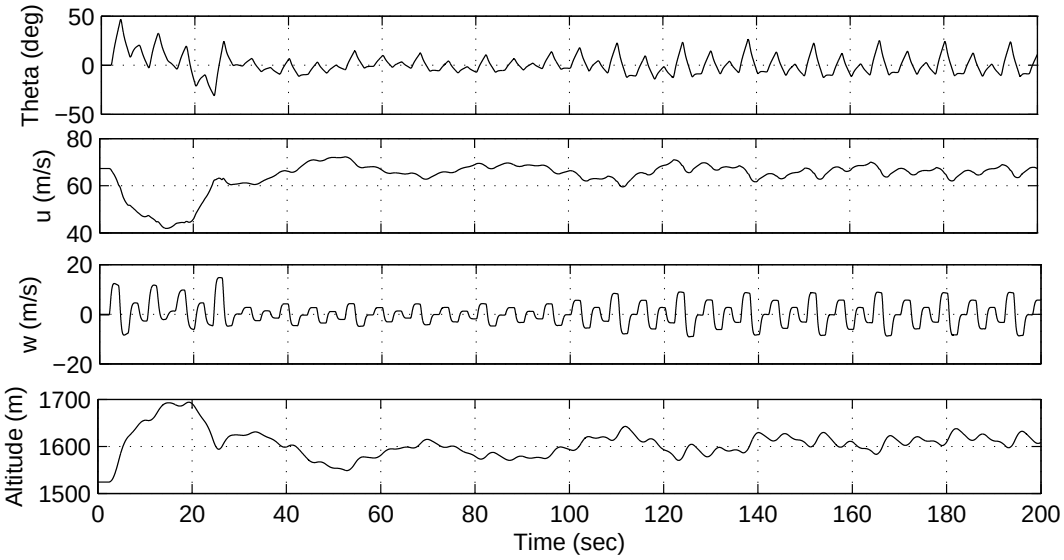

Figure 4.34: Aircraft States during the Long Term Simulation

Table 4.6: Design Param. of High Learning Rate Simulations
Angle of Attack Control Limits

$A_{11} = -5$	$A_{14} = 0.1$	$B_1 = -3$	$K_1 = 12$	$L_{R_W} = 2000$
$L_{R_V} = 3500$	$L_{R_{W_{tr}}} = 0.01$	$L_{R_{V_{tr}}} = 0.01$	$L_{R_{W_s}} = 0.005$	$L_{R_{V_s}} = 0.005$
$\epsilon_{y_1} = 0.0008$	$\epsilon_{y_2} = 0.01$	$\epsilon_{z_1} = 0.0008$	$\epsilon_{z_2} = 0.01$	$\epsilon_{\vec{x}} = 0.002$

Table 4.7: Design Param. of High Learning Rate Simulations
Load Factor Control Limits

$A_{22} = -5$	$A_{24} = 0.5$	$B_2 = -3$	$K_2 = 30$	$L_{R_W} = 2000$
$L_{R_V} = 2000$	$L_{R_{W_{tr}}} = 0.01$	$L_{R_{V_{tr}}} = 0.01$	$L_{R_{W_s}} = 0.01$	$L_{R_{V_s}} = 0.01$
$\epsilon_{y_1} = 0.0005$	$\epsilon_{y_2} = 0.0006$	$\epsilon_{z_1} = 0.0005$	$\epsilon_{z_2} = 0.0006$	$\epsilon_{\vec{x}} = 0.03$

Table 4.8: Design Param. of High Learning Rate Simulations
Pitch Rate Control Limits

$A_{33} = -10$	$A_{34} = 0.8$	$B_3 = -15$	$K_3 = 30$	$L_{R_W} = 2000$
$L_{R_V} = 3000$	$L_{R_{W_{tr}}} = 0.0005$	$L_{R_{V_{tr}}} = 0.0005$	$L_{R_{W_s}} = 0.005$	$L_{R_{V_s}} = 0.005$
$\epsilon_{y_1} = 0.001$	$\epsilon_{y_2} = 0.002$	$\epsilon_{z_1} = 0.001$	$\epsilon_{z_2} = 0.002$	$\epsilon_{\vec{x}} = 0.05$

4.4 Discussion

In this chapter, implementation of concurrent learning to neural network based adaptive dynamic trim and control limit estimation algorithm is presented. Online data storing is done in such a way that the data from transient response and steady state responses of limit parameters are recorded independently into different history stacks. In order to record transient data, a storing criterion which has been used for the design of a long term learning adaptive flight controller [33, 34, 35] is used. Whereas, a new criterion is applied for recording steady state data into a different stack. It is shown that much more accurate dynamic trim predictions are obtained when steady state data as well as transient data are used for weight adaptations compared to adaptations using transient and instantaneous data only.

Moreover, it is concluded that dynamic trim and control limit predictions using instantaneous learning has a potential to cause inaccurate iterative solutions especially for long term simulations. Since for long term, neural networks have a potential to gain high nonlinearity while forgetting the nature of the past limit violations or modeling error compensations. Whereas by using concurrent learning, past violations of limits and modeling error compensations are recorded into long term memories (history stacks) and those information are used for further weight adaptations. As expected, much more global online models and accurate dynamic trim predictions as well as control limits are obtained in long term simulations.

CHAPTER 5

CONCLUSION

Operational limits are constraints for both manned and unmanned aerial vehicles. The task of envelope protection is to observe and ensure vehicle operation within these limits. It is obvious that an automatic envelope protection system will maintain the adjustment between safety and performance, especially during agrassive maneuvers where the limit violations are highly probable. Therefore, for an effective envelope protection, approaching aircraft limits should be predicted before the actual limiting states are reached. These predictions are mainly used by limit avoidance algorithms to avoid the probable violations of envelope limits.

In this thesis, *limit detection* was the main focus. For that purpose, a 6DOF aircraft model is developed in chapter-2. Then, in chapter-3, using this model, maneuvering steady states of angle of attack, load factor and pitch rate parameters are predicted online. Control limits due to known envelope limits are estimated online as well. For these predictions, online models of angle of attack, load factor and pitch rate dynamics are generated. In order to generate online models, neural network augmented linear models are used. Modeling errors between real and approximate dynamics are compansated by adaptive neural networks.

In chapter-3, a classical weight update law which was called instantaneous learning is used for updating the neural network weights for modeling error compensation. A key problem in that chapter was that the online models were generated only locally. Due to this fact, in chapter-3, a relearning process of limit parameter dynamics over time is seen. Whereas, in limit prediction, global online models are needed since a future response is going to be predicted at the times the controls are applied. Therefore, online modeling could be done in such a way that the modeling error cancellations of a maneuver, which was encountered in the past, could be recorded and used for further modeling error cancellations. This idea is also

used as a motivation for the fourth chapter.

In chapter-4, a much more intelligent weight update law, called concurrent learning, is implemented to adaptive dynamic trim and control limit prediction algorithm. Besides, the benefits of concurrent learning over the instantaneous learning of chapter-3 are demonstrated with various simulations. The main advantages of concurrent learning enabled limit detection algorithm are presented below:

- Previously encountered maneuvers are recorded in long term memories (history stacks) and these information are used for further modeling error compensations. It is shown that neural network weights have a much more potential to be located in steady values. Therefore, global online models are much more probable when concurrent learning is used.

- Using concurrent learning, a faster generation of accurate online models are obtained for newly introduced modeling errors.

- It is concluded that using concurrent learning, neural network has a much more potential to represent the limit dynamics and normal operating conditions simultaneously. This way iterative solutions of dynamic trim and control limit predictions become much more reliable and accurate.

- It is also shown that by using a stack of steady state data for concurrent learning, for all above items, the accuracy of dynamic trim and control limit predictions is enhanced.

The following items are also listed as recommendations for future works:

- The performance of concurrent learning is highly dependent on data selection criteria as expected. In chapter-4, whenever the history stacks become full, oldest data is replaced with a new one. Therefore, it is obvious that the results would get better by using more intelliget data storing criteria. Different data storing criteria can be developed and compared with the ones used in this thesis, as future work.

- Neural network adaptations are generally affected by the complexity of the neural networks. Less number of neurons may cause difficulties in modeling error compensations.

Whereas difficulties may also be seen when high number of neurons are used. Therefore, with an optimum number of neurons, modeling errors may be compensated more effectively. In this thesis, a highly nonlinear network is used. As future study, a network with optimum number of neurons can be selected and the results may improve.

- Throughout the thesis longitudinal maneuvers are performed by using elevator inputs only. More realistic scenarios may be performed by including lateral maneuvers. For this purpose linear models and neural networks could be written as functions of lateral inputs and lateral slow states as well. It would be possible to calculate critical lateral controls due to known limits of lateral or longitudinal limit parameters. Therefore, as future work, multiple limits including critical rudder and critical aileron controls can be found for different maneuver types.

- Simulations for which the limits are violated can be repeated using a known limit avoidance method. The differences between concurrent learning and instantaneous learning can also be compared in the scope of limit avoidance as future work.

- Command limiting architecture [16] of envelope protection systems, which are mainly applied for autonomous UAV's, can also be studied as future work, in the scope of concurrent learning and instantaneous learning.

- As a future work, neural networks may be instantaneously linearized at each simulation time step. Control limits and dynamic trim predictions can be solved using instantaneous linearized networks. In appendix section, a procedure for instantaneous linearization is presented.

REFERENCES

[1] Menon, P. K., Iragavarapu, V. R., Whalley, M. S., "Estimation of Rotorcraft Limit Envelopes using Neural Networks", American Helicopter Society 52nd Annual Forum, Washington, DC, June 4-6, 1996.

[2] Mulgund, S. S., Zacharias, G. L., "A Hybrid Neural Network-Fuzzy Logic Limit Protection System for Rotorcraft",AIAA Guidance, Navigation and Control Conference, San Diego, CA, July 29-31, 1996.

[3] Bateman, A., Ward, D., Barron, R., Whalley, M., "A Piloted Simulation Evaluation of a Neural Network Limit Avoidance System for Rotorcraft",AIAA Atmospheric Flight Mechanics Conference, Portland, Oregon, Aug. 1999.

[4] Horn, J. F., Calise, A. J., Prasad, J. V. R., "Flight Envelope Cueing on a Tilt-Rotor Aircraft using Neural Network Limit Prediction",American Helicopter Society 54th Annual Forum, Washington, DC, May 1998.

[5] Horn, J. F., Calise, A. J., Prasad, J. V. R., "Development of Envelope Protection Systems for Rotorcraft",American Helicopter Society 55th Annual Forum, Montreal, Canada, May 1999.

[6] Horn, J. F., Calise, A. J., Prasad, J. V. R., "Flight Envelope Limiting Systems Using Neural Networks",AIAA Atmospheric Flight Mechanics Conference, Boston, MA, Aug. 1998.

[7] Horn, J. F., "Flight Envelope Limit Detection and Avoidance",Ph.D. Thesis, Georgia Institute of Technology, May 1999.

[8] Horn, J. F., Calise, A. J., Prasad, J. V. R., "Flight Envelope Cueing on a Tilt-Rotor Aircraft Using Neural Network Limit Prediction",*Journal of the American Helicopter Society*, Vol. 46, No. 1, Jan. 2001.

[9] Horn, J. F., Calise, A. J., Prasad, J. V. R., "Flight Envelope Limit Detection and Avoidance for Rotorcraft", European Rotorcraft Forum, Rome, Italy, Sept. 1999.

[10] Horn, J. F., "Rotor State Feedback for High Bandwidth Control and Structural Load Limiting", AHS Flight Controls and Crew System Design Technical Specialists Meeting, Philadelphia, PA, Oct. 2002.

[11] Horn, J. F., Calise, A. J., Prasad, J. V. R., "Flight Envelope Limit Detection and Avoidance for Rotorcraft", *Journal of the American Helicopter Society* Vol. 47, No. 4, Oct. 2002.

[12] Sahani, N. A. "Envelope Protection Systems for Piloted and Unmanned Rotorcraft", Phd Thesis, The Pennsylvania State University, December 2005.

[13] Yavrucuk, I., Prasad, J.V.R, Unnikrishnan, S., "Envelope Protection for Autonomous Unmanned Aerial Vehicles", *Journal of Guidance, Control and Dynamics*, Vol. 32, No. 1, January-February 2009.

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

[15] Calise, A. J., Rysdyk, R. T., "Adaptive Model Inversion Flight Control For Tiltrotor Aircraft", AIAA Guidance, Navigation and Control Conference, Paper No. 97-3758, 1997.

[16] Yavrucuk, I., Kutay, A. T., "Adaptive Neural Network Based Control System Applications for Autonomous UAV's", AIAC, 2005-024.

[17] Etkin, B., Reid, L. D., "Dynamics of Flight, Stability and Control", John Wiley and Sons, 1996.

[18] Raol, J. R., Singh, J., "Flight Mechanics Modeling and Analysis", CRC Press, 2009.

[19] McLean, D., "Automatic Flight Control Systems", Prentice Hall, 1990.

[20] Stevens, B. L., Lewis, F. L., "Aircraft Control and Simulation", John Wiley and Sons, 1992.

[21] Rauw, M. O., "A Simulink environment for Flight Dynamics and Control analysis - application to the DHC-2 Beaver", Graduate Thesis, TU Delft, 1993.

[22] FAA, "Part 3, Cessna 172 180HP Conversion, C172P Flight Manual Supplement", FAA Approved, 1987.

[23] Galloway, J. D., "Optimization of Conceptual Aircraft Design for Stability/Control and Performance", MSc. Thesis, University of Washington, 2001.

[24] Galloway, J. D., "Optimization of Conceptual Aircraft Design for Stability/Control and Performance", MSc. Thesis, University of Washington, 2001.

[25] Filippone, A., "Flight Performance of Fixed Wing and Rotary Wing Aircraft", AIAA, 2006.

[26] Eshelby, M. E., "Aircraft Performance: Theory and Practice", AIAA, 2000.

[27] Yavrucuk, I., Prasad, J.V.R, Calise, A.J., "Carefree Maneuvering Using Adaptive Neural Networks", AIAA Atmospheric Flight Mechanics Conference and Exhibit, Monterey, CA, 2002.

[28] Yavrucuk, I., Prasad, J.V.R, Calise, A.J., Unnikrishnan, S., "Adaptive Limit and Control Margin Prediction and Limit Avoidance",American Helicopter Society 58th Annual Forum, Montreal, Canada, 2002.

[29] Yavrucuk, I., Prasad, J.V.R, Calise, A.J., "Adaptive Limit Detection and Avoidance for Carefree Maneuvering", AIAA Atmospheric Flight Mechanics Conference and Exhibit, Montreal, Canada, 2001.

[30] Yavrucuk, I., "Adaptive Limit Margin Detection and Limit Avoidance", Phd. Thesis, Georgia Institue of Technology, 2003.

[31] Johnson, E.N., “Limited Authority Adaptive Flight Control”, Phd. Thesis, Georgia Institue of Technology, 2000.

[32] Kim, N., “Improved Methods in Neural Network-Based Adaptive Output Feedback Control, with Applications to Flight Control”, Phd. Thesis, Georgia Institue of Technology, 2003.

[33] E. N. Johnson, S. M. Oh., “Adaptive Control Using Combined Online and Background Learning Neural Network”,In Proceedings of the 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas, 2004.

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

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

[36] G. Chowdhary and E. N. Johnson, “Flight Test Validation of Long Term Adaptive Flight Controller”, In Proceedings of AIAA Guidance, Navigation and Control Conference, Chicago, Illinois, 2009.

[37] Hadad, W., Chellebaonia V., “Nonlinear Dynamical Systems and Control, A Lyapunov Based Approach”, Preprint, 2006, copyrighted by the Authors.

[38] Khalil H., “Nonlinear Systems”, Prentice Hall USA, 3rd Edition, December 18, 2001.

APPENDIX A

ITEMS RELATED TO MATHEMATICAL MODEL AND NEURAL NETWORKS

A.1 Mathematical Derivation of Linearized Dynamic Trim and Control Limit Equations

Single hidden layer neural networks can easily be linearized around the current neural network input vector by use of the following taylor expansion:

$$\Delta(\bar{x}) \approx \Delta(\bar{a}) + \frac{d\Delta}{d\bar{x}}|_{\bar{x}=\bar{a}}(\bar{x}-\bar{a}) + (HOT) \tag{A.1}$$

since, $\Delta(\bar{x}) = W^T\beta(V^T\bar{x})$ for a SHL network,

$$\frac{d\Delta}{d\bar{x}}|_{\bar{x}=a} = (W^T\frac{d\beta}{dz}\frac{dz}{d\bar{x}})|_{\bar{x}=\bar{a}} \tag{A.2}$$

where, $\frac{dz}{d\bar{x}} = V^T$ and $\frac{d\beta}{dz}$ is a function of $V^T\bar{x}$ therefore equation A.1 becomes with omitting the higher order terms,

$$\Delta(\bar{x}) = W^T\beta(V^T\bar{a}) + W^T\frac{d\beta}{dz}|_{z=V^T\bar{a}}V^T(\bar{x}-\bar{a}) \tag{A.3}$$

$$\Delta(\bar{x}) = W^T\beta(V^T\bar{a}) - W^T\frac{d\beta}{dz}|_{z=V^T\bar{a}}V^T\bar{a} + W^T\frac{d\beta}{dz}|_{z=V^T\bar{a}}V^T\bar{x} \tag{A.4}$$

The last term in equation A.4 can be written by use of associative property of matrix multiplications, in other words $W^T\frac{d\beta}{dz}|_{z=V^T\bar{a}}V^T\bar{x} = (W^T\frac{d\beta}{dz}|_{z=V^T\bar{a}}V^T)\bar{x}$,

Here, $\bar{x} = [bias\ \hat{x}_f\ x_s\ u]$ and the multiplication $(W^T \frac{d\beta}{dz}|_{z=V^T\bar{a}} V^T)$ becomes a 1 by 4 matrix, can be called as matrix L. And, equation A.4 can be written as:

$$\Delta(\bar{x}) = W^T\beta(V^T\bar{a}) - W^T\frac{d\beta}{dz}|_{z=V^T\bar{a}}V^T\bar{a} + [L_{11}\ L_{12}\ L_{13}\ L_{14}]\begin{bmatrix} bias \\ \hat{x}_f \\ x_s \\ u \end{bmatrix} \tag{A.5}$$

Note that the signals such as $W^T\beta(V^T\bar{a})$, $W^T\frac{d\beta}{dz}|_{z=V^T\bar{a}}V^T\bar{a}$ and $[L_{11}\ L_{12}\ L_{13}\ L_{14}]$ are all available signals at each simulation time step.

Dynamic trim equation, 3.15, may be written in terms of linearized neural network, as follows

$$A[\hat{x}_{f_{DT}}\ x_s]^T + Bu + W^T\beta(V^T\bar{a}) - W^T\frac{d\beta}{dz}|_{z=V^T\bar{a}}V^T\bar{a} + [L_{11}\ L_{12}\ L_{13}\ L_{14}]\begin{bmatrix} bias \\ \hat{x}_{f_{DT}} \\ x_s \\ u \end{bmatrix} + Ke = 0. \tag{A.6}$$

also, an iterative dynamic trim solution can be writtten in the following form,

$$\hat{x}_{f_{DT_{i+1}}} = -(A_{11} + L_{11})^{-1}((A_{14} + L_{13})x_s + (B_1 + L_{14})\delta_e + W^T\beta(V^T\bar{a}) - W^T\frac{d\beta}{dz}|_{z=V^T\bar{a}}V^T\bar{a} + Ke) \tag{A.7}$$

control limits are expressed as well,

$$\hat{u}_{DT_{i+1}} = -(B_1+L_{14})^{-1}((A_{11}+L_{11})\hat{x}_{f_{DT_{lim}}} + (A_{14}+L_{13})x_s + \Delta_1(\hat{x}_{f_{DT_{lim}}}, x_s, \hat{u}_{DT_i}, bias) + Ke) \tag{A.8}$$

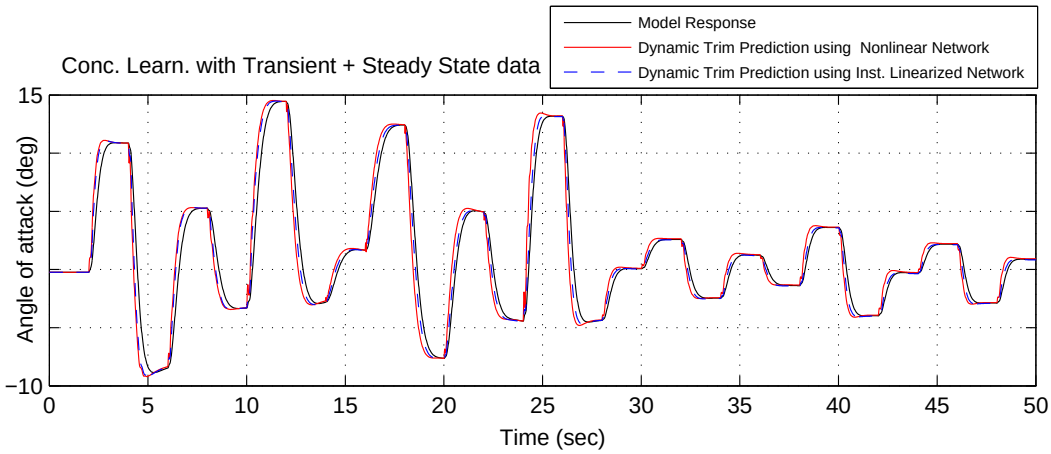

Figure A.1: Comparison of Dynamic Trim Predictions using Nonlinear and Instantaneously Linearized Networks

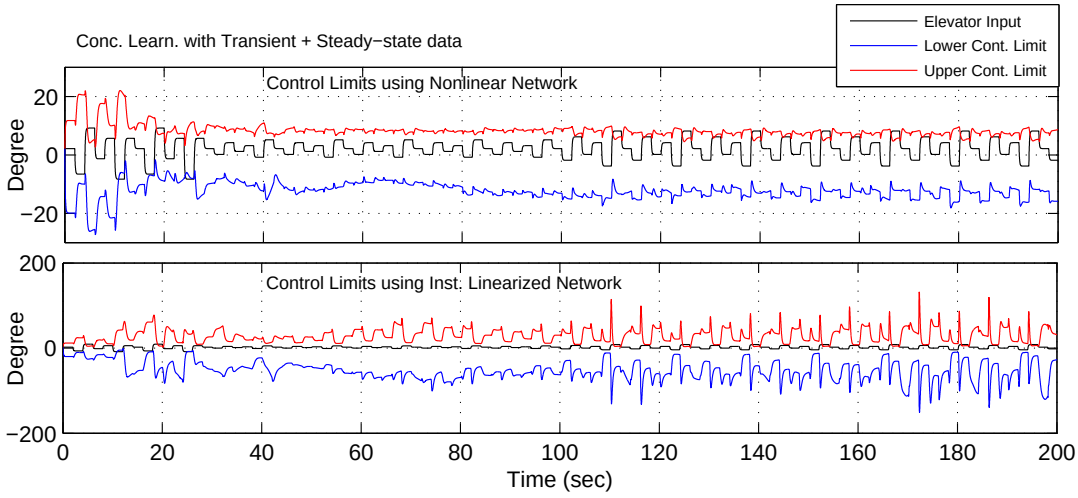

Figure A.2: Comparison of Control Limit Predictions using Nonlinear and Instantaneously Linearized Networks

A.2 Trimmed and Linearizad Model of Chapter-2

A.2.1 Trim Point

State vector and input vector of the nonlinear model have the following forms:

$$\vec{x} = [\phi\ \theta\ \psi p\ q\ r\ u\ v\ w\ X_e\ Y_e\ Z_e\ Alt]' \tag{A.9}$$

$$\vec{u} = [\delta_e\ \delta_a\ \delta_r\ \delta_{thr}]' \tag{A.10}$$

Level flight trim point at 67.2 m/s (true airspeed) is found by using the method mentioned in section 2.3.1. The following trim condition is obtained and used throughout the thesis:

$$\vec{x} = [0\ \ -0.004\ \ 0\ \ 0\ \ 0\ \ 0\ \ 67.27\ \ 0\ \ -0.2717\ \ 0\ \ 0\ \ -1524\ \ 1524]' \tag{A.11}$$

$$\vec{\dot{x}} = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 67.27\ \ 0\ \ 0]' \tag{A.12}$$

$$\vec{u} = [2.17\ \ 0\ \ 0\ \ 35.9]' \tag{A.13}$$

Note that, attitudes and rates are in radians and inputs are in degrees. Others are in metric system.

A.2.2 Linearized Model

The following linear model around the trim point of section 2.3.1 is obtained by using *linmod* command of Matlab.

$$\dot{x} = A \begin{Bmatrix} u \\ w \\ q \\ \theta \\ v \\ p \\ r \\ \phi \end{Bmatrix} + B \begin{Bmatrix} \delta_e \\ \delta_a \\ \delta_r \\ \delta_{thr} \end{Bmatrix} \tag{A.14}$$

where, A and B are found to be as:

$$A = \begin{Bmatrix} -0.0453 & 0.0797 & 0.2637 & -9.8099 & 0 & 0 & 0 & 0 \\ -0.2976 & -2.1089 & -65.2924 & 0.0392 & 0 & 0 & 0 & 0 \\ 0.0101 & -0.2081 & -6.8265 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.1877 & -0.4683 & -66.7113 & 9.8099 \\ 0 & 0 & 0 & 0 & -0.4512 & -13.0178 & 2.1357 & 0 \\ 0 & 0 & 0 & 0 & 0.1395 & -0.3349 & -1.2185 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.0040 & 0 \end{Bmatrix} \tag{A.15}$$

$$B = \begin{Bmatrix} -0.001 & 0 & 0 & 0.0416 \\ -0.2390 & 0 & 0 & 0 \\ -0.6099 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1049 & 0 \\ 0 & 1.3173 & 0.0831 & 0 \\ 0 & -0.0625 & -0.1791 & 0 \\ 0 & 0 & 0 & 0 \end{Bmatrix} \tag{A.16}$$

A.3 Derivation of Concurrent Learning Weight Update Law

Error dynamics of chapter-3 is written in the following forms:

$$\dot{e} = (A_1 - K)e - \hat{W}^T\beta(\hat{V}^T\bar{x}) + W^{*^T}\beta(V^{*^T}\bar{x}) + \epsilon(t) \tag{A.17}$$

$$\tilde{W} = \hat{W} - W^* \tag{A.18}$$

$$\tilde{V} = \hat{V} - V^* \tag{A.19}$$

We can expand the last three terms of equation A.17, and error dynamics take the following form [36]:

$$\dot{e} = (A_1 - K)e - [\tilde{W}^T(\beta(\hat{V}^T\bar{x}) - \beta^{'}(\hat{V}^T\bar{x})\hat{V}^T\bar{x}) + \hat{W}^T\beta^{'}(\hat{V}^T\bar{x})\tilde{V}^T\bar{x} + w] \tag{A.20}$$

where,

$$w = \tilde{W}^T\beta^{'}(V^{*^T}\bar{x}) + (HigherOrderTerms) + \epsilon(t) \tag{A.21}$$

It is shown in reference [36] that w signal is bounded. In order to obtain the bounds on w, reader may refer to [36].

Now we may consider the following Lyapunov [37, 38] candidate of the form:

$$L(e, \tilde{W}, \tilde{V}) = \frac{1}{2}e^T Pe + \frac{1}{2}tr\{\tilde{W}L_{R_W}^{-1}\tilde{W}^T\} + \frac{1}{2}tr\{\tilde{V}^T L_{R_V}^{-1}\tilde{V}\} \tag{A.22}$$

Take the time derivative of the function,

$$\dot{L}(e, \tilde{W}, \tilde{V}) = \frac{1}{2}\dot{e}^T Pe + \frac{1}{2}e^T P\dot{e}\frac{1}{2}tr\{\dot{\tilde{W}}L_{R_W}^{-1}\tilde{W}^T\} + \frac{1}{2}tr\{\tilde{V}^T L_{R_V}^{-1}\dot{\hat{V}}\} \tag{A.23}$$

then substitude equation A.20 into equation A.23,

$$\dot{L}(e,\tilde{W},\tilde{V}) = \frac{1}{2}e^T Qe - r^T[\tilde{W}^T(\beta(\hat{V}^T\bar{x}) - \beta'(\hat{V}^T\bar{x})\hat{V}^T\bar{x}) + \hat{W}^T\beta'(\hat{V}^T\bar{x})\tilde{V}^T\bar{x} + w]$$
$$... + tr\{(\dot{W}_i + \dot{W}_h)L_{R_W}^{-1}\tilde{W}^T\} + tr\{\tilde{V}^T L_{R_V}^{-1}(\dot{V}_i + \dot{V}_h)\} \quad \text{(A.24)}$$

$\dot{W}_h$ and $\dot{W}_i$ are the weight update contributions of recorded data and instantaneous data respectively.

We can add and subtract $\sum_{i=1}^{p}(\triangle_i - \xi_i)^T(\triangle_i - \xi_i)$ from (A.24), where $(\triangle_i - \xi_i)^T = r_{c_i}$ and $e^T P = r^T$, $\dot{L}(e,\tilde{W},\tilde{V})$ becomes:

$$\dot{L}(e,\tilde{W},\tilde{V}) = \frac{1}{2}e^T Qe - r^T[\tilde{W}^T(\beta(\hat{V}^T\bar{x}) - \beta'(\hat{V}^T\bar{x})\hat{V}^T\bar{x}) + \hat{W}^T\beta'(\hat{V}^T\bar{x})\tilde{V}^T\bar{x} + w]$$
$$... + tr\{(\dot{W}_i + \dot{W}_h)L_{R_W}^{-1}\tilde{W}^T\} + tr\{\tilde{V}^T L_{R_V}^{-1}(\dot{V}_i + \dot{V}_h)\}$$
$$... - \sum_{i=1}^{p} r_{c_i}^T(\triangle_i - \xi_i) + \sum_{i=1}^{p} r_{c_i}^T(\triangle_i - \xi_i) \quad \text{(A.25)}$$

Expanding the last term of equation A.25, we can set the following terms to zero:

$$tr\{[(-\beta(\hat{V}^T\bar{x}) + \beta'(\hat{V}^T\bar{x})\hat{V}^T\bar{x})r^T + \dot{W}_i L_{R_W}^{-1}]\tilde{W}^T\} = 0 \quad \text{(A.26)}$$

$$tr\{\tilde{V}^T(-\bar{x}r^T W^T\beta' + L_{R_V}^{-1}\dot{V}_i)\} = 0 \quad \text{(A.27)}$$

$$tr\{(\sum_{i=1}^{p}(\beta_i - \beta_i' V^T\bar{x}_i)r_{c_i}^T + \dot{W}_h L_{R_W}^{-1})\tilde{W}^T\} = 0 \quad \text{(A.28)}$$

$$tr\{\tilde{V}^T(\bar{x}_i r_{c_i}^T W^T\beta_i' + L_{R_V}^{-1}\dot{V}_h)\} = 0 \quad \text{(A.29)}$$

Now, we can solve for $\dot{W}_i$,$\dot{W}_h$,$\dot{V}_i$ and $\dot{V}_h$. Weight update law of chapter-4 can be written in the following forms:

$$\dot{W} = \dot{W}_i + \dot{W}_h \quad \text{(A.30)}$$

$$\dot{V} = \dot{V}_i + \dot{V}_h \tag{A.31}$$

User may refer to [36] for detailed Lyapunov based proof of ultimate boundedness of error signals.