

DIRECT ADAPTIVE FLIGHT ENVELOPE PROTECTION

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 DOCTOR OF PHILOSOPHY  
IN  
AEROSPACE ENGINEERING

MAY 2016



Approval of the thesis:

**DIRECT ADAPTIVE FLIGHT ENVELOPE PROTECTION**

submitted by **GÖNENÇ GÜRSOY** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver \_\_\_\_\_  
Dean, Graduate School of **Natural and Applied Sciences**

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

Assoc. Prof. Dr. İlkey Yavrucuk \_\_\_\_\_  
Supervisor, **Aerospace Engineering, METU**

**Examining Committee Members:**

Prof. Dr. Ozan Tekinalp \_\_\_\_\_  
Department of Aerospace Engineering, METU

Prof. Dr. Metin U. Salamcı \_\_\_\_\_  
Department of Mechanical Engineering, Gazi University

Assoc. Prof. Dr. İlkey Yavrucuk \_\_\_\_\_  
Department of Aerospace Engineering, METU

Assist. Prof. Dr. Ali Türker Kutay \_\_\_\_\_  
Department of Aerospace Engineering, METU

Assist. Prof. Dr. Yiğit Taşcıoğlu \_\_\_\_\_  
Department of Mechanical Engineering, TOBB ETÜ

**Defense 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

## DIRECT ADAPTIVE FLIGHT ENVELOPE PROTECTION

Gürsoy, Gönenç

Ph.D., Department of Aerospace Engineering

Supervisor : Assoc. Prof. Dr. İlkey Yavrucuk

May 2016, 130 pages

In this thesis, two vital signals to enable flight envelope protection, namely the onset to the flight envelope, i.e. limit margin, and the available control travel to reach the limit boundary, i.e. control margin, are estimated using adaptive neural-network-based approximate models. An adaptive learning method, known as concurrent learning, is used to update the adaptive weights online with guaranteed signal bounds. Current and previously recorded data are used concurrently in the weight update. Minimum singular value maximization method is used to record necessary data online for concurrent learning. Results showed better convergence properties of the network weights compared with results in the literature in which only the current data is used for network weight updates. New methodologies are introduced to calculate limit and control margins from approximate online models. None of the introduced methods require on-line iterations and therefore remove a previously introduced assumption related to iteration convergence. Nonlinear fixed wing and rotary wing aircraft models are used to show the effectiveness in simulation for estimating limit and control

margins and avoiding the limit through artificial control saturation.

Keywords: flight envelope protection, limit prediction, limit avoidance, carefree maneuvering, active controls, fly-by-wire

# ÖZ

## DOĞRUDAN ADAPTİF UÇUŞ ZARFI KORUMASI

Gürsoy, Gönenç

Doktora, Havacılık ve Uzay Mühendisliği Bölümü

Tez Yöneticisi : Doç. Dr. İlkey Yavrucuk

Mayıs 2016 , 130 sayfa

Bu tezde uçuş zarfı koruması için gerekli olan iki önemli sinyal: limit marjini olarak bilinen zarf limitlerine uzaklık ve kontrol marjini olarak bilinen kontrol limitlerine uzaklık sinyalleri, adaptif sinir ağı tabanlı yaklaşık modeller kullanılarak tahmin edilmiştir. Eş zamanlı öğrenme olarak bilinen ve sinyallerinin sınırlı olduğu garanti edilen bir adaptif öğrenme yöntemi, yapay sinir ağı katsayılarını güncellemek için kullanılmıştır. Anlık veriler ve geçmişte kaydedilen veriler eş zamanlı olarak katsayı güncellemesinde kullanılmıştır. Eş zamanlı öğrenme için gerekli olan veriler, en küçük tekil değeri büyütme yöntemi kullanılarak çevrimiçi kaydedilmiştir. Literatürde bulunan, anlık verileri kullanmaya dayalı öğrenme yöntemlerine göre daha iyi adaptasyon sonuçları elde edilmiştir. Elde edilen çevrimiçi modeller kullanılarak limit ve kontrol marjinleri tahmin edebilecek yeni yöntemler sunulmuştur. Sunulan yeni yöntemler, çevrimiçi iterasyonları ve önceki adaptif yöntemlerde ihtiyaç duyulan çevrimiçi iterasyon tabanlı varsayımları gerektirmemektedir. Geliştirilen yeni yöntemlerin verimliliği, doğrusal olmayan

uak ve helikopter modelleri kullanılarak limit ve kontrol marjinleri tahmininde ve kontrol limitleme yntemi ile limitlerden kaınarak gsterilmiřtir.

Anahtar Kelimeler: uuř zarfı koruma, limit belirleme, limitlerden kaınma, aktif kontroller

*Dedicated to*

*my lovely wife Funda*

## ACKNOWLEDGMENTS

I would like to acknowledge all my teachers and professors. I am thankful to all of them.

I would like to thank my committee members, consisting of Prof. Ozan Tekinalp, Prof. Metin U. Salamcı, Asst. Prof. Ali Türker Kutay and Asst. Prof. Yiğit Taşçıođlu. They provided helpful feedback throughout the development of this work.

I would like to express my special appreciation and thanks to my advisor Assoc. Prof. İlkey Yavrucuk, who has been a tremendous mentor for me. I would like to thank him for encouraging my research. I am grateful for his patience and understanding. He provided a focused work environment and had the ability to point me always in the right direction. His advice on both research as well as on my career have been priceless.

Thanks to all my friends and colleagues at Aerotim Engineering LLC, who provided valuable interactions and discussions. They include but are not limited to Onur Tarımcı, Eser Kubalı, Serdar Üşenmez, Nil Ercan, Sinan Ekinci, Figen Kılıç, Feyyaz Güner and Ercan Özgönül.

I would also like to thank to all my friends here in Ankara, who made my life here more enjoyable. Thanks to Kenan Cengiz, Mehmet Burak Sayar, Melike Guzey and İbrahim Yiğit.

Finally, I would like to express my deepest gratitude to my father and mother, Lütfullah and Sermin, who gave love and all necessary tools for life, my wife Funda Guzey Gürsoy, for her unrequited love, help and patience. I couldn't have succeeded without them.

# TABLE OF CONTENTS

ABSTRACT . . . . .	v
ÖZ . . . . .	vii
ACKNOWLEDGMENTS . . . . .	x
TABLE OF CONTENTS . . . . .	xi
LIST OF FIGURES . . . . .	xv
NOMENCLATURE . . . . .	xix
CHAPTERS	
1 INTRODUCTION . . . . .	1
1.1 Preliminaries . . . . .	1
1.1.1 Basic Methods for Flight Envelope Protection . . . . .	2
1.1.2 The Use of Fly-By-Wire Systems . . . . .	5
1.1.3 Limit Parameters and Response Types . . . . .	9
1.2 A Survey of Existing Limit Detection Methodologies . . . . .	11
1.3 Focus of This Research . . . . .	17
1.4 Contributions of This Research . . . . .	18
1.5 Structure of The Thesis . . . . .	20

2	ADAPTIVE LIMIT AND CONTROL MARGIN ESTIMATION	23
2.1	Problem Definition . . . . .	23
2.2	Adaptive Modeling Loops . . . . .	24
2.3	Limit and Control Margin Estimation . . . . .	25
2.4	Validation Studies . . . . .	28
2.5	Fixed-Point Solution Assumption in Literature . . . . .	28
2.6	Issues with Implementation . . . . .	30
2.7	Conclusions . . . . .	32
3	DIRECT ADAPTIVE LIMIT MARGIN AND SENSITIVITY ESTIMATION BASED CONTROL MARGIN ESTIMATION . . .	35
3.1	Methodology . . . . .	35
3.1.1	Direct Adaptive Limit Margin Estimation . . .	36
3.1.2	Sensitivity Estimation Based Control Margin Estimation . . . . .	44
3.2	Online Model Generation . . . . .	47
3.2.1	Linear in the Parameter Neural Network Structure	47
3.2.2	Concurrent Learning Weight Update Law . . .	49
3.2.3	Concurrent Learning Implementation . . . . .	50
3.3	Implementation and Simulation Results . . . . .	53
3.3.1	Flight Envelope Protection for Aircraft . . . . .	53
3.3.1.1	Example-1: Longitudinal Pull-up & Push-over . . . . .	54

	3.3.1.2	Example-2: Pull-up & Push-over during a Turn . . . . .	62
	3.3.2	Vertical Speed Limit Protection for Helicopters	65
	3.3.2.1	Example-1: Hover . . . . .	68
	3.3.2.2	Example-2: Low Speed Forward Flight	72
	3.3.2.3	Example-3: Forward Flight . . . . .	74
	3.4	Concluding Remarks . . . . .	79
4		DIRECT ADAPTIVE CONTROL MARGIN ESTIMATION . .	83
	4.1	Methodology . . . . .	83
	4.1.1	Direct Adaptive Control Margin Estimation . .	83
	4.2	Online Model Generation . . . . .	89
	4.3	Implementation and Simulation Results . . . . .	89
	4.3.1	Flight Envelope Protection for Aircraft . . . . .	89
	4.3.1.1	Example-1: Longitudinal Pull-up & Push-over . . . . .	90
	4.3.1.2	Example-2: Pull-up & Push-over during a Turn . . . . .	94
	4.3.2	Vertical Speed Limit Protection Helicopters . .	95
	4.3.2.1	Example-1: Hover . . . . .	97
	4.3.2.2	Example-2: Low Speed Forward Flight	100
	4.3.2.3	Example-3: Forward Flight . . . . .	102
	4.4	Concluding Remarks . . . . .	104
5		CONCLUSIONS . . . . .	107

REFERENCES . . . . .	113
APPENDICES	
A APPENDIX . . . . .	119
A.1 Proof of Theorem . . . . .	119
A.2 Concurrent Learning for Online Parameter Estimation .	123
CURRICULUM VITAE . . . . .	127

## LIST OF FIGURES

### FIGURES

Figure 1.1	Maximization of Operational Envelope to the boundaries of Flight Envelope . . . . .	2
Figure 1.2	Exceedance of a Conservative Envelope Boundary . . . . .	3
Figure 1.3	Modification of the Pilot Input through AFCS/SAS . . . . .	4
Figure 1.4	Schematic of a Fly-By-Wire Control System . . . . .	6
Figure 1.5	Demonstration of Limit Detection . . . . .	7
Figure 1.6	Limit Parameter Response Types for a given Step Input . . . . .	10
Figure 1.7	Control Margin Estimation using PNNs [8] . . . . .	12
Figure 1.8	Multilayer Neural Network used in [10] . . . . .	13
Figure 1.9	Adaptive Dynamic Trim Estimation Algorithm [16] . . . . .	14
Figure 2.1	Adaptive Online Loop [13, 27] . . . . .	25
Figure 2.2	Adaptive Output Feedback Control Architecture of [24] . . . . .	29
Figure 3.1	Direct Adaptive Limit Margin Estimation . . . . .	40
Figure 3.2	Direct Adaptive Limit Margin Estimation with Relative Degree Formulation . . . . .	44
Figure 3.3	Concurrent Learning Implementation . . . . .	51

Figure 3.4	Limit Prediction, Example-1 . . . . .	57
Figure 3.5	Weight Update Time Histories and Predicted Control Sensi- tivities, Example-1 . . . . .	58
Figure 3.6	Modeling Error Compensation and Minimum Singular Values of the History Stacks, Example-1 . . . . .	58
Figure 3.7	Limit Prediction, Nonzero Initial Condition for Weights, Example- 1 . . . . .	60
Figure 3.8	Modeling Error Compensation and Minimum Singular Val- ues of the History Stacks, Nonzero Initial Condition for Weights, Example-1 . . . . .	60
Figure 3.9	Weight Update Time Histories and Predicted Control Sensi- tivities, Nonzero Initial Condition for Weights, Example-1 . . . . .	61
Figure 3.10	Limit Avoidance, Example-1 . . . . .	62
Figure 3.11	Limit Prediction, Example-2 . . . . .	63
Figure 3.12	Limit Avoidance, Example-2 . . . . .	64
Figure 3.13	Simulation Block Diagram . . . . .	67
Figure 3.14	Vertical Speed Limit Profile . . . . .	68
Figure 3.15	Limit Prediction, Example-1 . . . . .	69
Figure 3.16	Modeling Error and NN Output, NN Weights, Predicted Col- lective Sensitivity, Minimum Singular Value of the History Stack, Example-1 . . . . .	70
Figure 3.17	Euler Angles and Ground Speeds, Example-1 . . . . .	70
Figure 3.18	Control Limiting for Limit Avoidance . . . . .	71
Figure 3.19	Limit Avoidance, Example-1 . . . . .	71
Figure 3.20	Limit Prediction, Example-2 . . . . .	72

Figure 3.21 Modeling Error and NN Output, NN Weights, Predicted Collective Sensitivity, Minimum Singular Value of the History Stack, Example-2 . . . . .	73
Figure 3.22 Euler Angles and Ground Speeds, Example-2 . . . . .	73
Figure 3.23 Limit Avoidance, Example-2 . . . . .	74
Figure 3.24 Limit Prediction, Example-3 . . . . .	75
Figure 3.25 Modeling Error and NN Output, NN Weights, Predicted Collective Sensitivity, Minimum Singular Value of the History Stack, Example-3 . . . . .	76
Figure 3.26 Euler Angles and Ground Speeds, Example-3 . . . . .	77
Figure 3.27 Limit Avoidance, Example-3 . . . . .	77
Figure 3.28 Modeling Error and NN Output, NN Weights, Predicted Collective Sensitivity, Minimum Singular Value of the History Stack, Example-3 . . . . .	78
Figure 4.1 Direct Adaptive Control Margin Estimation . . . . .	86
Figure 4.2 Direct Adaptive Control Margin Estimation with Relative Degree Formulation . . . . .	88
Figure 4.3 Limit Prediction, Example-1 . . . . .	91
Figure 4.4 Modeling Error Compensation and Minimum Singular Values, Example-1 . . . . .	92
Figure 4.5 Weight Update Time Histories, Example-1 . . . . .	92
Figure 4.6 Weight Update Time Histories, Long Term Response, Example-1 . . . . .	93
Figure 4.7 Limit Avoidance, Example-1 . . . . .	93
Figure 4.8 Limit Prediction, Example-2 . . . . .	94

Figure 4.9 Limit Avoidance, Example-2 . . . . .	95
Figure 4.10 Limit Prediction, Example-1 . . . . .	97
Figure 4.11 Modeling Error and NN Output, NN Weights, Minimum Singular Value of the History Stack, Example-1 . . . . .	98
Figure 4.12 Limit Avoidance, Example-1 . . . . .	99
Figure 4.13 Control Limiting for Limit Avoidance . . . . .	100
Figure 4.14 Limit Prediction, Example-2 . . . . .	101
Figure 4.15 Modeling Error and NN Output, NN Weights, Minimum Singular Value of the History Stack, Example-2 . . . . .	101
Figure 4.16 Limit Avoidance, Example-2 . . . . .	102
Figure 4.17 Limit Prediction, Example-3 . . . . .	103
Figure 4.18 Modeling Error and NN Output, NN Weights, Minimum Singular Value of the History Stack, Example-3 . . . . .	103
Figure 4.19 Limit Avoidance, Example-3 . . . . .	104
Figure A.1 NN Weights for $\sigma_{min}$ Maximization ON and OFF . . . . .	124
Figure A.2 $\sigma_{min}(Z)$ for $\sigma_{min}$ Maximization ON and OFF . . . . .	124

## NOMENCLATURE

$A_1, A_2, B$	Matrices of linear system
$A_{11}, B_{11}$	Elements of $A_1$ and $B$ matrices
$C_1, C_2, C_3, C_4$	Coefficients of a linear approximation
$e$	Model tracking error
$f, f_1, f_2$	Vectors of nonlinear functions
$h, h_n, g_n$	Nonlinear functions
$h$	Height above terrain
$n_z$	Load factor
$q$	Pitch rate
$S$	Vector of control input sensitivities
$tr(\cdot)$	Trace operator
$u$	Vector of control inputs
$u_e$	Single effective control input
$u'$	Vector of control inputs without $u_e$
$V$	Lyapunov function
$V_s$	Vertical speed
$V_e$	True airspeed
$W$	Weight matrix
$x$	Vector of system states
$\bar{x}$	Vector of neural network inputs
$y_p$	Limit Parameter
$Z, E$	History Stacks
$\alpha$	Angle of attack
$\Gamma$	Neural network learning gain
$\Delta$	Vector of neural networks
$\delta_e$	Elevator input
$\hat{\delta}_{e1}, \hat{\delta}_{e2}$	Elevator input approximations
$\bar{\partial}(\cdot)$	Central difference operator

$\mathcal{D}(\cdot)$	Average sum operator
$\epsilon$	Reconstruction error
$\theta$	Pitch angle
$\xi$	Modeling error
$\phi$	Neural network basis
$\phi$	Activation function
$\sigma_{min}(\cdot)$	Minimum singular value operator

### Subscripts

$DT$	Dynamic trim
$d$	Variable or calculation of a delayed moving time step
$f$	Fast
$prev$	Previous value
$upd$	Updated value
$p$	Size of a history stack
$s$	Slow
$lim$	Limit
$lim_{\alpha}$	$\alpha$ control limit
$lim_{n_z}$	$n_z$ control limit
$marg$	Limit or control margin
$marg_{\alpha}$	$\alpha$ control margin
$marg_{n_z}$	$n_z$ control margin
$\alpha$	Neural network, modeling error, history stack, model tracking error and basis of $\alpha$ channel
$n_z$	Neural network, modeling error, history stack, model tracking error and basis of $n_z$ channel

### Superscripts

$\hat{\cdot}$	Approximation
$l, k, n, p, r$	Dimension of a vector or matrix

## **Abbreviations**

AFCS	Automatic Flight Control System
ANN	Adaptive Neural Network
CFMS	Carefree Maneuvering System
EPS	Envelope Protection System
ERITS	Equivalent Retreating Indicated Tip Speed
FBW	Fly-By-Wire
HMD	Helmet Mounted Display
LPV	Linear Parameter Varying
MFD	Multi Functional Display
MRAC	Model Reference Adaptive Control
NN	Neural Network
PNN	Polynomial Neural Network
RPM	Revolution Per Minute
SAS	Stability Augmentation System



# CHAPTER 1

## INTRODUCTION

Aircraft have limits that should be avoided during flight to increase the overall confidence and safety. For a piloted aircraft pilot has to monitor the limits using the visuals in the cockpit and ensure safety in the cost of increased workload. Flight envelope protection is an area of research in literature and its focus is to develop algorithms or methods that can effectively cue pilots during flight to stay within the boundaries of a flight envelope. Such algorithms or methods are used to develop Envelope Protection Systems (EPS) to improve handling qualities and safety for both manned and unmanned aircraft.

### 1.1 Preliminaries

Boundaries of a flight envelope are generally restricted by aerodynamic, structural, power, control and operation specific limits (Fig. 1.1). For instance, the boundaries of a V-n diagram may define the aerodynamic limits such as angle of attack and airspeed limits. Related with structural integrity, maximum and minimum normal load factor limits define the structural operating boundaries. The limits associated with maximum engine torque and power are some examples of power related limits. Most of the aerodynamic, structural and power limits are associated with allowable sets of control positions, therefore, are often regarded as control limits [1]. There are additional limits such as vertical speed limits, typically defined to increase safety for a helicopter in a low height operation. Several operations or missions may require additional limits to increase safety. All these limits and others define the restricted flight conditions,

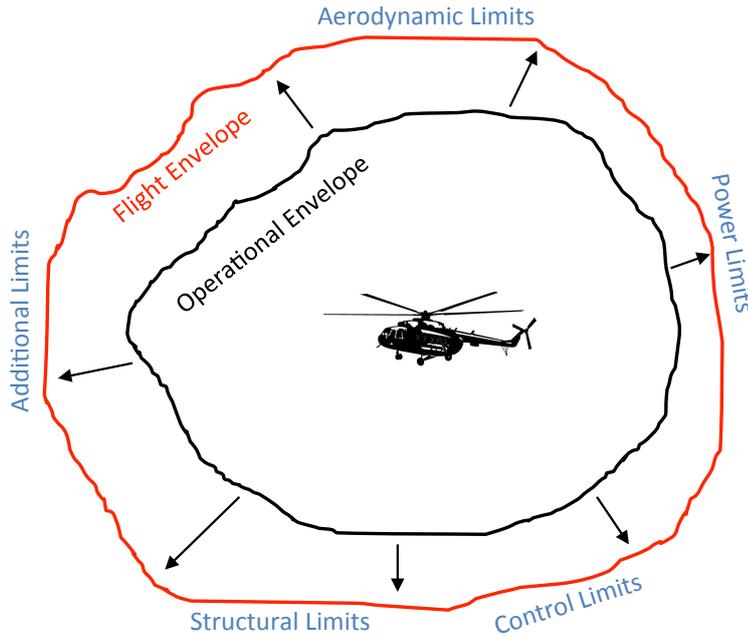


Figure 1.1: Maximization of Operational Envelope to the boundaries of Flight Envelope

hence, are regarded as the boundaries of a flight envelope. Note that in the terminology of flight envelope protection, a state or a parameter which limits the boundary of a flight envelope is called as *limit parameter*. Therefore, the parameters summarized above are only a few examples of limit parameters in the area.

### 1.1.1 Basic Methods for Flight Envelope Protection

Envelope Protection Systems are not only designed to warn a pilot to avoid flight envelope boundaries but also designed to enable effective maneuvering along the boundaries of a flight envelope. In literature, Envelope Protection Systems are also known as Carefree Maneuvering Systems (CFMS). Carefree maneuvering is a term used to describe a handling quality for modern aircraft such that the pilot is enabled to fly along the edges of a flight envelope. In the absence of an envelope protection or carefree maneuvering system, the monitoring of aircraft limits may significantly increase the workload on the pilot and may result in degraded aircraft performance and poor handling qualities. Whereas, in the presence of an EPS or CFMS, the allowable operational envelope, which is a

safe envelope that pilots typically use, can be maximized to the boundaries of an actual flight envelope (Fig. 1.1), hence the full performance of an aircraft can be maintained. It is apparent that systems which enable carefree maneuvering will become more important as the design requirements related with agility and maneuverability dominate the design of future aircraft.

The task of envelope protection is done in the past using simple devices or methods. Aural warnings and visual cues in the cockpit are just some ways of cueing the pilot for the task of envelope protection. A simple way is to visualize the distance of the considered parameters to the envelope limits through the Multi Functional Displays (MFDs), where the colour of the visuals might change according to the current distance to the limiting value. Also, visual cues can be sustained with aural cues as the aircraft fly closer to the envelope limits. Other than visual or aural aids, some devices such as stick shakers are often used to vibrate pilot controls as a stall warning for fixed wing aircraft. These simple cues can be initiated when a conservative envelope limit, also known as a *pre-limit*, is exceeded as in Figure 1.2.

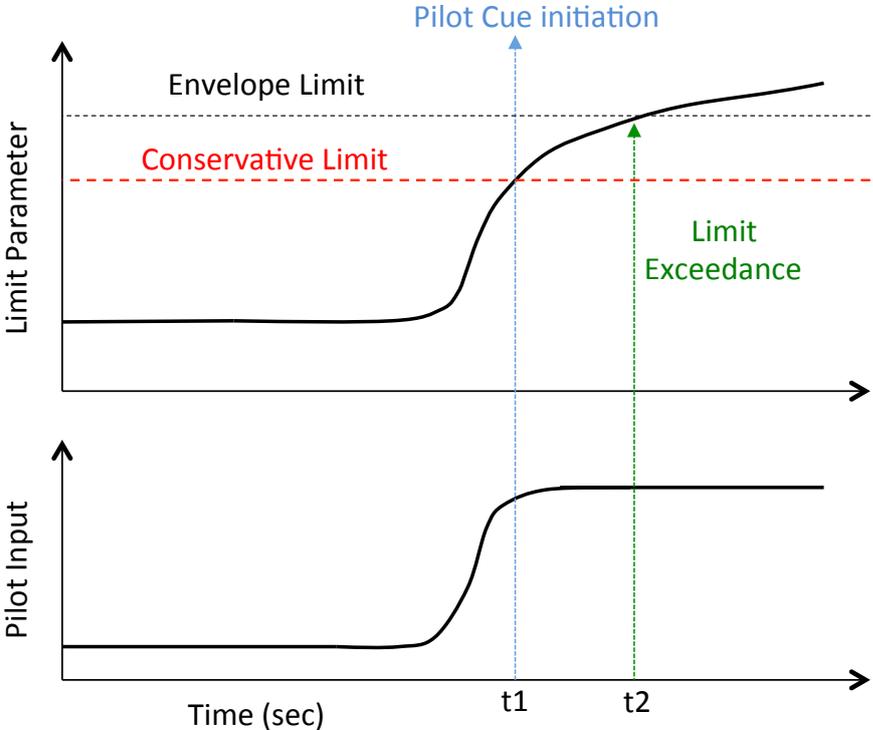


Figure 1.2: Exceedance of a Conservative Envelope Boundary

A concern in the selection of conservative limits is that a selected limit might be too preventive to maneuver at some flight conditions. Hence, the performance and the handling qualities might degrade due to the over protection and care-free maneuvering along the actual boundaries might not be possible. On the other hand, at some flight conditions, aircraft may have relatively fast transient dynamics. In that case, the time of a conservative limit exceedance might be too late for a pilot to react against the actual limits. Therefore, the selection of a conservative limit requires flight testing activities to predict the dynamic behaviour of the aircraft around the flight envelope boundaries and tuning activities with the pilots. Some examples of conservative limits for envelope cueing can be found in the Commanche RAH-66 helicopter [2]. In RAH-66, visual cues are given through the Helmet Mounted Display (HMD), and aural cues are given through the pilot headset. Conservative limits for load factor, main rotor shaft bending and engine torque limits are provided to the pilot. Instead of enabling the carefree handling capability, the featured envelope cueing system is designed to maintain safety and increase situational awareness during flight.

Other than providing the pilot with a variety of cues, the inputs of the pilot can be reshaped for automatic envelope protection. This idea is applicable if there exist an Automatic Flight Control System (AFCS) or a Stability Augmentation System (SAS) integrated to the flight control system of the aircraft. As depicted simply in Figure 1.3, the pilot input  $u$  can be modified by the flight control system for not to violate a flight envelope limit. In this technique, AFCS/SAS system gains (controller gains) can be manipulated as the aircraft fly closer to the limit parameter boundaries. This concept is first introduced in [3, 4]

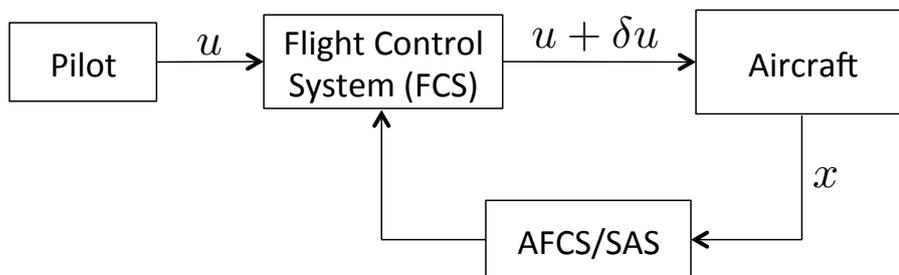


Figure 1.3: Modification of the Pilot Input through AFCS/SAS

for load factor limiting. Even with the full deflection of the stick, maximum load factor is shown to be avoided by manipulating the controller gains. Later, similar approaches are applied for Eurofighter and Airbus A319/320 aircraft. In the Eurofighter, pilot inputs are manipulated automatically in the flight control system to avoid the angle of attack and the load factor boundaries [5]. For the Airbus A319/320, relatively simple feedback laws are used for the maximum load factor and the maximum angle of attack protection [6]. Feedback laws are kept simple since the aircraft is not maneuvered aggressively.

Another example where the controller gains are manipulated for envelope protection is the RAH-66 helicopter. In the flight tests of RAH-66 program [7], very large longitudinal hub moments are reported for given longitudinal cyclic inputs. Instead of developing a hub moment cueing system, the solution is done through modifying the AFCS laws. At the end, the AFCS laws are manipulated to maintain the hub moments in a structurally safe region.

Note that the manipulation of the controller gains or the pilot inputs according to the limits or changing flight conditions require a careful investigation since the handling qualities for a conventional aircraft or helicopter can easily be degraded by the modification of the controller gains. An additional care may also be required if the pilot is not allowed to override the protection system as in the Eurofighter and A319/320 aircraft [5, 6]. Therefore, the design of such a system, based on the manipulations of the controller gains and pilot inputs, may require a complete information and analysis about the dynamics of the real aircraft. Such an engineering activity may not be straightforward, and in general require flight testing and evaluation.

### **1.1.2 The Use of Fly-By-Wire Systems**

For aircraft that have Fly-By-Wire (FBW) control systems, more advanced envelope protection methods are applicable. In a FBW system, pilot controls and control surfaces are not mechanically linked. Typically, an active control stick driven by a software is used to transfer pilot inputs to the control surfaces, as in Figure 1.4. The active control software is programmed to simulate the aero-

dynamic and mechanical loads on the pilot stick in order to match the real feel of the handling of the aircraft. Other than simulating the aerodynamic and mechanical loads, active controls in a FBW aircraft can be programmed to apply tactile cues for envelope protection.

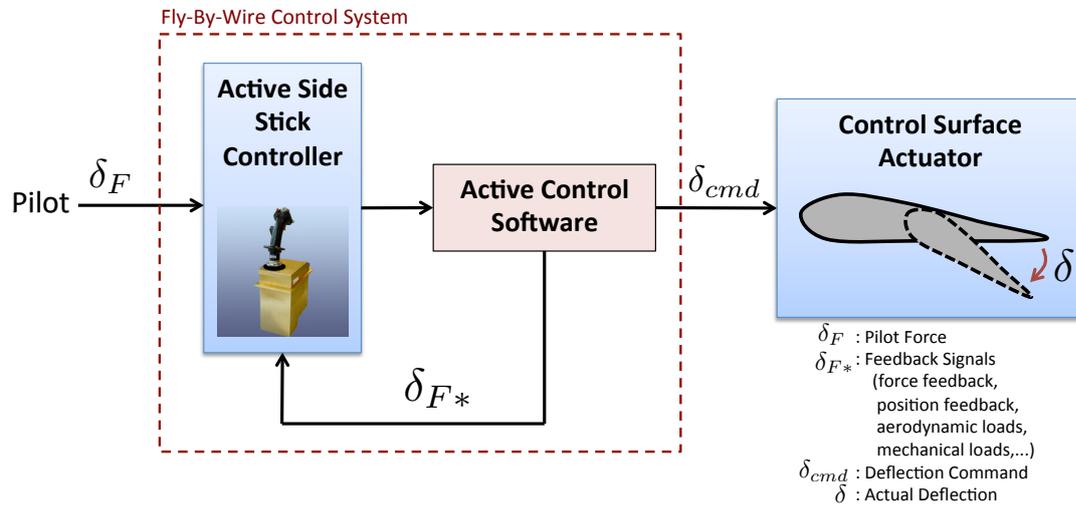


Figure 1.4: Schematic of a Fly-By-Wire Control System

The term *tactile cues* refers to the cues given in a FBW control system. Using the programmable force-feedback feature of the active controls, various control travel limiting and control rate limiting cues or several combinations of both can be given through the pilot stick. In general, the cues are known as *hard stops* and *soft stops* in active control terminology. Hard stops are generally used for the aircraft parameters that have limits which should never be exceeded. These limits are also known as *hard limits*. Structural limits such as maximum normal load factor for a fixed wing aircraft and maximum resultant hub moment for a rotorcraft might be some examples of hard limits. Soft stops are typically used for the aircraft parameters that have relatively more tolerance after a limit exceedance. These limits are known as *soft limits*. Transient torque and temperature limits for a helicopter engine might be some examples of soft limits. In previous studies [8, 9, 10, 11], tactile cues given through a FBW system are shown to be more effective for envelope protection than the cues given through aural and visual protection methods. It is also demonstrated that the effectiveness of the tactile cues can be significantly enhanced if they are provided before

the time of an actual limit exceedance.

Tactile cues can be enabled through an active control stick when the proximity of a future state of the aircraft to the envelope boundaries, and its mapping to the control axes are available. Proximity of a future state to the envelope boundaries is known as *limit margin* in flight envelope protection. Limit margin can be thought as an indication of closeness to the envelope limits. When the limit margin information is translated to the control axes, the translated margin is known as *control margin*. Control margins are the allowable control travels that would result in flight envelope boundaries. Both margins are the two vital signals for flight envelope protection and the determination or the prediction of both information is known as *limit detection*.

Limit detection concerns with the estimation of limit and control margins. An example for limit detection is demonstrated in Figure 1.5. In the figure, response of a limit parameter for a given control input along with the predicted limit and control margins are presented. Typically, estimation of a limit or control margin

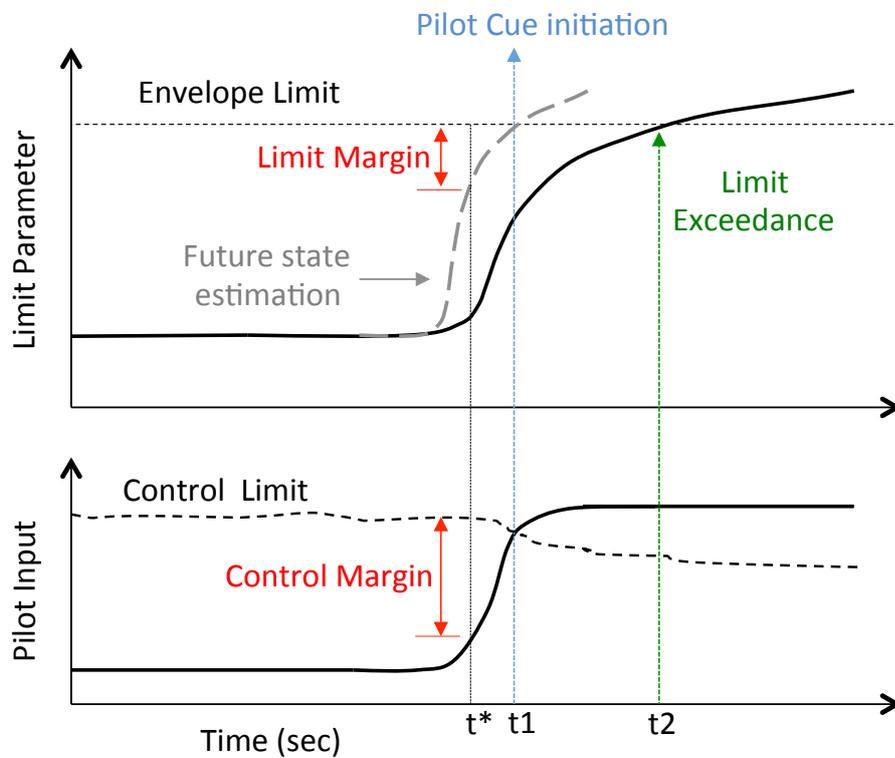


Figure 1.5: Demonstration of Limit Detection

require estimation of a future state, hence require lead time estimations. When a future state is detected to reach the flight envelope boundary, that is when the limit or control margin is zero (at time  $t_1$  in Figure 1.5), the cueing of the pilot can be initiated.

Following the task of limit detection, estimated margins can be used to cue a pilot to avoid approaching envelope boundaries. This task is called *limit avoidance*. For a FBW aircraft limit avoidance can be done through *control limits* (Figure 1.5). Control limits can be thought as the locations of the hard and soft stops on the control axes for envelope cueing. Also, control limits can be considered as the artificial limits on the control axes, which are constructed using the control margin estimation. Estimation of control limits using the control margin estimation will be detailed in the following chapters. When a control margin is detected as zero, that is when a pilot input exceeds an allowable control position, hard stops and soft stops can be given as tactile cues through the active control stick for limit avoidance.

A real implementation of such an envelope cueing system, including the estimation of limit and control margins and their integration to FBW controls, is recently done for CH-53K heavy lift helicopter. A brief summary of the methods applied are published in [12]. In the application, tactile cues are provided to the pilot for the avoidance of power and load factor limits. For the avoidance of power limits, soft stops are provided on the collective axis when the power signal is detected to reach the MCP (Maximum Continuous Power) limit. After the MCP level, collective shaker cues are provided to avoid MAP (Maximum Available Power) limit. Power calculation is made as a function of the torque signal, therefore, an approximate predictive model between the torque response and the collective input is used for the estimation of collective limits. The predictive model is mentioned to be a function fit generated through the flight tests which are done specifically for the generation of the predictive model. For the avoidance of load factor limits, soft stops are applied on the longitudinal axis of the cyclic when the load factor is detected to reach the limit value. After exceeding the load factor limit by a specified tolerance a shaker is triggered to provide further cues. A predictive model which relates the load factor and the

longitudinal cyclic input is used to estimate the longitudinal cyclic limits which are also the soft stop locations. Again, specific flight tests were performed for the generation of such a model. Power limit cues and load factor cues were found to be usable and effective by the pilots at the end of the tests.

To conclude, the estimation of limit and control margin information is a major task for envelope protection. If the aircraft doesn't have a FBW control system, then at least both margins can be displayed in the cockpit visuals to generate visual cues. Whereas, if the aircraft is a FBW aircraft, control margins can be used for effective protection and cueing. Note that the active control systems are typically programmed to avoid a control limit or a zero control margin.

### 1.1.3 Limit Parameters and Response Types

In the area of flight envelope protection limit parameters are classified with respect to their response types [13]. This is because the type of response of a limit parameter ascertains which limit detection and avoidance approach to implement. Reference [9] is an example in which envelope cueing methodologies are implemented according to the response types of limit parameters.

Limit parameter response types are summarized in Figure 1.6. For a given step input, some limit parameters exceed their limits during a transient response. These parameters are known as *peak response critical* limit parameters in the literature. Hub moments, main rotor angular speed and flapping amount of the main rotor blades are some examples of the peak response critical limit parameters in the literature for rotorcraft. Another type of a limit parameter is called *steady state critical* limit parameter. For a given step input, these type of parameters reach their maximum value at their steady state response as shown in Figure 1.6b. In general, the angle of attack and the load factor are some examples for the steady state critical limit parameters of a fixed wing aircraft. Yet, some limit parameters have both steady state critical and peak response critical limits as in Figure 1.6c. In general, power related limits of a helicopter have both type of limits. For instance, the torque and the temperature signals for the turboshaft engines of a helicopter are known to have peak and

steady state critical limits. However, it is important to note that for some limit parameters the peak limit exceedance may not be as critical as the exceedance of the steady state limit. In such a case, the considered limit parameter can be treated as steady state critical only. Another response type, shown in Figure 1.6, is called *integral response critical* type. Typically, the attitude angles such as pitch and bank angles are examples of that response type. In [9], bank angle is used as an integral response critical limit parameter and a related envelope cueing methodology is proposed.

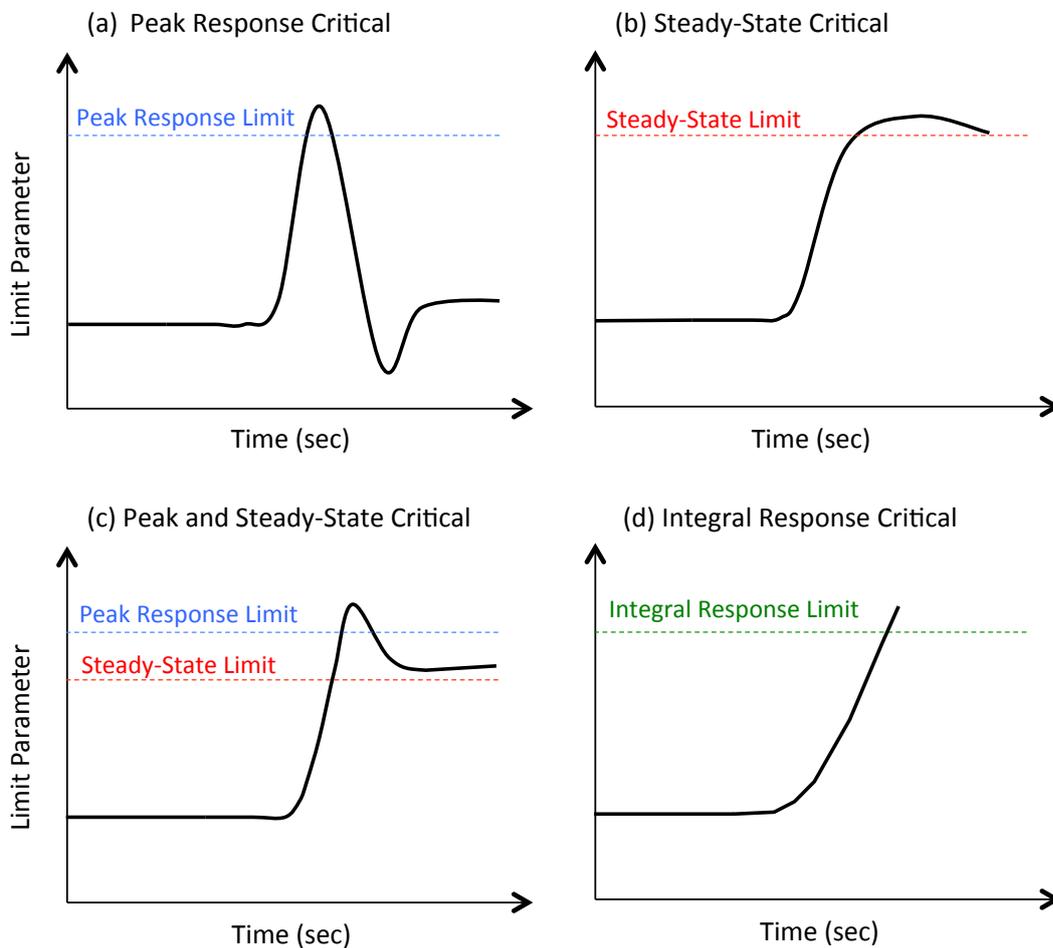


Figure 1.6: Limit Parameter Response Types for a given Step Input

## 1.2 A Survey of Existing Limit Detection Methodologies

Up to this section a variety of methods for flight envelope cueing have been introduced. Those methods can be classified as the ones that use a lead time estimation and the ones that do not use. The methods [2, 5, 6, 7] given in Subsection 1.1.1 are based on the detection of conservative limits, or the manipulation of controller gains. Note that these methods do not take into account the proximity of a future state estimate to the envelope boundaries, hence do not use lead time predictions. Whereas, the studies [8, 9, 10, 11] and [12] given in Subsection 1.1.2 are based on the proximity of a future state estimate to the limits, therefore use lead time estimations for envelope cueing. In this section, existing limit detection methodologies in the literature, which are based on lead time estimations of the limit parameters or controls, are presented.

Envelope protection studies related with lead time estimations begin with [14] and the references therein. Reference [14] is related with RPM cueing in the collective axis for rotorcraft. A linear adaptive element is used for time series prediction. In the approach, the current and the preceding values of the collective input and the current value of the RPM of the main rotor are used to predict the next time step value of the RPM. Constructed time series predictor is inverted to calculate artificial limits on the collective. In the inverted model, the term related with the future time step prediction, i.e. the next time step value of the RPM, is replaced with the RPM limit for control limit prediction. Note that in that study, the prediction horizon of the limit parameter is limited to one time step ahead.

In [15], polynomial neural networks (PNNs) are used to provide the pilot with main rotor hub moment and torque limit cues. In the approach, PNNs are trained offline and used to make online predictions in a fixed time horizon. Offline trained PNNs are augmented online with a bias correction to increase accuracy. Compared to [14], a larger prediction horizon which is between 0.25 to 0.5 seconds is achieved using PNNs. PNNs are also used in [8] for lead time predictions. Reference [8] concerns with the comparison of conventional inceptors and active sidesticks. Conventional inceptors are the ones that have large dis-

placements as the typical helicopter controls and active sidesticks are the short displacement controls which are already mentioned in Subsection 1.1.2. In the comparison study, torque and equivalent retreating indicated tip speed (ERITS) limit cues are provided to the pilot through both type of active controls. Fixed time horizon predictions are obtained with offline trained PNNs. In Figure 1.7 the control margin estimation algorithm of [8] is presented. In the approach, PNNs are corrected with the actual measurements in the low frequency region, i.e. using low pass filters, to increase the prediction accuracy at the steady state. Reference [11] is another example in which similar fixed time horizon prediction methods are implemented for lead time estimations.

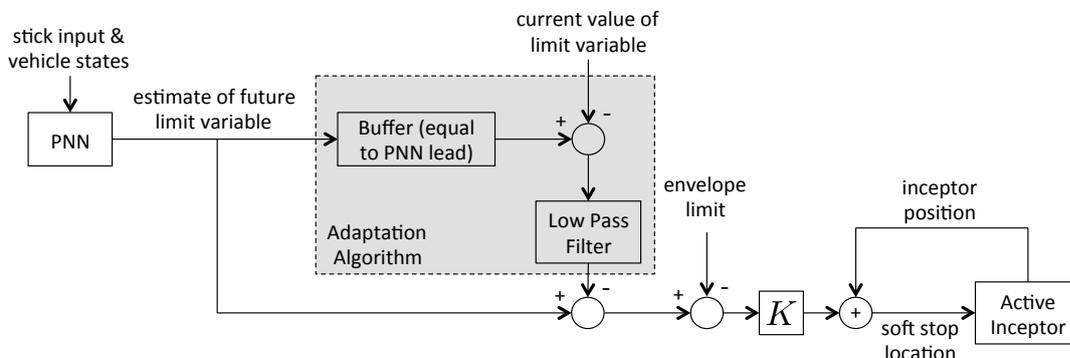


Figure 1.7: Control Margin Estimation using PNNs [8]

Fixed time horizon prediction methods are often limited by neural network's prediction time step [13]. In general, the available lead time of a step response is not guaranteed to be covered by a selected time horizon. This is a concern in an approach based on fixed time horizon prediction. Note that the selection of a time horizon may also depend on the size of the neural network and the dynamics of the considered limit parameter. Therefore, the selection of a correct fixed time horizon for a specific limit parameter may not be straight forward.

Another way to estimate a future response for the aircraft dynamics can be done using a maneuvering steady state condition called *dynamic trim*. Dynamic trim condition, which is analytically the trimming of fast aircraft states [10], is very often used to predict future response of aircraft. Fast aircraft states such as angle of attack, load factor or angular rates reach an equilibrium quickly for a

given pilot control input, whereas the slow states of aircraft such as airspeed may continue to change. Therefore, the estimation of the dynamic trim state of fast aircraft parameters for a given set of control inputs, provide an estimation about the future response and enables the calculation of the proximity to the limit values at the onset of limit violations. In addition, the dynamic trim condition can be used to estimate the allowable control travels which are the maximum control travels that result in the flight envelope limits.

Dynamic trim concept for envelope protection is first used in [10] in which the dynamic trim state and the allowable control travels of a tilt-wing aircraft are estimated using function approximations. Angle of attack and the load factor are taken as the limit parameters. Large datasets are obtained using high fidelity simulation data to represent the dynamic trim state of the limit parameters, and the datasets are fit into a multilayer neural network. In Fig. 1.8 the neural network structure used in [10] is presented. In the approach the dynamic trim data is used to represent a limit parameter as a function of slow states and control inputs.

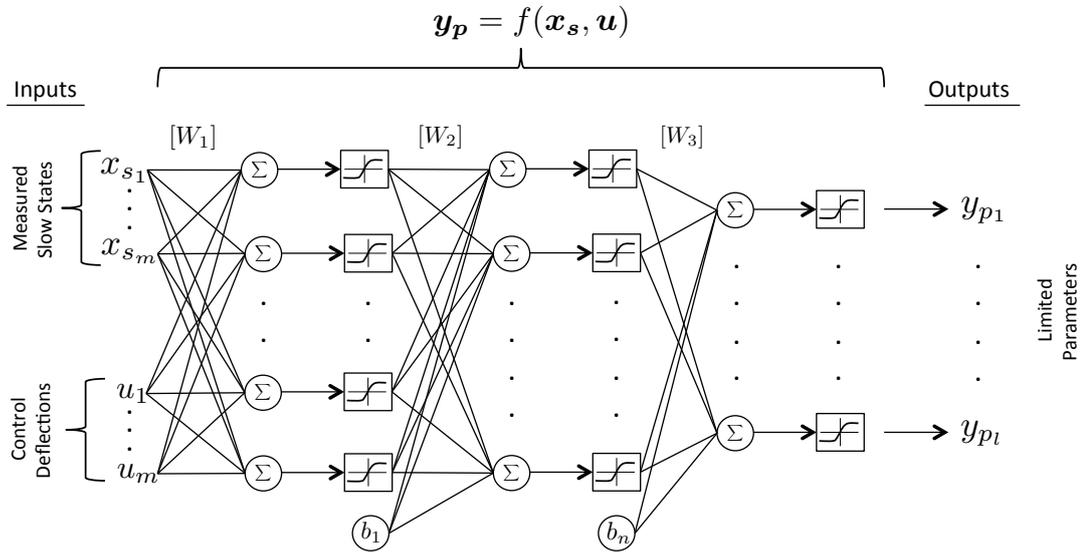


Figure 1.8: Multilayer Neural Network used in [10]

The network is first trained offline with a-priori data and later used in simulations to predict the dynamic trim state of aircraft. In parallel, allowable control travels are estimated online by calculating the sensitivities of the neural network

outputs with respect to the control inputs. Neural network is perturbed by each control input and slow state to obtain the partial derivatives, hence the sensitivities. In [16], the algorithm is improved and named as *Adaptive Dynamic Trim Estimation* by adding an online trained neural network. The network trained online is used to compensate for the error bias between the network approximation and the limit parameter measurement (Fig. 1.9). The online correction for the dynamic trim predictions are made through a low pass filter, hence the low frequency region of the prediction response is corrected.

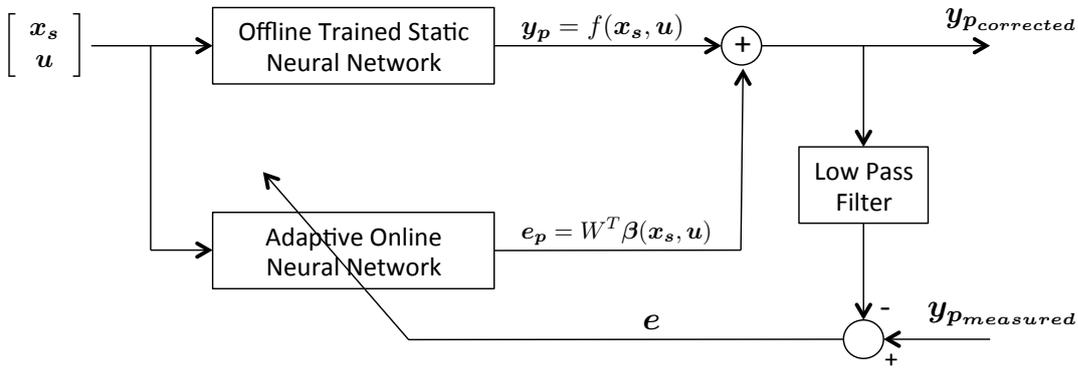


Figure 1.9: Adaptive Dynamic Trim Estimation Algorithm [16]

Note that adaptive dynamic trim estimation algorithm is applicable for the limit parameters that reach their maximum value at the steady state response and is not capable of estimating the peaks that may exist for the transient response critical limit parameters. In [16], a peak response estimation algorithm is developed using linear fast state models that are obtained offline using high fidelity simulation data.

The algorithms developed in [10, 16] are applied for various envelope protection applications for both fixed and rotary wing aircraft. In [17], a comprehensive collective axis cueing system is proposed for the avoidance of engine and transmission limits of rotorcraft. Steady state torque limits are estimated using offline trained neural networks, whereas the transient RPM and transient torque limits are predicted using linear models. Collective cueing for one engine inoperative and autorotation cases are studied. In [18], evaluations of the studies presented in [17] are tested with pilots. The study shows that collective axis tactile cueing

is effective in reducing pilot workload. In [19], main rotor hub moment limits which are often critical in the transient response are studied. Later in [20], the methods are tested with pilots in a high fidelity simulation environment.

Another study which requires a-priori datasets for dynamic trim estimation is [9]. In that study, various tactile cueing experiments are presented for rotorcraft. The lead time estimations of the limiting variables such as hub moments are obtained with offline trained neural networks. Predicted future response is used with linear models to estimate control limits, i.e. soft stop locations. The capabilities of the active controls for the envelope protection task are demonstrated through a variety of examples.

A recent study is [21] in which a technique based on receding horizon control is introduced. In the approach, Linear Parameter Varying (LPV) models of aircraft dynamics are obtained offline using simulation models and later used online to estimate control margins.

The main difficulty in the approaches above is that it is a demanding process to obtain necessary datasets to generate data tables or approximate models that are correct for a large region of actual flight conditions including the boundaries of a flight envelope. Moreover, it is a challenging task to collect dynamic trim data for a large region of flight conditions, various control positions, aircraft mass and balance combinations, etc.

Advances in neural network based adaptive control algorithms [22, 23, 24, 25] and adaptive observer solutions [26] has allowed the development of new methods. In [13] and [27], an adaptive envelope protection approach that requires limited a-priori plant information is introduced for online limit and control margin estimation. Compared to previous studies, the motivation in that approach is to obtain the information required for envelope protection through online adaptive learning even when the datasets representing the dynamic trim condition are not available. In the proposed methodology, considered aircraft dynamics are approximated online using nonlinear dynamic models [27]. Adaptive neural networks are used in the approach to cancel out the modeling errors due to the uncertainties between the actual dynamics and the dynamics approximated by

the local linear models. Limit and control margins are estimated by inverting and iterating the online generated approximate dynamics. Simulations are used for the demonstration of the method in which a linear model of a helicopter and a nonlinear model of the XV-15 tiltrotor are implemented. Dynamic trim estimations are shown to be performed with a minimum amount of information about the aircraft, therefore, the method is found to be an effective method. Large datasets or flight test data representing the dynamic trim condition are shown to be not required. For unmanned aircraft, the method is used to predict allowable commands, i.e. command margins, to the autopilot [28] and tested in real flight. In [29], unmanned aerial vehicles are used to demonstrate automatic limit avoidance through establishing finite-time horizon prediction methods. In that work, the online models developed in [28] and [27] are used.

The methods developed in [13, 27] and [28] are known to be the initial attempts for the implementation of the online parameter estimation algorithms/laws for flight envelope protection. A summary of the online parameter estimation laws that exist in the literature can be found in [30]. Online parameter estimation is a function approximation problem in which a set of parameters of an approximator function are tuned online to fit the actual measurements. The tuned parameters are in general called *weights* and the tuning is called *weight update*. Weights are updated using certain mathematical rules known as *weight update laws* in the literature. For the task of flight envelope protection [13, 27, 28], online weight update laws are used to capture the unmodeled dynamics between approximate models and actual measurements. Although many of the update laws in literature are shown to achieve asymptotic convergence of the approximation error to zero, the convergence of the weights to the optimal values, i.e. to the ideal neural network weights, can not be guaranteed [31]. This is related to a well-known condition called persistency of excitation (PE) in the adaptive element. In general it is required to have persistently exciting signals in the inputs of the adaptive element to achieve convergence to the optimal values. Apparently in practice, it may not be possible to give such inputs, i.e. PE type inputs, to an aircraft online due to safety issues. Therefore, because of the limitations of the weight update laws, the previous online models generated for the task of flight

envelope protection [13, 27, 28] are locally valid online models.

Later, the methods presented in [28] and [27] are improved in [32] by adding a recent neural network weight update law in the adaptation called concurrent learning [33]. Concurrent learning is introduced in [34] to improve convergence properties of the adaptive controllers for the scheme of Model Reference Adaptive Control (MRAC). The goal in concurrent learning is to locate the adaptive weights around a set of ideal values without requiring persistency of excitation in the inputs of the adaptive element. This case is demonstrated by using both instantaneous and online recorded data in the weight update. When both data is used concurrently, adaptation in the long term is shown to be possible without requiring PE type signals. For the problem of flight envelope protection, the tools developed in [34] and [33] are applied in [32] in which online dynamic models that are not only valid at the current state but also valid far from the current state are obtained. A comparison study of using only instantaneous data and using both instantaneous and recorded data in the weight update law is demonstrated through simulations performed for a fixed wing aircraft model. The accuracy of the limit and control margin estimations is shown to increase in the long term [32]. The method is also implemented for the avoidance of the engine limits such as engine outlet temperature and gas generator speed limits of a helicopter [35].

### **1.3 Focus of This Research**

Previously in the area of adaptive flight envelope protection [13, 28, 32, 27], approximate dynamic models of relevant aircraft states and limit parameters are generated online for the estimation of limit and control margins. One of the challenges of these methods is the inversion of the online dynamic model. Online dynamic models are first evaluated at the dynamic trim condition, and second inverted to estimate limit and control margins. Since an adaptive element of an online model can also be a function of limit parameters and aircraft states, as will also be shown in Chapter-II, a direct inversion may not be possible. An option in that case is to use online iterations to find an inverse. Therefore, in [13, 28, 32, 27]

online iterations are proposed for the estimation of the dynamic trim state, hence the limit and control margins. The online iterations of [13, 28, 32, 27] are assumed to give a minimum of one fixed point solution. A drawback related with the fixed point iterations of the previous methodologies is that a fixed point solution obtained from the online iterations is not guaranteed to be the dynamic trim condition, that is, no guarantee of convergence to the dynamic trim state is given using the online iterations of the previous methodologies.

This thesis concerns with the removal of the *fixed point solution assumption* made in the previous references [13, 28, 32, 27] for the problem of adaptive flight envelope protection. Therefore, the focus is given to the development of new methodologies that does not require online iterative solutions for the estimation of limit and control margins.

#### 1.4 Contributions of This Research

In this thesis new methodologies for the removal of the fixed point solution assumption, hence the iterations required in the previous studies, are proposed for adaptive limit and control margin estimation. Developed methodologies are similar to online neural network based methods of [27, 28], yet in the proposed methodologies the limit and control margin estimations are constructed in a different way. In the new methods online adaptation is also improved. A list of overall contributions of the thesis are given below:

- A new methodology, *Direct Adaptive Limit Margin Estimation*, is proposed. In the approach, critical aircraft states are modeled online using adaptive neural networks. Modeling is performed at a delayed moving time step to make use of the state derivatives in the adaptive elements. Also, a set of central difference expressions around the delayed moving time are used to represent the state derivatives in the approximation. The models constructed at the delayed moving time step are evaluated at the current time to estimate the limit margins. Here, the central differences are made zero using the dynamic trim condition, therefore, the dynamic

trim state and the limit margins are obtained. The method is called direct adaptive estimation since the limit margins can be estimated without requiring online iterations. The fixed point solution assumption is shown to be removed.

- A new methodology, *Direct Adaptive Control Margin Estimation*, is proposed. In the approach, instead of modeling the limiting states, the control inputs of interest are modeled online at a delayed moving time step. Similar to the limit margin estimation case, central differences around the delayed moving time is used to represent the derivatives in the adaptive element. Using the dynamic trim condition at the current time step, in which the central differences are zero, control margins are obtained. Fixed point solution assumption and related online iterations are removed for the problem of adaptive control margin estimation.
- In this thesis, concurrent learning neural networks which use a *Minimum Singular Value Maximization* approach to record necessary data for concurrent adaptation are implemented for the proposed direct adaptive limit and control margin estimation methodologies. A proof for the ultimate weight error bounds of the proposed estimators are established. The proof of [34] for the adaptive parameter estimation is extended for the case where the structure of the uncertainty is unknown.
- Sensitivity estimation based control margin estimation methods are revisited in this thesis. Since concurrent learning is used for online adaptation, parameter convergence and a correct estimation of the control sensitivities become possible. Therefore, the sensitivity estimation based methods for envelope protection are improved using concurrent learning and made applicable for the task of control margin estimation.
- Control input saturation is directly used in this thesis for the verification of the critical margins. When a control input is limited with the predicted control limits, the limiting state is expected to be at the envelope boundary. Unlike the previous adaptive methods, control input saturation can be directly applied without using additional smoothing algorithms or logics

at the limit boundary. The problem of chattering at the limit boundary during control input saturation, existing in the previous approaches, is also removed.

- Online adaptation, hence *learn while flying* capability, of adaptive limit and control margin estimation algorithms are improved. Long term learning is made possible using concurrent adaptation.
- In this thesis, central difference expressions are brought into picture, at least once again, for the online modeling and estimation of dynamical systems. In the proposed methodologies, central difference expressions are used in the adaptive element to represent the information of change, i.e. the derivatives, of the related states or limit parameters. Derivatives are not estimated but represented with difference expressions around a delayed moving time.

## 1.5 Structure of The Thesis

This thesis is organized as follows: In Chapter II, an overview of the previous methodologies of [13, 32, 27] is presented. Motivation of this thesis and related issues with the previous approach are mentioned as well. In Chapter III, the *Direct Adaptive Limit Margin Estimation* and *Sensitivity Estimation Based Control Margin Estimation* methodologies are presented. In that chapter, a methodology is developed for limit margin estimation. The developed methodology is used to estimate control sensitivities which are also used to establish functional relations between the limit margins and the control margins. In that chapter, neural network augmentation is explained too. The implementation of a Linear in the Parameter Neural Network (LPNN) for the developed online estimators and the Minimum Singular Value Maximization approach for concurrent learning are presented. In Chapter IV, the *Direct Adaptive Control Margin Estimation* approach is proposed to generate an online estimate of the limiting control input. The estimation model is used at the dynamic trim condition and evaluated at the envelope limits to estimate the control margins.

Note that none of the calculations in the proposed methods require online iterations as in [13, 27, 32] and [28]. In Chapter III and IV, effectiveness of the proposed estimation algorithms are demonstrated using fixed wing and rotary wing aircraft models. In the Appendix, a Lyapunov based proof for the boundedness of the proposed adaptive estimators and an example case which demonstrates the concept of Minimum Singular Value Maximization for concurrent adaptation are presented.



## CHAPTER 2

### ADAPTIVE LIMIT AND CONTROL MARGIN ESTIMATION

In this chapter, adaptive limit and control margin estimation methodologies of the previous works [13, 27, 32, 36, 37, 38] for the task of flight envelope protection, and the reasons behind the motivation of this thesis are presented.

#### 2.1 Problem Definition

Dynamic trim condition and dynamic trim state are directly related with the dynamics of the fast aircraft states or limit parameters. Dynamic trim condition is a quasi-steady condition in which the fast states are in their steady state while the slow states are free to change [10]. The steady state value of the fast states in this condition are called the dynamic trim state. Therefore, following [13, 27] dynamics of the fast aircraft states are represented with the following nonlinear state equation:

$$\dot{\mathbf{x}}_f = \mathbf{f}_1(\mathbf{x}_f, \mathbf{x}_s, \mathbf{u}), \quad (2.1)$$

in which,  $\mathbf{x}_f \in \mathfrak{R}^l$  is the vector of fast states with a known initial condition,  $\mathbf{x}_{f0}$ . Here,  $\mathbf{x}_s \in \mathfrak{R}^{n-l}$  and  $\mathbf{u} \in \mathfrak{R}^p$  are the known vectors of slow aircraft states and control inputs, respectively. It is assumed that the vector function  $\mathbf{f}_1 : \mathfrak{R}^n \times \mathfrak{R}^p \rightarrow \mathfrak{R}^l$  has continuous functions that satisfy a global Lipschitz condition. The solution  $\mathbf{x}_f(t)$  of Eq. (2.1) is assumed to be bounded and exists in finite time.

Dynamic trim condition is defined as a condition in which the time derivatives

of the fast aircraft states are zero, that is

$$\dot{\mathbf{x}}_f = 0, \quad (2.2)$$

and the dynamic trim state is defined for the states and controls when  $\dot{\mathbf{x}}_f = 0$  is enforced in Eq. (2.1) as

$$0 = \mathbf{f}_1(\mathbf{x}_{f_{DT}}, \mathbf{x}_s, \mathbf{u}_{DT}). \quad (2.3)$$

Here,  $\mathbf{x}_{f_{DT}}$  and  $\mathbf{u}_{DT}$  are respectively the dynamic trim state values of the fast states and the control inputs.

The main goal hereafter is the online estimation of the vectors  $\mathbf{x}_{f_{DT}}$  and  $\mathbf{u}_{DT}$  that satisfies the dynamic trim condition,  $\dot{\mathbf{x}}_f = 0$ . The methodologies given in [13, 28, 32, 27] have been proposed to solve this problem. When  $\mathbf{x}_{f_{DT}}$  and  $\mathbf{u}_{DT}$  are estimated, the limit and the control margins can be calculated as will be shown in the following section.

## 2.2 Adaptive Modeling Loops

In [13, 27], actual dynamics of the fast aircraft states, represented by Eq. (2.1), is approximated online using adaptive modeling loops. Below is a brief explanation of the methodology developed in [13, 27]. If we let  $\hat{\mathbf{f}}_1$  be an approximation of  $\mathbf{f}_1$ , then the actual dynamics of the fast states can be written as

$$\dot{\mathbf{x}}_f = \hat{\mathbf{f}}_1(\mathbf{x}_f, \mathbf{x}_s, \mathbf{u}) + \boldsymbol{\xi}. \quad (2.4)$$

Here,  $\boldsymbol{\xi}$  is the modeling error in the fast state dynamics. Using an adaptive element  $\boldsymbol{\Delta}$  in order to cancel out the modeling errors, an approximate dynamic model of the following form is established in [13, 27] as:

$$\dot{\hat{\mathbf{x}}}_f = \hat{\mathbf{f}}_1(\mathbf{x}_f, \mathbf{x}_s, \mathbf{u}) + \boldsymbol{\Delta}(\mathbf{x}_f, \mathbf{x}_s, \mathbf{u}) + K(\mathbf{x}_f - \hat{\mathbf{x}}_f). \quad (2.5)$$

Above,  $\boldsymbol{\Delta}$  is an adaptive element such as neural networks and often written as a function of aircraft states and controls. By subtracting Eq. (2.5) from Eq. (2.4) and noting that  $\mathbf{e} = \mathbf{x}_f - \hat{\mathbf{x}}_f$ , the following error dynamics is obtained:

$$\dot{\mathbf{e}} = -K\mathbf{e} + \boldsymbol{\xi} - \boldsymbol{\Delta} \quad (2.6)$$

Thus, if the modeling error,  $\xi$ , can be ideally cancelled through  $\Delta$ , with a positive definite matrix  $K$ , any estimation error of the actual plant dynamics will decay asymptotically to zero. Neural networks can be used to generate the signal  $\Delta$ . Also note that the error gain matrix,  $K$  in Eq. (2.5), is used to obtain a stable error dynamics in Eq. (2.6). Block diagram representation of the online model given by Eq. (2.5) is presented in Figure 2.1.

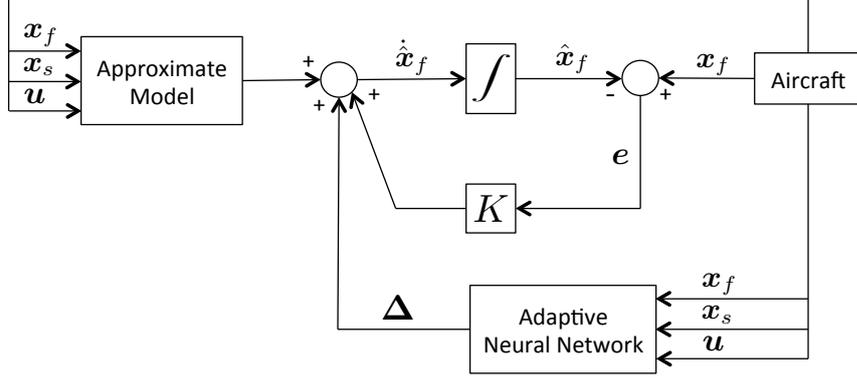


Figure 2.1: Adaptive Online Loop [13, 27]

In [13], two different neural network types such as *Linear in the Parameter* and *Single Hidden Layer* neural networks are used in the adaptive loop of Figure 2.1 to cancel out the modeling errors. Various theorems and proofs related with the stability of the modeling loops using both neural network types are provided in [13]. Ultimate bounds on the weight error estimation and related bounds on the system signals are also demonstrated through a Lyapunov based stability approach. Reader should refer to [13] and [27] for further details.

### 2.3 Limit and Control Margin Estimation

Previously, Eq. (2.5) is used to obtain the dynamic trim state for the task of adaptive envelope protection. Using the dynamic trim condition, Eq. (2.5) takes the following form:

$$0 = \hat{f}_1(\mathbf{x}_{f_{DT}}, \mathbf{x}_s, \mathbf{u}_{DT}) + \Delta(\mathbf{x}_{f_{DT}}, \mathbf{x}_s, \mathbf{u}_{DT}) + K(\mathbf{x}_{f_{DT}} - \hat{\mathbf{x}}_{f_{DT}}). \quad (2.7)$$

Here,  $\mathbf{x}_{f_{DT}}$  and  $\mathbf{u}_{DT}$  are a set of vectors which satisfy the above algebraic equation. Note that both vectors are the unknowns. When  $K(\mathbf{x}_{f_{DT}} - \hat{\mathbf{x}}_{f_{DT}})$  is approximated with  $K(\mathbf{x}_f - \hat{\mathbf{x}}_f) = K(\mathbf{e})$ , then Eq. (2.7) is reduced to the following form:

$$0 = \hat{\mathbf{f}}_1(\mathbf{x}_{f_{DT}}, \mathbf{x}_s, \mathbf{u}_{DT}) + \Delta(\mathbf{x}_{f_{DT}}, \mathbf{x}_s, \mathbf{u}_{DT}) + K(\mathbf{e}). \quad (2.8)$$

Since the goal is to estimate the dynamic trim state for a given set of inputs and slow states,  $\mathbf{u}_{DT}$  can be replaced with the current value of the input vector,  $\mathbf{u}$ . Hence, the following equation is obtained:

$$0 = \hat{\mathbf{f}}_1(\hat{\mathbf{x}}_{f_{DT}}, \mathbf{x}_s, \mathbf{u}) + \Delta(\hat{\mathbf{x}}_{f_{DT}}, \mathbf{x}_s, \mathbf{u}) + K(\mathbf{e}). \quad (2.9)$$

In the above equation,  $\hat{\mathbf{x}}_{f_{DT}}$  is an estimate of the dynamic trim state of the aircraft for a given set of control inputs and slow states. If  $\hat{\mathbf{x}}_{f_{DT}}$  can be calculated using Eq. (2.9) then an estimation of the limit margin vector can be calculated as follows:

$$\hat{\mathbf{x}}_{f_{marg}} = \hat{\mathbf{x}}_{f_{lim}} - \hat{\mathbf{x}}_{f_{DT}}, \quad (2.10)$$

in which  $\hat{\mathbf{x}}_{f_{lim}}$  is a known limiting fast state vector.

Eq. (2.8) can also be used to estimate control margins. In [13, 27],  $\hat{\mathbf{x}}_{f_{DT}}$  of Eq. (2.8) is replaced with  $\hat{\mathbf{x}}_{f_{DTlim}}$ , which is a known limiting fast state vector corresponding to a point on the boundary of the flight envelope, and  $\mathbf{u}$  is replaced with  $\hat{\mathbf{u}}_{DT}$ , which is an estimate of the control limit corresponding to that particular limiting state, hence the following algebraic equation,

$$0 = \hat{\mathbf{f}}_1(\hat{\mathbf{x}}_{f_{DTlim}}, \mathbf{x}_s, \hat{\mathbf{u}}_{DT}) + \Delta(\hat{\mathbf{x}}_{f_{DTlim}}, \mathbf{x}_s, \hat{\mathbf{u}}_{DT}) + K(\mathbf{e}), \quad (2.11)$$

is obtained. Note that the control position,  $\hat{\mathbf{u}}_{DT}$ , is the unknown variable in the above equation. If  $\hat{\mathbf{u}}_{DT}$  can be calculated using Eq. (2.11) then an estimation to control margin vector can be calculated as follows:

$$\hat{\mathbf{u}}_{marg} = \hat{\mathbf{u}}_{DT} - \mathbf{u}, \quad (2.12)$$

in which  $\mathbf{u}$  is the current control input vector.

Note that Eqs. (2.9) and (2.11) are in general nonlinear algebraic equations and may not be solvable analytically. Therefore, in order to solve the dynamic trim state,  $\hat{\mathbf{x}}_{f_{DT}}$ , and the critical control position that would result in the envelope

limits,  $\hat{\mathbf{u}}_{DT}$ , various iterative schemes have been introduced in [13]. One type of iterations has been obtained when the approximate model  $\hat{\mathbf{f}}_1$  is taken as a local linear model such as

$$\hat{\mathbf{f}}_1(\mathbf{x}_f, \mathbf{x}_s, \mathbf{u}) = A_1\mathbf{x}_f + A_2\mathbf{x}_s + B\mathbf{u}. \quad (2.13)$$

Using the linear approximation above, Eqs. (2.9) and (2.11) can be written respectively as

$$0 = A_1\hat{\mathbf{x}}_{f_{DT}} + A_2\mathbf{x}_s + B\mathbf{u} + \Delta(\hat{\mathbf{x}}_{f_{DT}}, \mathbf{x}_s, \mathbf{u}) + K(\mathbf{e}), \quad (2.14)$$

$$0 = A_1\hat{\mathbf{x}}_{f_{DT_{lim}}} + A_2\mathbf{x}_s + B\hat{\mathbf{u}}_{DT} + \Delta(\hat{\mathbf{x}}_{f_{DT_{lim}}}, \mathbf{x}_s, \hat{\mathbf{u}}_{DT}) + K(\mathbf{e}). \quad (2.15)$$

In order to obtain a solution for Eqs. (2.14) and (2.15), the following fixed point iterations are established in [13, 27] as

$$\hat{\mathbf{x}}_{f_{DT_{k+1}}} = -\frac{1}{A_1}(A_2\mathbf{x}_s + B\mathbf{u} + \Delta(\hat{\mathbf{x}}_{f_{DT_k}}, \mathbf{x}_s, \mathbf{u}) + K\mathbf{e}), \quad (2.16)$$

$$\hat{\mathbf{u}}_{DT_{k+1}} = -\frac{1}{B}(A_1\hat{\mathbf{x}}_{f_{DT_{lim}}} + A_2\mathbf{x}_s + \Delta(\hat{\mathbf{x}}_{f_{DT_{lim}}}, \mathbf{x}_s, \hat{\mathbf{u}}_{DT_{k+1}}) + K\mathbf{e}). \quad (2.17)$$

The above iterations are called fixed point iterations since all the variables including the neural network weights and the error bias,  $K\mathbf{e}$ , are kept constant, i.e. fixed, during the iteration process. The variables iterated are only the unknowns,  $\hat{\mathbf{x}}_{f_{DT}}$  and  $\hat{\mathbf{u}}_{DT}$ . Note that the online iterations, Eqs. (2.16) and (2.17), have to be performed at each time instant to estimate the critical margins. Therefore, once a convergence is achieved with an error tolerance at a time instant, the limit and control margins of that time instant can be estimated using Eqs. (2.10) and (2.12).

It is important to note that Eqs. (2.16) and (2.17) are assumed to give a minimum of one fixed point solution previously [13, 27, 28]. This is a valid theoretical assumption since a solution is always guaranteed to be obtained when bounded activation functions are used in the inputs of the adaptive element [24]. Note that, when the iterations are not convergent, the signals are at least guaranteed to be saturated at the bounds, which is also a fixed point solution. Therefore, the bounds of the activation functions have been used in the previous studies in order to guarantee the theoretical existence of solutions for the fixed point iterations.

## 2.4 Validation Studies

The methodology presented above including Subsections 2.2 and 2.3 have been applied in a variety of simulation studies [36, 37, 38, 35, 39] for the avoidance of several critical envelope limits of fixed and rotary wing aircraft. Whereas, the implementation of the methodology for a real flight is performed in [27]. In that work, command margins of a closed loop controller are estimated for not to violate a set of pre-specified envelope limits for an autonomous unmanned aerial vehicle. Command margins are often defined as the controller commands that would result in envelope limits. Note that command margin estimation for a closed loop controller is similar to the control margin estimation of Subsections 2.2 and 2.3. A main difference is that the control input  $\mathbf{u}$  is replaced with a command limit vector in the adaptive elements and in the linear approximations of equations between (2.1) and (2.17). Reader should refer to [27] for further details. To briefly conclude, online iterative equations similar to Eqs. (2.14) and (2.15) are obtained in [27] and related fixed point iterations, i.e. Eqs. (2.16) and (2.17), are applied online. The effectiveness of the adaptive modeling loop (Fig. 2.1) and the fixed point iterations similar to Eqs. (2.16) and (2.17) have been demonstrated through a set of aggressive maneuvers in real flight.

Although the methodology of Subsections 2.2 and 2.3 have been implemented for a considerable amount of simulation studies and validated in real time for an unmanned aerial vehicle with promising results, an improvement in the approach is still required due to the existence of online fixed point iterations and related implementation difficulties. Before moving on to the implementation difficulties of the method, a background of the fixed point solution assumption of the related works in the literature is summarized in the following subsection.

## 2.5 Fixed-Point Solution Assumption in Literature

Fixed point solution assumption of adaptive loops including neural networks have also been an issue for neural network based adaptive flight control [22, 23, 24, 33, 34, 40, 41] since 1997 and before. In the adaptive control problems of [22,

23, 24, 40] and the references therein, neural networks are typically used to cancel out reference tracking errors posed by a-priori designed stable linear controllers. In those studies, adaptive control architectures similar to the adaptive output feedback controller of [24] are used with minor modifications. In Figure 2.2, the design of [24] is presented. In the considered output feedback controller design, the neural network is made a function of its output through a loop,  $L$ , as roughly sketched in Figure 2.2. Hence, an algebraic loop between the output,  $v_{ad}$ , and the input,  $v$ , of the adaptive element is created intentionally by design. Please refer to [24] for the details of the adaptive controller algorithm given in that figure. Note that, in order for the adaptive element output to exist theoretically, the considered algebraic loop,  $L$ , is required to have a fixed point solution [24]. Therefore, the fixed point solution assumptions of [22, 23, 24, 40] are made for the theoretical existence of the neural network output. In [40], at least one fixed point solution is shown to be guaranteed with the use of bounded activation functions in the inputs of the adaptive element.

Note that the fixed point solution assumption made previously in the area of neural network based adaptive control is only a theoretical assumption that guarantees the existence of the output,  $v_{ad}$ , of the adaptive element. In addition to that, the fixed point iterations are not required in practice and have never been used online in adaptive control designs. Related algebraic loops in such adaptive control studies are simply avoided using filters or delay operators in the input signal,  $v$ , as done in [41].

If we return to the problem of online limit and control margin estimation of

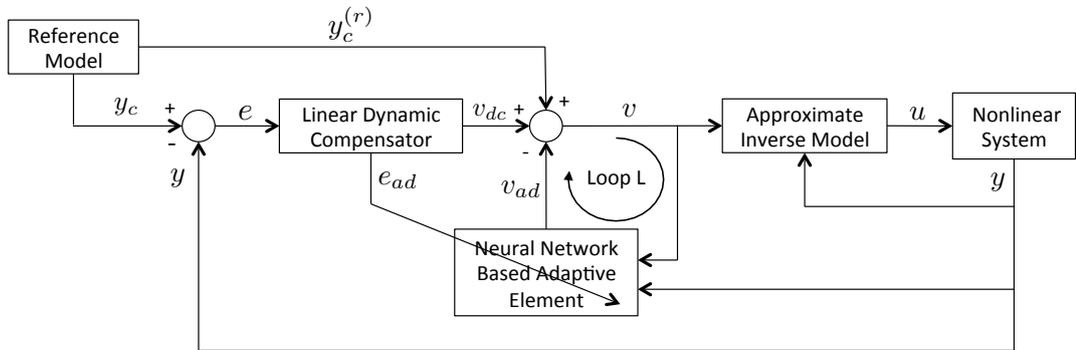


Figure 2.2: Adaptive Output Feedback Control Architecture of [24]

Subsections 2.2 and 2.3, it is clear that the fixed point solution assumption and a resulting fixed point solution are not only required for the theoretical reasons but also required for the estimation of limit and control margins. Therefore, the fixed point iterations, i.e. Eqs. (2.16) and (2.17), used in the methodology of this chapter are required to converge to a solution at each time instant. Whereas in the area of neural network based adaptive control [22, 40, 23, 24, 41, 33, 34], an online iteration is not necessary.

## 2.6 Issues with Implementation

As it is already mentioned, the previous method is based on online generation of approximate aircraft dynamics and that approximation is Eq. (2.5). One issue of the previous method is related with the number of required adaptive loops for the online approximation. Aircraft states are in general coupled to each other, hence, there exists dynamic couplings inherent in the state equations. Therefore, one fast aircraft state is often treated as functions of other fast aircraft states for a physically correct approximation. In the following form of Eq. (2.5), an online approximation for a fast aircraft state is presented:

$$\hat{x}_f = \hat{f}_1(\mathbf{x}_f, \mathbf{x}_s, \mathbf{u}) + \Delta(\mathbf{x}_f, \mathbf{x}_s, \mathbf{u}) + K(x_f - \hat{x}_f), \quad (2.18)$$

in which, a single state,  $\hat{x}_f$ , is written as a function of the fast state vector,  $\mathbf{x}_f$ . Note that, Eq. (2.18) is an adaptive loop for the online modeling of a single fast state. Here, that adaptive loop is typically not sufficient to estimate the dynamic trim state since Eq. (2.18) becomes a single algebraic equation with multiple unknowns at the dynamic trim condition. For instance, at the dynamic trim condition Eq. (2.18) takes the following form:

$$0 = \hat{f}_1(\mathbf{x}_{f_{DT}}, \mathbf{x}_s, \mathbf{u}) + \Delta(\mathbf{x}_{f_{DT}}, \mathbf{x}_s, \mathbf{u}) + Ke, \quad (2.19)$$

in which,  $\mathbf{x}_{f_{DT}}$  is the unknown fast state vector and contains the dynamic trim values of the fast aircraft states used in the approximation. Note that, in order to solve for the unknown,  $\mathbf{x}_{f_{DT}}$ , as many algebraic equations, that is, as many online adaptive loops as the size of the vector  $\mathbf{x}_{f_{DT}}$  is required and that size of

the fast state vector can be chosen if the number of dynamic couplings between the fast states is known or estimated a-priori.

As described in the above text, multiple adaptive loops are typically required in the previous adaptive framework for the online estimation of the dynamic trim state. The use of multiple adaptive loops brings additional complexity into the approach since the design parameters to deal with increases with the number of required adaptive loops. Note that for each additional adaptive loop, the structure of the network, basis functions, scaling factors and learning gains have to be reselected by the designer. Also note that, those design parameters are often specific for a single fast state, thus, a design effort is typically required for each considered state. Therefore, the minimization of the number of required adaptive loops for the dynamic trim estimation is a motivation in this thesis to improve the existing adaptive flight envelope protection methods. In the following chapters, new adaptive methodologies are introduced and the dynamic trim condition of a single fast state is shown to be estimated using only one adaptive loop in the approach, i.e. without requiring the dynamic trim state estimates of other fast states. Therefore, the necessity of multiple adaptive loops of the previous adaptive framework is alleviated using the new methodologies proposed in the following chapters.

Another issue of the previous adaptive framework is related with the convergence of the fixed point iterations. There are multiple adaptive loops and therefore multiple algebraic equations to iterate in the approach. Although the methodology presented in this chapter is guaranteed to give an estimation for the limit and control margins through online fixed point iterations, the iterations are not guaranteed to convergence to the dynamic trim state, i.e. there is no theoretical guarantee for the convergence. Note that, the method is expected to give an accurate result only in practice.

The convergence issue of the fixed point iterations is related with the adaptation and learning properties of the adaptive elements. For an accurate estimation of the dynamic trim state, the adaptive elements of the multiple loops are required to be physically correct models. Therefore, the weights of the adaptive elements

are required to converge to the optimal sets of weights during online adaptation. Note also that, when the optimal sets of values are non-unique, the weights may typically oscillate between different optimal solutions, indicating that the bounds of the weight errors are relatively large. Therefore, the considered fixed point iterations can typically converge to solutions different than the dynamic trim state, especially when the ideal sets of weights in the adaptive elements are non-unique, that is, when the weights oscillate in large bounds and exchange their locations during online adaptation. Note also that, since there are multiple adaptive loops, the oscillations can occur in each adaptive element of the online model, therefore, the convergence to the dynamic trim state can be even difficult in multiple algebraic equations. Also, the nonlinearity of the adaptive elements (nonlinearity of the basis functions used) can increase the complexity in the convergence. In order to overcome those problems in practice, additional terms, i.e. relaxation terms, have been used in the fixed point iterations of the previous works. Yet, none of the introduced terms established a guarantee in theory for the convergence to the dynamic trim state. Therefore, the removal of the fixed point iterations from the existing adaptive flight envelope protection methods is the basic motivation of this thesis. In the following chapters, new methodologies are proposed to remove the fixed point iterations, therefore to alleviate all the issues mentioned above.

Another issue is related with the control margin estimation. Note that, Eq. (2.17) requires the number of pilot controls be equal to the size of the fast state vector in order to make the control matrix,  $B$ , invertible. When the number of inputs are less than the number of fast states, estimation of the control limit typically become non-unique, hence additional algorithms is often required and have been used in the previous works to obtain a correct solution. Therefore, the inversion of the control matrix is another issue in the approach.

## 2.7 Conclusions

In this chapter, a brief summary of the previous adaptive approach for flight envelope protection and the fixed point solution assumption in which the theory

is built on and an analogy, between the considered flight envelope protection method and the previous adaptive control designs, are presented.

Although the methodology presented in this chapter is guaranteed to give an estimation for the limit and control margins through online fixed point iterations, the iterations are not guaranteed to convergence to a dynamic trim state in theory. Also considering the issues mentioned in the previous subsection, it is essential to improve the adaptive envelope protection algorithms and seek for new methodologies for adaptive estimation of limit and control margins.



## CHAPTER 3

# DIRECT ADAPTIVE LIMIT MARGIN AND SENSITIVITY ESTIMATION BASED CONTROL MARGIN ESTIMATION

In this chapter, an adaptive limit margin estimation methodology, that doesn't require online fixed point iterations, is developed. The proposed online limit margin estimator is used to obtain control margins through control sensitivity estimations. Effectiveness of the proposed algorithm is demonstrated through fixed wing and rotary wing aircraft simulation models.

### 3.1 Methodology

Let the following nonlinear state equations represent the aircraft dynamics:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}); \quad \mathbf{x}(t_0) = \mathbf{x}_0; \quad (3.1)$$

where  $\mathbf{x} \in \mathfrak{R}^n$  is the state vector with known initial condition,  $\mathbf{x}_0$ , and  $\mathbf{u} \in \mathfrak{R}^p$  is a known control input vector. It is assumed that the vector function  $\mathbf{f} : \mathfrak{R}^n \times \mathfrak{R}^p \rightarrow \mathfrak{R}^n$  has continuous functions that satisfy a global Lipschitz condition. The solution  $\mathbf{x}(t)$  of Eq. (3.1) is assumed to be bounded and exists in finite time.

Similar to [27], the state vector  $\mathbf{x}$  can be divided into fast and slow states as

$$\mathbf{x} = [\mathbf{x}_f \quad \mathbf{x}_s]^T, \quad \mathbf{x}_f \in \mathfrak{R}^l, \mathbf{x}_s \in \mathfrak{R}^{n-l} \quad (3.2)$$

where, the actual fast and slow state dynamics are represented by

$$\dot{\mathbf{x}}_f = \mathbf{f}_1(\mathbf{x}_f, \mathbf{x}_s, \mathbf{u}), \quad (3.3)$$

$$\dot{\mathbf{x}}_s = \mathbf{f}_2(\mathbf{x}_f, \mathbf{x}_s, \mathbf{u}). \quad (3.4)$$

At the dynamic trim condition, fast aircraft states reach a steady state faster than the slow states [27]. The dynamic trim condition is defined as

$$\dot{\mathbf{x}}_f = 0. \quad (3.5)$$

Assumption (3.1): Considered aircraft dynamics given by Eq. (3.1) is a stable plant.

### 3.1.1 Direct Adaptive Limit Margin Estimation

Fast aircraft states are of significant importance in envelope protection since they exhibit a relatively quick response to pilot inputs. Angle of attack, load factor and angular velocities are a few examples of fast aircraft states. The relation of the fast aircraft states to the envelope limits are often considered separately for the task of flight envelope protection. Using Eq. (3.3), an actual representation of the fast states is written as

$$\mathbf{x}_f = \mathbf{f}_1^{-1}(\dot{\mathbf{x}}_f, \mathbf{x}_s, \mathbf{u}). \quad (3.6)$$

Here,  $\mathbf{f}_1^{-1}$  is obtained inverting the actual plant dynamics. Since the actual plant,  $\mathbf{f}_1$ , of Eq. (3.6) is not usually available or invertible, an approximate inverted model is required. Let  $\hat{\mathbf{f}}_1^{-1}$  be an approximate inverted model, then the actual fast states is written as a summation of the approximate inversion and the modeling error,  $\boldsymbol{\xi}$ :

$$\mathbf{x}_f = \hat{\mathbf{f}}_1^{-1}(\dot{\mathbf{x}}_f, \mathbf{x}_s, \mathbf{u}) + \boldsymbol{\xi}. \quad (3.7)$$

$\boldsymbol{\xi}$  can be compensated using adaptive elements such as neural networks. Using neural networks, the following approximation for  $\mathbf{x}_f$  can be established:

$$\hat{\mathbf{x}}_f = \hat{\mathbf{f}}_1^{-1}(\dot{\mathbf{x}}_f, \mathbf{x}_s, \mathbf{u}) + \boldsymbol{\Delta}(\dot{\mathbf{x}}_f, \mathbf{x}_s, \mathbf{u}). \quad (3.8)$$

$\Delta$  is a vector function containing neural networks and  $\hat{\boldsymbol{x}}_f$  is the estimated signal.  $\Delta$  is trained online to minimize the model tracking error. Subtracting Eq. (3.8) from (3.7) the model tracking error is obtained as

$$\boldsymbol{e} = \boldsymbol{\xi} - \Delta(\hat{\boldsymbol{x}}_f, \boldsymbol{x}_s, \boldsymbol{u}). \quad (3.9)$$

Thus, if  $\boldsymbol{\xi}$  can be ideally cancelled by  $\Delta$ , then the model tracking,  $\boldsymbol{e}$ , will be zero. Although neural networks are universal approximators, depending on the network structure there exists an estimation error between  $\boldsymbol{\xi}$  and  $\Delta$ .

Assumption (3.2): Fast states, slow states and control inputs, that is the signals  $\boldsymbol{x}_f$ ,  $\boldsymbol{x}_s$  and  $\boldsymbol{u}$ , are known, i.e measured, and sampled with the same sampling rate.

Assumption (3.3): Sampling rate of the signals,  $\boldsymbol{x}_f$ ,  $\boldsymbol{x}_s$  and  $\boldsymbol{u}$ , is fixed and chosen as sufficient as to approximate the considered aircraft dynamics.

Assumption (3.4): A moving time window, i.e. delay line, containing the previous consecutive samples of the signals  $\boldsymbol{x}_f$ ,  $\boldsymbol{x}_s$  and  $\boldsymbol{u}$ , is available at the time of current sampling. Therefore, the following matrix is constructed as the delay line matrix:

$$DL = \begin{bmatrix} \boldsymbol{x}_{f_{t_c-1}} & \boldsymbol{x}_{f_{t_c-2}} & \cdots & \boldsymbol{x}_{f_{t_c-w}} \\ \boldsymbol{x}_{s_{t_c-1}} & \boldsymbol{x}_{s_{t_c-2}} & \cdots & \boldsymbol{x}_{s_{t_c-w}} \\ \boldsymbol{u}_{t_c-1} & \boldsymbol{u}_{t_c-2} & \cdots & \boldsymbol{u}_{t_c-w} \end{bmatrix}, \quad (3.10)$$

in which,  $t_c$  is the time index for the current sampling and  $t_c - w$  is the index of the most previous data. Note that,  $DL \in \mathfrak{R}^{(n+p) \times w}$  and  $w \in \mathfrak{N}^+$  is the number of samples in the delay line.

Remark (3.1): A column of  $DL$  in Eq. (3.10) represents a delayed moving time step. Therefore, the data in a considered column of  $DL$  are the values of the variables at that delayed moving time step.

The estimator and the model tracking error in Eqs. (3.8) and (3.9) require  $\hat{\boldsymbol{x}}_f$ . However, measuring  $\hat{\boldsymbol{x}}_f$  might be difficult or not possible at all. When the state measurement is available, it is possible to estimate the state derivatives at a delayed time step using optimal smoothing methods [42] or central difference approximations [43]. Therefore, it is feasible to establish the online model at

a delayed moving time step,  $d$ , and perform the estimations,  $\hat{\mathbf{x}}_{f_d}$ , at that time accordingly. Hence, Eq. (3.8) is re-written at a delayed moving time as

$$\hat{\mathbf{x}}_{f_d} = \hat{\mathbf{f}}_1^{-1}(\dot{\mathbf{x}}_{f_d}, \mathbf{x}_{s_d}, \mathbf{u}_d) + \mathbf{\Delta}(\dot{\mathbf{x}}_{f_d}, \mathbf{x}_{s_d}, \mathbf{u}_d), \quad (3.11)$$

where, the subscript  $d$  denotes the values of the variables at a chosen delayed moving time step. Using a central difference approximation [43], a smoothed estimate of  $\dot{\mathbf{x}}_{f_d}$  can be written as a function of available central differences as follows:

$$\dot{\mathbf{x}}_{f_d} = \mathbf{g}(\mathbf{x}_{f_{d+1}} - \mathbf{x}_{f_{d-1}}, \mathbf{x}_{f_{d+2}} - \mathbf{x}_{f_{d-2}}, \dots, \mathbf{x}_{f_{d+k}} - \mathbf{x}_{f_{d-k}}), \quad (3.12)$$

where,  $k \in \mathbb{N}^+$  is the number of central differences.

Remark (3.2): The signals,  $\mathbf{x}_f$ ,  $\mathbf{x}_s$  and  $\mathbf{u}$ , are the samples obtained at the time of current sampling,  $t_c$ . Therefore, the subscript  $t_c$  is omitted from the variables intentionally.

Remark (3.3): Hereafter, the wording *current time* refers to the *time of current sampling*.

Let  $\bar{\partial}(\cdot)_t : \mathfrak{R}^l \rightarrow \mathfrak{R}^{l \times k}$  be an operator that takes a vector at a time  $t$  and outputs a matrix of central differences, such that;

$$\bar{\partial}(\cdot)_t = [(\cdot)_{t+1} - (\cdot)_{t-1}, (\cdot)_{t+2} - (\cdot)_{t-2}, \dots, (\cdot)_{t+k} - (\cdot)_{t-k}]. \quad (3.13)$$

Using  $\bar{\partial}(\cdot)_t$  at  $t = d$  to represent  $\dot{\mathbf{x}}_{f_d}$ , Eq. (3.11) is re-written as

$$\hat{\mathbf{x}}_{f_d} = \hat{\mathbf{f}}_1^{-1}(\bar{\partial}(\mathbf{x}_f)_d, \mathbf{x}_{s_d}, \mathbf{u}_d) + \mathbf{\Delta}(\bar{\partial}(\mathbf{x}_f)_d, \mathbf{x}_{s_d}, \mathbf{u}_d). \quad (3.14)$$

Note that the estimator is now a function of central differences at a delayed moving time step,  $d$ . The model tracking error may also be written at the delayed moving step using central differences:

$$\mathbf{e}_d = \boldsymbol{\xi}_d - \mathbf{\Delta}(\bar{\partial}(\mathbf{x}_f)_d, \mathbf{x}_{s_d}, \mathbf{u}_d). \quad (3.15)$$

In Eq. (3.14), the approximate inversion,  $\hat{\mathbf{f}}_1^{-1}$ , is augmented with  $\mathbf{\Delta}$  and the resulting operator,  $\hat{\mathbf{f}}_1^{-1} + \mathbf{\Delta}$ , is called the augmented inversion operator.

Assumption (3.5): The operator,  $(\hat{\mathbf{f}}_1^{-1} + \mathbf{\Delta})$ , is a one-to-one mapping such that  $(\hat{\mathbf{f}}_1^{-1} + \mathbf{\Delta}) : \mathfrak{R}^{l \times (k-1) + n + p} \rightarrow \mathfrak{R}^l$  exists and is unique.

Since the derivatives are represented with central differences,  $\bar{\partial}(\mathbf{x}_f)_t$  is expected to be zero at the dynamic trim condition at any given time,  $t$ . Hence, the dynamic trim condition in Eq. (3.5) also implies

$$\bar{\partial}(\mathbf{x}_f)_t = 0. \quad (3.16)$$

In order to estimate the dynamic trim state at the current time, the estimator of Eq. (3.14) is used at the current time step,  $t_c$ . This is applicable when the network weights reach a compact neighborhood around optimal weights or the delayed time is short compared to the aircraft dynamics. As a result, using Eqs. (3.14), (3.15) and (3.16), the fast states in dynamic trim,  $\hat{\mathbf{x}}_{fDT}$ , at the current time can be estimated as follows:

$$\hat{\mathbf{x}}_{fDT} = \hat{\mathbf{f}}_1^{-1}(0, \mathbf{x}_s, \mathbf{u}) + \mathbf{\Delta}(0, \mathbf{x}_s, \mathbf{u}) + \mathbf{e}_d. \quad (3.17)$$

Then the limit margin vector  $\mathbf{x}_{f_{marg}}$  can then be calculated as

$$\mathbf{x}_{f_{marg}} = \mathbf{x}_{f_{lim}} - \hat{\mathbf{x}}_{fDT}. \quad (3.18)$$

Limit margins can be used in various ways in carefree maneuvering. As it may be used in the cueing algorithms, in this chapter, it is used to determine control sensitivities and control margins, see Eqs. (3.38) and (3.39).

The proposed limit margin estimation algorithm is presented in Fig. 3.1. Note that the variables and computations used at the delayed moving time step are shown with the dashed lines, whereas, the parts of the algorithm used at the current time are shown with the solid lines. The delay operator,  $z^{-d}(\cdot)$ , is used to obtain the value of the operator input at the delayed moving time step,  $d$ , such that  $z^{-d}(\mathbf{x}_f) = \mathbf{x}_{f_d}$ . The aircraft block represents the plant, but may also include any control system. If a control system is present the aircraft dynamics of Eq. (3.1) will simply include the closed loop dynamics.

A special case is when the approximate model is chosen as a linear model,  $\dot{\mathbf{x}}_f = A_1\mathbf{x}_f + A_2\mathbf{x}_s + B\mathbf{u}$ . In that case Eq. (3.11) is written as

$$\hat{\mathbf{x}}_{f_d} = -A_1^{-1}(-\dot{\mathbf{x}}_{f_d} + A_2\mathbf{x}_{s_d} + B\mathbf{u}_d) + \mathbf{\Delta}(\dot{\mathbf{x}}_{f_d}, \mathbf{x}_{s_d}, \mathbf{u}_d). \quad (3.19)$$

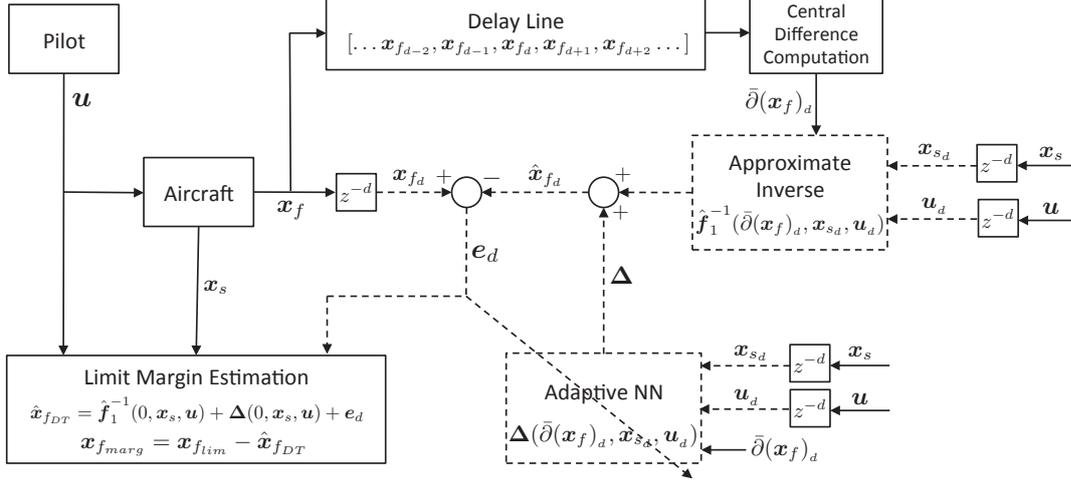


Figure 3.1: Direct Adaptive Limit Margin Estimation

In this equation the input,  $\hat{\mathbf{x}}_{f_d}$ , in  $\Delta$  is replaced with  $\bar{\partial}(\mathbf{x}_f)_d$ . However, the derivative,  $\hat{\mathbf{x}}_{f_d}$ , which is the first term of the inverted the linear model, is approximated with an average sum of central differences. Note that any error resulting from this approximation will be compensated through the neural network. The operator,  $\partial(\cdot)_t = \frac{1}{k} \sum_{j=1}^k \bar{\partial}(\cdot)_t(:, j) : \mathfrak{R}^l \rightarrow \mathfrak{R}^l$ , is used to calculate the average sum:

$$\hat{\mathbf{x}}_{f_d} \cong \partial(\mathbf{x}_f)_d = \frac{1}{k} \sum_{j=1}^k \bar{\partial}(\mathbf{x}_f)_d(:, j). \quad (3.20)$$

Hence Eq. (3.19) is re-written as

$$\hat{\mathbf{x}}_{f_d} = -A_1^{-1}(-\partial(\mathbf{x}_f)_d + A_2 \mathbf{x}_{s_d} + B \mathbf{u}_d) + \Delta(\bar{\partial}(\mathbf{x}_f)_d, \mathbf{x}_{s_d}, \mathbf{u}_d). \quad (3.21)$$

Using Eqs. (3.15), (3.16) and (3.21), the dynamic trim at the current time can be estimated:

$$\hat{\mathbf{x}}_{f_{DT}} = -A_1^{-1}(A_2 \mathbf{x}_s + B \mathbf{u}) + \Delta(0, \mathbf{x}_s, \mathbf{u}) + e_d. \quad (3.22)$$

Note that  $e_d$  and  $\Delta$  are known signals, and Eqs. (3.17) and (3.22) do not require online iterations to find  $\hat{\mathbf{x}}_{f_{DT}}$  as it was the case in [27, 28] and [32]. Therefore this approach is called the *Direct Adaptive Limit Margin Estimation*.

## Relative Degree Formulation

Note that the above methodology is developed assuming that the fast and the slow aircraft states are measured. An alternative methodology can also be developed considering the measurements of a single limit parameter and the slow states. In that case, the measured limit parameter,  $y_p \in \mathfrak{R}$ , can be written as a nonlinear function of aircraft states:

$$y_p = h(\mathbf{x}_f, \mathbf{x}_s). \quad (3.23)$$

Using the definition of relative degree given in [44], the  $n$ -th order derivative of  $y_p$  can be established in the following form:

$$y_p^{(n)} = h_n(\mathbf{x}_f, \mathbf{x}_s, y_p, y_p^{(1)}, y_p^{(2)}, \dots, y_p^{(n-1)}, u_e). \quad (3.24)$$

Assumption (3.6): The limit parameter, slow states and the effective control input, that is the signals  $y_p$ ,  $\mathbf{x}_s$  and  $u_e$ , are known, i.e. measured, at a given time, and sampled with the same sampling rate.

Assumption (3.7): Sampling rate of the signals,  $y_p$ ,  $\mathbf{x}_s$  and  $u_e$ , is fixed and chosen as sufficient as to approximate the considered limit parameter dynamics.

Assumption (3.8): The number of differentiations in Eq. (3.24) is known, that is, the limit parameter has a known relative degree.

In Eq. (3.24),  $u_e \in \mathfrak{R}$  is a single control input or a linear combination of multiple controls obtained after  $n$  differentiations of  $h$ . Furthermore, when the relation in Eq. (3.23) is known and invertible, i.e.  $\mathbf{x}_f = h^{-1}(y_p, \mathbf{x}_s)$ , then Eq. (3.24) can be written as

$$y_p^{(n)} = g_n(\mathbf{x}_s, y_p, y_p^{(1)}, y_p^{(2)}, \dots, y_p^{(n-1)}, u_e). \quad (3.25)$$

Assumption (3.9): The control input  $u_e$  is a smooth and continuous signal.

In Eq. (3.25),  $u_e$  is the control input that results in a flight envelope violation of the limit parameter  $y_p$ . When Eq. (3.25) is inverted,  $y_p$  can be written as

$$y_p = g_n^{-1}(\mathbf{x}_s, y_p^{(1)}, y_p^{(2)}, \dots, y_p^{(n)}, u_e). \quad (3.26)$$

Here,  $g_n^{-1}$  is the actual inverse of the plant dynamics and is an unknown function and does not have to be a one-to-one mapping. Using an approximate model,  $\hat{g}_n$ ,  $y_p$  can be written as a summation of the approximation and a modeling error,  $\xi$ :

$$y_p = \hat{g}_n^{-1}(\mathbf{x}_s, y_p^{(1)}, y_p^{(2)}, \dots, y_p^{(n)}, u_e) + \xi. \quad (3.27)$$

In Eq. (3.27),  $\mathbf{x}_s$  represents slow aircraft states that are required for the approximation. If we let  $\mathbf{y}_p = [y_p \ y_p^{(1)} \ \dots \ y_p^{(n-1)}]^T \in \mathfrak{R}^n$ , then the derivatives in Eq. (3.27) can be written as

$$\dot{\mathbf{y}}_p = [y_p^{(1)} \ y_p^{(2)} \ \dots \ y_p^{(n)}]^T \in \mathfrak{R}^n. \quad (3.28)$$

Now, using an adaptive element,  $\Delta$ , the following estimation to  $y_p$  can be written:

$$\hat{y}_p = \hat{g}_n^{-1}(\mathbf{x}_s, \dot{\mathbf{y}}_p, u_e) + \Delta(\mathbf{x}_s, \dot{\mathbf{y}}_p, u_e). \quad (3.29)$$

Note that, Eq. (3.29) requires the derivative signals  $\dot{\mathbf{y}}_p$  to be available. It is possible to estimate the derivatives from available past data [42]. Therefore, the estimator of Eq. (3.29) can be established at a delayed moving time step,  $d$ , as follows:

$$\hat{y}_{p_d} = \hat{g}_n^{-1}(\mathbf{x}_{s_d}, \dot{\mathbf{y}}_{p_d}, u_{e_d}) + \Delta(\mathbf{x}_{s_d}, \dot{\mathbf{y}}_{p_d}, u_{e_d}) \quad (3.30)$$

where, the subscript,  $d$ , represents the values of the variables at a considered delayed moving time step.

Assumption (3.10): A moving time window, i.e. delay line, containing the previous consecutive samples of the signals  $y_p$ ,  $\mathbf{x}_s$  and  $u_e$ , is available at the time of current sampling. The following matrix is the delay line matrix:

$$DL = \begin{bmatrix} y_{p_{t_c-1}} & y_{p_{t_c-2}} & \dots & y_{p_{t_c-w}} \\ \mathbf{x}_{s_{t_c-1}} & \mathbf{x}_{s_{t_c-2}} & \dots & \mathbf{x}_{s_{t_c-w}} \\ u_{e_{t_c-1}} & u_{e_{t_c-2}} & \dots & u_{e_{t_c-w}} \end{bmatrix}, \quad (3.31)$$

here,  $t_c$  is the time index for the current sampling and  $t_c - w$  is the index of the most previous data. Note that,  $DL \in \mathfrak{R}^{(n-l+2) \times w}$  and  $w \in \mathbb{N}^+$  is the number of samples in the delay line. Also, a chosen column of  $DL$  represents a delayed moving time step, as in *Remark (3.1)*.

Note that the derivatives,  $\dot{\mathbf{y}}_p$ , can be written as a function of available central differences around the delayed moving time step,  $d$ . Using the central difference

expressions of [45], the following central difference operator,  $\bar{\partial}(\cdot)_t : \mathfrak{R} \rightarrow \mathfrak{R}^{n \times k}$ , is defined:

$$\bar{\partial}(\cdot)_t(:, j) = \begin{bmatrix} (\cdot)_{t+j} - (\cdot)_{t-j} \\ (\cdot)_{t+j} - 2(\cdot)_t + (\cdot)_{t-j} \\ (\cdot)_{t+j+1} - 2(\cdot)_{t+j} + 2(\cdot)_{t-j} - (\cdot)_{t-j-1} \\ \dots \end{bmatrix} \quad (3.32)$$

in which,  $j = 1, 2, \dots, k$  represent each column of the operator and  $k \in \mathbb{N}^+$  of Eq. (3.32) is the number of available central differences around the time,  $t$ . Note that the derivatives in Eq. (3.30) can be represented using Eq. (3.32) when  $t = d$  as  $\bar{\partial}(y_p)_d$ . Also, the  $n^{\text{th}}$  row of the operator is the central difference representation of the  $n^{\text{th}}$  derivative of the limit parameter. In [46], central difference representations of higher order derivatives are given.

Therefore, Eq. (3.30) can now be written as a function of central differences,

$$\hat{y}_{p_d} = \hat{g}_n^{-1}(\mathbf{x}_{s_d}, \bar{\partial}(y_p)_d, u_{e_d}) + \Delta(\mathbf{x}_{s_d}, \bar{\partial}(y_p)_d, u_{e_d}). \quad (3.33)$$

Subtracting Eq. (3.33) from  $y_{p_d} = \hat{g}_n^{-1}(\mathbf{x}_{s_d}, \bar{\partial}(y_p)_d, u_{e_d}) + \xi_d$ , the delayed approximation error,  $e_d$ , is obtained as

$$e_d = \xi_d - \Delta(\mathbf{x}_{s_d}, \bar{\partial}(y_p)_d, u_{e_d}). \quad (3.34)$$

Assumption (3.11): In Eq. (3.33), the approximate inversion,  $\hat{g}_n^{-1}$ , is augmented with  $\Delta$ . The resulting operator,  $(\hat{g}_n^{-1} + \Delta)$ , is a one-to-one mapping such that  $(\hat{g}_n^{-1} + \Delta) : \mathfrak{R}^{n(k-l)+1} \rightarrow \mathfrak{R}$  exists and is unique.

At the maneuvering steady state condition, i.e. the dynamic trim condition, the derivatives which are represented with central differences,  $\bar{\partial}(y_p)_d$ , are expected to be zero. Hence, the maneuvering steady state condition,  $\dot{\mathbf{y}}_p = 0$ , also indicates

$$\bar{\partial}(y_p)_t = 0. \quad (3.35)$$

Therefore, using Eq. (3.35), along with Eqs. (3.33) and (3.34), the maneuvering steady state,  $\hat{y}_{p_{DT}}$ , for a given limit,  $y_{p_{lim}}$ , can be estimated at the current time step as

$$\hat{y}_{p_{DT}} = \hat{g}_n^{-1}(\mathbf{x}_s, 0, u_e) + \Delta(\mathbf{x}_s, 0, u_e) + e_d. \quad (3.36)$$

In Eq. (3.36), the delayed approximation error,  $e_d$ , is used at the current time step for the estimation. Note that this is reasonable when the network weights reach a compact neighborhood of ideal weights.

The limit margin,  $y_{p_{\text{marg}}}$ , can be calculated as

$$y_{p_{\text{marg}}} = \hat{y}_{p_{\text{lim}}} - \hat{y}_{p_{DT}}. \quad (3.37)$$

In Fig. 3.2, the block diagram representation of the *Direct Adaptive Limit Margin Estimation* using the *Relative Degree Formulation* is presented. Dashed lines represent the variables or calculations that take place at the considered delayed moving time step.

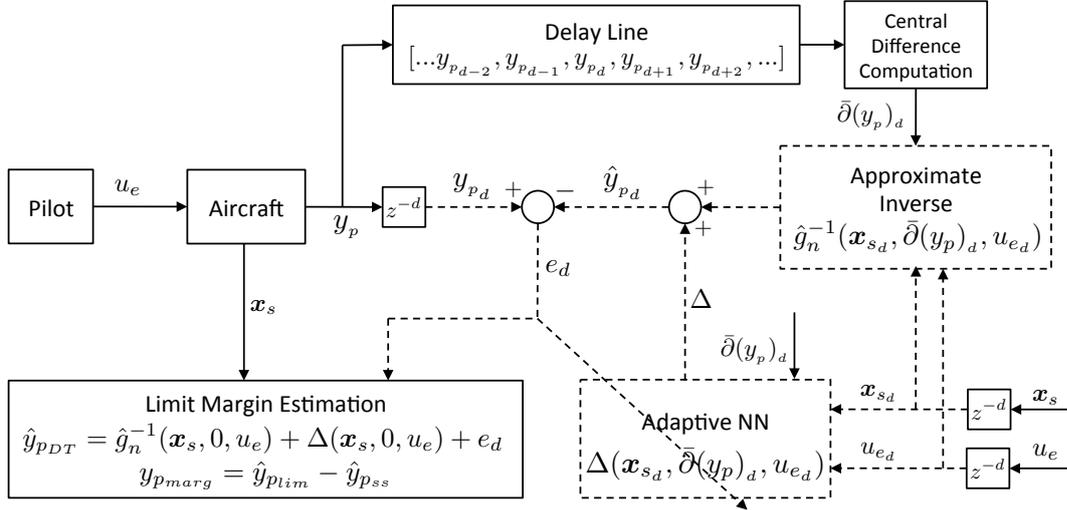


Figure 3.2: Direct Adaptive Limit Margin Estimation with Relative Degree Formulation

### 3.1.2 Sensitivity Estimation Based Control Margin Estimation

Control sensitivities can be used to estimate control margins using the estimator of Eq. (3.17). The sensitivity vector  $\mathbf{S} \in \mathfrak{R}^l$  of fast aircraft states to the effective control input,  $u_e \in \mathfrak{R}$ , can be found by taking the partial derivative of Eq. (3.17) with respect to  $u_e$ :

$$\mathbf{S} = \frac{\partial \hat{\mathbf{x}}_{fDT}}{\partial u_e} = \frac{\partial (\hat{\mathbf{f}}_1^{-1}(0, \mathbf{x}_s, \mathbf{u}) + \Delta(0, \mathbf{x}_s, \mathbf{u}))}{\partial u_e}. \quad (3.38)$$

Now, let the control limits on a single control axis be represented by the vector  $\mathbf{u}_{e_{lim}}$ , then the control margins,  $\mathbf{u}_{e_{marg}}$ , can be calculated by subtracting the control input of interest,  $u_e$ , from the control limits,  $\mathbf{u}_{e_{lim}}$ :

$$\mathbf{u}_{e_{marg}} = \mathbf{u}_{e_{lim}} - \mathbf{I}_v u_e \quad (3.39)$$

where,  $\mathbf{I}_v$  is given by  $\mathbf{I}_v = [1, 1, \dots, 1]^T \in \mathfrak{R}^l$ . The vectors  $\mathbf{u}_{e_{lim}}$  and  $\mathbf{u}_{e_{marg}}$  contain control limits and margins, respectively, for multiple limiting states.

The limit margin vector can be expressed as a function of the control margin vector using the Taylor series expansion around the point  $(\mathbf{x}_{f_{marg}}, \mathbf{u}_{e_{marg}})$  and neglecting the higher order terms:

$$\mathbf{x}_{f_{marg}} = \text{diag}\left(\frac{d\mathbf{x}_{f_{marg}}}{d\mathbf{u}_{e_{marg}}}\right)\mathbf{u}_{e_{marg}}. \quad (3.40)$$

Here, the operator  $\text{diag}(\cdot)$  outputs a matrix with diagonal elements only. Noting that

$\text{diag}\left(\frac{d\mathbf{x}_{f_{marg}}}{d\mathbf{u}_{e_{marg}}}\right) = \text{diag}(\mathbf{S}\mathbf{I}_v^T)$ , Eq. (3.40) takes the following form:

$$\mathbf{x}_{f_{marg}} = \text{diag}(\mathbf{S}\mathbf{I}_v^T)\mathbf{u}_{e_{marg}}. \quad (3.41)$$

Let  $M = \text{diag}(\mathbf{S}\mathbf{I}_v^T)$ , the control margins,  $\mathbf{u}_{e_{marg}}$ , on a single control axis,  $u_e$ , can be found as

$$\mathbf{u}_{e_{marg}} = M^{-1}\mathbf{x}_{f_{marg}}. \quad (3.42)$$

Note that a control margin becomes zero if and only if an associated limit margin becomes zero. Here, the diagonal elements of  $M$  in Eq. (3.42) are required to be nonzero. This is also compatible when the goal is to initiate pilot cueing at a time when a limit margin becomes zero,  $x_{f_{marg}} = 0$ . Therefore, the effectiveness of the control margin estimation is directly related to the limit margin estimation in the approach.

*Remark (3.4):* To avoid any singularity, when a diagonal element of  $M$  is zero, the sign of the approximate sensitivity  $\frac{\partial(\hat{\mathbf{f}}_1^{-1}(0, \mathbf{x}_s, \mathbf{u}))}{\partial u_e}$  of Eq. (3.38) should be selected with the correct sign and the magnitude of the estimated sensitivity for each fast state should have a nonzero lower bound.

Once  $\mathbf{u}_{e_{marg}}$  is calculated from Eq. (3.42), using Eq. (3.39) control limits on a particular control axis can be estimated. Note that the sensitivity vector,  $S$ ,

can be obtained for any control axis. Therefore, the method can be applied to estimate control limits of multiple control axes as well.

One shortcoming of this method is that there may exist large bounds in the neural network weight errors. When the bounds are large the sensitivities can be estimated with large errors. However, the sensitivities can be obtained with sufficient accuracy when the concurrent learning adaptive law is used. Note that, the weights can be located around a unique set of optimal values using concurrent learning [47, 34], therefore the sensitivities can be accurately estimated.

Control margins can also be estimated using the estimator given by Eq. (3.36). The sensitivity  $S \in \Re$  of the limit parameter to the effective control input,  $u_e$ , can be found by taking the derivative of Eq. (3.36) with respect to  $u_e$ :

$$S = \frac{\partial \hat{y}_{pDT}}{\partial u_e} = \frac{\partial(\hat{g}_n^{-1}(\mathbf{x}_s, 0, u_e) + \Delta(\mathbf{x}_s, 0, u_e))}{\partial u_e} \quad (3.43)$$

The control margin,  $u_{emarg}$ , can be calculated by subtracting the control input of interest,  $u_e$ , from the control limit,  $u_{elim}$ :

$$u_{emarg} = u_{elim} - u_e \quad (3.44)$$

The sensitivity,  $S$  can be used to establish a linear relation between the limit margin and the control margin as done similarly in Eq. (3.40):

$$y_{pmarg} = S u_{emarg} \quad (3.45)$$

Then the control limit,  $u_{elim}$ , can be found using Eqs. (3.43),(3.44) and (3.45):

$$u_{elim} = \frac{1}{S} y_{pmarg} + u_e \quad (3.46)$$

*Remark (3.5):* Note that to avoid  $S = 0$ , the sign of the approximate control sensitivity,  $\frac{\partial(\hat{g}_n^{-1}(\mathbf{x}_s, 0, u_e))}{\partial u_e}$ , should be selected with the correct sign and the magnitude of the estimated sensitivity,  $S$ , should have a nonzero lower bound.

Note that Eqs. (3.36), (3.37) and (3.38)-(3.46) do not require a fixed point solution assumption to find the limit and control margins as it was the case in [28], [27] and [32].

## 3.2 Online Model Generation

The main goal of the methodologies proposed above is to establish online models that can represent actual dynamics of the fast states or the limit parameters. For clarity, the online models in consideration are Eqs. (3.14) and (3.33). In order for these models to represent the actual dynamics, the model approximation errors of Eqs. (3.15) and (3.34) should be minimized online as much as possible. Therefore, the adaptive element  $\Delta$  is used for that purpose.

In this section, online implementation of the adaptive element, which is a neural network, is presented. In particular, the neural network structure and the weight update law for the function  $\Delta$ , and the implementation of a learning method, i.e. concurrent learning, are briefly summarized.

Although the online models, Eqs. (3.14) and (3.33), are established at a delayed moving time step, the subscript  $d$  is omitted from the variables and calculations for simplicity in the following subsection. Therefore in the next subsection, the equations and the variables are written as if they are valid at a time  $t$ .

### 3.2.1 Linear in the Parameter Neural Network Structure

For the methodology proposed above, the uncertainties are assumed to be linearly parameterizable. Therefore, a linear structure, i.e. Linear in the Parameter Neural Networks (LPNNs), are proposed to cancel out the modeling errors online.

*Assumption (3.12):* The uncertainty  $\xi$  of Eq. (3.9) is linearly parameterizable with a set of unique optimal weights and a reconstruction error,  $\epsilon$ , as

$$\xi = W^{*T} \phi(\bar{\mathbf{x}}) + \epsilon. \quad (3.47)$$

Here,  $\bar{\mathbf{x}} \in \mathfrak{R}^r$  is the neural network input vector and includes the central differences of the fast states, the slow states and controls, that is  $\bar{\mathbf{x}} = [\bar{\partial}(\mathbf{x}_f)_t, \mathbf{x}_s, \mathbf{u}]^T$ .  $W^* \in \mathfrak{R}^{m \times l}$  represents a unique optimal set of weights and,  $\phi(\bar{\mathbf{x}}) = [\phi_1(\bar{\mathbf{x}}), \phi_2(\bar{\mathbf{x}}), \dots, \phi_m(\bar{\mathbf{x}})]$ , in which  $\phi(\cdot) : \mathfrak{R}^r \rightarrow \mathfrak{R}^m$ . Here,  $\phi_i : \mathfrak{R}^r \rightarrow \mathfrak{R}$ ,  $i = 1, 2, \dots, m$  are known and bounded functions over the domain of approximation.

Since the basis functions are bounded by definition, the upper bound of the norm of the basis vector can be defined using a constant,  $\alpha_0$ , as

$$\|\phi(\bar{\mathbf{x}})\| \leq \alpha_0. \quad (3.48)$$

Since  $\xi$  is assumed to be a linear combination of a known basis, an LPNN with the same basis can be constructed for  $\Delta$ :

$$\Delta(\bar{\mathbf{x}}) = W^T \phi(\bar{\mathbf{x}}). \quad (3.49)$$

Note that, LPNNs are universal approximators [24]. Using Eqs. (3.47) and (3.49), the model tracking error given by Eq. (3.9) takes the following form:

$$\mathbf{e} = \tilde{W}^T \phi(\bar{\mathbf{x}}) + \epsilon \quad (3.50)$$

in which,  $\tilde{W}^T = W^{*T} - W^T$  is the error between the optimal and the approximate weights.

The reconstruction error,  $\epsilon$ , is shown to be bounded using the universal approximation property of LPNNs, that is, for a positive definite  $\epsilon$  and for a compact set  $D$  such that all  $\bar{\mathbf{x}} \in D$ , the largest  $\|\epsilon\|$  can be given as [33]:

$$\bar{\epsilon} = \sup_{\bar{\mathbf{x}} \in D} \|\xi - W^T \phi(\bar{\mathbf{x}})\|. \quad (3.51)$$

Similarly, for the case of Eqs. (3.33) and (3.34) presented in the relative degree formulation, Eqs. (3.47), (3.49) and (3.50) should be written in scalar form. For that case,  $\bar{\mathbf{x}}$  consists of the central differences of the limit parameter, slow states, and the control of interest, such that  $\bar{\mathbf{x}} = [\bar{\partial}(y_p)_t, \mathbf{x}_s, u_e]^T$ .

The goal hereafter is to minimize  $\mathbf{e}$  of Eq. (3.50). Note that the model tracking error,  $\mathbf{e}$ , is a function of two error components such as; the reconstruction error,  $\epsilon$ , and the weight error component,  $\tilde{W}^T \phi(\bar{\mathbf{x}})$ . Theoretically,  $\mathbf{e}$  can be made equal to  $\epsilon$  only when the weight errors are made exactly zero, that is  $\mathbf{e}|_{W=W^*} = \epsilon$ . Here,  $\epsilon$  can be reduced only with using a sufficiently larger basis,  $\phi$ . Once a basis with necessary functions inside is decided to be used for online modeling, the only remaining task is to use an online weight update law to minimize the weight error component, hence the modeling error.

In this thesis, a weight update law, known as concurrent learning in the literature [33, 32], is used. Concurrent learning is a concept through neural network based adaptive control in which both instantaneous data and online recorded data are used simultaneously in the weight update. Next, the concurrent learning weight update law for the neural networks used in the proposed limit margin estimation methodology is presented.

### 3.2.2 Concurrent Learning Weight Update Law

*Theorem:* At a delayed moving time, consider the plants given by Eqs. (3.7) and (3.27) with the uncertainty of Eq. (3.47) and the derivative representations obtained using Eqs. (3.13) and (3.32), which are  $\bar{\partial}(\mathbf{x}_f)_d$  and  $\bar{\partial}(y_p)_d$ . Using the *Assumptions (3.1-3.12)*, let the following equation be used to update the weights of the LPNNs,  $\Delta$ , of Eqs. (3.14) and (3.33):

$$\dot{W} = \Gamma(\phi(\bar{\mathbf{x}}_d)\mathbf{e}_d^T + \sum_{j=1}^p \phi(\bar{\mathbf{x}}_j)\mathbf{e}_j^T) \quad (3.52)$$

where,  $\Gamma$  is a positive definite learning gain matrix. The weight update law of Eq. (3.52) guarantees that the error in the weight estimation  $\tilde{W} = W^* - W$  for the estimators of Eqs. (3.14) and (3.33) are ultimately bounded.

*Proof:* See the Appendix.

In Eq. (3.52), two main terms are used to update the weights online. One term is the instantaneous error signal,  $\phi(\bar{\mathbf{x}}_d)\mathbf{e}_d^T$ , which is the contribution of the instantaneous data in the weight update. Subscript  $d$  is used since the adaptation is performed at the delayed moving time. Note that the error signal,  $\mathbf{e}_d^T$ , and the basis function,  $\phi(\bar{\mathbf{x}}_d)$ , are the data used for instantaneous adaptation. Whereas the second term of Eq. (3.52), which is  $\sum_{j=1}^p \phi(\bar{\mathbf{x}}_j)\mathbf{e}_j^T$ , is the contribution of the previously recorded data, hence the information of past, in the weight update. The data with subscript  $j$  refers to the  $j^{\text{th}}$  recorded error signal and the basis in the recorded data set, which are  $\mathbf{e}_j^T$  and  $\phi(\bar{\mathbf{x}}_j)$ . Mathematically, recorded data sets are the matrices that contain recorded error signals and neural network bases. Such matrices are called *history stacks* in the concept of concurrent learning.

The weight update law given by Eq. (3.52) is found in [48, 34], and shown to be asymptotically stable for a special case of adaptive parameter estimation problem [34]. This special case is when the reconstruction error,  $\epsilon$ , of Eq. (3.50) is assumed to be zero. For a general case when the reconstruction error,  $\epsilon$ , of Eq. (3.50) is nonzero, the ultimate bounds of the LPNN weight estimation,  $\tilde{W} = W^* - W$ , can be established. In the proof in the Appendix, the related error bounds of the LPNN weight estimation is presented.

Note that the concurrent learning weight update law given by Eq. (3.52) doesn't require persistency of excitation (PE) in the inputs of the adaptive element for parameter convergence, i.e. the convergence of weights around a compact neighbourhood of the optimal values is possible without requiring PE. In [47], the related weight error bounds of MRAC are shown to be minimized when the minimum singular value of the history stack of the recorded bases is increased, i.e. maximized, online. In the proof given in the Appendix, the weight error bounds for the considered estimators are established and similar to [34], it is demonstrated that the bounds can be made even smaller with increasing the minimum singular value of the history stack of the considered bases.

In the simulations presented in this chapter, the weight update law given by Eq. (3.52) is used to update the weights of LPNN's given by Eqs. (3.14) and (3.33). The necessary data for concurrent adaptation is also recorded online using the methods described in the following subsection.

### 3.2.3 Concurrent Learning Implementation

The method in concurrent learning is to record data online to be used later to cancel out the modeling errors during flight. The recorded data is used with the instantaneous data concurrently in the weight update of the neural network. Model tracking error,  $e_d^T$ , of Eq. (3.50) and the network basis,  $\phi(\bar{x}_d)$ , are the data to be recorded into history stacks. If  $Z$  is the history stack in which the bases are recorded into, then  $Z$  is constructed as

$$Z = [\phi_1, \phi_2, \phi_3, \dots, \phi_p]. \quad (3.53)$$

Here,  $\phi_j \in \mathfrak{R}^m$ ,  $j = 1, 2, \dots, p$ , and  $Z$  is an  $m \times p$  history stack matrix. In addition, if  $E$  represents the history stack of the model tracking errors, then  $E$  is constructed as

$$E = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \dots, \mathbf{e}_p], \quad (3.54)$$

in which,  $\mathbf{e}_j \in \mathfrak{R}^l$ ,  $j = 1, 2, \dots, p$ , and  $E$  is an  $l \times p$  history stack matrix. Note that a necessary data is recorded into the columns of the matrices  $Z$  and  $E$  to generate a long term memory in the approach.

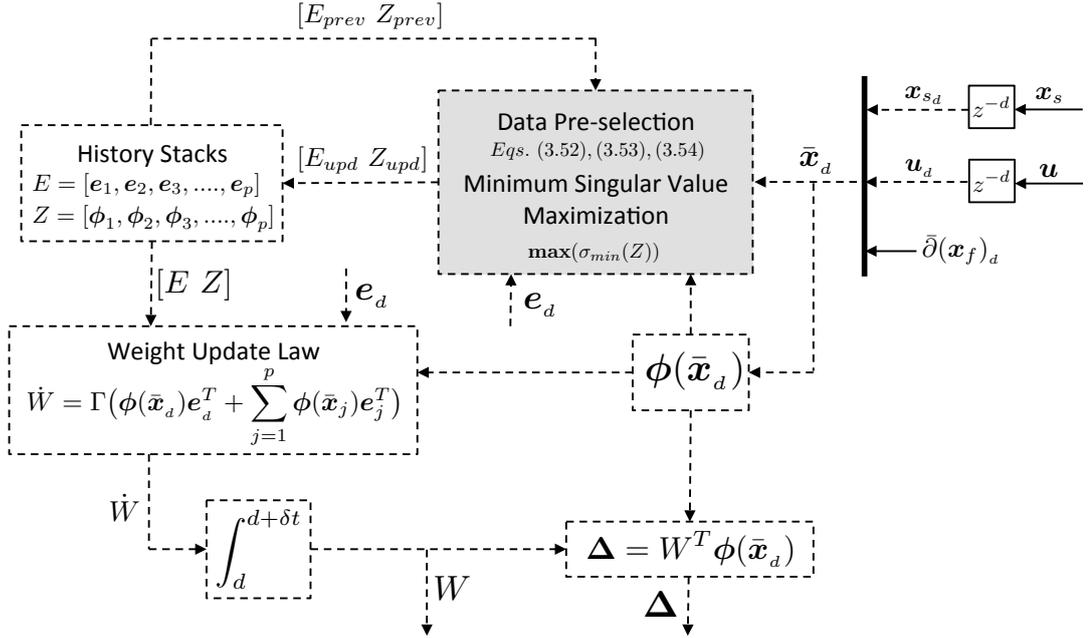


Figure 3.3: Concurrent Learning Implementation

The selection of the data to be recorded is a challenging task in concurrent learning. A typical selection criteria, also used in [33], is:

$$\frac{(\bar{\mathbf{x}}_d - \bar{\mathbf{x}}_p)^T (\bar{\mathbf{x}}_d - \bar{\mathbf{x}}_p)}{\bar{\mathbf{x}}_d^T \bar{\mathbf{x}}_d} > \epsilon_x. \quad (3.55)$$

Here,  $p$  denotes the latest point stored in the history stack and  $\epsilon_x$  is a design parameter. Therefore, if  $\bar{\mathbf{x}}_d$  is sufficiently different than  $\bar{\mathbf{x}}_p$  such that Eq. (3.55) is satisfied, then  $\bar{\mathbf{x}}_d$  can be recorded into  $Z$ . The following criteria is used in [32] to capture the data of a steady state condition:

$$\epsilon_{y1} < \sqrt{\sum_{a=1}^N (x_d(k) - x_d(k-a))^2} < \epsilon_{y2} \quad (3.56)$$

$$\epsilon_{z_1} < \sqrt{\sum_{a=1}^N (u_{e_d}(k) - u_{e_d}(k-a))^2} < \epsilon_{z_2} \quad (3.57)$$

Here,  $x_d$  is a state and  $u_{e_d}$  is a control input at the delayed moving time.  $x_d(k)$  is the state at the delayed time and  $x_d(k-N)$  is the state at the  $N^{th}$  time step before the delayed time.  $\epsilon_{y_1}$ ,  $\epsilon_{y_2}$ ,  $\epsilon_{z_1}$  and  $\epsilon_{z_2}$  are selected to capture an approximate steady state condition.

Note that a goal in concurrent learning is to decrease the weight error bounds in order to place the adaptive weights in a compact neighborhood of the ideal weights. Following the proof given in the Appendix, that goal is shown to be achieved by recording data into the history stack such that the minimum singular value of  $Z$  is maximized online. In [47] and [48] a minimum singular value maximization algorithm has also been introduced to record necessary data for concurrent learning. Note also that Eqs. (3.55), (3.56) and (3.57) do not guarantee the maximization of the minimum singular value of  $Z$ , therefore in this thesis, an algorithm similar to those of [47] and [48] is developed and used to decrease the weight error bounds online.

The overall simulation block diagram of the concurrent learning implementation is presented in Fig. 3.3. That figure also represents the inside of the Adaptive NN block shown previously in Fig. 3.1.

In Figure 3.3, data recording is performed inside the gray colored box. First, using Eqs. (3.55), (3.56) and (3.57) a necessary data is pre-selected. Then, the neural network basis, which is calculated with that pre-selected data, is replaced with each column of the history stack matrix of the previous time step,  $Z_{prev}$ . Once the considered basis is found to increase the minimum singular value of  $Z_{prev}$ , that is when a replacement is detected to increase  $\sigma_{min}(Z_{prev})$ , then the associated column is updated with that basis and the history stack is updated as  $Z_{upd}$ . For the implementation, that algorithm is run inside the gray colored box at each simulation time step.

In the Appendix, a comparative example case is presented as well to demonstrate concurrent learning with and without using minimum singular value maximization in the weight update.

### 3.3 Implementation and Simulation Results

The envelope cueing methodologies and the concurrent learning implementation presented above are evaluated in this section using simulation studies. Fixed wing and rotary wing aircraft models are used for the demonstration of the adaptive limit and control margin estimations.

#### 3.3.1 Flight Envelope Protection for Aircraft

In this section the effectiveness of the proposed limit and control margin estimation algorithms is evaluated through simulations. Simulations are performed using a fixed-wing aircraft model based on the model described in [49]. The aircraft model used is similar to the Cessna182 fixed-wing airplane. The aerodynamic model described in [49] is extended with a set of nonlinear aerodynamic data obtained from an open source flight dynamics library, known as *JSBSim*. JSBSim is available to public over the web. The nonlinear lift and drag curves of the same aircraft are extracted from the open source library (JSBSim) and integrated into the aerodynamic model of [49]. The resulting aircraft model has similar dynamic characteristics, i.e. phugoid, short period and dutch roll modes, as the model described in [49].

Two examples are presented next. In both examples, the *Direct Adaptive Limit Margin Estimation* is used to predict the limit margins for both angle of attack and load factor. In the examples, the *Sensitivity Estimation Based Control Margin Estimation* is applied to predict the control sensitivities and control margins on the elevator control. In the first example, purely longitudinal pull-up and push-over maneuvers are performed to violate envelope limits in different adaptation schemes. In the second example, similar pull-up and push-over maneuvers are applied during a transient turn. Control saturation is used to keep the aircraft at the limit boundaries, hence is used to demonstrate the effectiveness of the proposed methodologies.

The angle of attack and the load factor are expressed using the following equations:

$$\alpha = \text{atan}\left(\frac{w}{U}\right), \quad n_z = 1 + \frac{Uq}{g} \quad (3.58)$$

where  $U$  is the forward velocity,  $q$  and  $w$  are the pitch rate and the z-axis body frame velocity and  $g$  is the gravitational acceleration. Here,  $q$  and  $w$  are fast states and  $U$  is a slow state. Therefore, a dynamic trim condition of  $q$  and  $w$  will also correspond to a trimmed out angle of attack and load factor response. Hence, both parameters are assumed to be fast aircraft states, and are known to reach their critical limits at the steady state condition. The following flight envelope boundaries are assumed: -5 to 12 degrees for the angle of attack, and -0.5 to 3.5 g's for the load factor.

In the examples, angle of attack, load factor and other signals required for the implementation of the algorithms are assumed to be smooth and available signals. Gust or sensor noise are not included.

### 3.3.1.1 Example-1: Longitudinal Pull-up & Push-over

In this example, the aircraft model is subjected to filtered step elevator inputs starting from a trimmed flight condition to simulate pitch up and pitch down maneuvers. A first order filter with 0.2 seconds of a time constant is used to represent the actuator dynamics. Angle of attack, load factor, pitch rate, airspeed and pitch angle are assumed to be accurately measured or estimated. The delayed moving time step,  $d$ , is taken as  $d = t_c - 0.1s$  where  $t_c$  is the current time and the simulation time step is  $\Delta t = 0.01s$ . The delayed moving time step limits the number of central differences to be used in the approximation. It is essential to have enough data points for central differences to represent the derivatives of the signal. In this example, 4 data points are used in a time window of 0.1s to generate central differences. Time derivatives of angle of attack, pitch rate and load factor are represented by central differences at the delayed time. Elevator control, airspeed and pitch angle are used to generate an approximate model. Eq. (3.52) is used for the weight update. Using Eq. (3.21) the following model is constructed at the delayed time step,  $d$ , for the angle of

attack and the load factor outputs:

$$\begin{bmatrix} \hat{\alpha}_d \\ \hat{n}_{zd} \end{bmatrix} = -A_1^{-1}[-[\partial_\alpha \ \partial_q]^T + B\delta_{ed}] + \begin{bmatrix} \Delta_\alpha(\bar{\partial}(\mathbf{x}_{fd}), V_{ed}, \theta_d, \delta_{ed}, b_1) \\ \Delta_{n_z}(\bar{\partial}(\mathbf{x}_{fd}), V_{ed}, \theta_d, \delta_{ed}, b_2) \end{bmatrix}, \quad (3.59)$$

where,  $\bar{\partial}(\mathbf{x}_f)_d$  is constructed using Eq. (3.13) with  $\mathbf{x}_f = [\alpha \ q \ n_z]^T \in \mathfrak{R}^3$ .  $\partial_\alpha$  and  $\partial_q$  are the average sums of central differences used in the linear model inverse and given below:

$$\begin{aligned} \partial_\alpha &= \frac{1}{k} \sum_{j=1}^k \bar{\partial}(\mathbf{x}_f)_d(1, j), \\ \partial_q &= \frac{1}{k} \sum_{j=1}^k \bar{\partial}(\mathbf{x}_f)_d(2, j), \end{aligned} \quad (3.60)$$

and the following matrices are used for  $A_1$  and  $B$  of Eq. (3.59):

$$A_1 = \begin{bmatrix} -7.50 & 0.20 \\ -129 & -6.50 \end{bmatrix}, \quad B = \begin{bmatrix} -0.71 \\ -0.60 \end{bmatrix}. \quad (3.61)$$

In this example,  $k$  of Eq. (3.13) is taken as  $k = 4$ , hence  $\bar{\partial}(\mathbf{x}_f)_d \in \mathfrak{R}^{3 \times 4}$ . The parameter  $k$  can be larger, depending on the computation time. Note that, each central difference expression corresponds to a weight. Therefore, the following basis is constructed for the linear neural networks,  $\Delta_\alpha$  and  $\Delta_{n_z}$ :

$$\begin{aligned} \phi(i) &= \phi_i(\bar{\partial}(\mathbf{x}_f)_d(1, i)), \quad i = 1 : 4 \\ \phi(i + 4) &= \phi_{i+4}(\bar{\partial}(\mathbf{x}_f)_d(2, i)), \quad i = 1 : 4 \\ \phi(i + 8) &= \phi_{i+8}(\bar{\partial}(\mathbf{x}_f)_d(3, i)), \quad i = 1 : 4 \\ \phi(13 : 18) &= [\phi_{13}(\delta_e) \ \phi_{14}(V_e) \ \phi_{15}(\theta) \ \phi_{16}(\delta_e V_e) \ \phi_{17}(\delta_e \theta) \ b_1]^T. \end{aligned} \quad (3.62)$$

Note that,  $\phi(i)$  is the  $i^{th}$  element of the basis vector and  $\phi_i(\cdot)$  is the corresponding activation function. The following activation function is used to ensure boundedness:

$$\phi_i(\cdot) = a_i \tanh\left(\frac{\cdot}{a_i}\right) \in \mathfrak{R}, \quad i = 1, 2, \dots, 18 \quad (3.63)$$

where,  $a$ 's are design parameters. In the simulations, only the linear regime of the function is used, that is, all the inputs are scaled approximately to the linear regime of  $\tanh(\cdot)$ .

The dynamic trim state,  $[\hat{\alpha}_{DT} \ \hat{n}_{zDT}]^T$ , can be found by evaluating Eq. (3.59) when the central differences and the average sums are zero, such that  $\bar{\partial}(\mathbf{x}_f)_d = 0$  and

$\partial_\alpha = \partial_q = 0$ :

$$\begin{bmatrix} \hat{\alpha}_{DT} \\ \hat{n}_{zDT} \end{bmatrix} = -A_1^{-1}B\delta_e + \begin{bmatrix} \Delta_\alpha(0, V_e, \theta, \delta_e, b_1) \\ \Delta_{n_z}(0, V_e, \theta, \delta_e, b_2) \end{bmatrix} + \begin{bmatrix} e_{\alpha_d} \\ e_{n_{zd}} \end{bmatrix}, \quad (3.64)$$

where, the model tracking errors,  $e_{\alpha_d}$  and  $e_{n_{zd}}$ , are given by

$$e_{\alpha_d} = \alpha_d - \hat{\alpha}_d, \quad e_{n_{zd}} = n_{zd} - \hat{n}_{zd}. \quad (3.65)$$

The limit margins are obtained as

$$\hat{\alpha}_{margin} = \hat{\alpha}_{lim} - \hat{\alpha}_{DT}, \quad \hat{n}_{zmargin} = \hat{n}_{zlim} - \hat{n}_{zDT}. \quad (3.66)$$

Control sensitivities of fast states to elevator input at the dynamic trim condition can be found by taking the derivative of Eq. (3.64) with respect to  $\delta_e$ . The sensitivities  $S_\alpha = \frac{\partial \hat{\alpha}_{DT}}{\partial \delta_e}$  and  $S_{n_z} = \frac{\partial \hat{n}_{zDT}}{\partial \delta_e}$  are found as

$$\begin{bmatrix} S_\alpha \\ S_{n_z} \end{bmatrix} = -A_1^{-1}B + \left[ \frac{\partial \Delta_\alpha}{\partial \delta_e} \quad \frac{\partial \Delta_{n_z}}{\partial \delta_e} \right]_{\partial(\mathbf{x}_f)_d=0}^T. \quad (3.67)$$

Then, the control margins are

$$\hat{\delta}_{emarg_\alpha} = \frac{1}{S_\alpha} \hat{\alpha}_{margin}, \quad \hat{\delta}_{emarg_{n_z}} = \frac{1}{S_{n_z}} \hat{n}_{zmargin}. \quad (3.68)$$

Using the control margins, the control limits are calculated as

$$\hat{\delta}_{elim_\alpha} = \hat{\delta}_{emarg_\alpha} + \delta_e, \quad \hat{\delta}_{elim_{n_z}} = \hat{\delta}_{emarg_{n_z}} + \delta_e. \quad (3.69)$$

In Figure 3.4, angle of attack, load factor, airspeed response and the control input along with the dynamic trim and the control limit estimations are presented. Open loop elevator inputs are provided that result in  $\alpha$  or  $n_z$  limit exceedances. Both margins are predicted within a sufficient time range for a pilot to react to the limits of a flight envelope. In fact, the points  $(\hat{\alpha}_{margin}, \hat{\delta}_{emarg_\alpha}) = (0, 0)$  and  $(\hat{n}_{zmargin}, \hat{\delta}_{emarg_{n_z}}) = (0, 0)$  are predicted using the estimation algorithm at the onset of the actual violations of the angle of attack and the load factor limits.

The weight update time histories are shown in Fig. 3.5. Since the model is a nonlinear model, the structure of the uncertainty is not exactly known. Hence the weights can only be bounded around the optimal weights, which might be

non-unique depending on the selection of the LPNN basis. Compared to [27, 28], the weight time history exhibits a tendency of steady state convergence (Fig. 3.5). This is also the case for the control sensitivity estimations.

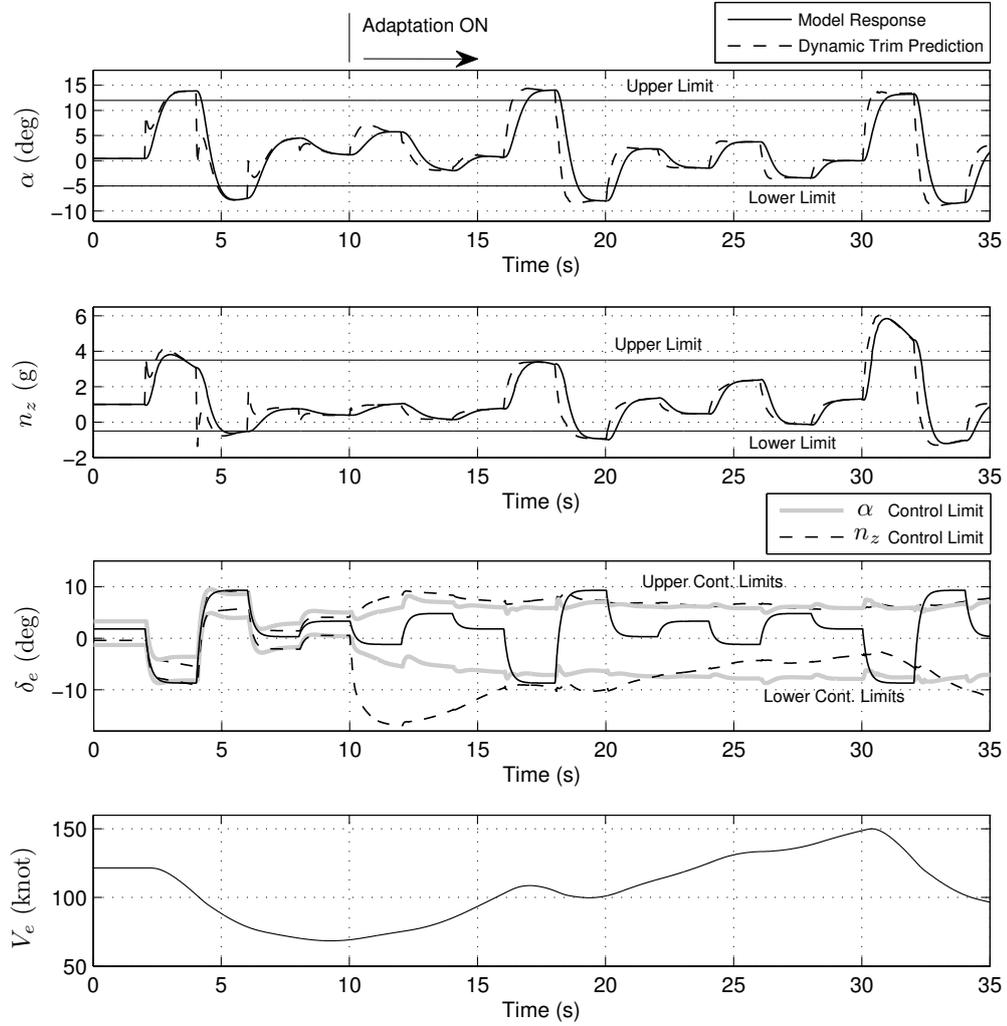


Figure 3.4: Limit Prediction, Example-1

The modeling errors in the angle of attack and the load factor estimations,  $\xi_\alpha$ ,  $\xi_{n_z}$ , and the neural net outputs,  $\Delta_\alpha$ ,  $\Delta_{n_z}$ , are presented in Fig. 3.6. Note that without adaptation (0-10s), the modeling errors and the delayed approximation errors,  $e_\alpha$ ,  $e_{n_z}$ , have the same magnitude. After adaptation is switched on, delayed error signals evolve to zero as the adaptive elements capture the modeling errors.

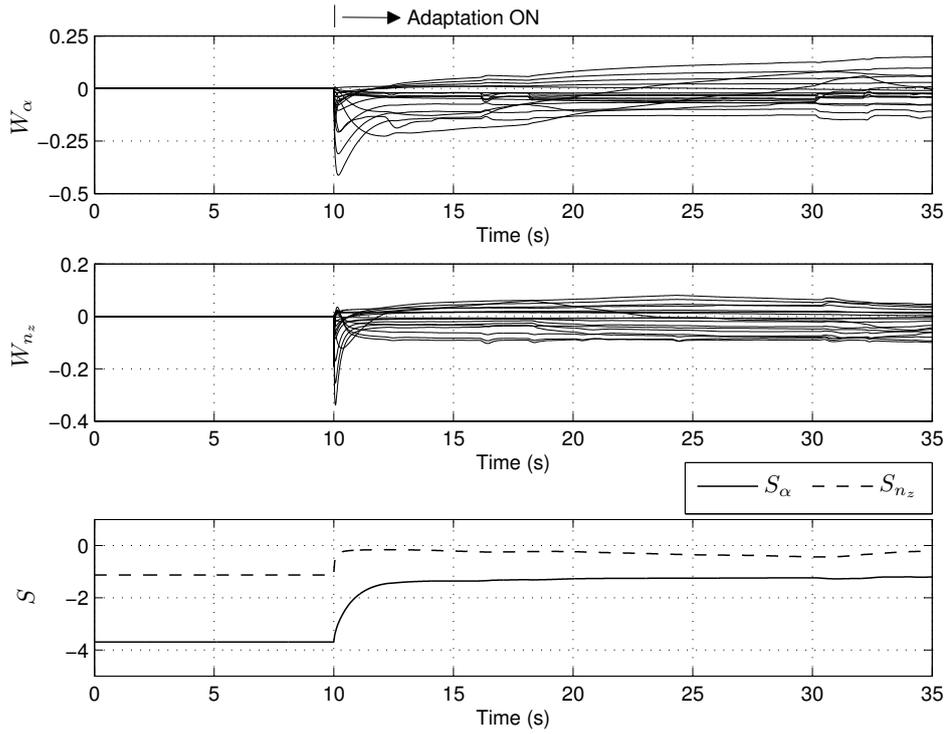


Figure 3.5: Weight Update Time Histories and Predicted Control Sensitivities, Example-1

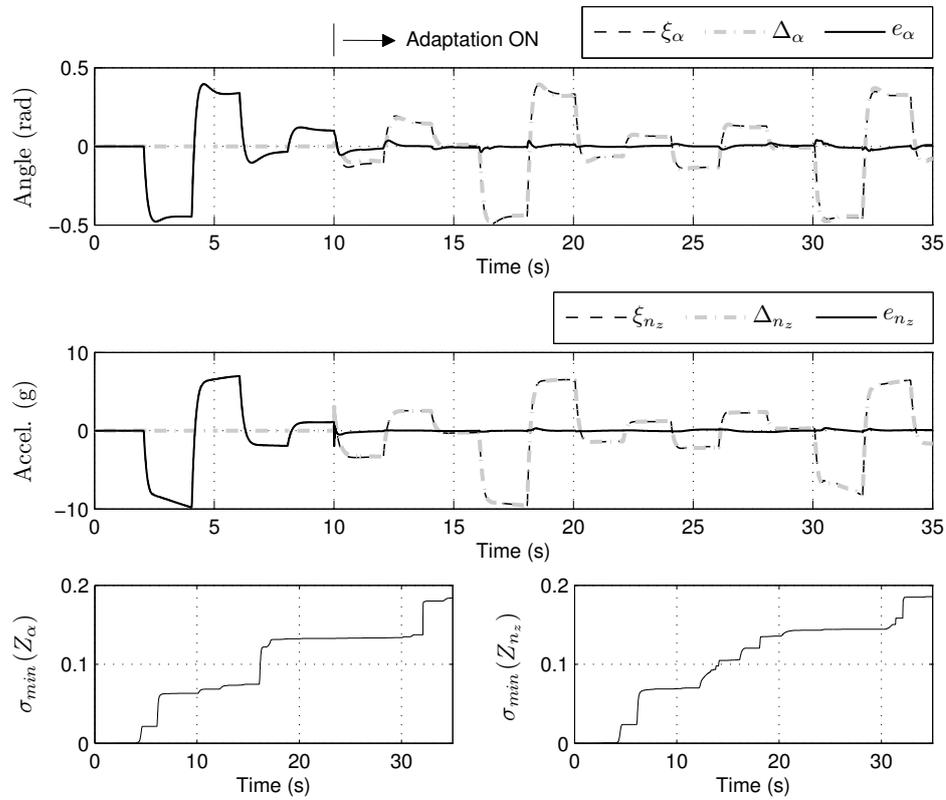


Figure 3.6: Modeling Error Compensation and Minimum Singular Values of the History Stacks, Example-1

In Figure 3.6, the modeling errors are approximated with sufficient error bounds using the adaptive elements,  $\Delta_\alpha$ ,  $\Delta_{n_z}$ . Maximized singular values of the history stack matrices,  $Z_\alpha$  and  $Z_{n_z}$  are shown in Fig. 3.6. A maximum number of 30 data points are used for history stack and only data increasing the minimum singular value is added into the stack.

The results of Fig. 3.4 are similar to a physical model behavior since as the airspeed of the model increases the control margins of the load factor become smaller, around  $t = 20 - 30s$ . Also, at the low speed region around  $t = 10 - 20s$ , the angle of attack control margin become more critical.

The simulation is repeated when the weights start from nonzero initial conditions. Results are given in Fig. 3.7. The related modeling error compensation and maximized singular values are presented in Fig. 3.8. Although the modeling error is large before the adaptation, the adaptive element compensates for that error after adaptation is turned on. Although the weight matrix converged to a different optimal set as in Fig. 3.9, the estimated sensitivities are very close, indicating that the weights related with control sensitivities are around the same optimal weights.

The reason why the weight matrix is converged to an optimal set, which is different from the results in Fig. 3.5, is that the central difference expressions have numerically close values in the neural network basis. Since the values are numerically close, the ideal set of weights associated with the central difference expressions of that basis can be non-unique, therefore, the weight error bounds associated with those weights can be relatively large. That also makes the overall weight error bound (weight error bound of the full basis) larger. Although there exist linearly dependent elements in the basis such as central differences, the minimum singular value of the history stack can still be increased due to the existence of the linearly independent elements. Therefore, the larger weight error bounds can still be minimized theoretically when the minimum singular value of the history stack is made as maximum as possible.

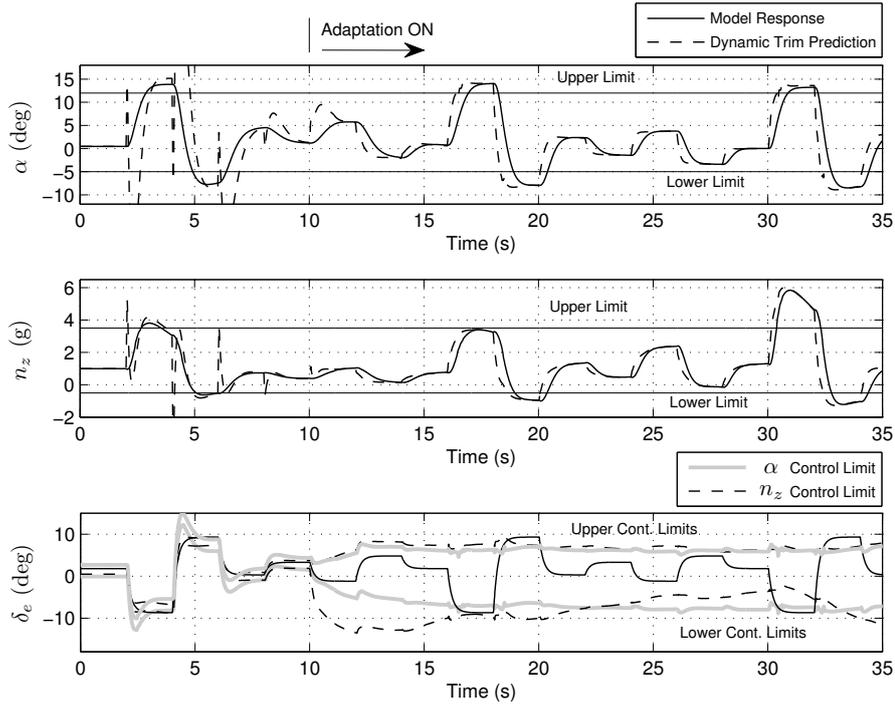


Figure 3.7: Limit Prediction, Nonzero Initial Condition for Weights, Example-1

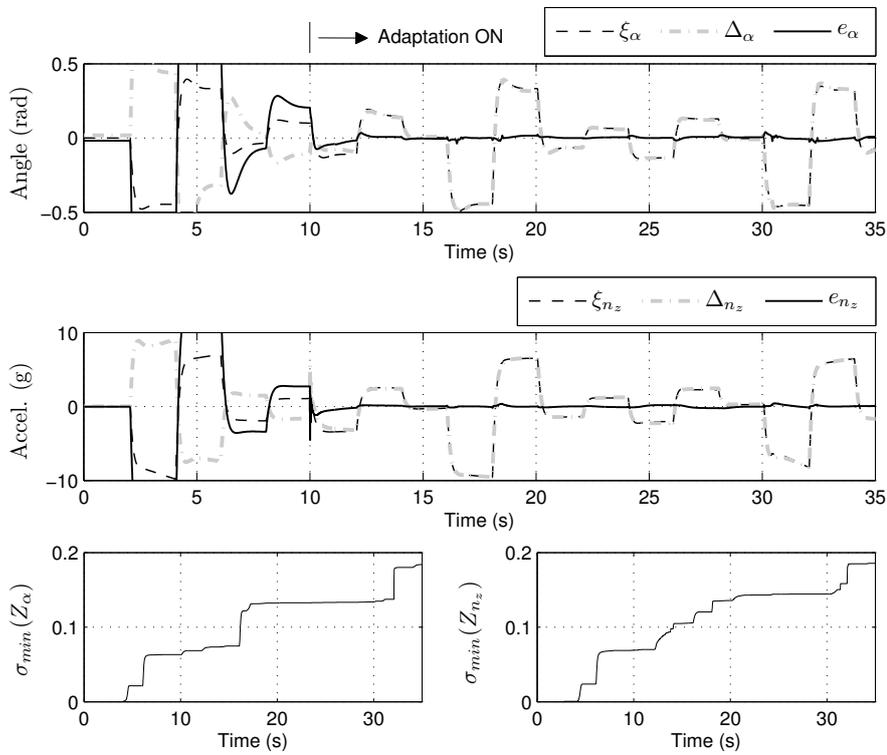


Figure 3.8: Modeling Error Compensation and Minimum Singular Values of the History Stacks, Nonzero Initial Condition for Weights, Example-1

Note also that, the ideal set of weights associated with the control sensitivities are unique (Figs. 3.5 and 3.9), since the considered basis functions are linearly independent functions for online approximation. Therefore, the weight error bounds associated with the sensitivities are expected to be made sufficiently small in the approach. Whereas, the bounds of the weights related with the central difference expressions are expected to be relatively larger.

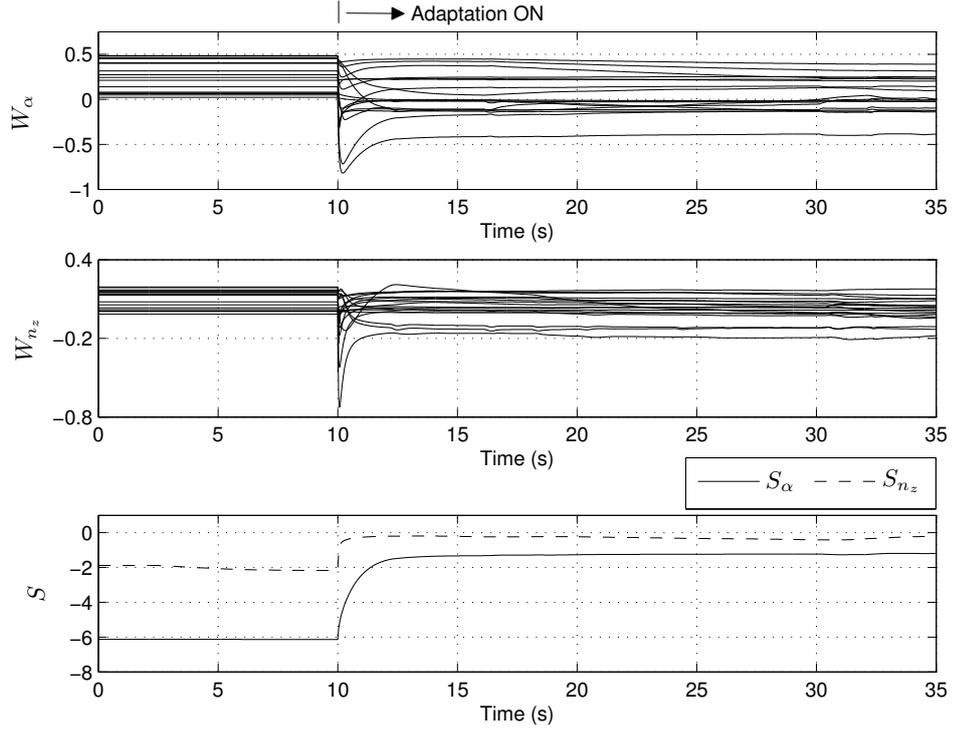


Figure 3.9: Weight Update Time Histories and Predicted Control Sensitivities, Nonzero Initial Condition for Weights, Example-1

When the control input is limited with the estimated control limits the limiting states are expected to follow the flight envelope boundary. Therefore, a limit avoidance simulation is performed such that the actual control inputs are saturated by the estimated control limits throughout the simulation. The initial conditions of the weights start at zero. Results are presented in Figure 3.10 where the limiting states stay at the envelope boundaries during artificial control saturation with acceptable errors, indicating a verification of the online predicted margins. In Section 3.3.2, the limit avoidance method will further be detailed. Note that, in the previous work [28], additional filters and smoothing logics are

used to avoid chattering during control input saturation. Chattering was due to the fast instantaneous learning which resulted in oscillations in the weight update. Compared to [28], smoothers are not required during control saturation since the adaptive element has a steady state tendency, hence, parameter convergence using concurrent adaptation.

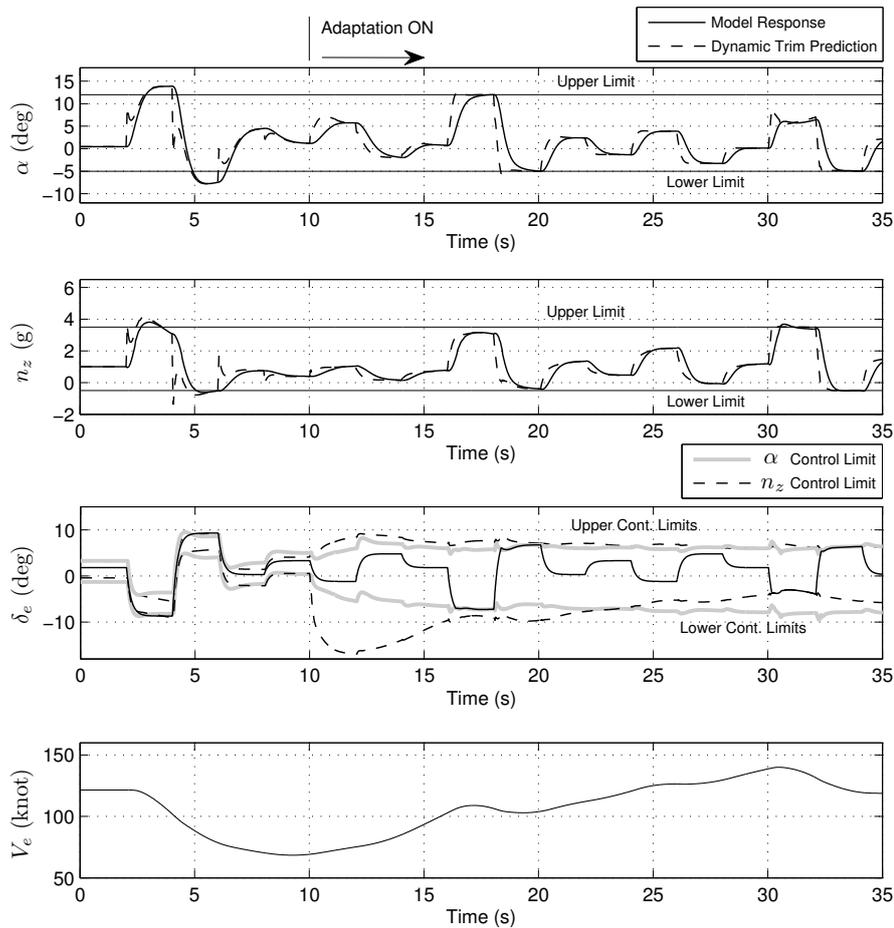


Figure 3.10: Limit Avoidance, Example-1

### 3.3.1.2 Example-2: Pull-up & Push-over during a Turn

In this example, the pitch up and pitch down maneuvers are repeated during a high bank angle turn maneuver. The initial conditions of the weights and the history stack matrix are the final values obtained from Example-1 (Fig. 3.4). Results of the predictions and the aileron input, roll angle and airspeed response are given in Fig. 3.11.

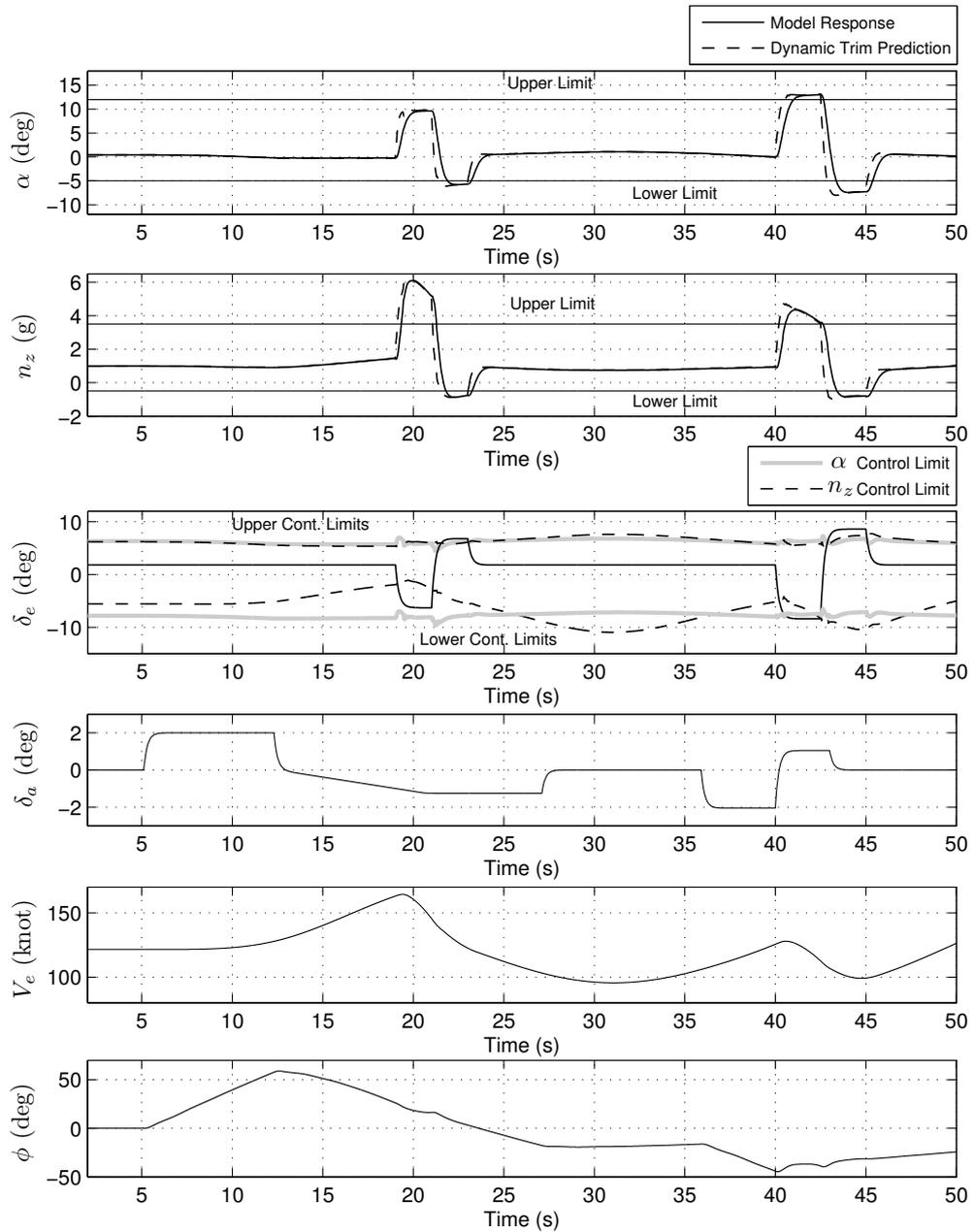


Figure 3.11: Limit Prediction, Example-2

The aircraft is banked to the right and left during the maneuver. Two elevator doublet inputs are applied. First, a doublet input is applied around  $t = 18$ s while having a bank angle of 30 degrees. The second doublet is applied while banking at an angle of -48 degrees ( $t = 40$ s). In both cases the dynamic trim of the critical states and related control limits are estimated prior to reaching the actual limits.

Next, limit avoidance is applied. The aileron input scenario of the previous simulation is repeated while the elevator inputs are limited with the estimated control limits. Results are shown in Fig. 3.12. Both elevator doublets are saturated and the aircraft is kept within the envelope limits. Compared to the previous simulation, the second doublet input is applied during a bank angle of -70 degrees.

Therefore, the obtained control margins can be used to cue the pilot of a fly-by-wire aircraft both to avoid approaching envelope limits and also to stay at the flight envelope limit to maximize the operational effectiveness.

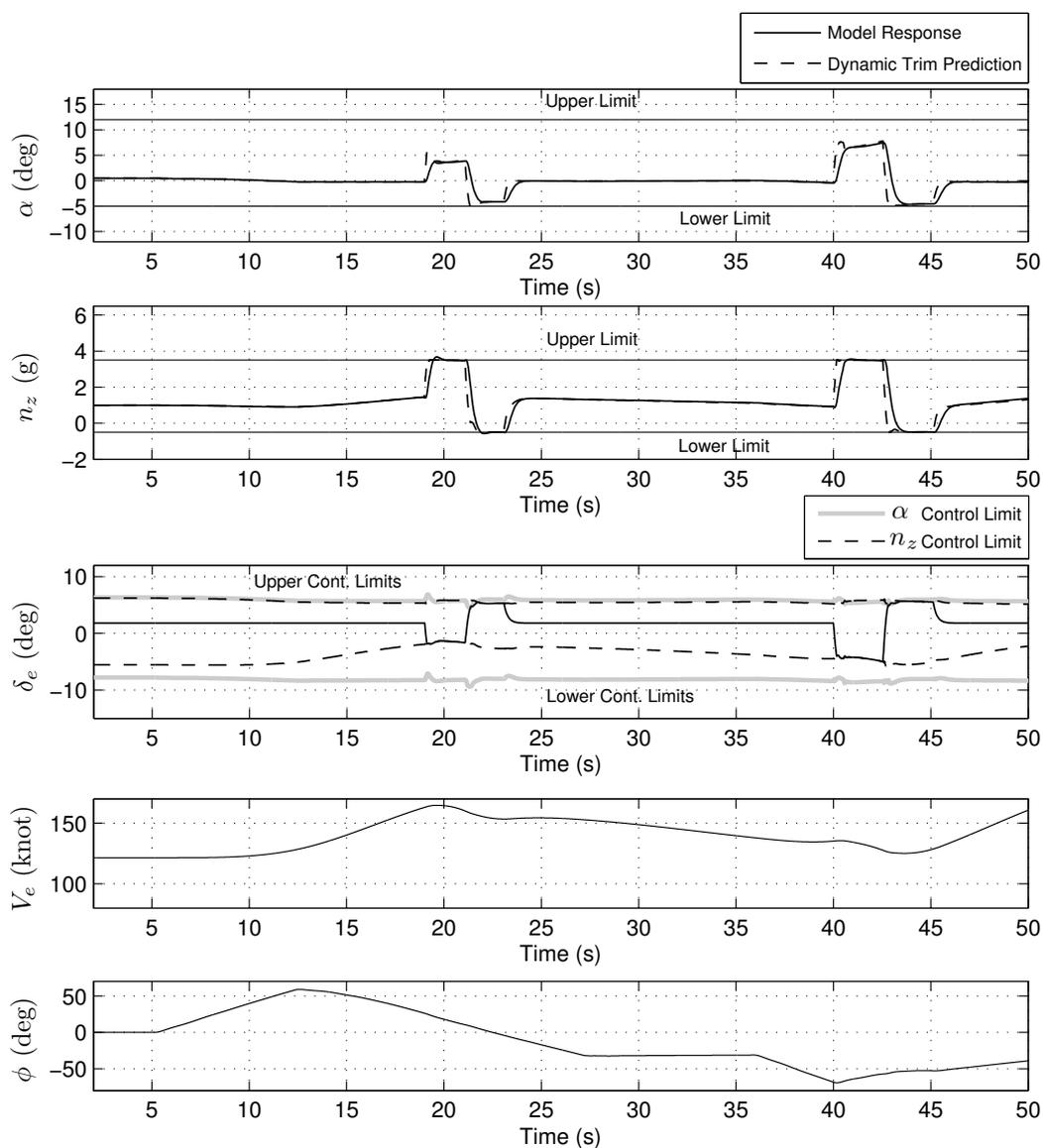


Figure 3.12: Limit Avoidance, Example-2

### 3.3.2 Vertical Speed Limit Protection for Helicopters

Vertical speed is a critical flight parameter for a helicopter operating at low height flight operations and sensitive to collective movement. A collective cueing system that gives feedback to the pilot at the onset of a vertical speed down limit would reduce the pilot workload and increase the overall safety of the helicopter during operations in close proximity to the ground.

In this example, vertical speed ( $V_s$ ) is treated as the measured limit parameter that limits the operational envelope and is known to have a steady state critical response, such that for a given collective input the maximum magnitude of the response occurs at the steady state condition.

The prediction algorithm shown in Fig. 3.2, is applied to predict vertical speed limit margins for a utility helicopter, operating at hover, low speed and low height above terrain flight conditions. The *Sensitivity Estimation Based Control Margin Estimation* is applied to predict allowable margins on the collective control. A ground speed controller is used to keep the helicopter in hover and in 20-60 kts forward flight conditions for the demonstration.

The simulation model is a high-fidelity nonlinear helicopter model with a 3-state dynamic inflow model and flapping dynamics in the main rotor, resembling a generic utility helicopter. The helicopter model is similar to the UH-1H utility helicopter and modeled using a modeling and simulation tool called *Heli-Dyn+*. *Heli-Dyn+* is a software tool in which various helicopter components such as main rotor, tail rotor, fuselage and etc. can be modeled individually and combined all together using a component build-up method [50]. For the simulations presented next, the main rotor of the considered helicopter is modeled in the software using the *Peters-He Finite State Dynamic Wake Model*, whereas the tail rotor is modeled using *Blade Element Momentum Theory*. Flat plate drag areas are used to model the fuselage in the software, and first order aerodynamic models are used for the aerodynamic surfaces. The final model is exported into the MATLAB/Simulink environment as a dynamic link library (.dll) using the export function of the software, and implemented for the proposed limit and

control margin estimation algorithms.

For the online approximation of the vertical speed signal, the parameters such as; airspeed ( $V_e$ ), height above terrain ( $h$ ), pitch angle ( $\theta$ ), roll angle ( $\phi$ ), the collective input ( $\delta_{coll}$ ) and the central difference expressions are used. Using Eq. (3.33) the following online model is constructed at the delayed time step,  $d$ , as

$$\hat{V}_{s_d} = A_{11}^{-1}(\partial_{V_s} - B_{11}\delta_{coll_d}) + \Delta(\bar{\partial}(V_s)_d, h_d, V_{e_d}, \theta_d, \phi_d, \delta_{coll_d}, b_1) \quad (3.70)$$

in which, the first term represents the inverse of a linear model approximation. Here, the central differences,  $\bar{\partial}(V_s)_d$ , are obtained using Eq. (3.32) and  $\partial_{V_s}$  is calculated using the average sum of central differences as below:

$$\partial_{V_s} = \frac{1}{k} \sum_{j=1}^k \bar{\partial}(V_s)_d(1, j). \quad (3.71)$$

The following basis is constructed for the neural network,  $\Delta = W^T \Phi_d$ :

$$\begin{aligned} \phi(i) &= \phi_i(\bar{\partial}(V_s)_d(1, i)), \quad i = 1 : 4 \\ \phi(i + 4) &= \phi_{i+4}(\bar{\partial}(V_s)_d(2, i)), \quad i = 1 : 4 \\ \phi(9 : 17) &= [\phi_9(\delta_{coll}) \quad \phi_{10}(V_e) \quad \phi_{11}(\theta) \quad \dots \\ &\dots \quad \phi_{12}(\phi) \quad \phi_{13}(\delta_{coll}V_e) \quad \phi_{14}(\delta_{coll}\theta) \quad \dots \\ &\dots \quad \phi_{15}(\delta_{coll}\phi) \quad \phi_{16}(V_e\theta) \quad b_1]^T \end{aligned} \quad (3.72)$$

Note that,  $\phi(i)$  is the  $i^{th}$  element of the basis vector and  $\phi_i(\cdot)$  is the corresponding activation function. The following activation function is used to ensure boundedness:

$$\phi_i(\cdot) = a_i \tanh\left(\frac{\cdot}{a_i}\right) \in \mathfrak{R}, \quad i = 1, 2, \dots, 17 \quad (3.73)$$

where,  $a$ 's are design parameters.

Using the variables of the current time step, the dynamic trim value,  $\hat{V}_{s_{DT}}$ , can be found evaluating Eq. (3.70) at the dynamic trim condition ( $\dot{V}_s = 0$ ) as

$$\hat{V}_{s_{DT}} = -A_{11}^{-1}B_{11}\delta_{coll} + \Delta(0, h, V_e, \theta, \phi, \delta_{coll}, b_1) + e_d \quad (3.74)$$

where, the delayed approximation error,  $e_d$ , is given by,

$$e_d = V_{s_d} - \hat{V}_{s_d} \quad (3.75)$$

and the limit margin is given as

$$\hat{V}_{smarg} = \hat{V}_{stim} - \hat{V}_{sDT}. \quad (3.76)$$

The control sensitivity between the vertical speed and the collective input at the steady state condition ( $\dot{V}_s = 0$ ) can be found by taking the derivative of Eq. (3.64) with respect to  $\delta_{coll}$ . The sensitivity of the vertical speed with respect to the collective input,  $S = \frac{\partial \hat{V}_{sDT}}{\partial \delta_{coll}}$ , is

$$S = -A_{11}^{-1}B_{11} + \left[ \frac{\partial \hat{V}_{sDT}}{\partial \delta_{coll}} \right]_{\dot{V}_s=0}. \quad (3.77)$$

Then the control margin and control limits are found as,

$$\hat{\delta}_{collmarg} = \frac{1}{S} \hat{V}_{smarg}, \quad \hat{\delta}_{colllim} = \frac{1}{S} \hat{V}_{smarg} + \delta_{coll}. \quad (3.78)$$

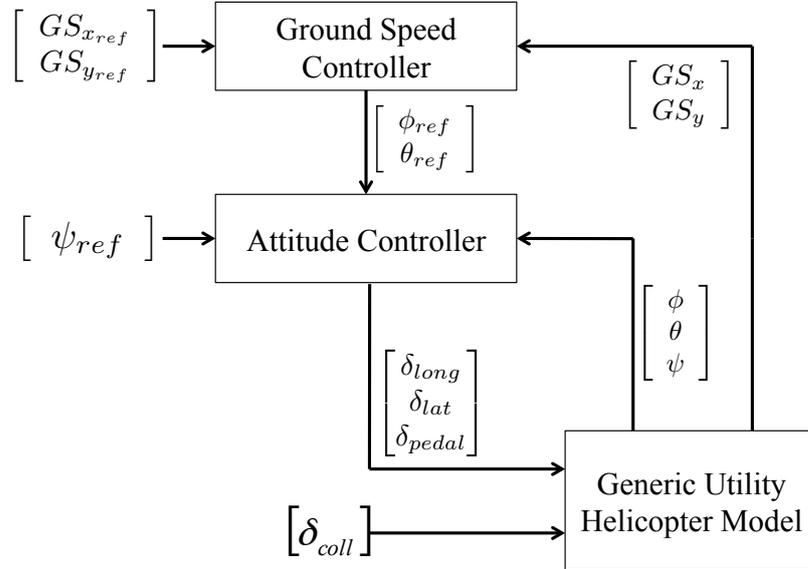


Figure 3.13: Simulation Block Diagram

In the following examples, the simulation block diagram given in Fig. 3.13 is used for the demonstration. Ground speed controllers in the outer loop and attitude controllers in the inner loop are used to maintain ground speed reference inputs. While the controllers maintain horizontal velocities, open loop collective inputs are given to helicopter. Three different examples are given. Examples are performed to demonstrate the effectiveness of the proposed algorithm during aggressive collective inputs given in hover and forward flight conditions. For the forward flight examples, 20 kts and 60 kts of ground speed conditions are used.

### 3.3.2.1 Example-1: Hover

The simulation is initialized at a hovering flight condition at 500 ft height above terrain. Controllers are used to keep the helicopter at zero ground speed throughout the simulation. Open loop collective inputs are given to result in vertical descent. In Fig. 3.15, the vertical speed response and steady state predictions are shown along with the vertical speed limit signal which is a function of the altitude. Vertical speed down limit is written as a function of helicopter's height above terrain,  $h$ , as shown in Fig. 3.14.

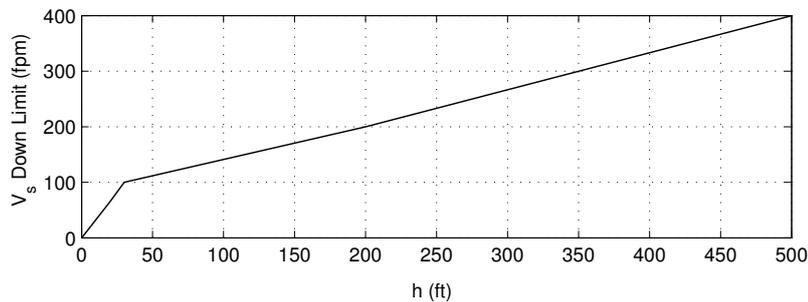


Figure 3.14: Vertical Speed Limit Profile

As shown, the future steady state values of vertical speed are predicted prior to actual violations with a significant lead time. Note that in Fig. 3.15 the positive sign of the vertical speed signal refers to a descending vertical speed. The collective inputs, estimated collective limits and the height above terrain information are presented as well. Note that the exceedances of the limits on the collective control happen at the same simulation time when the limit margin estimations of the vertical speed is zero (Fig. 3.15).

In Fig. 3.16, the modeling error compensation and the weight time histories of the neural network signal are presented. Since the weights reach approximate steady values rather fast, the modeling error compensation is fast as well. The data recording for concurrent adaptation starts from the beginning of the simulation. The maximized singular value of the history stack and the estimated sensitivity parameters are shown as well. Note that the estimated sensitivity is closely around a steady value. Helicopter attitudes and ground speeds are presented in Fig. 3.17. Ground speeds are kept at zero to maintain a vertical flight.

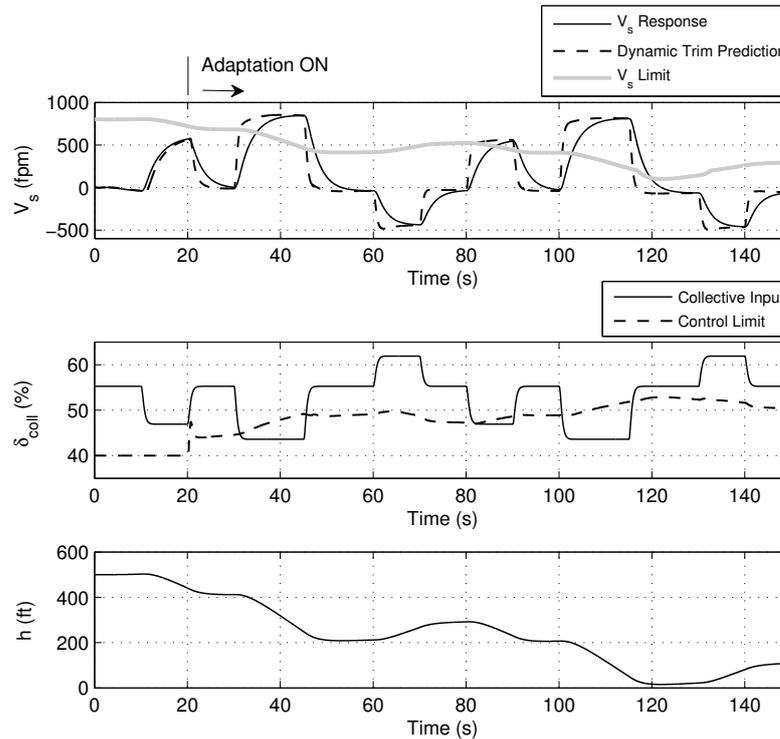


Figure 3.15: Limit Prediction, Example-1

Next, the limit avoidance algorithm given in Fig. 3.18 is added into the simulation. The limit avoidance algorithm is used to verify results. Note that, an algebraic loop exists during the control input saturation since an estimated control limit at the current time,  $\delta_{coll_{lim}}$ , is also required by the control margin estimation block at the current time. Therefore, a low pass filter is used to avoid that algebraic loop, A, of Fig. 3.18.

Here, the pilot input is saturated with the online obtained control limits. Therefore, the vertical speed is expected to be at the limit boundary during the control saturation. The limit margin is expected to be zero when the control margin is zero, or vice a versa. The result is presented in Fig. 3.19. The vertical speed is kept at the boundary with a significant lead time when the control limit is followed.

In the examples of Section 3.3.1, the limit avoidance technique given in Fig. 3.18 is used as well. In Figures 3.10 and 3.12, the elevator control inputs are saturated using the estimated elevator control limits through a low pass filter.

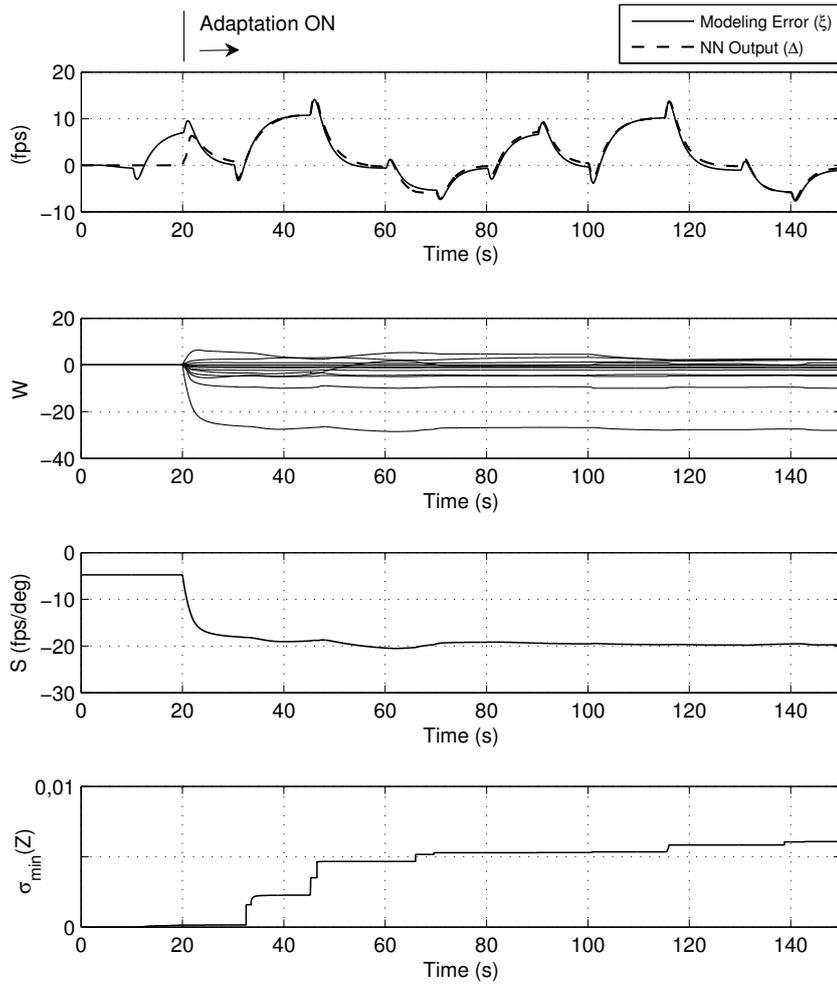


Figure 3.16: Modeling Error and NN Output, NN Weights, Predicted Collective Sensitivity, Minimum Singular Value of the History Stack, Example-1

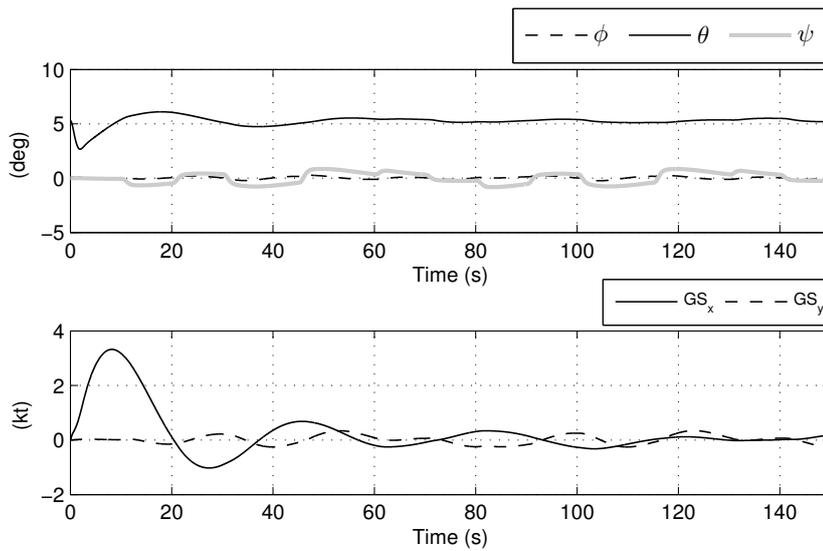


Figure 3.17: Euler Angles and Ground Speeds, Example-1

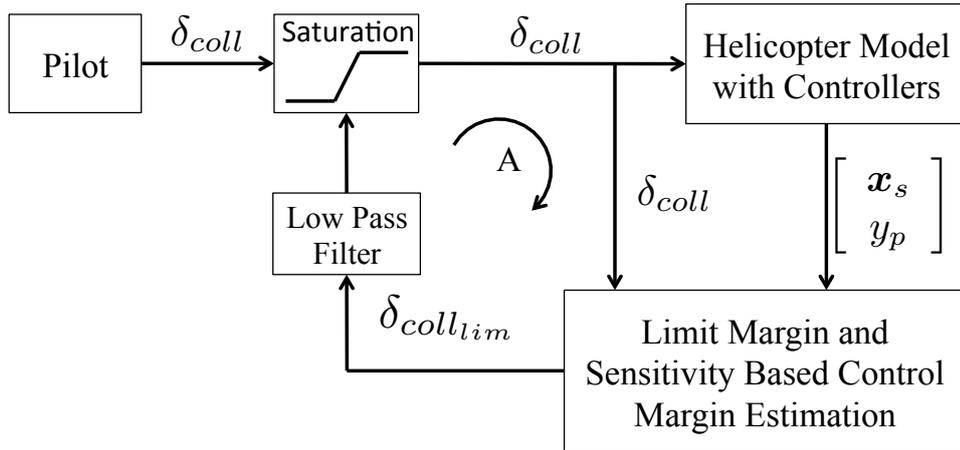


Figure 3.18: Control Limiting for Limit Avoidance

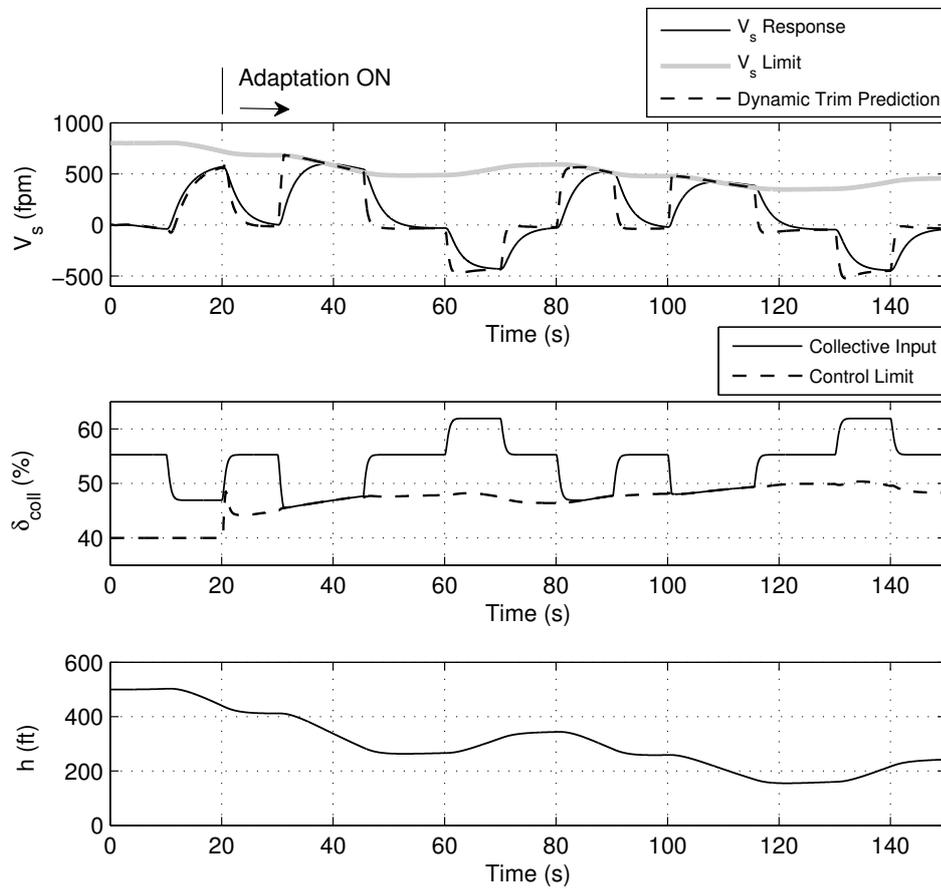


Figure 3.19: Limit Avoidance, Example-1

### 3.3.2.2 Example-2: Low Speed Forward Flight

Next, the algorithm is used in a low speed forward flight simulation. First, the simulation is initialized at the hovering condition. Then, the ground speed controller is used to bring the helicopter's ground speed to 20 kts. When  $t = 20 \text{ sec}$ , adaptation is turned ON. Data recording for concurrent learning is used from the beginning of the simulation. In Fig. 3.20, prediction results are presented. The predictions indicate the capability of the algorithm for adapting to different flight conditions.

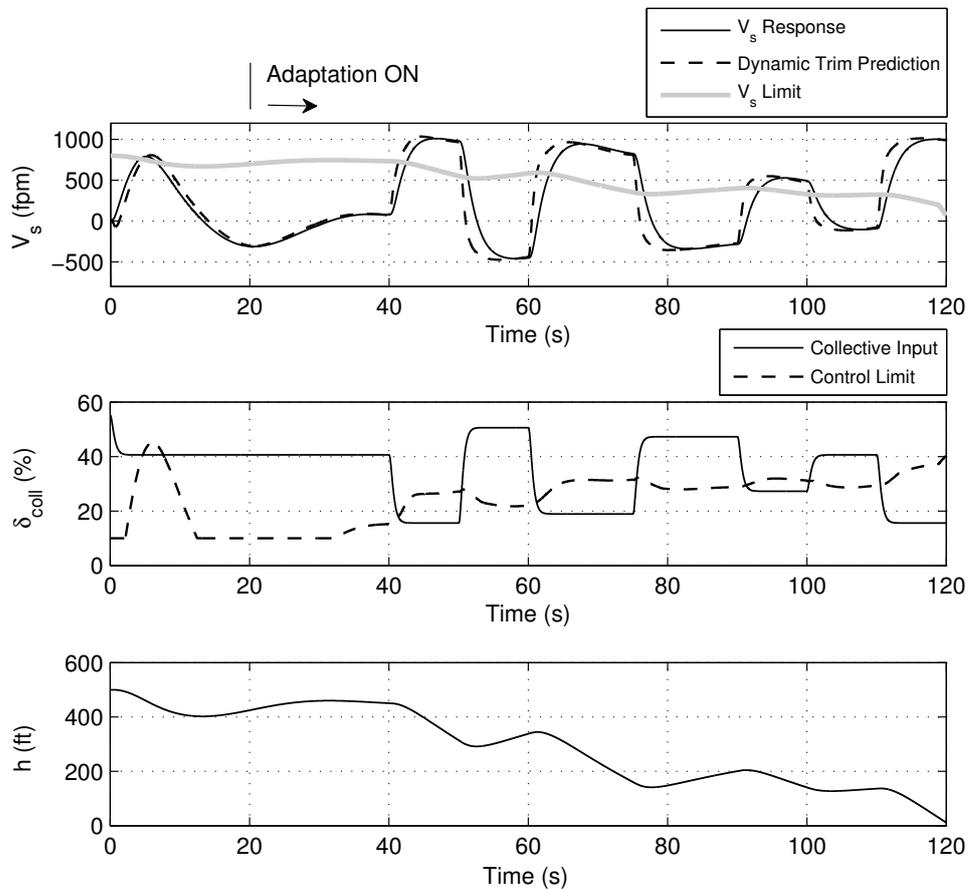


Figure 3.20: Limit Prediction, Example-2

In Fig. 3.21, the modeling error compensation, weight update time history, sensitivity estimation and singular value maximization results are presented. The fast convergence of weights to steady values are observed in the figure. Helicopter attitudes and ground speeds are shown in Fig. 3.22.

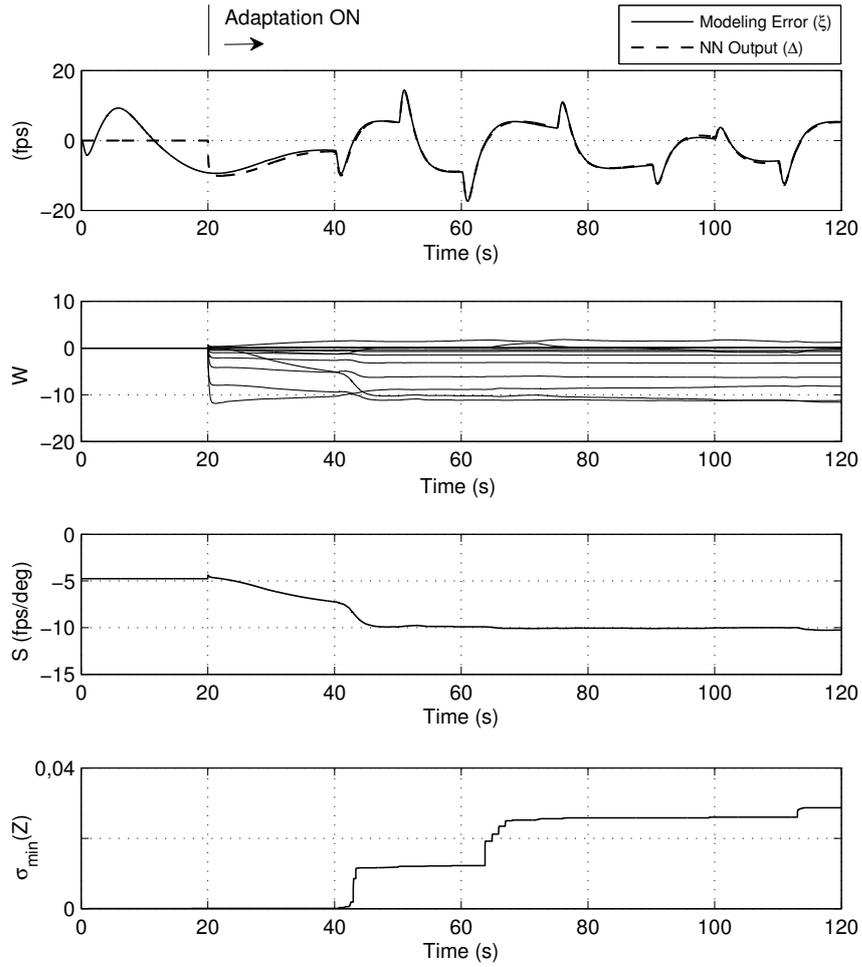


Figure 3.21: Modeling Error and NN Output, NN Weights, Predicted Collective Sensitivity, Minimum Singular Value of the History Stack, Example-2

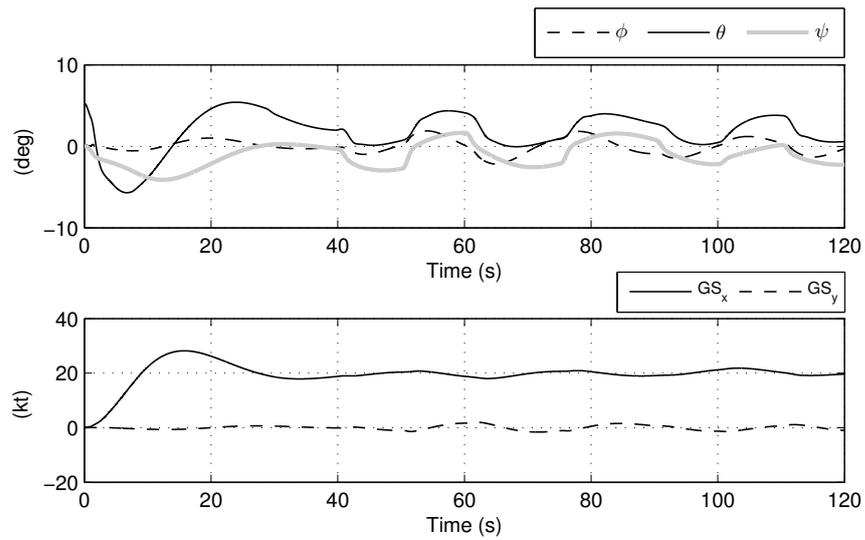


Figure 3.22: Euler Angles and Ground Speeds, Example-2

When the collective is limited by the online estimated control limits, the vertical speed is expected to be within its boundaries. The algorithm of Fig. 3.18 is used for this simulation. Results are shown in Fig. 3.23. The helicopter follows the estimated vertical speed limit when the collective control follows the control limit.

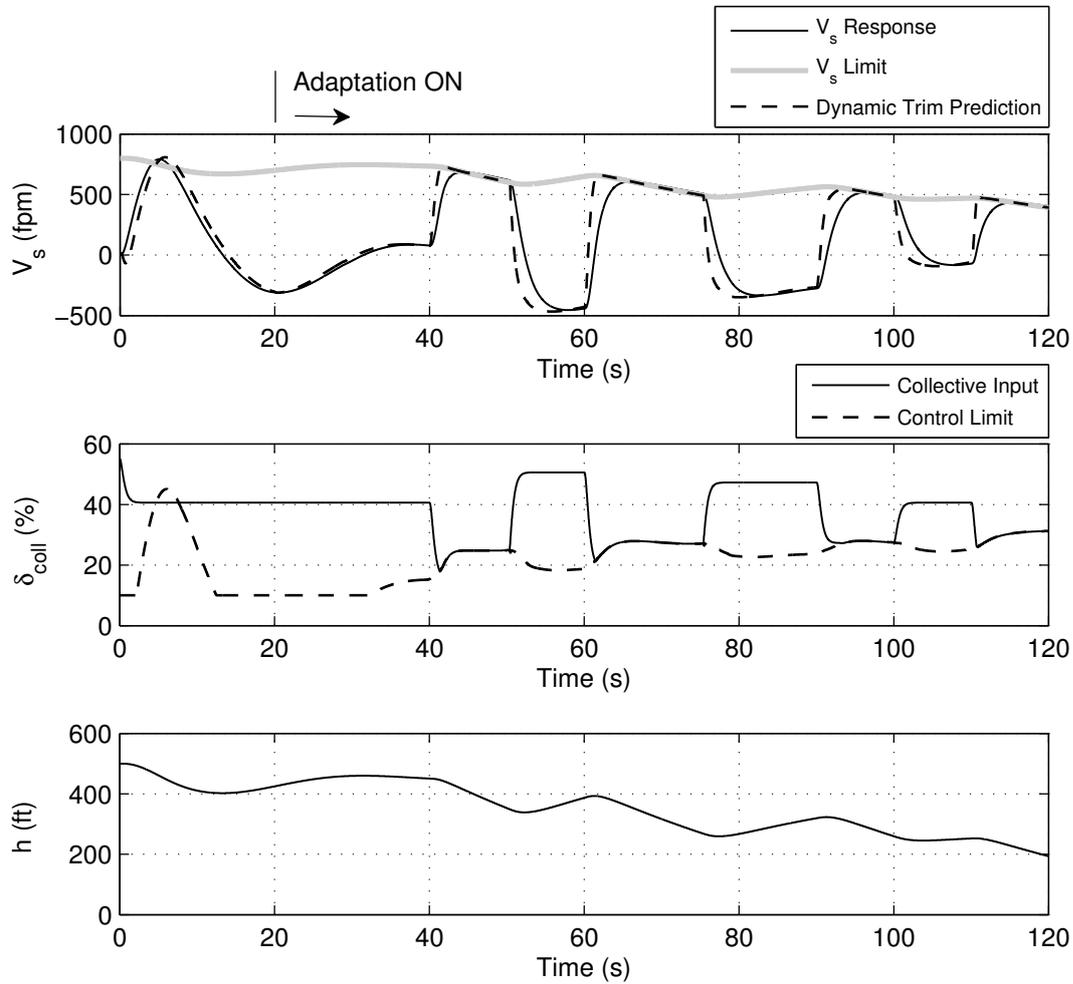


Figure 3.23: Limit Avoidance, Example-2

### 3.3.2.3 Example-3: Forward Flight

In Figures 3.24-3.26, simulation results are presented for the 60 kts forward flight condition. Here, the helicopter model is initialized at the hovering flight condition at 500 ft ground altitude and then accelerated to 60 kts forward speed using the ground speed controller. Adaptation is turned on at  $t = 20s$ , and

at  $t = 110s$  collective inputs are given to the trimmed helicopter to result in excessive vertical speed. In Figure 3.24, the vertical speed response and the steady state estimations are presented. When the steady state prediction is at the envelope boundary, the collective control input is also at the allowable control boundary. As shown in Figure 3.24, the helicopter model is close to the ground at about  $t = 200s$  with an excessive vertical speed. At the instants when the control limits are violated the pilot could have been cued through active inceptors of a fly-by-wire helicopter.

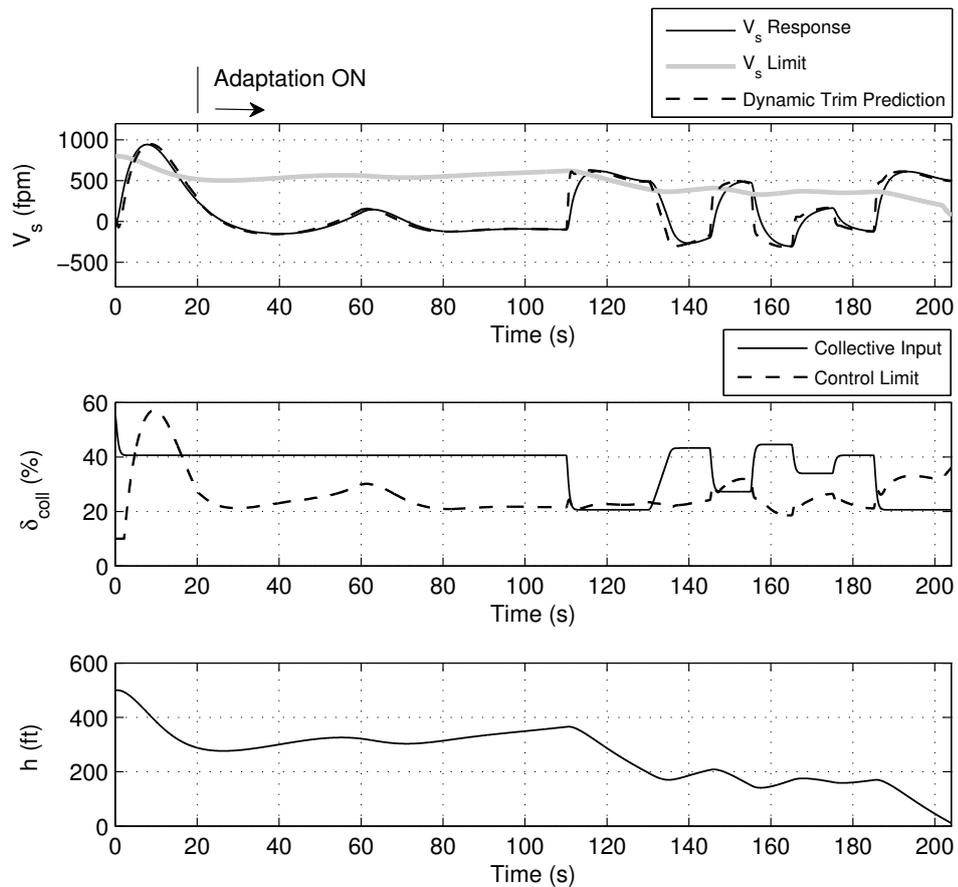


Figure 3.24: Limit Prediction, Example-3

In Figure 3.25, the estimator error compensation, adaptive weights, control sensitivity estimation and the maximized singular value of the concurrent learning adaptive law are presented. Data recording for concurrent learning is initialized at the beginning of the simulation and at  $t = 110s$  the minimum singular value increases (Figure 3.25). The recorded data in the history stack becomes different enough at that time step so that the minimum singular value is maximized.

The estimator error ( $\xi$ ) is compensated by the adaptive element ( $\Delta$ ) with acceptable errors. Note that the weights and the sensitivity estimations are slowly changing compared to the system response and the collective input, indicating a tendency to convergence steady state. Corresponding helicopter states are given in Figure 3.26. When the control input is limited to the estimated control limit, the vertical speed is expected to follow the envelope boundary. Therefore, a limit avoidance simulation is presented next in which the collective inputs are saturated with the online estimated collective limits. Results are presented in Figures 3.27 and 3.28. The vertical speed is at the limit boundary when the control is at the boundary. Although aggressive inputs are given to the helicopter, vertical speed is kept at the limit with a significant lead time.

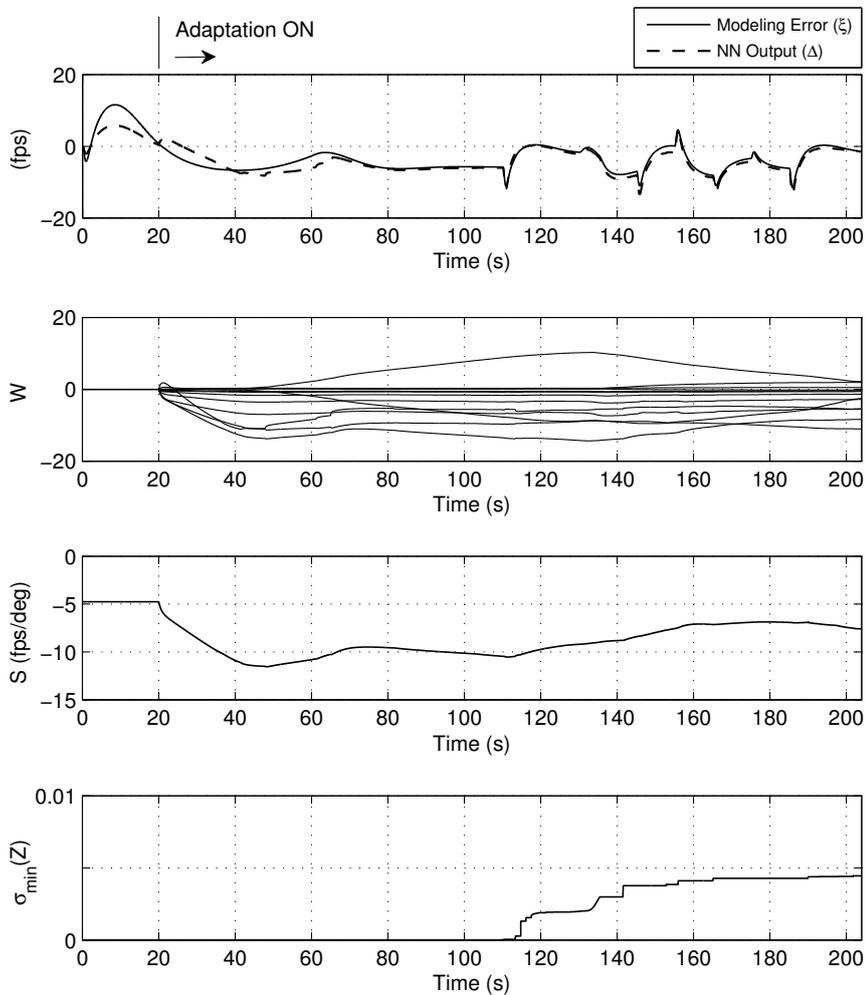


Figure 3.25: Modeling Error and NN Output, NN Weights, Predicted Collective Sensitivity, Minimum Singular Value of the History Stack, Example-3

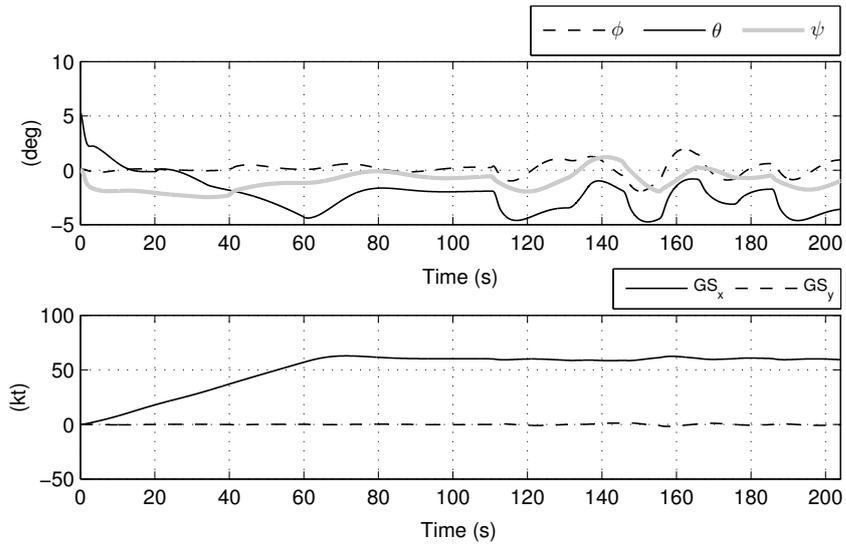


Figure 3.26: Euler Angles and Ground Speeds, Example-3

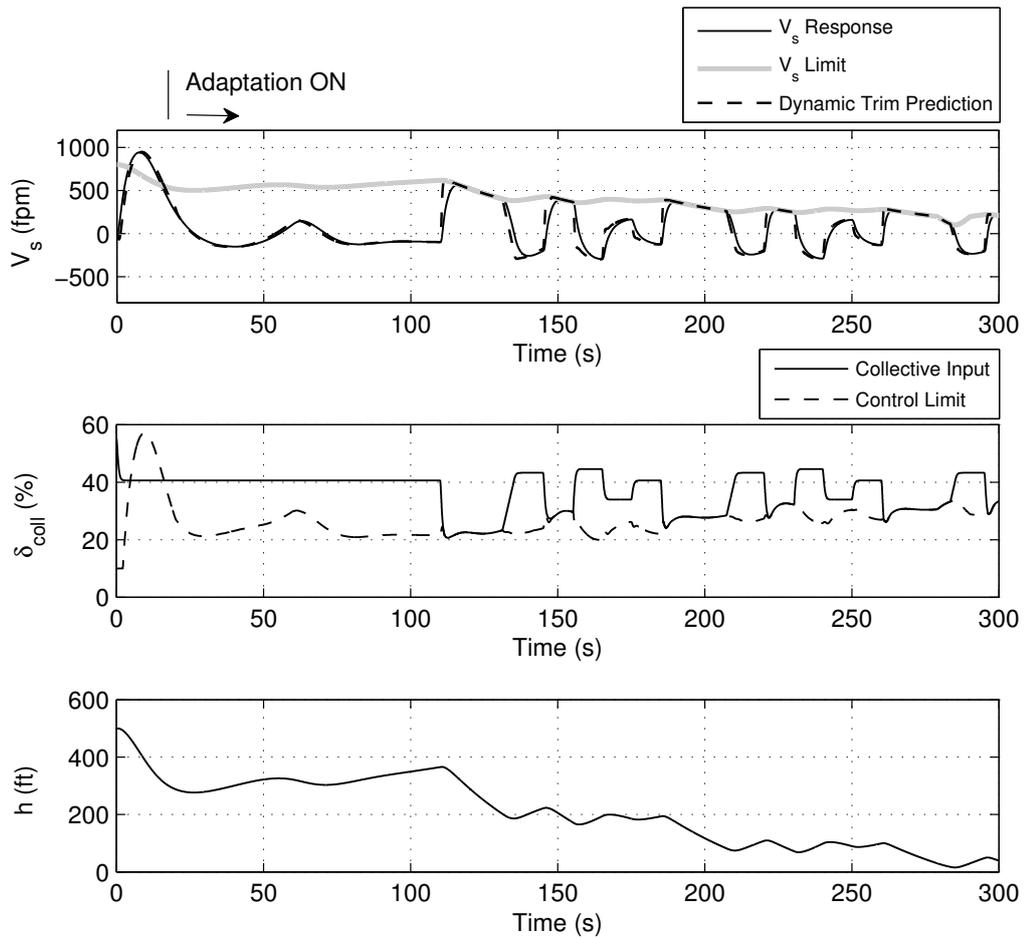


Figure 3.27: Limit Avoidance, Example-3

In Figure 3.28, estimator error compensation, adaptive weights, control sensitivity estimation and the maximized singular values of the concurrent learning adaptive law are presented for the limit avoidance simulation. Note that the weights and the estimated sensitivity are close to steady values after  $t = 250s$ , indicating a long term learning in the estimations. Therefore, a better estimation accuracy is expected over time. According to that, the estimation error is compensated with better accuracy over time (Figure 3.28). Minimum singular value of the history stack is maximized.

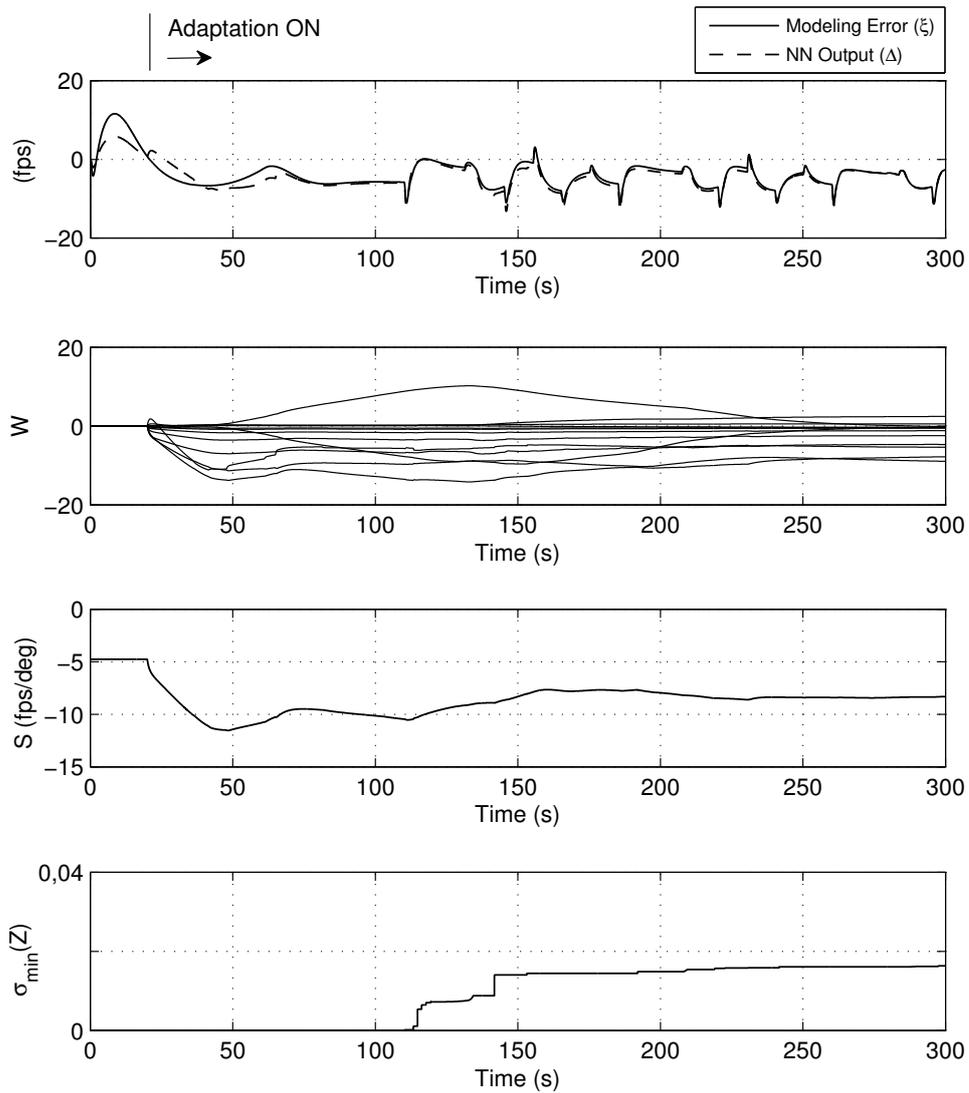


Figure 3.28: Modeling Error and NN Output, NN Weights, Predicted Collective Sensitivity, Minimum Singular Value of the History Stack, Example-3

### 3.4 Concluding Remarks

In this chapter, the previous adaptive framework for the online estimation of limit margins is improved. Sensitivity estimation based control margin estimation technique is revisited and shown to be improved using the proposed limit margin estimation methodology and concurrent learning in the weight update.

For the limit margin estimation, approximate models of fast aircraft states are constructed using online trained adaptive neural networks. A delayed moving time step is chosen for the generation of the online models. A sufficient set of central differences around the delayed moving time is used to represent the time derivatives of the fast aircraft states in the approximation. Models constructed in the past are used in the current time for the limit margin estimation. Central differences used in the approximation are set to zero at the current time to estimate the dynamic trim state. Estimations are performed without requiring online fixed point iterations. Note that this was an implementation difficulty in the previous methodologies. Since the parameters of the adaptive element are found to reach the steady values at a delayed moving time, the delayed models are found to be effective to estimate the dynamic trim state at the current time.

For the control margin estimation, the proposed online limit margin estimator is used to calculate control sensitivities at the dynamic trim condition. Estimated limit margins and control sensitivities are used to construct control margins. The sensitivity estimation based control margin estimation method is found to be effective since the control sensitivity estimations are observed to have approximately constant values during aggressive maneuvering. No large changes in the sensitivity estimations are obtained as it was a drawback of the previous methods based on control sensitivity estimations. Therefore in this chapter, previous shortcomings of the sensitivity estimation based methods in the literature are also improved using concurrent learning in the adaptive element.

In the methodology, linear neural network structures, i.e. LPNNs, are used along with concurrent learning in the weight update. For concurrent adaptation, the data that increase the minimum singular value of a history stack are recorded

online. Minimum singular value maximization method is shown to locate the adaptive weights around optimal values. A Lyapunov based proof is presented in the Appendix to show the ultimate bounds of the weight errors of the proposed estimators. It is shown that the bounds on the weight errors can be decreased even further if the minimum singular value of the data history stack is maximized online. Therefore, using concurrent learning in the weight update, bounded control sensitivities and adaptive weights in a compact neighborhood of the optimal values are obtained and used online to construct limit and control margin estimations.

The effectiveness of the proposed methodologies is evaluated in simulations using fixed wing and rotary wing aircraft dynamic models. Critical parameters such as; load factor, angle of attack and vertical speed are considered for the implementation of the methodologies. It is shown that limit margins can be estimated with sufficient lead times to react on approaching angle of attack, load factor and vertical speed limit boundaries. The control sensitivities are shown to be accurately estimated using the limit margin estimator in which the estimator weights are ultimately bounded. Using concurrent learning in the weight update, sensitivity estimation based control margin estimation methodology is shown to be effective. Also, using the minimum singular value maximization to record necessary data for concurrent adaptation, the sensitivity estimations are shown to reach the expected (physically correct) values fast.

Estimated limit and control margins are verified using control input saturation at the limit boundary. It is observed that the parameters stay at the limit boundary, enabling carefree maneuvering, when the control inputs are saturated artificially with the estimated control limits. No chattering is observed at the limit boundary during control input saturation. Note that the chattering was a drawback of the previous adaptive frameworks due to the oscillations triggered by instantaneous learning during the control input saturation.

In the proposed methodologies, the limit margin and the control margin estimations are directly related using the estimated control sensitivities, i.e. linear algebraic relations are established between the two critical margins. Therefore in

the proposed approach, a control margin is guaranteed to be zero only when an associated limit margin become zero. That constraint is beneficial since, in theory, both margins are expected to be zero, that is at the limit boundary, at the same time instant. Using the sensitivity and the limit margin estimations, that theoretical constraint is made practical in the proposed methodologies. Whereas in the previous adaptive frameworks for envelope protection, no constraints have been established between the two critical margins. Therefore, using the previous approach it is possible for both margins to be at the limit boundary at different time instants, which can be confusing while avoiding the limits or establishing mathematical rules for limit avoidance.

Based on the simulation results, the new adaptive framework is shown to be effective to estimate limit and control margins to be used to warn pilots at the onset of the limiting flight conditions. Note that the limit margin cues can be displayed through cockpit visuals or given through aural indications for pilot cueing. Whereas, the control margin cues can be given to pilots through the active controls in a fly-by-wire aircraft.



## CHAPTER 4

# DIRECT ADAPTIVE CONTROL MARGIN ESTIMATION

In this chapter, an adaptive control margin estimation methodology, that doesn't require online fixed point iterations, is proposed for the task of envelope cueing for Fly-By-Wire aircraft. Compared to the previous chapter, limit margin and control sensitivity estimations are not required for the estimation of the control margins in this chapter. Fixed and rotary wing aircraft dynamic models are used to show the effectiveness in the simulation.

### 4.1 Methodology

Using the plant definitions of the previous chapter, which are Eqs. (3.1) and (3.2), and the assumptions made in Section 3.1, dynamics of the fast aircraft states and the dynamic trim condition are redefined here as

$$\dot{\boldsymbol{x}}_f = \boldsymbol{f}_1(\boldsymbol{x}_f, \boldsymbol{x}_s, \boldsymbol{u}), \quad (4.1)$$

$$\dot{\boldsymbol{x}}_f = 0. \quad (4.2)$$

#### 4.1.1 Direct Adaptive Control Margin Estimation

In this section, an inverse modeling approach is used to estimate the control margins similar to the limit margin estimations. In order to find control margins on a single control axis, the control of interest,  $u_e \in \mathfrak{R}$ , can be formulated using

Eq. (4.1) with the following representation:

$$u_e = f_1^{-1}(\dot{\mathbf{x}}_f, \mathbf{x}_f, \mathbf{x}_s, \mathbf{u}'). \quad (4.3)$$

Here,  $\mathbf{u}' \in \mathfrak{R}^{p-1}$  is the control input vector excluding  $u_e$  and  $f_1^{-1}$  is an inverse of the actual plant dynamics. When the relation given by Eq. (3.23) is inverted such that  $\mathbf{x}_f = g^{-1}(y_p, \mathbf{x}_s)$ , then Eq. (4.3) can be approximated as a function of  $y_p$  as:

$$u_e = \hat{f}_1^{-1}(\dot{\mathbf{x}}_f, y_p, \mathbf{x}_s, \mathbf{u}') + \xi. \quad (4.4)$$

where,  $\hat{f}_1^{-1}$  is an approximate inversion and  $\xi$  is the modeling error. Using the neural network function approximation,  $\Delta$ , the following estimation to  $u_e$  can be written:

$$\hat{u}_e = \hat{f}_1^{-1}(\dot{\mathbf{x}}_f, y_p, \mathbf{x}_s, \mathbf{u}') + \Delta(\dot{\mathbf{x}}_f, y_p, \mathbf{x}_s, \mathbf{u}'), \quad (4.5)$$

and, subtracting Eq. (4.5) from (4.4), the function approximation error,  $e$ , is obtained:

$$e = \xi - \Delta(\dot{\mathbf{x}}_f, y_p, \mathbf{x}_s, \mathbf{u}'). \quad (4.6)$$

Assumption (4.1): The limit parameter, fast states, slow states and control inputs, that is the signals  $y_p$ ,  $\mathbf{x}_f$ ,  $\mathbf{x}_s$  and  $\mathbf{u}$ , are known, i.e. measured, and sampled with the same sampling rate.

Assumption (4.2): Sampling rate of the signals,  $y_p$ ,  $\mathbf{x}_f$ ,  $\mathbf{x}_s$  and  $\mathbf{u}$ , is fixed and chosen as sufficient as to approximate the considered pilot control input ( $u_e$ ).

Assumption (4.3): A moving time window, i.e. delay line, containing the previous consecutive samples of the signals  $y_p$ ,  $\mathbf{x}_f$ ,  $\mathbf{x}_s$  and  $\mathbf{u}$ , is available at the time of current sampling. Therefore, the following matrix is constructed as the delay line matrix:

$$DL = \begin{bmatrix} y_{p_{t_c-1}} & y_{p_{t_c-2}} & \cdots & y_{p_{t_c-w}} \\ \mathbf{x}_{f_{t_c-1}} & \mathbf{x}_{f_{t_c-2}} & \cdots & \mathbf{x}_{f_{t_c-w}} \\ \mathbf{x}_{s_{t_c-1}} & \mathbf{x}_{s_{t_c-2}} & \cdots & \mathbf{x}_{s_{t_c-w}} \\ \mathbf{u}_{t_c-1} & \mathbf{u}_{t_c-2} & \cdots & \mathbf{u}_{t_c-w} \end{bmatrix}, \quad (4.7)$$

in which,  $t_c$  is the time index for the current sampling and  $t_c - w$  is the index of the most previous data. Note that,  $DL \in \mathfrak{R}^{(n+p+1) \times w}$  and  $w \in \mathfrak{N}^+$  is the number

of samples in the delay line. Also, a chosen column of  $DL$  in Eq. (4.7) represents a delayed moving time step, as in *Remark (3.1)*.

When Eqs. (4.5) and (4.6) are established at a delayed moving time step,  $d$ , the derivatives,  $\dot{\mathbf{x}}_{fd}$ , can be represented using a central difference operator as done in the previous chapter. Therefore, using Eqs. (3.12) and (3.13), the following input estimator can be constructed:

$$\hat{u}_{ed} = \hat{f}_1^{-1}(\bar{\partial}(\mathbf{x}_f)_d, y_{pd}, \mathbf{x}_{sd}, \mathbf{u}'_d) + \Delta(\bar{\partial}(\mathbf{x}_f)_d, y_{pd}, \mathbf{x}_{sd}, \mathbf{u}'_d). \quad (4.8)$$

The model tracking error,  $e_d$ , can now be written at the delayed moving time step as:

$$e_d = \xi_d - \Delta(\bar{\partial}(\mathbf{x}_f)_d, y_{pd}, \mathbf{x}_{sd}, \mathbf{u}'_d). \quad (4.9)$$

*Assumption (4.4)*: In Eq. (4.8), the approximate inversion  $\hat{f}_1^{-1}$  is augmented with  $\Delta$ . The resulting operator,  $(\hat{f}_1^{-1} + \Delta)$ , is a one-to-one mapping such that  $(\hat{f}_1^{-1} + \Delta) : \mathfrak{R}^{l(k-1)+n+p} \rightarrow \mathfrak{R}$  exists and is unique.

*Remark (4.1)*: *Assumption (3.9)* of the previous chapter is required hereafter.

Using the dynamic trim condition,  $\bar{\partial}(\mathbf{x}_f)_t = 0$ , along with Eqs. (4.8) and (4.9), the control limit,  $\hat{u}_{elim}$ , for a given output limit,  $y_{plim}$ , can be estimated at the current time step as

$$\hat{u}_{elim} = \hat{f}_1^{-1}(0, y_{plim}, \mathbf{x}_s, \mathbf{u}') + \Delta(0, y_{plim}, \mathbf{x}_s, \mathbf{u}') + e_d. \quad (4.10)$$

In practice, Eq. (4.10) is used to estimate  $u_{elim}$  with  $y_{plim}$ , the known limit boundary of the aircraft. Then, the control margin,  $u_{emarg}$ , can be calculated:

$$u_{emarg} = \hat{u}_{elim} - u_e. \quad (4.11)$$

In Fig. 4.1 the block diagram of the control margin estimation algorithm is presented. The dashed lines represent the variables or calculations of the delayed moving time step,  $d$ , and the solid lines are the ones that run at the current time step.

If a reduced order linear plant model,  $\dot{x}_f = A_{11}x_f + B_{11}u_e$ , is available and  $y_p = x_f$ , then the approximate inversion,  $\hat{f}_1^{-1}$ , can be obtained by inverting similar to  $u_e = B_{11}^{-1}(\dot{y}_p - A_{11}y_p)$ .

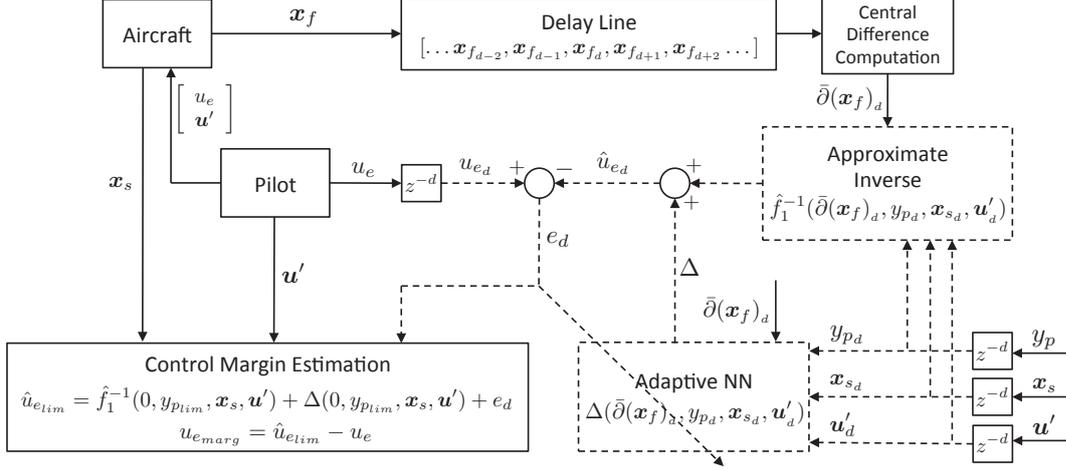


Figure 4.1: Direct Adaptive Control Margin Estimation

The methodology presented here does not require sensitivity and limit margin estimations to obtain control margins. Also no iterations are required. Therefore the method is called, the *Direct Adaptive Control Margin Estimation*.

### Relative Degree Formulation

Using Eqs. (3.23), (3.24) and (3.25) of Section (3.1.1), an alternative methodology can be developed for the task of control margin estimation.

In Eq. (3.25),  $u_e$  is the effective control input that results in a flight envelope violation of the limit parameter  $y_p$ . Hereafter, the goal is to establish a functional approximation between  $u_e$  and  $y_p$ . When Eq. (3.24) is inverted,  $u_e$  can be written as in Eq. (4.12).

$$u_e = g_n^{-1}(\mathbf{x}_s, y_p, y_p^{(1)}, y_p^{(2)}, \dots, y_p^{(n)}) \quad (4.12)$$

Here,  $g_n^{-1}$  is the actual inverse of the plant dynamics and is an unknown function. Using an approximate model,  $\hat{g}_n$ ,  $u_e$  can be written as a summation of the approximation and a modeling error,  $\xi$ :

$$u_e = \hat{g}_n^{-1}(\mathbf{x}_s, y_p, y_p^{(1)}, y_p^{(2)}, \dots, y_p^{(n)}) + \xi. \quad (4.13)$$

In Eq. (4.13),  $\mathbf{x}_s$  represents slow aircraft states that might be required for the approximation. If we let  $\mathbf{y}_p = [y_p \ y_p^{(1)} \ \dots \ y_p^{(n-1)}]^T \in \mathfrak{R}^n$  then the derivatives in

Eq. (4.13) can be written as

$$\dot{\mathbf{y}}_p = [y_p^{(1)} \ y_p^{(2)} \ \dots \ y_p^{(n)}]^T \in \mathfrak{R}^n, \quad (4.14)$$

and using an adaptive element,  $\Delta$ , the following estimation to  $u_e$  can be written:

$$\hat{u}_e = \hat{g}_n^{-1}(\mathbf{x}_s, y_p, \dot{\mathbf{y}}_p) + \Delta(\mathbf{x}_s, y_p, \dot{\mathbf{y}}_p). \quad (4.15)$$

Remark (4.2): Assumptions (3.6-3.10) of the previous chapter are required hereafter.

Note that, Eq. (4.15) requires the derivative signals  $\dot{\mathbf{y}}_p$  to be available. Similar to Eq. (3.30) of Section (3.1.1), the estimator of Eq. (4.15) can be established at a delayed moving time step,  $d$ , as follows:

$$\hat{u}_{e_d} = \hat{g}_n^{-1}(\mathbf{x}_{s_d}, y_{p_d}, \dot{\mathbf{y}}_{p_d}) + \Delta(\mathbf{x}_{s_d}, y_{p_d}, \dot{\mathbf{y}}_{p_d}). \quad (4.16)$$

where, the subscript,  $d$ , represents the variables of a delayed moving time step.

Note that the derivatives,  $\dot{\mathbf{y}}_{p_d}$ , can be written as a function of available central differences around the delayed moving time step,  $d$ . As in Eq. (3.32) of Section (3.1.1), the same operator can be used here to take the limit parameter,  $y_p$ , at time  $d$  and output a matrix of central differences around that time. Therefore, Eq. (4.16) can now be written as a function of central differences as

$$\hat{u}_{e_d} = \hat{g}_n^{-1}(\mathbf{x}_{s_d}, y_{p_d}, \bar{\partial}(y_p)_d) + \Delta(\mathbf{x}_{s_d}, y_{p_d}, \bar{\partial}(y_p)_d). \quad (4.17)$$

Subtracting Eq. (4.17) from  $u_{e_d} = \hat{g}_n^{-1}(\mathbf{x}_{s_d}, y_{p_d}, \bar{\partial}(y_p)_d) + \xi_d$ , the delayed approximation error,  $e_d$ , is obtained as

$$e_d = \xi_d - \Delta(\mathbf{x}_{s_d}, y_{p_d}, \bar{\partial}(y_p)_d). \quad (4.18)$$

Assumption (4.5): In Eq. (4.17), the approximate inversion,  $\hat{g}_n^{-1}$ , is augmented with  $\Delta$ . The resulting operator,  $(\hat{g}_n^{-1} + \Delta)$ , is a one-to-one mapping such that  $(\hat{g}_n^{-1} + \Delta) : \mathfrak{R}^{n(k+1)-l+1} \rightarrow \mathfrak{R}$  exists and is unique.

Since the derivatives are represented with central differences,  $\bar{\partial}(y_p)_d$  is expected to be zero at the maneuvering steady state condition. Hence, the maneuvering steady state condition,  $\dot{\mathbf{y}}_p = 0$ , also indicates

$$\bar{\partial}(y_p)_t = 0. \quad (4.19)$$

Therefore, using Eq. (4.19), along with Eqs. (4.17) and (4.18), the control limit,  $\hat{u}_{e_{lim}}$ , for a given limit,  $y_{plim}$ , can be estimated at the current time step,  $t$ , as

$$\hat{u}_{e_{lim}} = \hat{g}_n^{-1}(\mathbf{x}_s, y_{plim}, 0) + \Delta(\mathbf{x}_s, y_{plim}, 0) + e_d. \quad (4.20)$$

In Eq. (4.20), the delayed approximation error,  $e_d$ , is used at the current time step. Note that this is applicable when the network weights reach a compact neighborhood of the ideal weights.

The control margin,  $u_{e_{marg}}$ , can be calculated as

$$u_{e_{marg}} = \hat{u}_{e_{lim}} - u_e. \quad (4.21)$$

A block diagram representation of the proposed control margin estimation algorithm is presented in Fig. 4.2. Dashed lines represent the values of the variables or calculations that take place at the delayed moving time step and the straight lines are the calculations of the current time step.

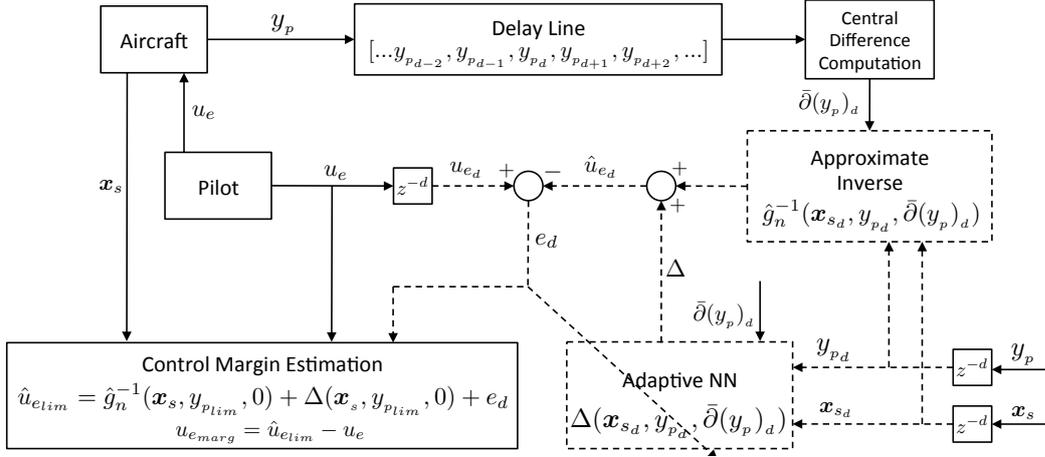


Figure 4.2: Direct Adaptive Control Margin Estimation with Relative Degree Formulation

It should be noted that the methodologies presented in [13, 27, 28] and [32] as well as the approaches proposed in the previous chapter and in this chapter are applicable for the aircraft states or limit parameters that reach their limiting value in the steady state response as opposed to the ones that reach their limiting values in their transient response. Using Eqs. (4.10), (4.11), (4.20) and (4.21) control limits and control margins can be estimated.

## 4.2 Online Model Generation

In this chapter, an effective pilot control input,  $u_e$ , is aimed to be modeled online. Therefore, the modeling errors are in that control input channel and the related online estimators, Eqs. (4.8) and (4.17), are established to cancel out the modeling errors. Note that the neural network augmentation of the errors is similar to the implementation presented in Section 3.2, hence, the reader should refer to Section 3.2 for the online compensation of the modeling errors.

## 4.3 Implementation and Simulation Results

In the simulation examples of this chapter, LPNNs are used in both estimators to cancel out the modeling errors. Also, the data for concurrent learning is recorded through maximizing the minimum singular values of the history stacks. Next, the implementation of the proposed algorithms for the fixed wing and rotary wing aircraft models is presented.

### 4.3.1 Flight Envelope Protection for Aircraft

In this example, simulation scenarios of Section 3.3.1 of the previous chapter are repeated for the same aircraft model. The *Direct Adaptive Control Margin Estimation* methodology of this chapter is used to calculate related control margins.

The following online models are constructed at the delayed time step  $d$  using Eq. (4.8):

$$\hat{\delta}_{e_{1d}} = (1/C_1)(C_2[\partial_\alpha \ \partial_q]^T - \alpha_d) + \Delta_\alpha(\bar{\partial}(\mathbf{x}_f)_d, \alpha_d, V_{e_d}, \theta_d, b_1), \quad (4.22)$$

$$\hat{\delta}_{e_{2d}} = (1/C_3)(C_4[\partial_\alpha \ \partial_q]^T - n_{z_d}) + \Delta_{n_z}(\bar{\partial}(\mathbf{x}_f)_d, n_{z_d}, V_{e_d}, \theta_d, b_1). \quad (4.23)$$

Here,  $\partial_\alpha$  and  $\partial_q$  are constructed using Eq. (3.60) and  $\bar{\partial}(\mathbf{x}_f)_d$  is obtained using Eq. (3.13), in which  $\mathbf{x}_{f_d} = [\alpha_d \ q_d \ n_{z_d}]^T \in \mathfrak{R}^3$  and  $k = 4$ . Note that, the first terms of Eqs. (4.22) and (4.23) are obtained using reduced order linear models.

In Eq. (4.22), the angle of attack is used as the limit parameter, and in Eq. (4.23), the limit parameter is the the load factor.

The following bases are constructed for the neural networks given in Eqs. (4.22) and (4.23), respectively:

$$\begin{aligned}\phi_\alpha(13 : 18) &= [\phi_{13}(\alpha) \ \phi_{14}(V_e) \ \phi_{15}(\theta) \ \phi_{16}(\alpha V_e) \ \phi_{17}(\alpha\theta) \ b_2]^T, \\ \phi_{n_z}(13 : 18) &= [\phi_{13}(n_z) \ \phi_{14}(V_e) \ \phi_{15}(\theta) \ \phi_{16}(n_z V_e) \ \phi_{17}(n_z\theta) \ b_3]^T,\end{aligned}\quad (4.24)$$

where, the first 12 elements of each basis,  $\phi_\alpha$  and  $\phi_{n_z}$ , are constructed similar to Eq. (3.62).

Control limits are estimated at the current time step using Eqs. (4.22) and (4.23) at the dynamic trim condition ( $\bar{\partial}(\mathbf{x}_f)_d = 0$ ,  $\partial_\alpha = \partial_q = 0$ ) and by substituting the envelope limits into the equations as:

$$\begin{aligned}\hat{\delta}_{e_{1_{lim}}} &= (1/C_1)(-\alpha_{lim}) + \Delta_\alpha(0, \alpha_{lim}, V_e, \theta, b_1) + e_{1_d}, \\ \hat{\delta}_{e_{2_{lim}}} &= (1/C_3)(-n_{z_{lim}}) + \Delta_{n_z}(0, n_{z_{lim}}, V_e, \theta, b_1) + e_{2_d}.\end{aligned}\quad (4.25)$$

in which,  $e_{1_d}$  and  $e_{2_d}$  are delayed model tracking errors of each estimator as in Eq. (4.10), and calculated at the delayed moving time as

$$e_{1_d} = \delta_{e_d} - \hat{\delta}_{e_{1_d}}, \quad e_{2_d} = \delta_{e_d} - \hat{\delta}_{e_{2_d}} \quad (4.26)$$

#### 4.3.1.1 Example-1: Longitudinal Pull-up & Push-over

In Fig. 4.3, control margin predictions for the angle of attack and the load factor response are presented along with the airspeed response. As in the examples of the previous chapter, the limit boundaries on the control travel are estimated prior to actual limit violations. Here, the estimation of control limits do not require the estimation of limit margins or control sensitivities. Yet, results are similar to the ones shown in Fig. 3.4 of Section 3.3.1.

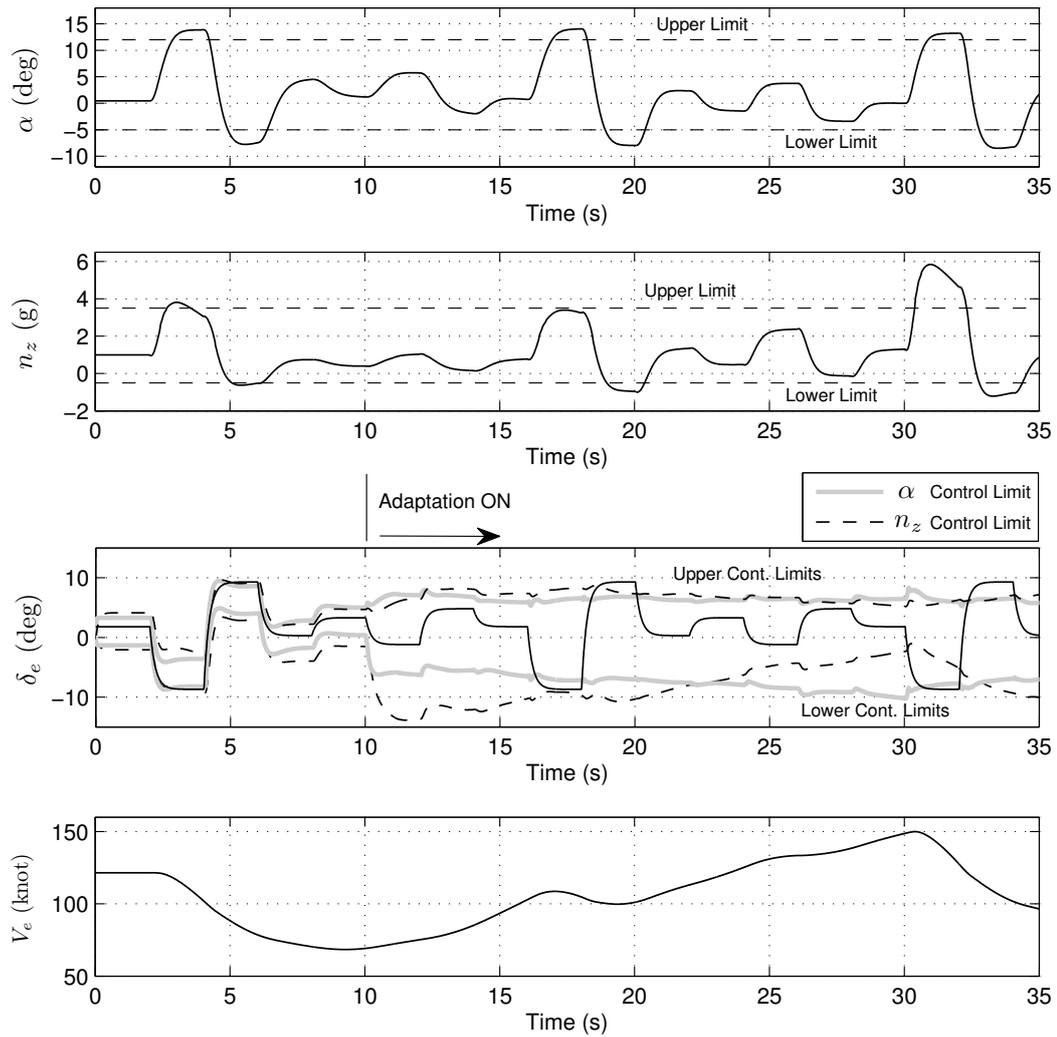


Figure 4.3: Limit Prediction, Example-1

In Fig. 4.4, the modeling error compensation and the maximized minimum singular values are shown. Both online models have reasonable tracking performance where the approximation error is kept within sufficient bounds. Minimum singular values are maximized in time. At each increment in Fig. 4.4 a new basis is recorded into the history stacks.

In Fig. 4.5, the neural network weight update time histories are shown. Note that in these results and in the results of Section 3.3.1, the weights have a tendency to move slowly compared to the changes in other flight parameters. This indicates a long term adaptation in the adaptive element.

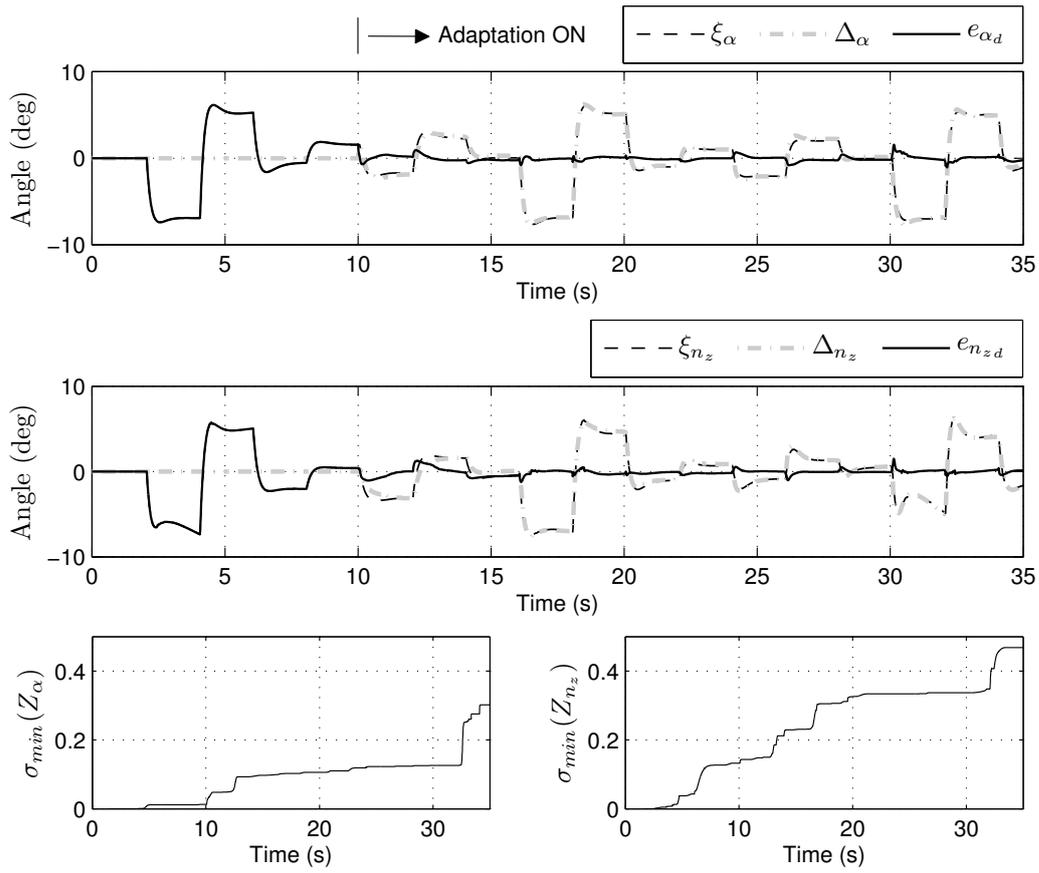


Figure 4.4: Modeling Error Compensation and Minimum Singular Values, Example-1

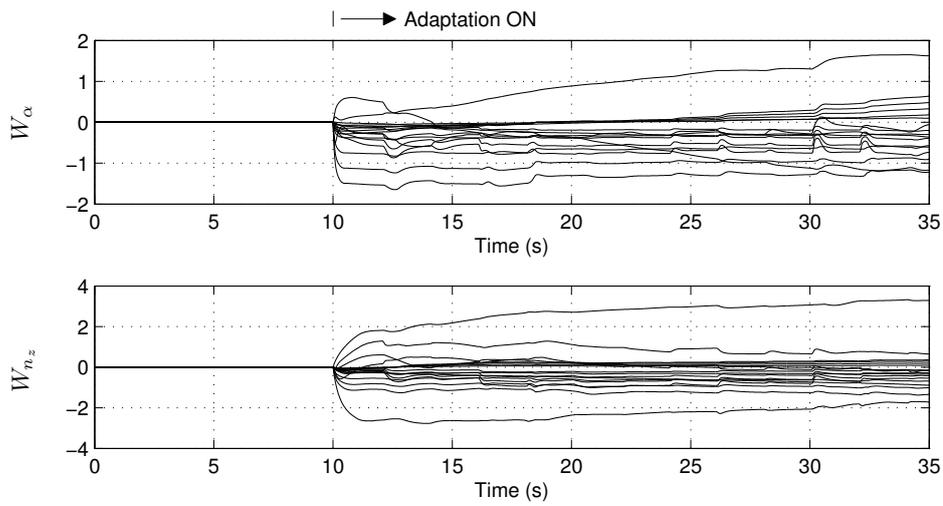


Figure 4.5: Weight Update Time Histories, Example-1

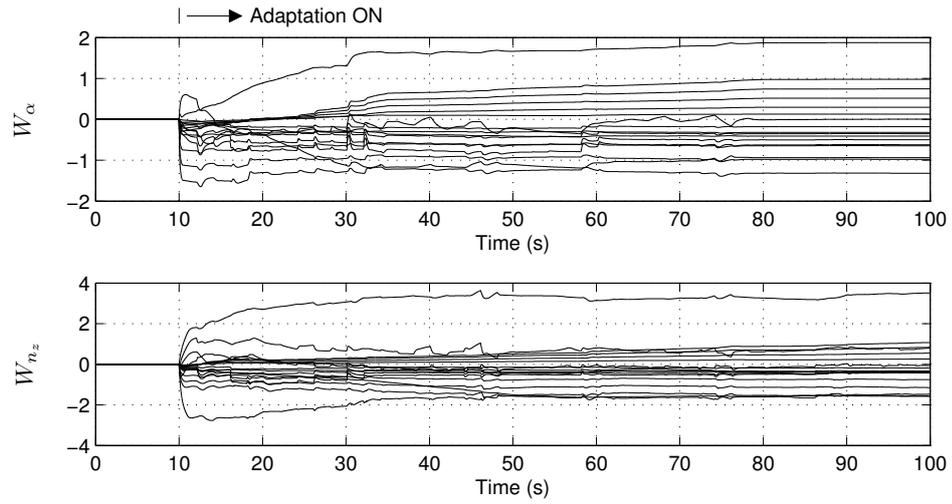


Figure 4.6: Weight Update Time Histories, Long Term Response, Example-1

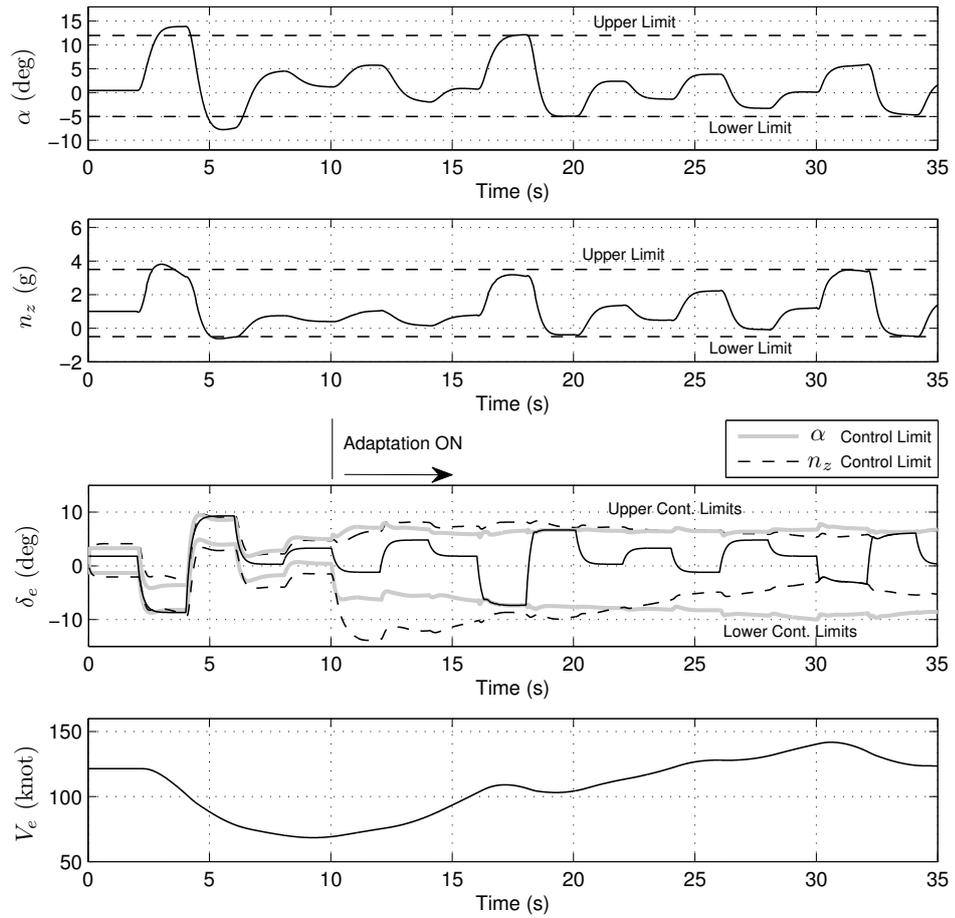


Figure 4.7: Limit Avoidance, Example-1

Next, the maneuver is repeated for a control input that repeats itself after 35s until about 100 seconds to demonstrate the long term response of the weight update. The weight update time history is shown in Fig. 4.6, also further indicating a tendency of steady state convergence in the long term.

Limit avoidance is achieved by saturating the controls at the estimated control limits. Results are presented in Fig. 4.7. The limiting states are also at the envelope boundary when the control is saturated.

#### 4.3.1.2 Example-2: Pull-up & Push-over during a Turn

In this example, the simulation scenario of Example 2 of Section 3.3.1 is repeated. Weights and the history stacks obtained at the end of Fig. 4.5 are used as the initial conditions in this example. Estimated control limits are shown in Fig. 4.8.

Limit avoidance is applied and results including the aircraft response are shown in Fig. 4.9. In the simulation, the aircraft is kept in the flight envelope.

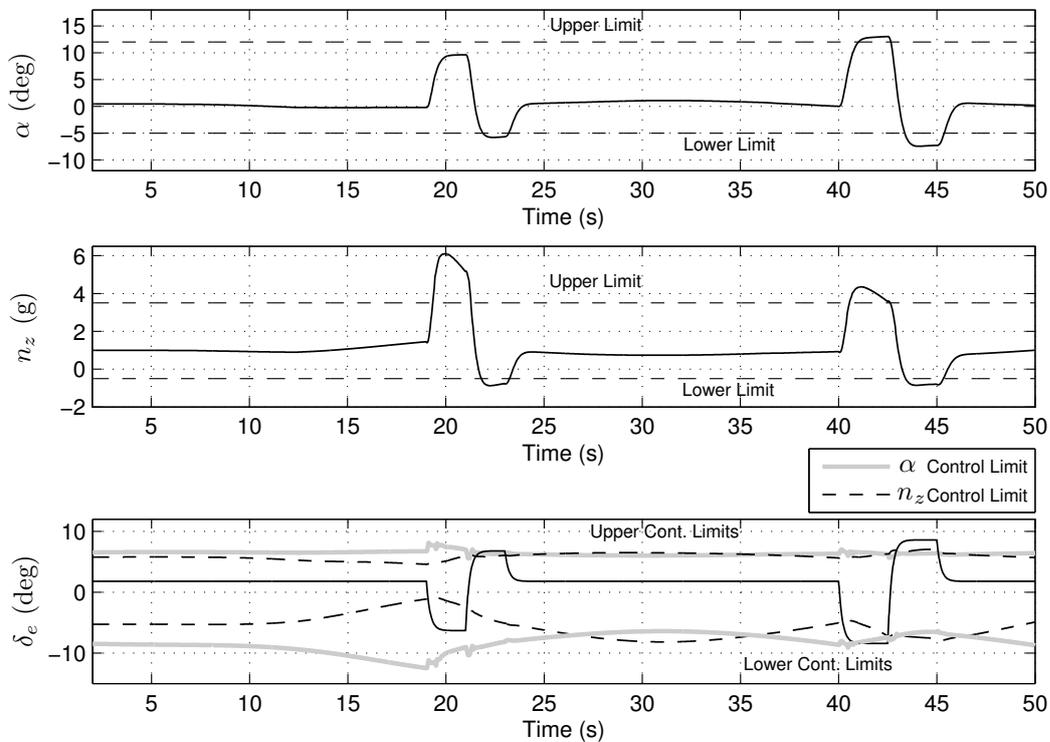


Figure 4.8: Limit Prediction, Example-2

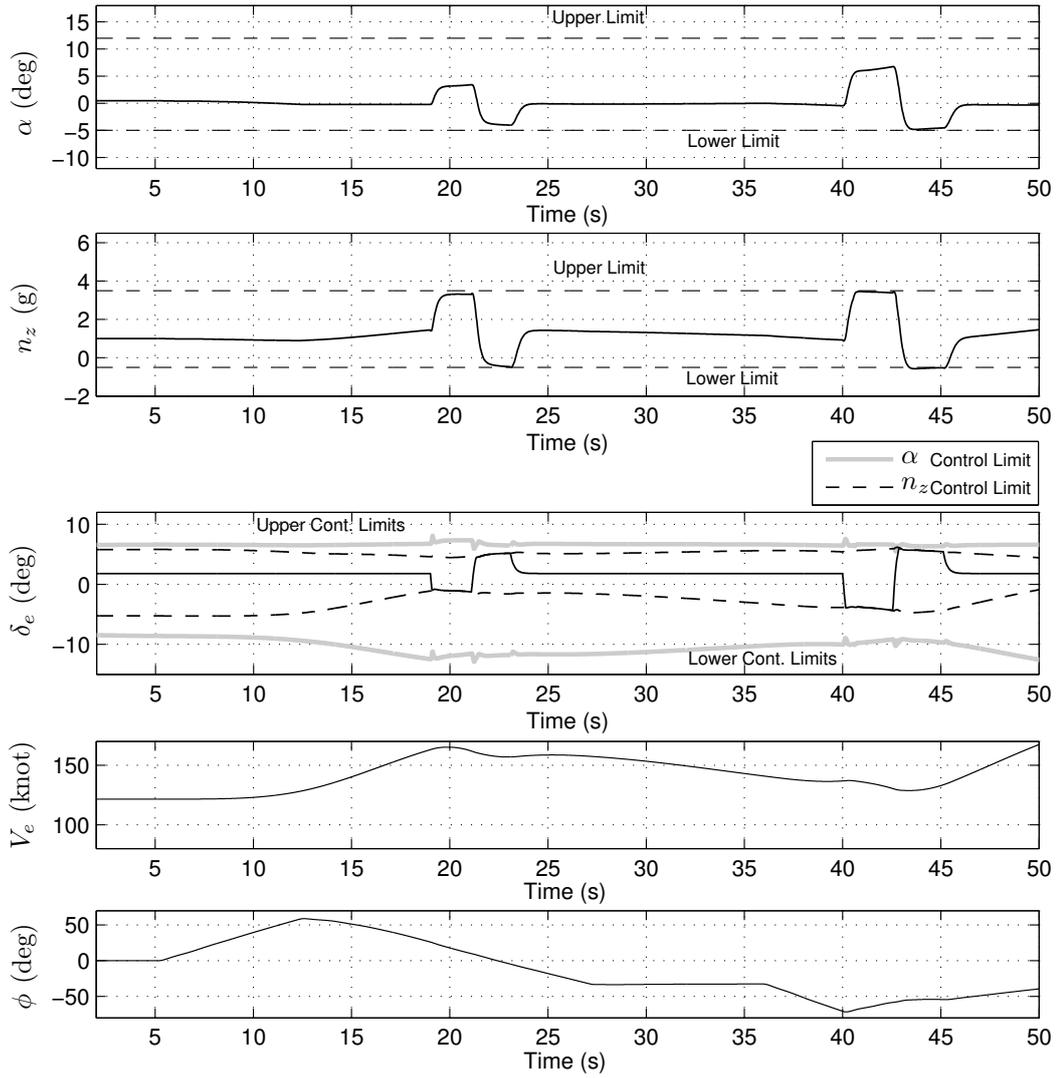


Figure 4.9: Limit Avoidance, Example-2

### 4.3.2 Vertical Speed Limit Protection Helicopters

In this section, collective control limits for a helicopter operating in hover and forward flight conditions are estimated online using the *Direct Adaptive Control Margin Estimation* methodology. For the implementation, Eqs. (4.12), (4.21) and between are used.

For the online approximation of the collective input ( $\delta_{coll}$ ), the parameters such as; vertical speed ( $V_s$ ), airspeed ( $V_e$ ), height above terrain ( $h$ ), pitch angle ( $\theta$ ), roll angle ( $\phi$ ) and the central difference expressions of the vertical speed are used.

Using Eq. (4.17), the following online function approximation is constructed at a delayed time step,  $d$ :

$$\hat{\delta}_{coll_d} = B_{11}^{-1}(\partial_{V_s} - A_{11}V_{s_d}) + \Delta(\bar{\partial}(V_s)_d, h_d, V_{e_d}, \theta_d, \phi_d, V_{s_d}, b_1) \quad (4.27)$$

in which,  $\bar{\partial}(V_s)_d$  is calculated using Eq. (3.32) and, the first term represents the inverse of a linear model approximation.  $\partial_{V_s}$  is the average sum of central differences used in the linear model inverse:

$$\partial_{V_s} = \frac{1}{k} \sum_{j=1}^k \bar{\partial}(V_s)_d(1, j). \quad (4.28)$$

The following basis is constructed for the neural network,  $\Delta = W^T \Phi_d$ :

$$\begin{aligned} \phi(i) &= \phi_i(\bar{\partial}(V_s)_d(1, i)), \quad i = 1 : 4 \\ \phi(i+4) &= \phi_{i+4}(\bar{\partial}(V_s)_d(2, i)), \quad i = 1 : 4 \\ \phi(9 : 17) &= [\phi_9(V_s) \quad \phi_{10}(V_e) \quad \phi_{11}(\theta) \quad \dots \\ &\dots \quad \phi_{12}(\phi) \quad \phi_{13}(V_s V_e) \quad \phi_{14}(V_s \theta) \quad \dots \\ &\dots \quad \phi_{15}(V_s \phi) \quad \phi_{16}(V_e \theta) \quad b_1]^T \end{aligned} \quad (4.29)$$

Note that,  $\phi(i)$  is the  $i^{th}$  element of the basis vector and  $\phi_i(\cdot)$  is the corresponding activation function. The following activation function is used to ensure boundedness:

$$\phi_i(\cdot) = a_i \tanh\left(\frac{\cdot}{a_i}\right) \in \mathfrak{R}, \quad i = 1, 2, \dots, 17 \quad (4.30)$$

where,  $a$ 's are design parameters.

Using the variables of the current time step, the collective limit,  $\hat{\delta}_{coll_{lim}}$ , can be estimated evaluating Eq. (4.27) at the dynamic trim condition ( $\dot{V}_s = 0$  or  $\partial_{V_s} = 0$ ) as

$$\hat{\delta}_{coll_{lim}} = -B_{11}^{-1}A_{11}V_{s_{lim}} + \Delta(0, h, V_e, \theta, \phi, V_{s_{lim}}, b_1) + e_d \quad (4.31)$$

where, the delayed approximation error,  $e_d$ , is given by,

$$e_d = \delta_{coll_d} - \hat{\delta}_{coll_d} \quad (4.32)$$

and the control margin is given as

$$\hat{\delta}_{coll_{marg}} = \hat{\delta}_{coll_{lim}} - \delta_{coll}. \quad (4.33)$$

Note that compared to the methodologies given in the previous chapter, in this chapter, control sensitivity estimations is not required for control margin estimation. Yet, for the cases in which the limit margin information is required, i.e. for the visualization of the limits on the cockpit visuals, the limit margin estimation methodology of the previous chapter can be used.

#### 4.3.2.1 Example-1: Hover

In this example helicopter model is initialized at a hovering flight condition at 500 ft height above terrain. Simulation scenario for the hover example of the Section 3.3.2 is repeated. Controllers demonstrated previously in Fig. 3.13 are used to keep the helicopter at zero ground speed throughout the simulation and the limit profile given in Fig. 3.14 is used to define the vertical speed down limits as a function of helicopter's height above terrain. Open loop collective inputs are given to result in vertical descent. In Fig. 4.10, the vertical speed response and steady state predictions are shown along with the vertical speed limit signal.

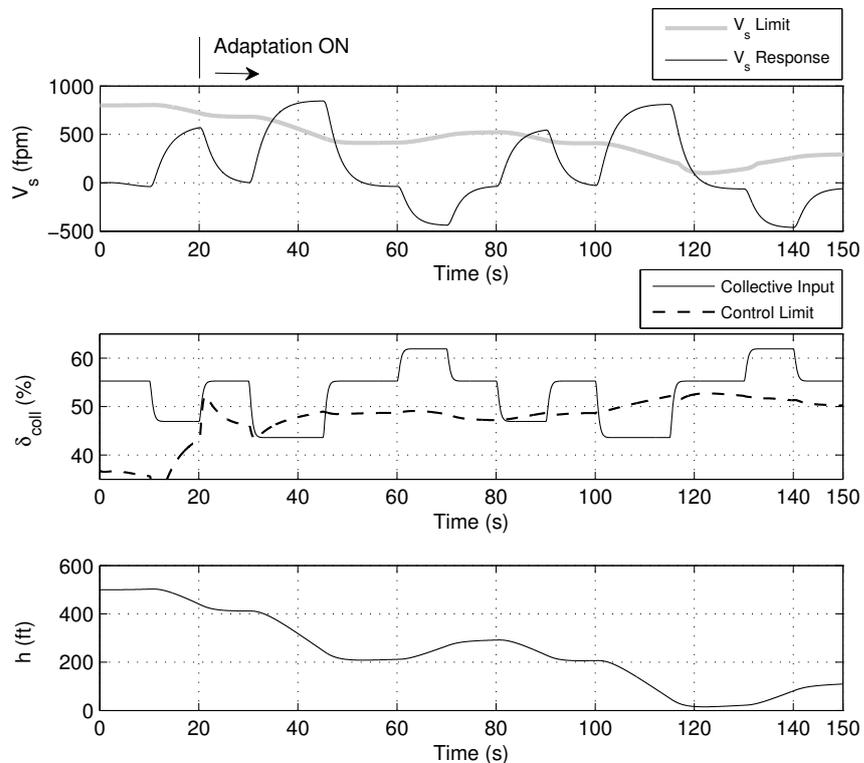


Figure 4.10: Limit Prediction, Example-1

In Figure 4.10, limit margins and control sensitivities are not estimated, yet the control limits are estimated. In addition to that the exceedance of control limits are estimated prior to the actual exceedance of the vertical speed down limits. An advantage of the *Direct Adaptive Control Margin Estimation* methodology is that the estimated control margins can be used without considering limit margin or sensitivity estimations. Also, the control limits can be used directly as the soft stop locations in the active control system of a fly-by-wire aircraft.

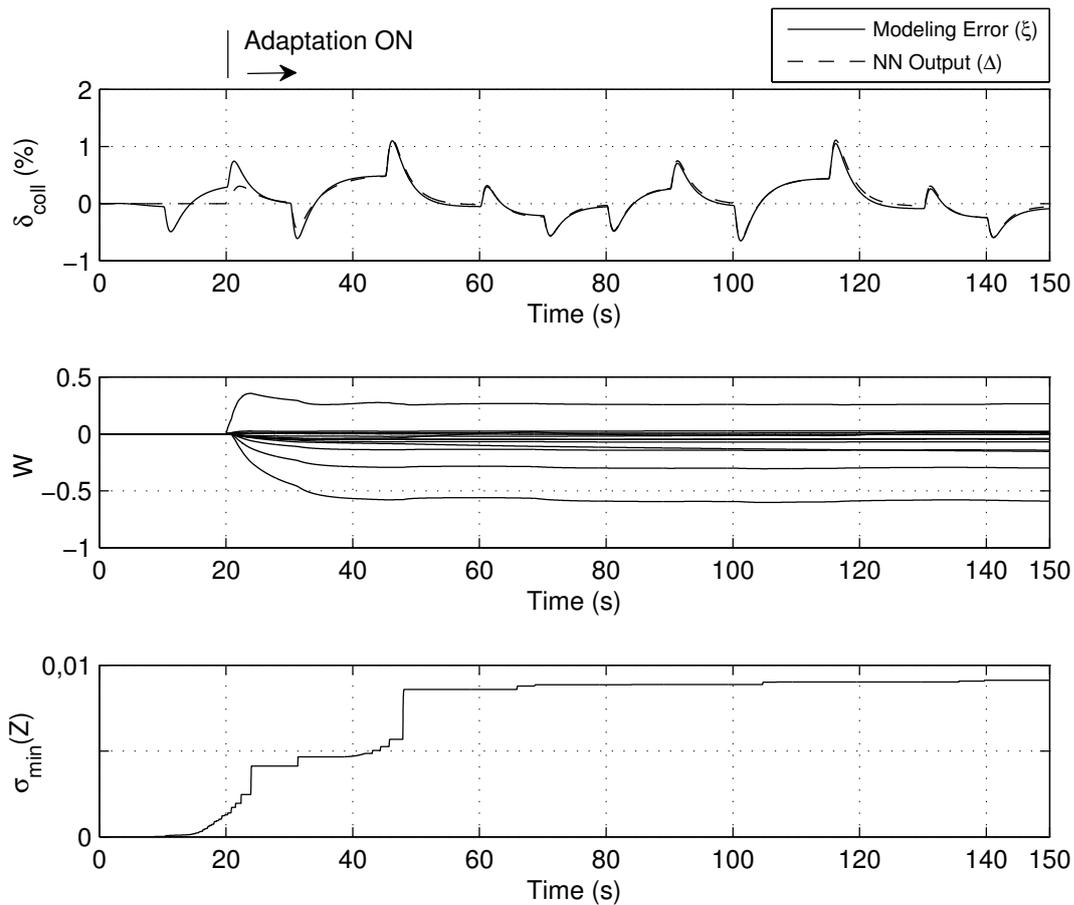


Figure 4.11: Modeling Error and NN Output, NN Weights, Minimum Singular Value of the History Stack, Example-1

In Figure 4.11, modeling error compensation, weight time history of the adaptive element and maximized minimum singular values of the history stack are presented. Modeling errors are augmented in sufficient error bounds using the adaptive element with a fast convergence of weights around a steady region.

Next, limit avoidance is applied through saturating the pilot collective inputs at the estimated collective limits. A block diagram of the limit avoidance scheme is presented in Fig. 4.13. A similar limit avoidance scheme has also been presented in Fig. 3.18 of Section 3.3.2 in which a low pass filter is used to avoid an algebraic loop existing in the approach. Whereas, using the *Direct Adaptive Control Margin Estimation* methodology such an algebraic loop do not exist during control input saturation as in Fig. 4.13 since the pilot input is not required for the estimation of the control limits at the current time step. In the approach, the collective input is only required to calculate the model tracking error at the delayed moving time.

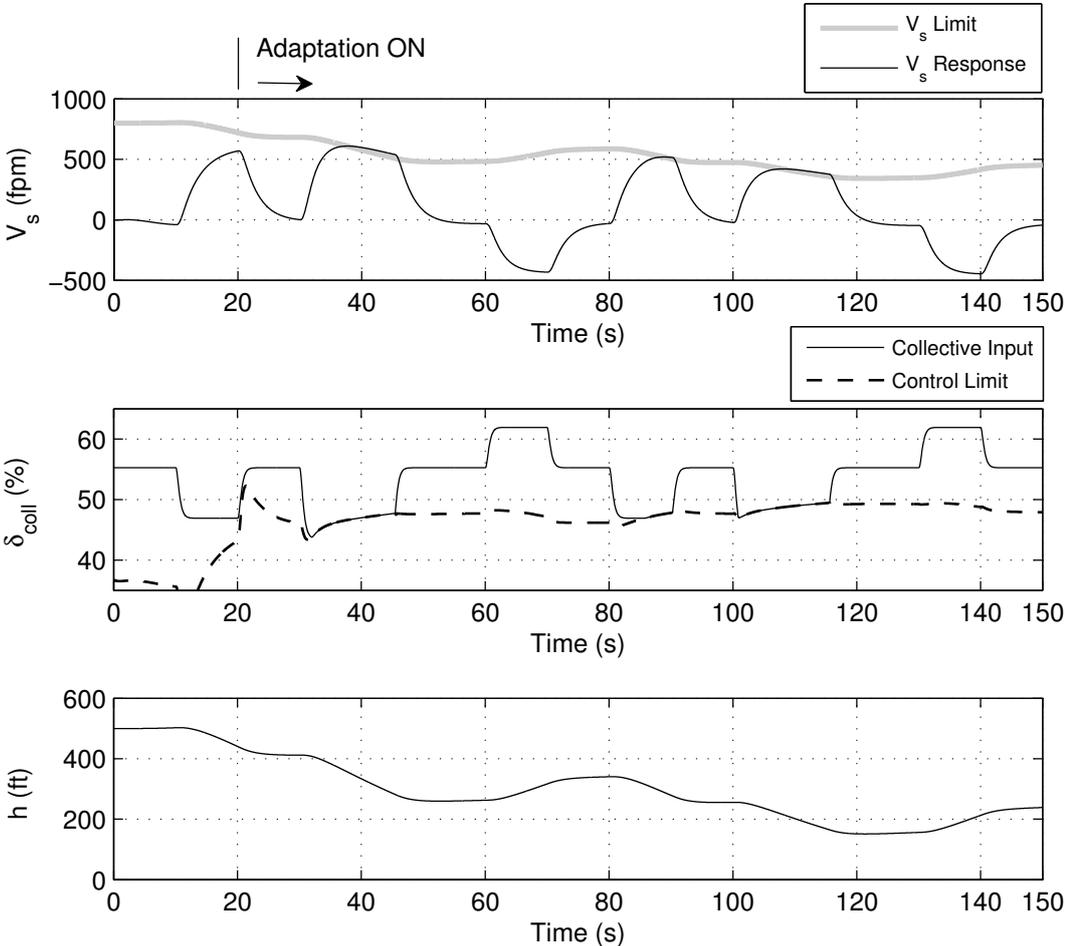


Figure 4.12: Limit Avoidance, Example-1

In Fig. 4.12, results for the limit avoidance example are presented. As shown, the vertical speed is at the limit boundary whenever the controls are saturated with the estimated collective limits.

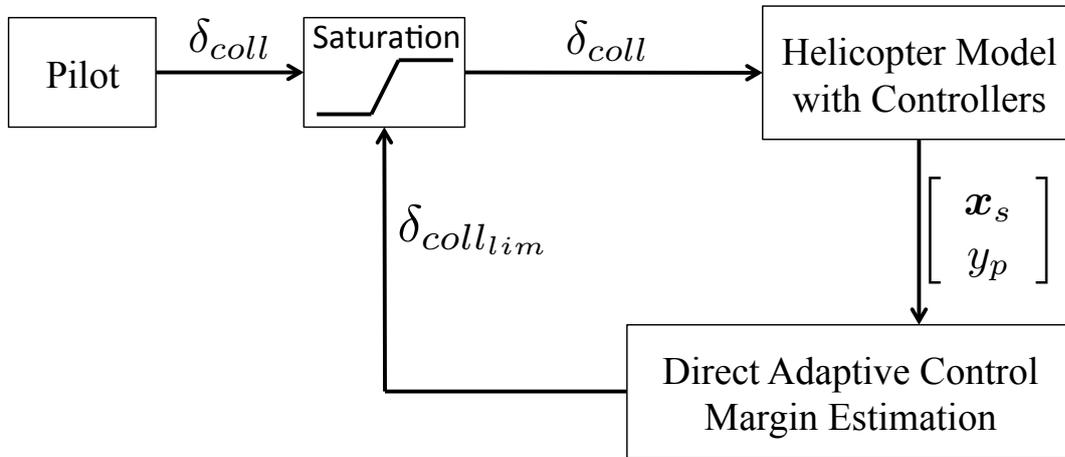


Figure 4.13: Control Limiting for Limit Avoidance

#### 4.3.2.2 Example-2: Low Speed Forward Flight

In this example, helicopter model is accelerated to 20 kts of ground speed and later automatic controllers are used to keep the helicopter at that ground speed. Excessive collective inputs are applied to exceed the vertical speed limits. Prediction results are presented in Fig. 4.14 in which the collective limits are estimated at the onset of actual limit exceedances. Modeling error compensation and the weight time history of the adaptive element are shown in Fig. 4.15. Results agree with the previous example, the hover case. Note that the response of the helicopter to collective inputs at 20 kts is different than the the response of the helicopter at the hover case. Therefore, using the learn while flying capability of the adaptive element, modeling errors associated with different flight conditions are canceled out and correct estimations are obtained.

Limit avoidance is presented next. Results are presented in Fig. 4.16. As in the previous examples, vertical speed limits are avoided at the limit boundary using the control input saturation. Note that the helicopter is above the ground, i.e. at 200 feet, at the end of the simulation.

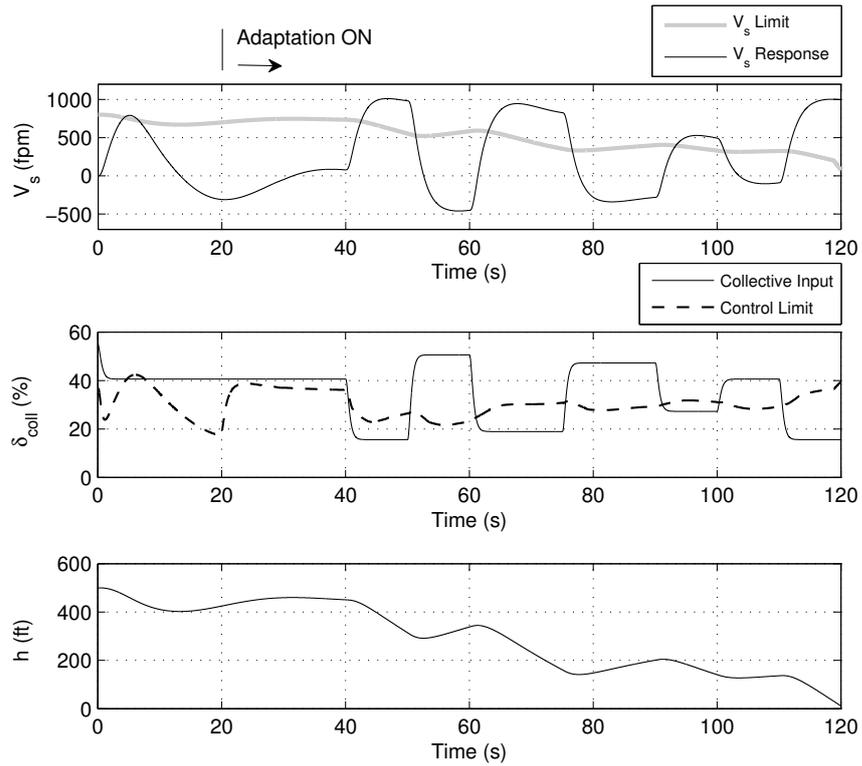


Figure 4.14: Limit Prediction, Example-2

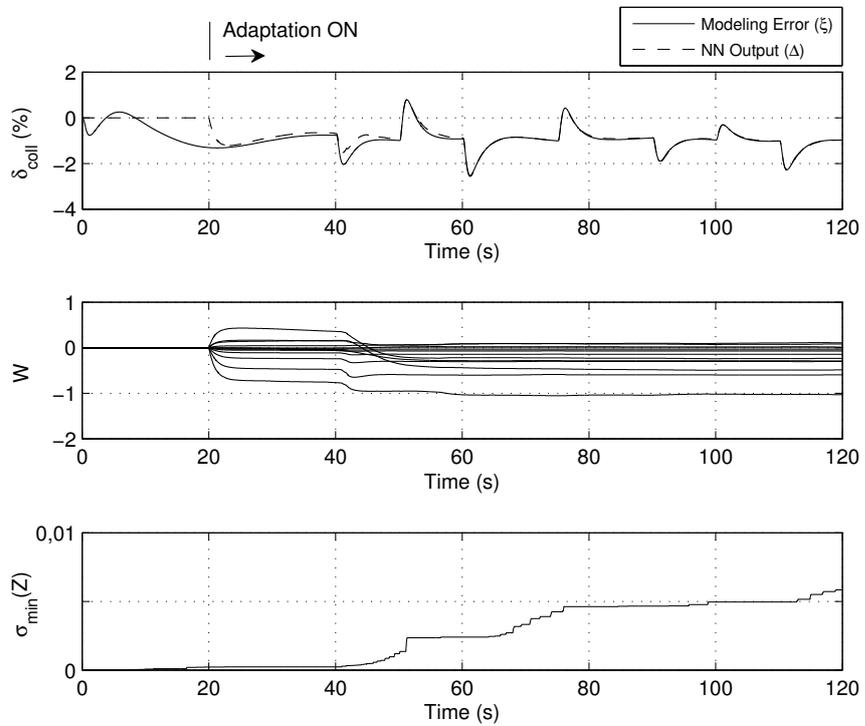


Figure 4.15: Modeling Error and NN Output, NN Weights, Minimum Singular Value of the History Stack, Example-2

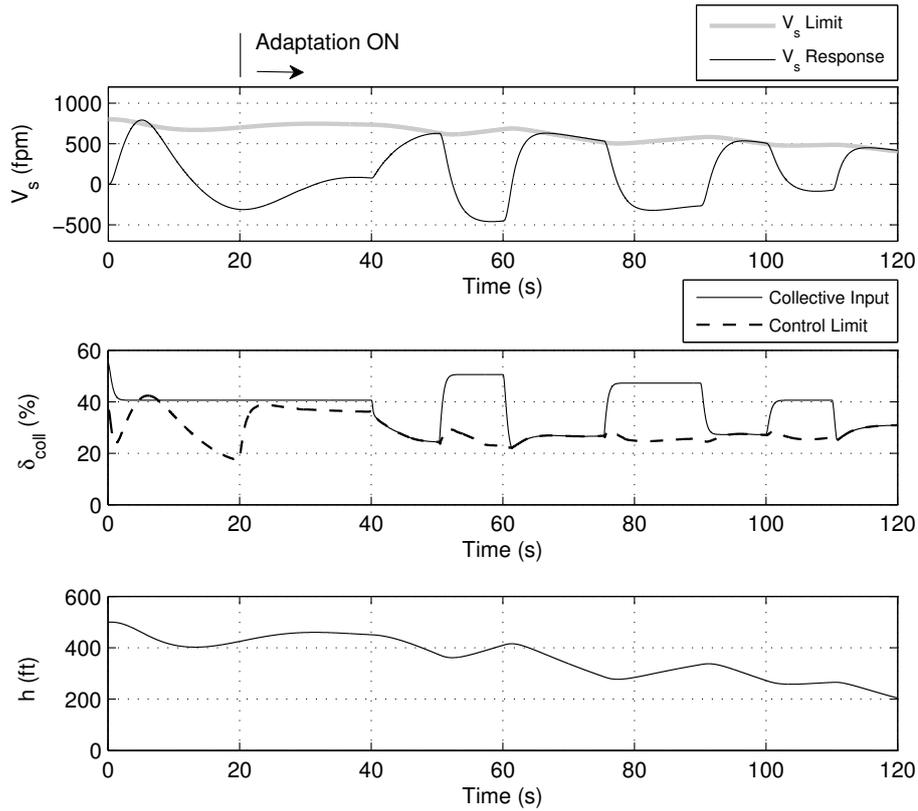


Figure 4.16: Limit Avoidance, Example-2

### 4.3.2.3 Example-3: Forward Flight

In this example, helicopter model is accelerated to 60 kts of ground speed and later, i.e. after 100 seconds, aggressive collective inputs are applied. Prediction results are shown in Fig. 4.17. Until the time is 100 seconds of the simulation, necessary data for concurrent adaptation is recorded online and later used concurrently after that time for the control margin estimation. Note that at the end of the simulation the helicopter hits the ground with an excessive vertical speed since the limits are not avoided by the estimated collective control limits. Related modeling error compensation and weight time histories along with the maximized minimum singular values are presented in Fig. 4.18. Results agree with the results of the previous chapter and agree with the results of the previous examples of this chapter in which modeling errors are approximated by the adaptive elements within adequate error bounds.

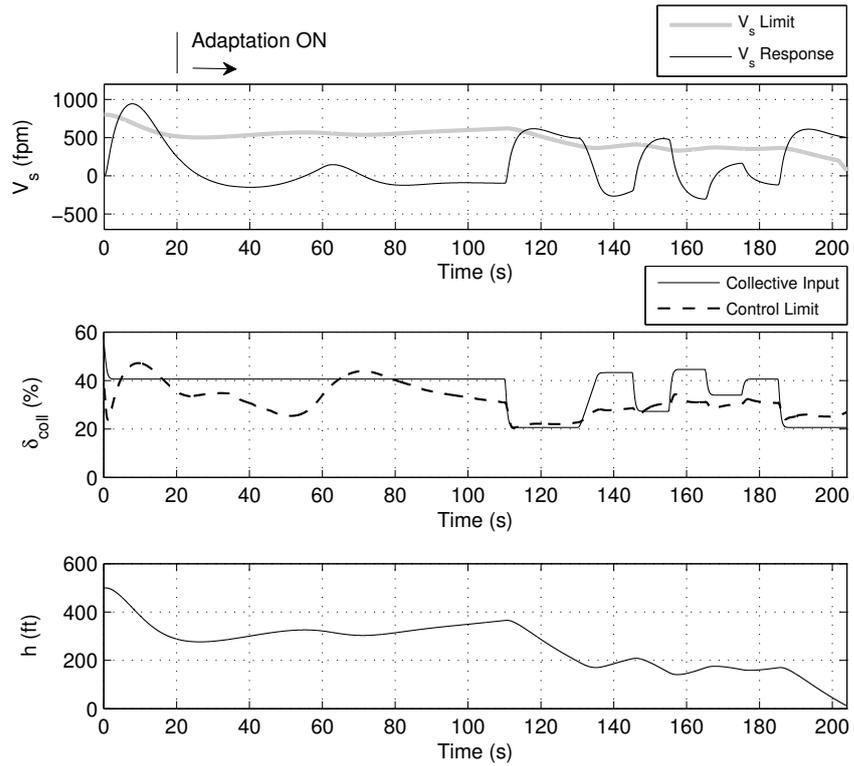


Figure 4.17: Limit Prediction, Example-3

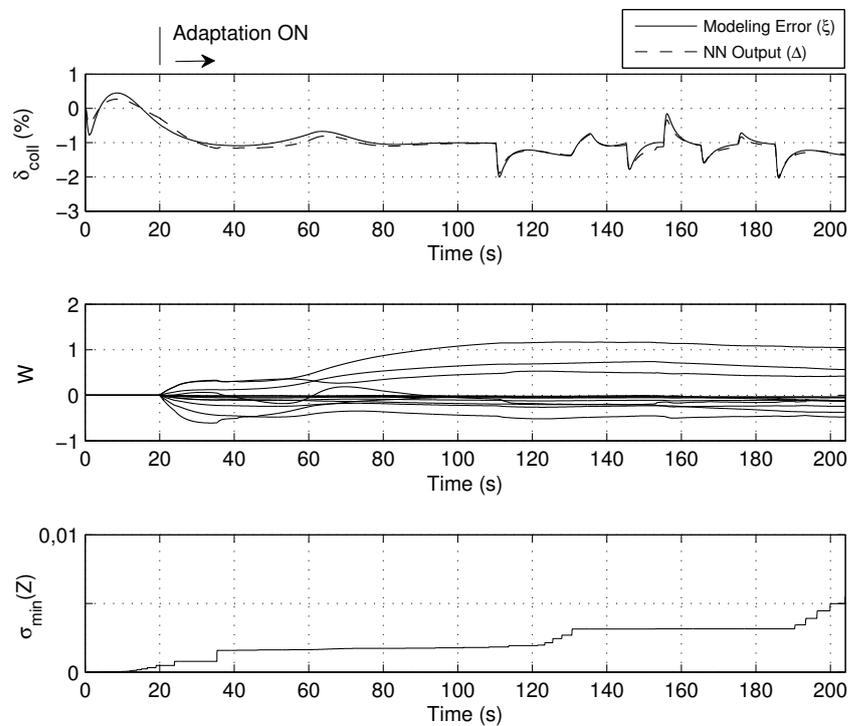


Figure 4.18: Modeling Error and NN Output, NN Weights, Minimum Singular Value of the History Stack, Example-3

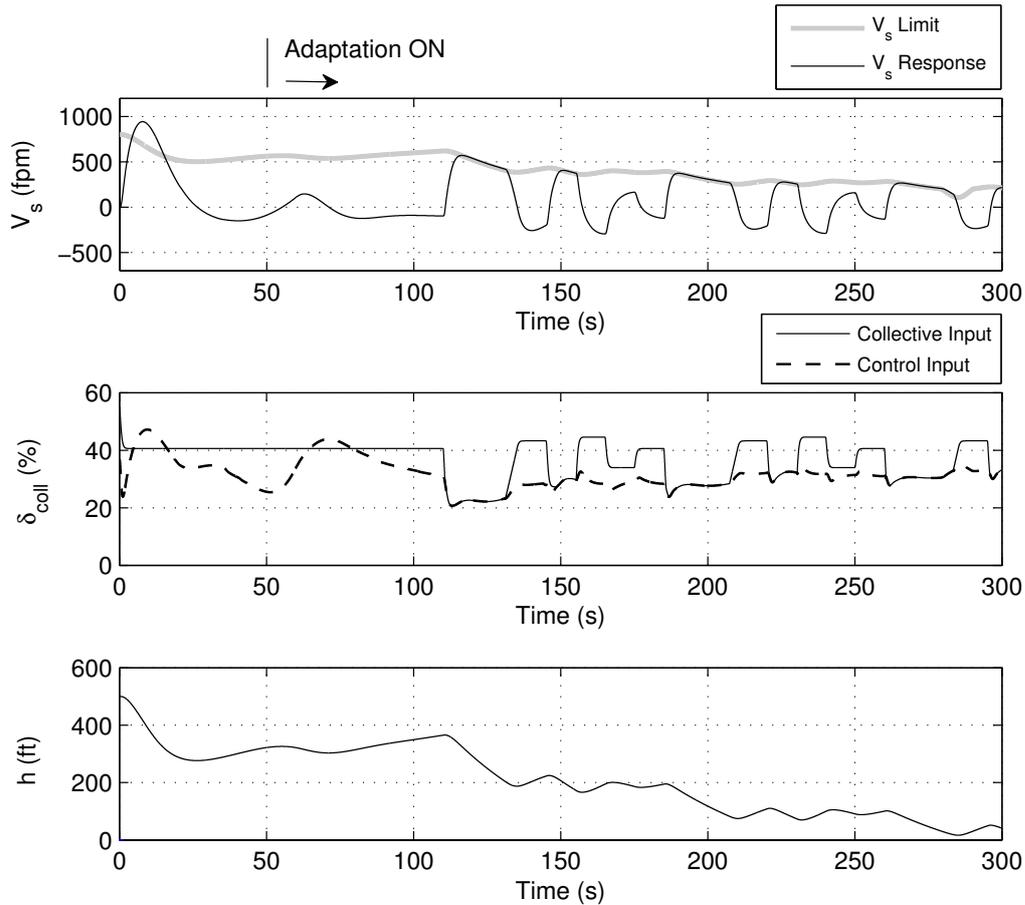


Figure 4.19: Limit Avoidance, Example-3

#### 4.4 Concluding Remarks

In this chapter, an adaptive control margin estimation methodology is proposed for the task of control margin cueing for fly-by-wire aircraft. The previous adaptive framework for the online estimation of control margins is improved.

Proposed method is based on online generation of approximate models of pilot control inputs. Compared to the methodology presented in the previous chapter, control margins are calculated without requiring limit margin and control sensitivity estimations. In the proposed approach, adaptive neural networks are trained online to generate an approximate model of the effective pilot control input at a delayed moving time. A one-to-one relation between the limit pa-

parameter of interest and the effective control input is enforced including only the limit parameter of interest as being a fast state in the online model. Central difference expressions of the considered limit parameter are used at the delayed time to represent the time derivatives of the limit parameter and used in the network for online adaptation. Models generated at the delayed time are then used in the current time. The central differences are set to zero at the current time to estimate the control limits associated with the known envelope boundaries. In the approach, the use of delayed input models at the current time step to estimate the control limits, is found to be effective since the network parameters are observed to converge to steady, i.e. approximately constant, values during aggressive maneuvering. Note that, the online iterations required in the previous adaptive frameworks, are removed using the proposed control margin estimation methodology.

The effectiveness of the proposed technique is simulated using nonlinear fixed wing and rotary wing aircraft models exposed to aggressive maneuvers. It is shown that using the *Direct Adaptive Control Margin Estimation*, accurate control margins which result in the envelope boundaries can be estimated with sufficient lead times to react on approaching load factor, angle of attack and vertical speed limits. In the method, envelope limits are avoided by limiting the actual control input using the estimated control limits, which is done for the verification of the control limit estimations. In that case, no chattering in the limit boundary is observed as it was a drawback of the previous adaptive methods. Based on simulation results, the new adaptive framework is shown to be effective to find control limits and margins to be used to warn pilots in advance of approaching exceedances of flight envelope boundaries.



## CHAPTER 5

### CONCLUSIONS

Limit and control margins are the two vital signals sought in the area of flight envelope protection. One way of predicting both signals can be done through using adaptive methods. In this dissertation, adaptive frameworks are proposed for the online estimation of limit and control margins.

For limit margin estimation, the *Direct Adaptive Limit Margin Estimation* methodology is proposed. In that approach, adaptive elements are trained online to generate reduced order approximate models of fast aircraft states or limit parameters. At a maneuvering steady state condition called dynamic trim, the approximate models are evaluated and used to estimate limit margins. Based on the simulation results, the method is shown to be effective for limit margin cueing since approaching limit exceedances are estimated within effective lead times for a pilot to react.

For control margin estimation, two different approaches are proposed. In the first approach, the limit margin estimator, obtained using the *Direct Adaptive Limit Margin Estimation*, is used to calculate control sensitivities. Control margins are related to the limit margins linearly using the control sensitivity estimations. In that method, control margins are guaranteed to be estimated when the proximity of the limit parameters to the envelope boundary, i.e. limit margin, become zero. In the second approach, the *Direct Adaptive Control Margin Estimation* methodology is used to estimate control margins without requiring limit margin and sensitivity estimations. In that approach, adaptive elements are trained online to generate approximate models of an effective control input.

At the dynamic trim condition, approximate models are evaluated using the known envelope limits to estimate related control limits and margins. Both approaches are shown to be effective in simulations since control limit exceedances are estimated in advance of reaching the actual flight envelope boundaries.

Throughout the thesis, a recent advance in the area of Model Reference Adaptive Control, i.e. a concurrent learning adaptive law with minimum singular value maximization algorithm, is applied to estimate relevant aircraft states, limit parameters and control inputs online, and used to decrease the weight error bounds of the proposed online estimators. A tailored rule is used to collect data sets online for concurrent learning.

For the verification of the online obtained margins, a limit avoidance scheme is applied by artificially saturating the controls at the estimated control limits. In that case, the response reaches and rides the limit boundary and makes use of the actual flight envelope.

In the proposed methodologies, a critical design parameter is found to be as the adaptive element itself. Note that a considered adaptive element should be able to capture all the uncertainties recorded into a data history stack. Therefore, the resulting online models should be physically correct to represent the uncertainties. According to that, the size of the basis and the activation functions inside have to be selected carefully to obtain accurate adaptation results. In this thesis, a different online modeling strategy is used, in which the basis functions are classified and used as two different sets of functions. One set of basis functions is made as a function of central difference expressions and the other set is made as a function of the control inputs, slow states, other necessary parameters and their multiplications. Note that the functions associated with the difference expressions are typically excited relatively more compared to the other set of functions during a transient response. In addition, the set of functions associated with the aircraft states and inputs have typically nonzero values at the steady state response, whereas, the other set associated with the central differences have zero values at the steady state. By doing that, the functions associated with the difference expressions is used to model the transient response,

whereas, the other set is used to model the steady state response online. In this thesis, that strategy is found to be physically correct and effective for online modeling and supported in theory using the definition of relative degree in the previous chapters.

Another critical design parameter can be the learning gains. For the selection of the learning gains, relatively low values are found to be sufficient. Note that the concurrent learning adaptive law enabled to select low gains and maintain adaptation in long term. Therefore, after a few trials proper learning gains were selected easily.

The following specific conclusions can be made:

- Online models are established at a delayed moving time and later used at the current time to estimate the dynamic trim state and control limits. The use of delayed models at the current time to estimate both information is found to be effective.
- Central difference expressions used instead of state derivatives in the adaptive element are found to be sufficient in generating approximate models.
- The network parameters are seen to have an increased tendency to steady state convergence when concurrent learning is used. Moreover the use of singular value maximization in the collection of data sets improved weight convergence even further. A Lyapunov based proof is presented to show the boundedness of the weights of the proposed estimators.
- The direct methods presented in this work deal with the uncertainty in the states, not in its derivatives. In the related studies of MRAC and in the previous works of adaptive limit and control margin estimation, modeling errors were written as functions of state derivatives. Yet in this this work, the implementation of concurrent learning is found to be less involved since derivative estimations and related smoothing algorithms are not required directly.
- The *Sensitivity Estimation Based Control Margin Estimation* requires a model that is converged. If the model undergoes fast changes, the con-

trol sensitivities and therefore the control limit estimations would undergo large changes. The results in this thesis were adequate since the adaptive element parameters quickly converged to approximate steady states.

- When the control inputs are saturated in simulation by the estimated control limits, the limiting states are observed to ride the limit boundary. No signal chattering is observed at the limit boundaries. Therefore, no smoothing functions are used during control input saturation.
- The new adaptive frameworks introduced in this dissertation removed some of the restrictions and implementation difficulties that have already been present in the adaptive envelope protection methods. A major contribution of the new frameworks is that the fixed point iterations of the previous methods are not required. Note that, the fixed point solution assumption, which have been required for the existence of the limit and control margins estimations, is removed. Therefore, the new adaptive frameworks introduced in this dissertation do not require any iteration after the approximate models are established.

As a future research, the algorithms proposed in this thesis can be implemented in a high fidelity simulation environment including both fixed wing aircraft and helicopter dynamic models. For the piloted simulation of fixed wing models, the angle of attack and the load factor limit parameters can be considered again. For the helicopter models, collective axis cueing can be applied to avoid collective limits due to engine torque and temperature steady state limits. Vertical acceleration and vertical speed limits can be considered as well in piloted simulations. Online estimated control limits can be given through the active controllers as being the soft stop and the hard stop cues or a combination of both in the related control axis. Various handling quality assessments can be made with piloted simulations in normal and aggressive maneuvering flight conditions.

Another potential research can be related with the limit parameters that exceed their envelope limits during a transient response. The algorithms proposed in this thesis are based on the estimation of a dynamic trim

state. Therefore, the control margins presented are the allowable control travels that would result in the steady state envelope limits. Whereas, new algorithms or methods can be developed to estimate peak response of a transient critical limit parameter and its related control cue in the control axis. That research can focus on the estimation of the rate limits in the control axis since a peak in the response can be written as a function of the input rates.

The methodologies proposed in thesis are also applicable for command margin estimation of closed loop control systems or autopilots. When the control input signal is replaced with the command input signal, considered limit and command margins can be estimated online using the proposed algorithms. Estimation of command margins can be performed without requiring online fixed point iterations as it was a shortcoming of the previous adaptive frameworks of the envelop protection for UAVs and aircraft autopilots.



## REFERENCES

- [1] Wright, G. and Lappos, N., "Helicopter Maneuver Envelope Enhancement (HelMEE) Study," USAAVCOM TR-89-A-001, NASA CR 177542, Sept. 1989
- [2] Mayo, J. R., Iannacci, V. N., "The RAH-66 Comanche Flight Envelope Cueing System," *In Proceedings of the American Helicopter Society 55th Annual Forum*, Montreal, Canada, May 1999.
- [3] Sensburg, O., Bartsch, O., and Bergmann, H., "Reduction of the Ultimate Factor by Applying a Maximum Load Concept," *Journal of Aircraft, AIAA*, Vol. 24, No. 11, Nov. 1987.
- [4] Sensburg, O., Schmidinger, G., and Fullhas, K., "Integrated Design Structures," *Journal of Aircraft, AIAA*, Vol. 26, No. 3, March 1989.
- [5] Wolfdietrich, H., "Flying by Wire-Eurofighter Flight Control System," *Aerospace, AIAA*, No. 1, April 2000, pp. 34-37.
- [6] Corps, S. G., "Airbus A320 Side Stick and Fly By Wire-An Update," *In Proceedings of the SAE 5th Aerospace Behavioral Engineering Technology Conference*, Long Beach, CA., 1987.
- [7] Kothmann, B. D., and Armbrust, J., "RAH-66 Comanche Core AFCS Control Law Development: DEMVAL to EMD," *Proceedings of the American Helicopter Society 58th Annual Forum*, Montreal, Canada, June 2002.
- [8] Whalley, M. S., Hindson, W. S., and Thiers, G. G., "A Comparison of Active Sidestick and Conventional Inceptors for Helicopter Flight Envelope Tactile Cueing," *Proceedings of the American Helicopter Society 56th Annual Forum*, Virginia Beach, VA, May 2000, pp. 181-204.
- [9] Einthoven, P., Miller, D., Nicholas, J., and Margetich, S., "Tactile Cueing Experiments with a Three Axis Sidestick," *Proceedings of the American Helicopter Society 57th Annual Forum*, Washington D.C., May 2001.
- [10] Horn, J. F., Calise, A. J., and 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, pp. 23-31.  
doi:10.4050/JAHS.46.23

- [11] Jeram, G. J., "Open Design for Helicopter Active Control Systems," *Proceedings of the American Helicopter Society 58th Annual Forum*, Montreal, Canada, June 2002.
- [12] Greenfield, A., Kubik, S., Faynberg, A., Litwin, J., Marzella, S., Sahasrabudhe, V., Rucci, J., Corry, C., McCulley, S., Pavelka, E., Rhinehart, M., Roark, S., Fletcher, J. W. and Ott, C., "CH-53K Control Law Risk Reduction Flight Testing on the Rascal JUH-60A In-Flight Simulator," *American Helicopter Society 68th Annual Forum*, AHS International, Alexandria, VA, May 2012.
- [13] Yavrucuk, I., "Adaptive Limit Margin Detection and Limit Avoidance," *PhD. Thesis*, Georgia Institute of Technology, 2003.
- [14] Menon, P. K., Iragavarapu, V. R., and Whalley, M. S., "Estimation of Rotorcraft Limit Envelopes Using Neural Networks," June 1996, In Proceedings of the American Helicopter Society 52th Annual Forum, Washington D.C.
- [15] Bateman, A. J., Ward, D. G., Barron, R. L., and Whalley, M. S., "Piloted Simulation Evaluation of a Neural Network Limit Avoidance System for Rotorcraft," AIAA Paper 99-4252, Aug. 1999.  
doi: 10.2514/6.1999-4252
- [16] Horn, J. F., Calise, A. J., and Prasad, J. V. R., "Flight Envelope Limit Detection and Avoidance for Rotorcraft," *Journal of the American Helicopter Society*, Vol. 47, No. 4, Oct. 2002, pp. 253-262.  
doi:10.4050/JAHS.47.253
- [17] Sahani, N. A., Horn, J. F., "Neural Network Based Algorithms for Comprehensive Collective Axis Limit Avoidance on Rotorcraft," *Journal of Aerospace Computing, Information, and Communication*, Vol. 1, No. 11, 2004, pp. 432-451.  
doi: 10.2514/1.6259
- [18] Sahasrabudhe, V., Horn, J. F., Sahani, N., Faynberg, A., Spaulding, R., "Simulation Investigation of a Comprehensive Collective Axis Tactile Cueing System," *Journal of the American Helicopter Society*, Volume 51, Number 3, 2006, pp. 215-224(10).  
doi:10.4050/1.3092883
- [19] Horn, J. F., Sahani, N., "Detection and Avoidance of Main Rotor Hub Moment Limits on Rotorcraft," *Journal Aircraft*, Vol. 41, No. 2, March-April 2004, pp. 372-379, doi:10.2514/1.301
- [20] Sahani, N. A.; Horn, J. F.; Jeram, G. J. J.; Prasad, J. V. R., "Hub Moment Limit Protection Using Neural Network Prediction," *Journal of the*

- American Helicopter Society*, Vol. 51, No. 4, Oct. 2006, pp. 331-340, doi: 10.4050/JAHS.51.331
- [21] Bottasso, C. L., Montinari, P., “Rotorcraft Flight Envelope Protection by Model Predictive Control,” *Journal of the American Helicopter Society*, Vol. 60, No. 2, 2015, pp. 1-13.  
do: 10.4050/JAHS.60.022005
- [22] Kim, B. S., Calise, A. J., “Nonlinear Flight Control Using Neural Networks,” *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 26-33, doi: 10.2514/2.4029
- [23] Rysdyk, R. T., Calise, A. J., “Adaptive Model Inversion Flight Control for Tilt-Rotor Aircraft,” *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 3, 1999, pp. 402-407, doi: 10.2514/2.4411
- [24] Calise, A. J., Hovakimyan, N., and Idan, M., “Adaptive Output Feedback Control of Nonlinear Systems Using Neural Networks,” *Automatica*, Special issue on Neural Networks for Feedback Control, Vol. 37, No. 8, 2001, pp. 1201-1211.  
doi:10.1016/S0005-1098(01)00070-X
- [25] Johnson, E. N., Kannan, S. K., “Adaptive Trajectory Control for Autonomous Helicopters,” *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 3, 2005, pp. 524-538, doi: 10.2514/1.6271
- [26] Kim, Y. H., Lewis, F. L., and Abdallah, C. T., “A Dynamic Recurrent Neural-Network-Based Adaptive Observer for a Class of Nonlinear Systems,” *Automatica*, Vol. 33, No. 8, 1997, pp. 1539-1543., doi:10.1016/S0005-1098(97)00065-4
- [27] Yavrucuk, I., Prasad, J. V. R., “Online Dynamic Trim and Control Limit Estimation,” *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, 2012, pp. 1647-1656.  
doi: 10.2514/1.53116
- [28] Yavrucuk, I., Prasad, J. V. R., and Unnikrishnan, S., “Envelope Protection for Autonomous Unmanned Aerial Vehicles,” *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 248-261.  
doi:10.2514/1.35265
- [29] Unnikrishnan, S., Prasad, J. V. R., Yavrucuk, I., “Flight Evaluation of a Reactionary Envelope Protection System for UAVs,” *Journal of the American Helicopter Society*, Volume 56, Number 1, Jan. 2011, pp. 12009.  
doi: 10.4050/JAHS.56.012009
- [30] Ioannou, P. A., Sun, J., “Robust Adaptive Control,” Dover Publications, Inc., Mineola, NY, 2012, pp. 144-250.

- [31] Lu, S. and Basar, T., “Robust Nonlinear System Identification Using Neural Network Models,” *IEEE Transactions Neural Networks*, Vol. 9, No. 3, May 1998.
- [32] Gursoy, G., Yavrucuk, I., “Concurrent Learning Enabled Adaptive Limit Detection for Active Pilot Cueing,” *Journal of Aerospace Information Systems*, Vol.11, 2014, pp. 542-550.  
doi: 10.2514/1.I010205
- [33] Chowdhary, G. and Johnson, E. N., “Theory and Flight-Test Validation of a Concurrent-Learning Adaptive Controller,” *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 2, 2011, pp. 592-607. doi: 10.2514/1.46866
- [34] Chowdhary, G., “Concurrent Learning for Convergence in Adaptive Control Without Persistency of Excitation,” *PhD. Thesis*, Georgia Institute of Technology, 2010.
- [35] Gursoy, G., Yavrucuk, I., “Engine Limit Detection and Avoidance for Helicopters with Multiple Limits,” *AIAA Atmospheric Flight Mechanics (AFM) Conference*, Boston, MA, 2013. doi:10.2514/6.2013-4689
- [36] Yavrucuk, I., Prasad, J. V. R., and Calise, A. J., “Adaptive Limit Detection and Avoidance for Carefree Maneuvering,” Aug. 2001, In *Proceedings of the AIAA Atmospheric Flight Mechanics Conference*, Montreal, Canada.  
doi: 10.2514/6.2001-4003
- [37] Yavrucuk, I., Prasad, J. V. R., and Calise, A. J., “Carefree Maneuvering Using Adaptive Neural Networks,” Aug. 2002, In *Proceedings of the AIAA Atmospheric Flight Mechanics Conference*, Monterey, California.  
doi: 10.2514/6.2002-4495
- [38] Yavrucuk, I., Prasad, J. V. R., and Calise, A. J., “Adaptive Limit Prediction and Avoidance for Rotorcraft,” Sept. 2002, In *Proceedings of the 28th European Rotorcraft Forum*, Bristol, UK.
- [39] Yavrucuk, I., Gursoy, G., Novikov, Y., “Online Detection and Avoidance of Helicopter TGT Limits,” *American Helicopter Society 69th Annual Forum*, Phoenix, AZ, May 2013.
- [40] Rysdyk, R. T., Calise, A. J., “Adaptive Nonlinear Control for Tilt-Rotor Aircraft,” *IEEE International Conference on Control Applications*, Trieste, Italy, Sept. 1998.
- [41] Shin, Y., Calise, A. J., Johnson, M.D., “Adaptive Control of Advanced Fighter Aircraft in Nonlinear Flight Regimes,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1464-1477.  
doi:10.2514/1.30213

- [42] Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974, pp.156-174.
- [43] Jordan, C., “Calculus of Finite Differences,” New York, NY: Chelsea Pub. Co., 1960.
- [44] Khalil, Hassan K., *Nonlinear Systems*, Prentice Hall, NJ, 2002, pp. 509–511.
- [45] Mathews, J. H., *Numerical Methods for Mathematics, Science and Engineering*, Prentice-Hall International, Inc., Englewood Cliffs, NJ, 1992, pp 333. ISBN: 0-13-625047-5
- [46] Abramowitz, M., Stegun, I. A., *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, United States Department of Commerce, Washington, D.C., 1972, pp 877.
- [47] Chowdhary, G., Yucelen, T., Muhlegg, M., Johnson, E. N., “Concurrent Learning Adaptive Control of Linear Systems with Exponentially Convergent Bounds,” *International Journal of Adaptive Control and Signal Processing*, Vol. 27, No. 4, April 2013, pp. 280-301.  
doi: 10.1002/acs.2297
- [48] Chowdhary, G., Johnson, E. N., “Concurrent Learning for Convergence in Adaptive Control without Persistency of Excitation,” *IEEE Conference on Decision and Control*, GA, USA 2010.  
doi: 10.1109/CDC.2010.5717148
- [49] Roskam, J., “Airplane Flight Dynamics and Automatic Flight Controls, Part-I,” Design, Analysis and Research Corporation, Kansas, KS, 2003, pp. 480-486.
- [50] Yavrucuk, I. Tarimci, O., Katircioglu., M., Kubali, E., Yilmaz, D., *A new Helicopter Simulation and Analysis Tool: Helidyn+*, September 2010, In Proceedings of the Thirty-Sixth European Rotorcraft Forum, Paris, France.



## APPENDIX A

### APPENDIX

#### A.1 Proof of Theorem

In [34], the following proof was performed for the case when the structure of the uncertainty is exactly known, that is, the reconstruction error  $\epsilon$  of Eq. (3.47) is zero. In that case it is proven that the weight update law given by Eq. (3.52) guarantees asymptotic stability for the problem of adaptive parameter estimation. Here, the proof of [34] for adaptive parameter estimation is extended for the case where the structure of the uncertainty is unknown such that the reconstruction error  $\epsilon$  of Eq. (3.47) is nonzero. Hence, the ultimate bounds on the weight errors of LPNN are established.

First, consider the estimator of Eq. (3.8). When the modeling error  $\xi$  of Eq. (3.7) is replaced with the uncertainty of Eq. (3.47), and  $\Delta$  of Eq. (3.8) is replaced with Eq. (3.49), one can write the actual fast states and the estimator as:

$$\mathbf{x}_f = \mathbf{f}_1^{-1}(\dot{\mathbf{x}}_f, \mathbf{x}_s, \mathbf{u}) + W^{*T} \boldsymbol{\phi}(\bar{\mathbf{x}}) + \boldsymbol{\epsilon} \quad (\text{A.1})$$

$$\hat{\mathbf{x}}_f = \hat{\mathbf{f}}_1^{-1}(\dot{\mathbf{x}}_f, \mathbf{x}_s, \mathbf{u}) + W^T \boldsymbol{\phi}(\bar{\mathbf{x}}) \quad (\text{A.2})$$

Second, consider the estimator of Eq. (4.5). When the modeling error  $\xi$  of Eq. (4.4) is replaced with the uncertainty expression of Eq. (3.47), and  $\Delta$  of Eq. (4.5) is replaced with Eq. (3.49), one can write the actual input of interest and the input estimator as:

$$u_e = f_1^{-1}(\dot{\mathbf{x}}_f, \mathbf{x}_f, \mathbf{x}_s, \mathbf{u}') + W^{*T} \boldsymbol{\phi}(\bar{\mathbf{x}}) + \epsilon \quad (\text{A.3})$$

$$\hat{u}_e = \hat{f}_1^{-1}(\dot{\mathbf{x}}_f, \mathbf{x}_f, \mathbf{x}_s, \mathbf{u}') + W^T \boldsymbol{\phi}(\bar{\mathbf{x}}) \quad (\text{A.4})$$

The function approximation error,  $\mathbf{e} = \mathbf{x}_f - \hat{\mathbf{x}}_f$  or  $e = u_e - \hat{u}_e$ , is written by subtracting Eq. (A.2) from Eq. (A.1) or Eq. (A.3) from Eq. (A.4) and can be written in the form:

$$\mathbf{e} = \tilde{W}^T \boldsymbol{\phi}(\bar{\mathbf{x}}) + \boldsymbol{\epsilon} \quad (\text{A.5})$$

The bounds on  $\boldsymbol{\epsilon}$  are shown using the universal approximation property of Neural Networks given in Eq. (3.51). Hereafter, the goal is to establish the bounds on the weight error dynamics.

Let's write the weight error dynamics as:

$$\dot{\tilde{W}}(t) = \dot{W}^* - \dot{W} \quad (\text{A.6})$$

Similar to [34], substituting Eq. (3.52) into Eq. (A.6) and using  $\dot{W}^* = 0$ , the parameter error dynamics takes the form:

$$\dot{\tilde{W}}(t) = -\Gamma \boldsymbol{\phi}(\bar{\mathbf{x}}) \mathbf{e}^T - \sum_{i=1}^p \Gamma \boldsymbol{\phi}(\bar{\mathbf{x}}_i) \mathbf{e}_i^T. \quad (\text{A.7})$$

Now, substitute Eq. (A.5) and  $\mathbf{e}_i = \tilde{W}^T \boldsymbol{\phi}(\bar{\mathbf{x}}_i) + \boldsymbol{\epsilon}_i$  into Eq. (A.7):

$$\dot{\tilde{W}}(t) = -\Gamma \boldsymbol{\phi}(\bar{\mathbf{x}}) (\tilde{W}^T \boldsymbol{\phi}(\bar{\mathbf{x}}) + \boldsymbol{\epsilon})^T - \sum_{i=1}^p \Gamma \boldsymbol{\phi}(\bar{\mathbf{x}}_i) [\tilde{W}^T \boldsymbol{\phi}(\bar{\mathbf{x}}_i) + \boldsymbol{\epsilon}_i]^T, \quad (\text{A.8})$$

when expanded:

$$\dot{\tilde{W}}(t) = -\Gamma \boldsymbol{\phi}(\bar{\mathbf{x}}) \boldsymbol{\phi}(\bar{\mathbf{x}})^T \tilde{W} - \Gamma \boldsymbol{\phi}(\bar{\mathbf{x}}) \boldsymbol{\epsilon}^T - \sum_{i=1}^p \Gamma \boldsymbol{\phi}(\bar{\mathbf{x}}_i) \boldsymbol{\phi}(\bar{\mathbf{x}}_i)^T \tilde{W} - \sum_{i=1}^p \Gamma \boldsymbol{\phi}(\bar{\mathbf{x}}_i) \boldsymbol{\epsilon}_i^T \quad (\text{A.9})$$

Consider the following positive definite and radially unbounded function as the Lyapunov candidate:

$$V(\tilde{W}) = \frac{1}{2} \text{tr}(\tilde{W}^T \Gamma^{-1} \tilde{W}) \quad (\text{A.10})$$

where,  $V(0) = 0$  and  $V(\tilde{W}) > 0 \forall \tilde{W} \neq 0$ .

The time derivative of  $V(\tilde{W})$  along the system trajectories can be written as:

$$\dot{V}(\tilde{W}) = \text{tr}(\tilde{W}^T \Gamma^{-1} \dot{\tilde{W}}) \quad (\text{A.11})$$

Substitute Eq. (A.9) into Eq. (A.11), then the following time derivative of the Lyapunov function is obtained:

$$\begin{aligned} \dot{V}(\tilde{W}) = & -\text{tr}\left(\tilde{W}^T[\boldsymbol{\phi}(\bar{\mathbf{x}})\boldsymbol{\phi}(\bar{\mathbf{x}})^T + \sum_{i=1}^p \boldsymbol{\phi}(\bar{\mathbf{x}}_i)\boldsymbol{\phi}(\bar{\mathbf{x}}_i)^T]\tilde{W}\right) + \dots \\ & \dots - \text{tr}\left(\tilde{W}^T[\boldsymbol{\phi}(\bar{\mathbf{x}})\boldsymbol{\epsilon}^T + \sum_{i=1}^p \boldsymbol{\phi}(\bar{\mathbf{x}}_i)\boldsymbol{\epsilon}_i^T]\right) \end{aligned} \quad (\text{A.12})$$

Note that  $\boldsymbol{\phi}(\bar{\mathbf{x}})\boldsymbol{\phi}(\bar{\mathbf{x}})^T \geq 0 \forall \boldsymbol{\phi}(\bar{\mathbf{x}})$ , and  $\Omega = \sum_{i=1}^p \boldsymbol{\phi}(\bar{\mathbf{x}}_i)\boldsymbol{\phi}(\bar{\mathbf{x}}_i)^T$  is a matrix. Then  $\dot{V}(\tilde{W})$  can be bounded as:

$$\dot{V}(\tilde{W}) \leq -\lambda_{\min}(\Omega)\|\tilde{W}\|^2 + \|\tilde{W}^T\|(\|\boldsymbol{\phi}\| \|\boldsymbol{\epsilon}^T\| + \sum_{i=1}^p \|\boldsymbol{\phi}_i\| \|\boldsymbol{\epsilon}_i^T\|) \quad (\text{A.13})$$

Using Eq. (3.48), the basis vector is further bounded as  $\|\boldsymbol{\phi}(\bar{\mathbf{x}})\| \leq \alpha_0$ . Noting that the reconstruction error is bounded using Eq. (3.51) as  $\|\boldsymbol{\epsilon}\| \leq \bar{\epsilon}$ , and all bounds are valid for the  $i^{\text{th}}$  recorded data, that is  $\|\boldsymbol{\phi}(\bar{\mathbf{x}}_i)\| \leq \alpha_0$  and  $\|\boldsymbol{\epsilon}_i\| \leq \bar{\epsilon}$ , then the bounds on  $\dot{V}(\tilde{W})$ , Eq. (A.13), become as:

$$\dot{V}(\tilde{W}) \leq -\lambda_{\min}(\Omega)\|\tilde{W}\|^2 + \|\tilde{W}\|(p+1)\alpha_0\bar{\epsilon} \quad (\text{A.14})$$

$$\dot{V}(\tilde{W}) \leq -\|\tilde{W}\|(\lambda_{\min}(\Omega)\|\tilde{W}\| - (p+1)\alpha_0\bar{\epsilon}) \quad (\text{A.15})$$

Note that  $\dot{V}(\tilde{W})$  is strictly negative when

$$\|\tilde{W}\| > \frac{(p+1)\alpha_0\bar{\epsilon}}{\lambda_{\min}(\Omega)} \quad (\text{A.16})$$

Now, within a domain,  $D_0$ , over which the neural network approximation is valid, let  $\boldsymbol{\zeta}$  represent  $\tilde{W}$ , then a ball,  $B^r$ , can be defined such that,

$$B^r = \{\boldsymbol{\zeta} : \|\boldsymbol{\zeta}\| \leq r\} \quad (\text{A.17})$$

Let  $\eta$  be the minimum value of the function  $V(\boldsymbol{\zeta})$  on the trajectories of  $B^r$ , that is

$$\eta = \min\left(\frac{1}{2}\boldsymbol{\zeta}^T\Gamma^{-1}\boldsymbol{\zeta}\right) = \frac{1}{2}r^2\lambda_{\min}(\Gamma^{-1}) \quad (\text{A.18})$$

Define the following set of  $\boldsymbol{\zeta}$ 's from  $B^r$ , where  $V(\boldsymbol{\zeta})$  is less than  $\eta$

$$\Theta_\eta = \{\boldsymbol{\zeta} \in B^r : \boldsymbol{\zeta}^T\Gamma^{-1}\boldsymbol{\zeta} \leq \eta\} \quad (\text{A.19})$$

Hence,  $\Theta_\eta \subset B^r$  is a positive invariant set of Eq. (A.7) since the time derivative of  $V$  is shown to be negative definite when Eq. (A.16) is satisfied. Next, using Eq. (A.16), define the set:

$$\Theta_\beta = \{\zeta \in B^r : \|\zeta\| \leq \frac{(p+1)\alpha_0\bar{\epsilon}}{\lambda_{\min}(\Omega)}\} \quad (\text{A.20})$$

If  $\Theta_\beta \subset \Theta_\eta$ , then it is required that

$$\frac{(p+1)\alpha_0\bar{\epsilon}}{\lambda_{\min}(\Omega)} < \eta \quad (\text{A.21})$$

Then the minimum size of  $B^r$  is given by

$$r^2 > \frac{2(p+1)\alpha_0\bar{\epsilon}}{\lambda_{\min}(\Omega)\lambda_{\min}(\Gamma^{-1})}. \quad (\text{A.22})$$

Here,  $D_0$  must be sufficiently large, hence  $B^r \subset D_0$ . When  $\zeta(t_0) \in \Theta_\eta$  and  $D_0$  is sufficiently large the weight errors  $\zeta$  will be ultimately bounded.

Since LPNNs are universal approximators,  $\bar{\epsilon}$  of Eq. (A.20), hence  $\Theta_\eta$ , can be made sufficiently small by choosing a larger basis in the network. Another way to make  $\Theta_\eta$  sufficiently small is to maximize  $\lambda_{\min}(\Omega)$  of Eq. (A.20). This can be done through recording data such that the minimum singular value of  $Z$  is maximized online [34, 47]. Note that,  $\Omega = \sum_{i=1}^p \phi(\bar{\mathbf{x}}_i)\phi(\bar{\mathbf{x}}_i)^T = ZZ^T$  where  $Z \in \mathfrak{R}^{m \times p}$  is the history stack matrix, therefore,  $\lambda_{\min}(\Omega) = \lambda_{\min}(ZZ^T)$  where  $ZZ^T \in \mathfrak{R}^{m \times m}$ . Also nothing that the singular values of a matrix  $A$  are the square roots of the nonzero eigenvalues of the matrix  $AA^*$ , in which  $A^*$  is the transpose-conjugate of  $A$ , therefore,  $\sigma(Z)^2 = \lambda(ZZ^T)$  holds using linear algebra in which  $\sigma(\cdot)$  is the singular value operator of a matrix. Therefore, an increase in the minimum singular value of  $Z$  corresponds to an increase in the minimum eigen value of  $\Omega$  and a decrease in the established bounds.

As in [34], the weights will be bounded in a compact neighbourhood of the ideal weights when  $\lambda_{\min}(\Omega)$  is maximized online such that the theorem holds. The bound is given when  $\dot{V}(\tilde{W}) \leq 0$  such that

$$\|\tilde{W}\| \geq \frac{(p+1)\alpha_0\bar{\epsilon}}{\lambda_{\min}(\Omega)}. \quad (\text{A.23})$$

## A.2 Concurrent Learning for Online Parameter Estimation

In this example, the weight update time histories using concurrent learning with and without minimum singular value maximization in the recorded data are compared. A linear model is used as the plant, and the linear coefficients of the plant are aimed to be estimated. The Direct Adaptive Limit Margin Estimation methodology is used to construct an estimator for the angle of attack. The derivatives in the linear model are assumed to be known at the delayed time. Hence, the central difference representation for the derivatives is not used in this example.

Using a reduced order longitudinal model, the angle of attack,  $\alpha$ , is linearly parameterized at a delayed time step,  $d$ , as follows:

$$\alpha_d = C_1 \dot{\alpha}_d + C_2 \dot{q}_d + C_3 \delta_{e_d}. \quad (\text{A.24})$$

An approximation to Eq. (A.24) is established using  $\Delta$ :

$$\hat{\alpha}_d = \hat{C}_1 \dot{\alpha}_d + \hat{C}_2 \dot{q}_d + \hat{C}_3 \delta_{e_d} + \Delta(\dot{\alpha}_d, \dot{q}_d, \delta_{e_d}) \quad (\text{A.25})$$

where,  $\hat{C}$ 's are the approximate coefficients. The model tracking error,  $e_d$ , can be written at the delayed time by subtracting Eq. (A.25) from Eq. (A.24) as:

$$e_d = C_1^* \dot{\alpha}_d + C_2^* \dot{q}_d + C_3^* \delta_{e_d} - \Delta(\dot{\alpha}_d, \dot{q}_d, \delta_{e_d}) \quad (\text{A.26})$$

where,  $C^* = C - \hat{C}$ 's are the optimal weights and  $\Delta = W^T \phi$ . If we let the network basis be

$$\phi_d = [\dot{\alpha} \quad \dot{q} \quad \delta_e \quad \dot{\alpha} \dot{q} \quad \delta_e \dot{\alpha} \quad \delta_e \dot{q} \quad b_1]_d^T, \quad (\text{A.27})$$

then, the weights,  $W$ , are expected to converge to the following ideal weights:

$$W^* = [C_1^* \quad C_2^* \quad C_3^* \quad 0 \quad 0 \quad 0 \quad 0]^T. \quad (\text{A.28})$$

In the simulation the plant model of Eq. (A.24) is used to simulate the  $\alpha$  response for a given elevator input and Eq. (A.25) is used as an approximation. In Fig. A.1a-b, adaptation results for the cases with and without minimum singular value maximization are compared. In both cases the necessary data for concurrent adaptation are pre-selected with criteria of Eqs. (3.55), (3.56) and (3.57).

Data recording is initiated at the beginning of the simulation, whereas the adaptation is initiated at  $t = 10$ s. Note that neural network weights converge to the ideal values faster using singular value maximization in the recorded data.

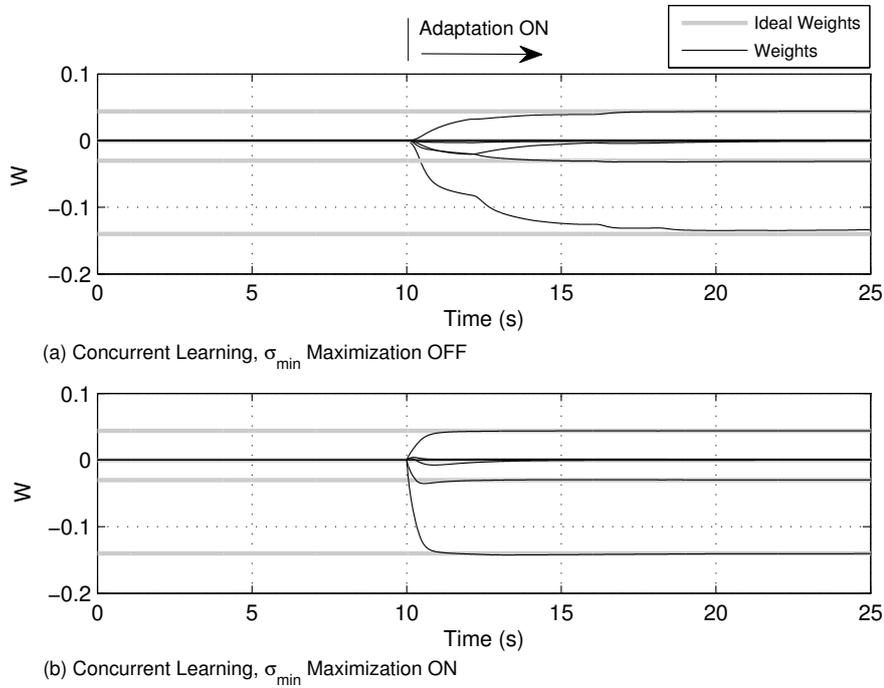


Figure A.1: NN Weights for  $\sigma_{min}$  Maximization ON and OFF

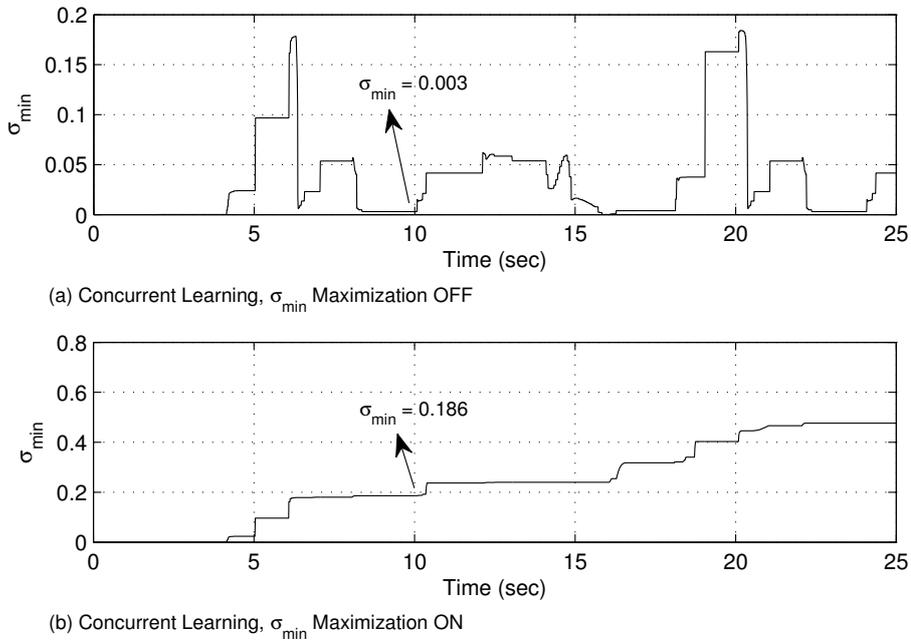


Figure A.2:  $\sigma_{min}(Z)$  for  $\sigma_{min}$  Maximization ON and OFF

In [34], it is shown that for the case of linearly parameterizable uncertainty, the minimum singular value of the history stack is a measure of the speed of weight convergence. In Fig. A.2, minimum singular values of the history stacks are compared. Here, every increment of  $\sigma_{min}$  corresponds to the instants that a necessary data is recorded. Note that at  $t = 10s$ , where the adaptation is initiated, the minimum singular value of the history stack with maximization is about 50 times higher than for the case without maximization. As a result, adaptations are faster with minimum singular value maximization.



# CURRICULUM VITAE

## PERSONAL INFORMATION

**Surname, Name:** Gürsoy, Gönenç

**Nationality:** Turkish (TC)

**Date and Place of Birth:** 20 Jun 1985, Kargı

**Marital Status:** Married

**E-mail:** gursoy@aerotim.com.tr

## EDUCATION

<b>Degree</b>	<b>Institution</b>	<b>Year of Graduation</b>
Ph.D.	METU, Dept. of Aerospace Engineering	2016
M.S.	METU, Dept. of Aerospace Engineering	2010
B.S.	METU, Dept. of Aerospace Engineering	2007
High School	Dörtyol Süleyman Demirel Anadolu Lisesi	2002

## PROFESSIONAL EXPERIENCE

<b>Year</b>	<b>Place</b>	<b>Enrollment</b>
2010 - 2016	Aerotim Engineering L.L.C.	Senior Engineer - Flight Mechanics
2007 - 2010	Turkish Aerospace Industries, Inc.	Design Engineer

## AWARDS

<b>Date</b>	<b>Event/Place</b>	<b>Award</b>
2-5 Jan. 2015	AIAA SciTech Forum, Kissimmee, FL	AFM Best Student Paper Award

## **PUBLICATIONS**

### **International Journal Publications**

Gursoy, G., Yavrucuk, I., Direct Adaptive Limit and Control Margin Estimation with Concurrent Learning, *Journal of Guidance, Control and Dynamics*, AIAA, May 2016,

DOI: 10.2514/1.G001515

Gursoy, G., Yavrucuk, I., Concurrent Learning Enabled Adaptive Limit Detection for Active Pilot Cueing, *Journal of Aerospace Information Systems*, AIAA, July 2014,

DOI:10.2514/1.I010205

### **National Journal Publications**

Gürsoy, G, Yavrucuk, İ, Paraşüt-Yük Sistemlerinin Dinamik Modellenmesi ve Yol-Nokta Takibi, EMO Bilimsel Dergi, SAVTEK Özel Sayısı Cilt 3, Sayı 5, Sayfa 49-54, Haziran 2013

### **International Conference Publications**

Gursoy, G., Yavrucuk, I., Online Sensitivity Estimation Based Non-Iterative Control Margin Estimation for Rotorcraft Vertical Speed Protection, 8th Ankara International Aerospace Conference, 10-12 Sept. 2015, Ankara, TR

Gursoy, G., Yavrucuk, I., Non-Iterative Adaptive Vertical Speed Limit and Control Margin Prediction for Fly-By-Wire Helicopter, 71st American Helicopter Society Forum, 5-7 May 2015, Virginia Beach, VA

Gursoy, G., Yavrucuk, I., Non-Iterative Direct Adaptive Control Margin Estimation, 3rd CEAS Specialist Conference on Guidance, Navigation & Control (EuroGNC 2015), 13-15 April 2015, Toulouse, FR

Gursoy, G., Yavrucuk, I., Non-Iterative Adaptive Limit and Control Margin

Estimation with Concurrent Learning , AIAA SciTech 2015 Forum (AFM Best Student Paper), 2-5 Jan. 2015, Kissimmee, FL

Okatan, M., Gursoy, G., Yavrucuk, I., Kalman Filter Based Modification on Helicopter Adaptive Control, AIAA SciTech 2015 Forum, 2-5 Jan. 2015, Kissimmee, FL

Gursoy, G., Novikov, Y., Yavrucuk, I., Engine Limit Detection and Avoidance for Helicopters with Multiple Limits, AIAA Atmospheric Flight Mechanics Conference, 19-22 Aug. 2013, Boston, MA

Yavrucuk, I., Gursoy, G., Novikov, Y., Online Detection and Avoidance of Helicopter TGT Limits, 69th American Helicopter Society Forum, 21-23 May 2013, Phoenix, AZ

Gursoy, G., Prach, A., Yavrucuk, I., Design of a Waypoint Tracking Control Algorithm for Parachute-Payload Systems, 2nd CEAS Specialist Conference on Guidance, Navigation & Control, 10-12 April 2013, TU Delft, Netherlands

Gursoy, G., Tarimci, O., Yavrucuk, I., Helicopter Slung Load Simulations Using Heli-Dyn+, AIAA Modeling and Simulation Technologies Conference, 13-16 August 2012, Minneapolis, MN

Yavrucuk, I., Gursoy, G., Limit Margin Prediction For Helicopters Using Long Term Learning Adaptive Neural Networks, European Rotorcraft Forum, 13-15 September 2011, Vergiate and Gallarate, ITALY

Gursoy, G., Yavrucuk, I., Long Term Learning Adaptive Neural Network Estimator Based Limit Detection, Adaptation and Learning in Control and Signal Processing & Periodic Control Systems Conference, 26-28 Aug. 2010, Antalya, TURKEY

Gursoy, G., Yavrucuk, I., Concurrent Learning Enabled Adaptive Limit Detection for Active Pilot Cueing, AIAA Atmospheric Flight Mechanics Conference, 2-5 August 2010, Ontario, CANADA

## National Conference Publications

Gürsoy, G., Yavrucuk, İ., Fly-By-Wire Helikopterler için Uçuş Zarfı Limitleri Koruması, SAVTEK Konferansı, ODTÜ, 25-27 Haziran 2014, ANKARA

Gürsoy, G., Tarımcı, O., Yavrucuk, İ., Çok Rotorlu Hava Araçlarının Dinamik Modellenmesi ve Adaptif Kontrolü, SAVTEK Konferansı, ODTÜ, 25-27 Haziran 2014, ANKARA

Tarımcı, O., Türkal, S. M., Gürsoy, G., Yavrucuk, İ., Konfigüre Edilebilir Helikopter Mühendislik ve Eğitim Simülatörü, SAVTEK Konferansı, ODTÜ, 25-27 Haziran 2014, ANKARA

Gürsoy, G., Tarımcı, O., Yavrucuk, İ., Helikopter Otopilotları için Yaklaşma Ust Modu Tasarımı ve Simülasyonu, USMOS Konferansı, ODTÜ, 10-11 Haziran 2013, ANKARA

Novikov ,Y., Gürsoy, G., Uzol, O., Yavrucuk, İ., Helikopter Simülatörleri için Dinamik Turboşaft Motor Modeli Geliştirilmesi ve Simülatöre Entegrasyonu, USMOS Konferansı, ODTÜ, 10-11 Haziran 2013, ANKARA

Gürsoy, G., Tarımcı, O., Kubalı, E., Yavrucuk, İ., Heli-Dyn+ Kullanılarak Helikopter-Yük Sistemi Simülasyonlarının Gerçeklenmesi, SAVTEK Konferansı, ODTÜ, 20-22 Haziran 2012, ANKARA

Gürsoy, G., Yavrucuk, İ., Paraşüt-Yük Sistemlerinin Dinamik Modellenmesi ve Nokta Kontrolü, SAVTEK Konferansı, ODTÜ, 20-22 Haziran 2012, ANKARA