

Closeout Notice Date 20-JAN-2000

Project Number E-16-N67

Doch Id 40743

Center Number 10/24-6-R0385-OA0

Project Director CALISE, ANTHONY

Project Unit AERO ENGR

Sponsor NASA/AMES RESEARCH CTR/ CA

Division Id 3386

Contract Number NCC 2-981

Contract Entity GTRC

Prime Contract Number

Title NONLINEAR FLIGHT CONTROL FOR TILT ROTOR AIRCRAFT OPERATING IN  
TERMINAL AREA

Effective Completion Date 31-AUG-1999 (Performance) 30-NOV-1999 (Reports)

Closeout Action:	Y/N	Date Submitted
Final Invoice or Copy of Final Invoice	Y	
Final Report of Inventions and/or Subcontracts	Y	
Government Property Inventory and Related Certificate	N	
Classified Material Certificate	N	
Release and Assignment	N	
Other	N	

Comments

Distribution Required:

Project Director/Principal Investigator	Y
Research Administrative Network	Y
Accounting	Y
Research Security Department	N
Reports Coordinator	Y
Research Property Team	N
Supply Services Department/Procurement	Y
Georgia Tech Research Corporation	Y
Project File	Y

NOTE: Final Patent Questionnaire sent to PDPI

E-16-N67  
#1

# Nonlinear Flight Control for Tilt-Rotor Aircraft Operating in the Terminal Area

Final Report  
November 1999

1 July 1997 - 31 August 1999

Research supported by the NASA Ames Research Center  
NASA Cooperative Agreement No. NCC 2-981

Anthony J. Calise, Principal Investigator  
Rolf T. Rysdyk, Investigator  
Robert T. Chen, William A. Decker, NASA Grant monitors

School of Aerospace Engineering  
Georgia Institute of Technology  
Atlanta, Georgia, 30332-0150

# Contents

List of Figures	3
<b>1 Introduction</b>	<b>1</b>
<b>2 Nomenclature</b>	<b>1</b>
<b>3 Experimental Setup</b>	<b>2</b>
<b>4 Discussion of Results</b>	<b>3</b>
<b>5 Recommendation</b>	<b>4</b>
References	10
Appendix	11
<b>A Full Authority Implementation</b>	<b>12</b>
<b>B Closed Loop Architecture</b>	<b>12</b>
B.1 Stick Input: Roll, Pitch, and Yaw Command . . . . .	13
B.2 Command Filter . . . . .	13
B.3 Linear controller; PI and PD Tracking Error Dynamics . . . . .	16
B.3.1 Roll channel (RCAH) . . . . .	16
B.3.2 Pitch channel (ACAH) . . . . .	16
B.3.3 Yaw channel (RCAH) . . . . .	17
B.4 Pseudo Control . . . . .	17
B.5 Model Inversion . . . . .	17
B.6 Actuator Input . . . . .	19
<b>C Linear-in-the-Parameters Neural Network Structure</b>	<b>20</b>
<b>D Creation of the Input Vector <math>\beta</math></b>	<b>21</b>
D.1 The Roll-Channel . . . . .	21
D.2 The Pitch-Channel . . . . .	21
D.3 The Yaw-Channel . . . . .	21
D.4 Vector of basisfunctions . . . . .	22
D.5 The weight vector, longitudinal channel . . . . .	22
D.6 The neural net output . . . . .	23
D.7 The weight vector update for the lateral channels . . . . .	23
D.8 The command generation for the yaw-channel: turn coordination . . . . .	24
<b>E Initializing the Architecture</b>	<b>25</b>
<b>F Stick Trim Position</b>	<b>27</b>

## List of Figures

1	The linear-in-the-parameters neural network. . . . .	2
2	Hover, $\widehat{M}_\delta = -1.0$ , $VN = (V_T - 30)/100$ , $\gamma = 0, 100$ . . . . .	5
3	150 Kts airplane mode, $\widehat{M}_\delta = -1.0$ , $VN = (V_T - 30)/100$ , $\gamma = 0, 100$ . . . . .	6
4	Hover, $\widehat{M}_\delta = -1.0$ , $VN = (V_T - 30)/30$ , $\gamma = 0, 100$ . . . . .	7
5	150 Kts airplane mode, $\widehat{M}_\delta = -1.0$ , $VN = (V_T - 30)/30$ , $\gamma = 0, 100$ . . . . .	8
6	150 Kts airplane mode, $\widehat{M}_\delta = -1.0$ , $VN = (V_T - 30)/30$ , $\gamma = 0.1, 1.0, 8.0, 16.0$ . . . . .	9
7	Original XV-15 and Full authority implementation. . . . .	12
8	Adaptive NN augmented model inversion architecture, in the longitudinal channel, configured for ACAH. . . . .	12
9	Adaptive NN augmented model inversion architecture, in the roll-channel, configured for RCAH. . . . .	13
10	Command filter for RCAH in roll and yaw-channels. . . . .	14
11	Command filter for ACAH in pitch channel. . . . .	14
12	Turn coordination through suppression of lateral acceleration. . . . .	25
13	NN augmented model inversion architecture, at ACAH trim in the longitudinal channel. . . . .	26
14	NN augmented model inversion architecture, at ACAH trim in the longitudinal channel. . . . .	27

# 1 Introduction

A neural-network (NN) augmented model inversion flight controller architecture was developed at Georgia Tech [1]. The architecture demonstrated consistent handling qualities in the approach to landing phase of a civil tiltrotor aircraft, in off-line simulations of the XV-15 tiltrotor [2]. The off-line simulations focused on the approach phase of flight, in general starting with zero NN weights, at 3000', 150 Kts, in airplane mode, converting to helicopter and approaching hover at 1000' [3].

A subsequent piloted simulation evaluation of this NN-augmented model inversion architecture at NASA Ames revealed

- negligible NN activity at hover, and
- numerical ringing at high velocities.

The observations pertained to the linearly parametrized NN used in [1, 2].

This report contains the investigation of these observations in off-line simulation. It is focused on performance in the longitudinal channel, and uses a degraded element in the control sensitivity matrix, (the B-matrix), to induce inversion error. The results found in piloted simulation were reproduced, and simple remedies are investigated.

# 2 Nomenclature

$M_\delta$	control sensitivity of longitudinal channel, element(2,2) of B-matrix
$\widehat{M}_\delta$	approximation of $M_\delta$ (This value is used in the inversion model.)
$\gamma$	learning rate
$u_b, w_b$	velocity components along body axis
$\theta_c$	commanded pitch attitude angle
$\theta$	pitch attitude angle
$thtN$	normalized pitch angle
$wN$	normalized velocity component along body $z$ -axis
$VT$	total velocity
$VN$	normalized total velocity
$V_{scale}$	scaling-factor of velocity input to neural network
$\beta_{mast}$	rotor mast angle in degrees
$betaN$	normalized mast angle
$U_{ad2}$	neural network output in the pitch channel
$U_{ad2N}$	'squashed' neural network output
$C_i$	neural network input layer categories
$\delta_E$	longitudinal actuator deflection
$\delta_{COL}$	collective deflection
$\Delta$	inversion error

### 3 Experimental Setup

The linear-in-the-parameters NN is shown in Fig 1. In terms of the nomenclature used in Ref [4] the NN inputs were selected as follows:

$$wN = w_b \quad (1)$$

$$thtN = \theta \quad (2)$$

$$thtdN = \dot{\theta} \quad (3)$$

$$VN = (V_T - 30)/V_{scale} \quad (4)$$

$$VN2 = VN \times VN \quad (5)$$

$$betadN = \beta_{mast}[deg]/90 \quad (6)$$

$$Uad2N = 2 * (1 - e^{-U_{ad2}})/(1 + e^{-U_{ad2}}) \quad (7)$$

The Kronecker products [2] which we used to form the network basis functions, are constructed using the following input categories:

$$C_1 = \{ 0.1 VN VN2 \} \quad (8)$$

$$C_2 = \{ 1 betadN wN thtN thtdN Uad2N \} \quad (9)$$

$$C_3 = \{ 1 thtN \} \quad (10)$$

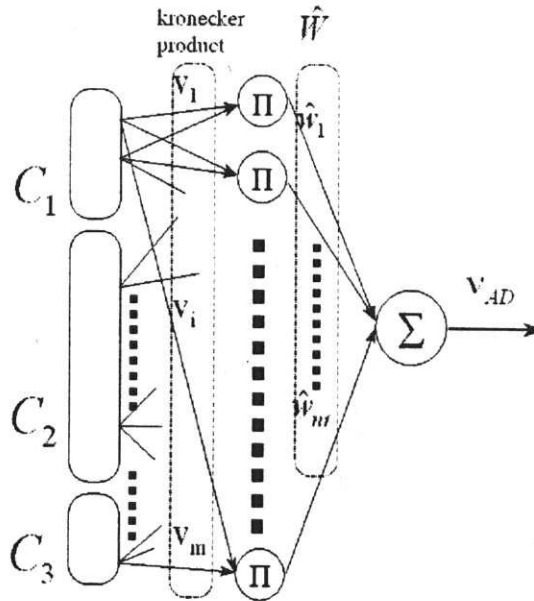


Figure 1: The linear-in-the-parameters neural network.

Two operating points and two values for  $V_{scale}$  are considered in this investigation. These operating points are at 'opposite' ends of the approach envelope, as shown in Table 1. The flight conditions are:

- Hover in helicopter mode at 1,000', with a simple pilot model to maintain altitude through control of the collective/throttle, while commanding a square wave in pitch. The magnitude of the pitch command was chosen to obtain responses about the hover condition. Other configuration parameters were set at: Flaps=3, RPM=589, G.W.=13,000, c.g. 295", Gear up.
- Level forward flight at 10,000' in full airplane mode, 150 Kts, Flaps=1, RPM=516.9, with pitch attitude commands.

The NN performance was evaluated in 30 sec runs. The command consists of a square wave input, shaped to provide a response that remains approximately centered about the operating point at the hover condition. To emphasize the effects of inversion error, the linear model obtained at 30 Kts operating point is degraded by using  $\widehat{M}_\delta^{-1} = -1.0$ , in stead of  $M_\delta^{-1} = -4.38$ .

## 4 Discussion of Results

The performance with and without the NN is compared. In all cases, the NN is initiated with zero weights. The results are displayed in Figs 2 through 6. An overview of the figures related to each case is given in Table 1.

**Table 1: Overview of results shown in figures**

$\widehat{M}_\delta^{-1} = -1.0$	Hover, at 1,000'	Fwd Flight, 150 Kts, at 10,000'	
	$\gamma = 0, 100$	$\gamma = 0, 100$	$\gamma = 0.1, 1.0, 8.0, 16.0$
$V_N = (V_T - 30)/100$	Fig 2	Fig 3	
$V_N = (V_T - 30)/30$	Fig 4	Fig 5	Fig 6

Fig 2 shows the response to a square wave commanded pitch attitude input at hover for  $V_{scale} = 100$ , which is the value used in the piloted simulations. The  $\Delta$  trace indicates that some inversion error is present. The linear NN does not provide significant compensation for the error within the 30 second run investigated here. Note that  $\nu_{ad}$  is essentially zero. The  $\theta$ ,  $u_b$ , and  $w_b$  responses are shown for both  $\gamma = 0$  (no adaptation) and  $\gamma = 100$ . The responses are nearly identical.

The same NN design applied to the forward flight case is shown in Fig 3. This result indicates improved performance of  $\gamma = 100$  in comparison with the  $\gamma = 0$  result. It also shows that  $\nu_{ad}$  nearly cancels  $\Delta$ . The combination of Figs 2 and 3 suggests that the performance of the NN is affected by the operating conditions.

To provide improved response for both operating conditions, we changed the value of  $V_{scale}$ . Fig 4 indicates the result for  $V_{scale} = 30$ , and  $\gamma = 0, 100$ . The results are quite satisfactory at this operating point.

Fig 5 shows the  $V_{scale} = 30$  response in forward flight. This result indicates a 'ringing' effect associated with a rapid growth of weights. A remedy may be provided by simple scaling of the learning rate. A simple design procedure, applied successfully in similar efforts, involved finding the minimum value for  $\gamma$  that provides reasonable NN compensation, and taking twice that value as the nominal design for  $\gamma$ . Fig 6 shows an example of the effect of different  $\gamma$ 's for the forward flight case. Very good response is obtained with  $\gamma = 0.1$ . Increasing the value for  $\gamma$  provides some improvement, but not significantly so. The value for  $\gamma$  may be increased by two orders of magnitude before leading to the 'ringing' effect. This result implies that the adaptation gain should be scheduled with flight condition, and that a large range of values for  $\gamma$  is available for proper design.

## 5 Recommendation

The observations made during piloted simulation at NASA Ames, i.e.

- negligible NN activity at hover, and
- numerical ringing at high velocities.

may both be avoided by lowering the value for  $V_{scale}$ , and introducing a simple scheduling of the learning rate. A straightforward design procedure consists of finding the minimum learning rate that provides desired response, and taking twice that value as the actual learning rate for that design point.

Although the circumstances of the above observations are inherent to most forms of adaptive control based on tracking error signals, it is likely that these effects will be reduced by using a so-called multi-layer perceptron Neural Network [3].

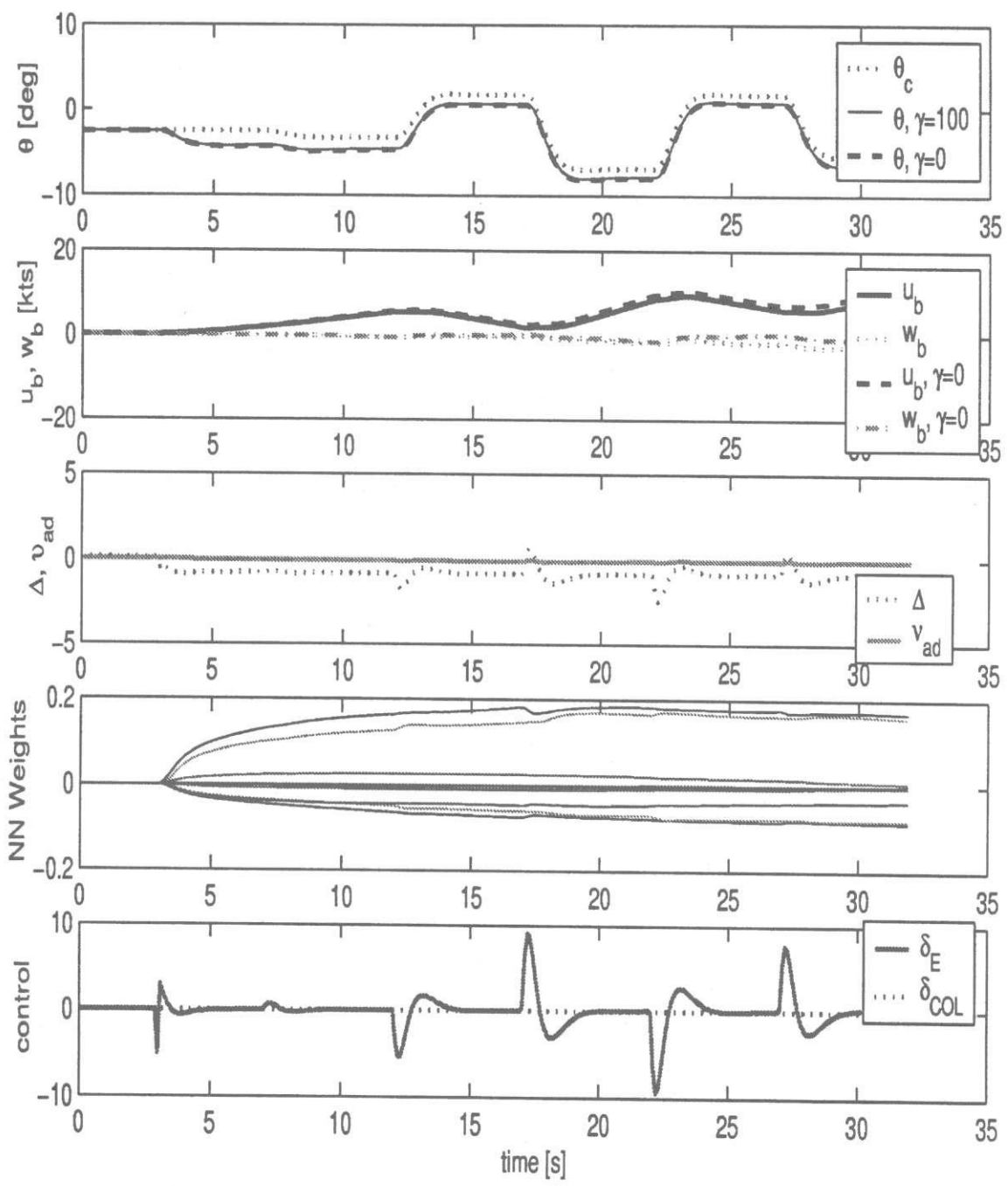


Figure 2: Hover,  $\widehat{M}_\delta = -1.0$ ;  $VN = (V_T - 30)/100$ ,  $\gamma = 0, 100$ .

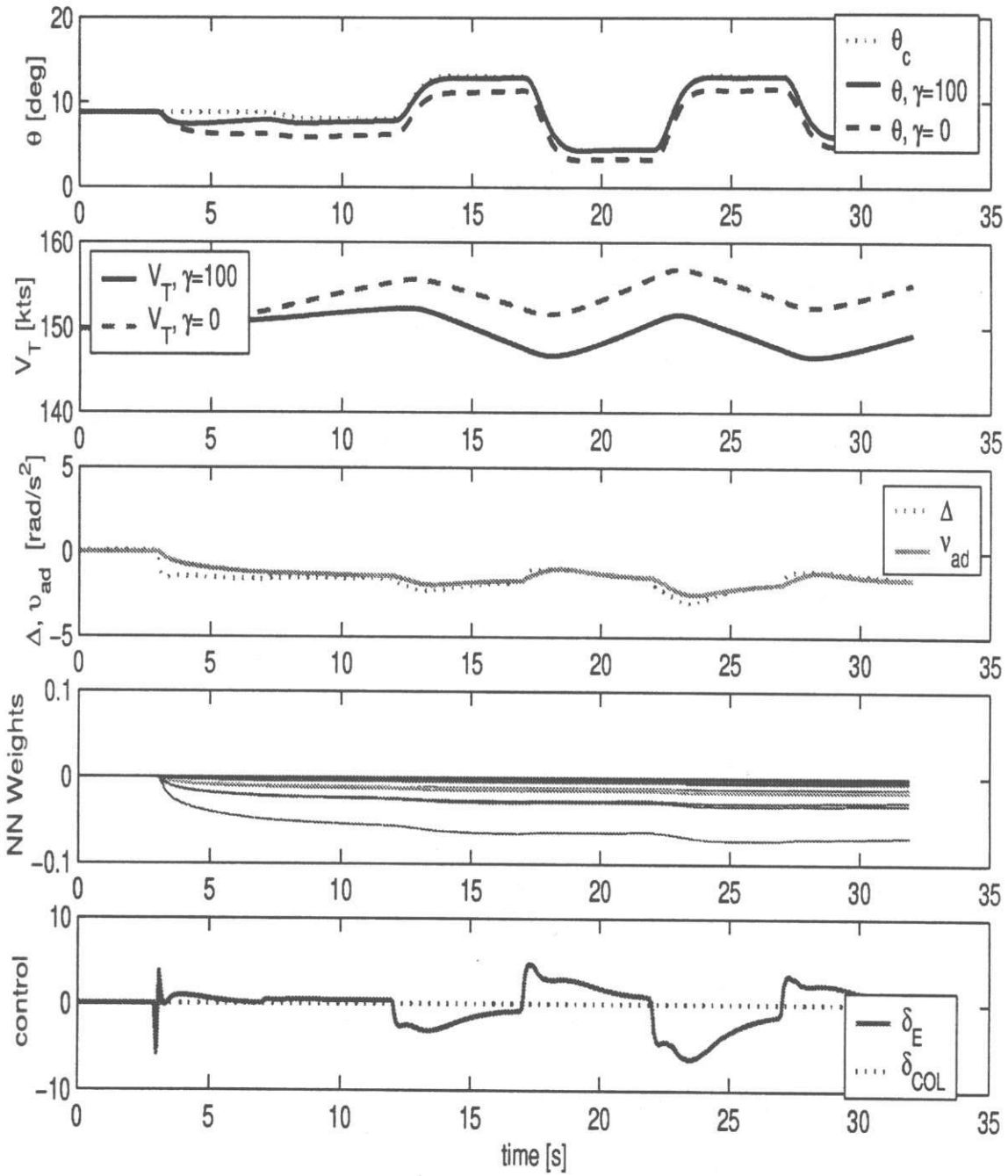


Figure 3: 150 Kts airplane mode,  $\widehat{M}_\delta = -1.0$ ,  $VN = (V_T - 30)/100$ ,  $\gamma = 0, 100$ .

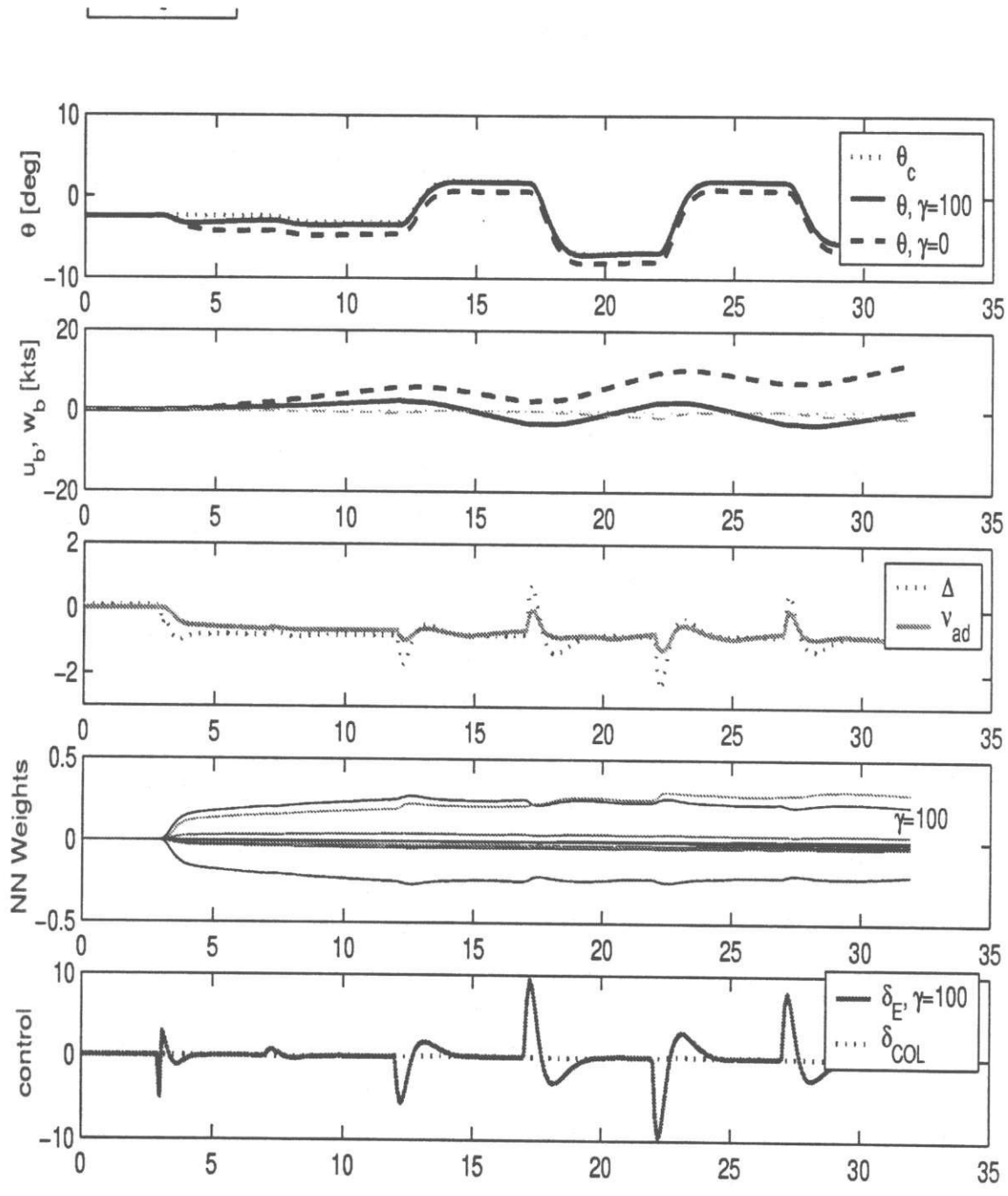


Figure 4: Hover,  $\widehat{M}_\delta = -1.0$ ,  $VN = (V_T - 30)/30$ ,  $\gamma = 0, 100$ .

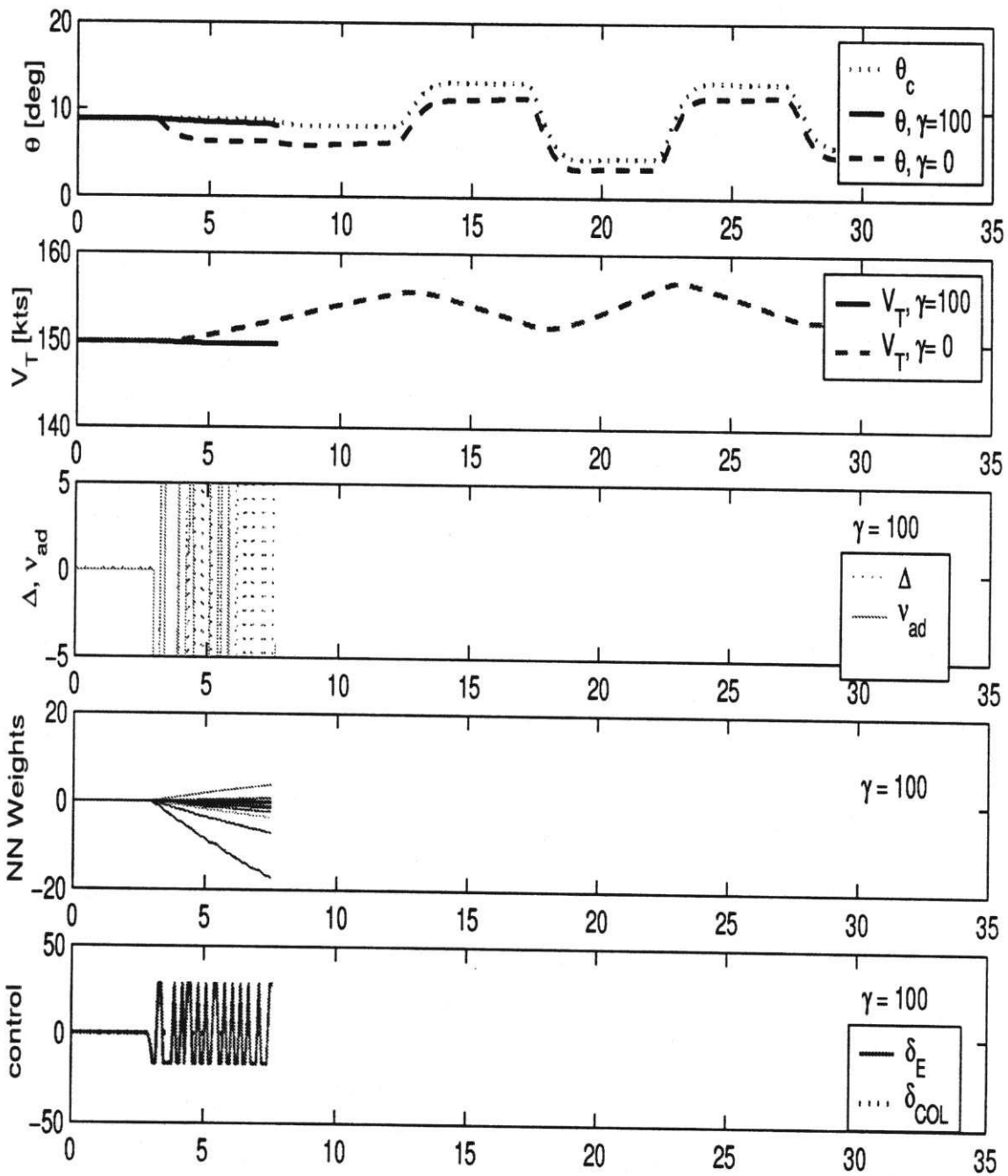


Figure 5: 150 Kts airplane mode,  $\widehat{M}_\delta = -1.0$ ,  $VN = (V_T - 30)/30$ ,  $\gamma = 0, 100$ .

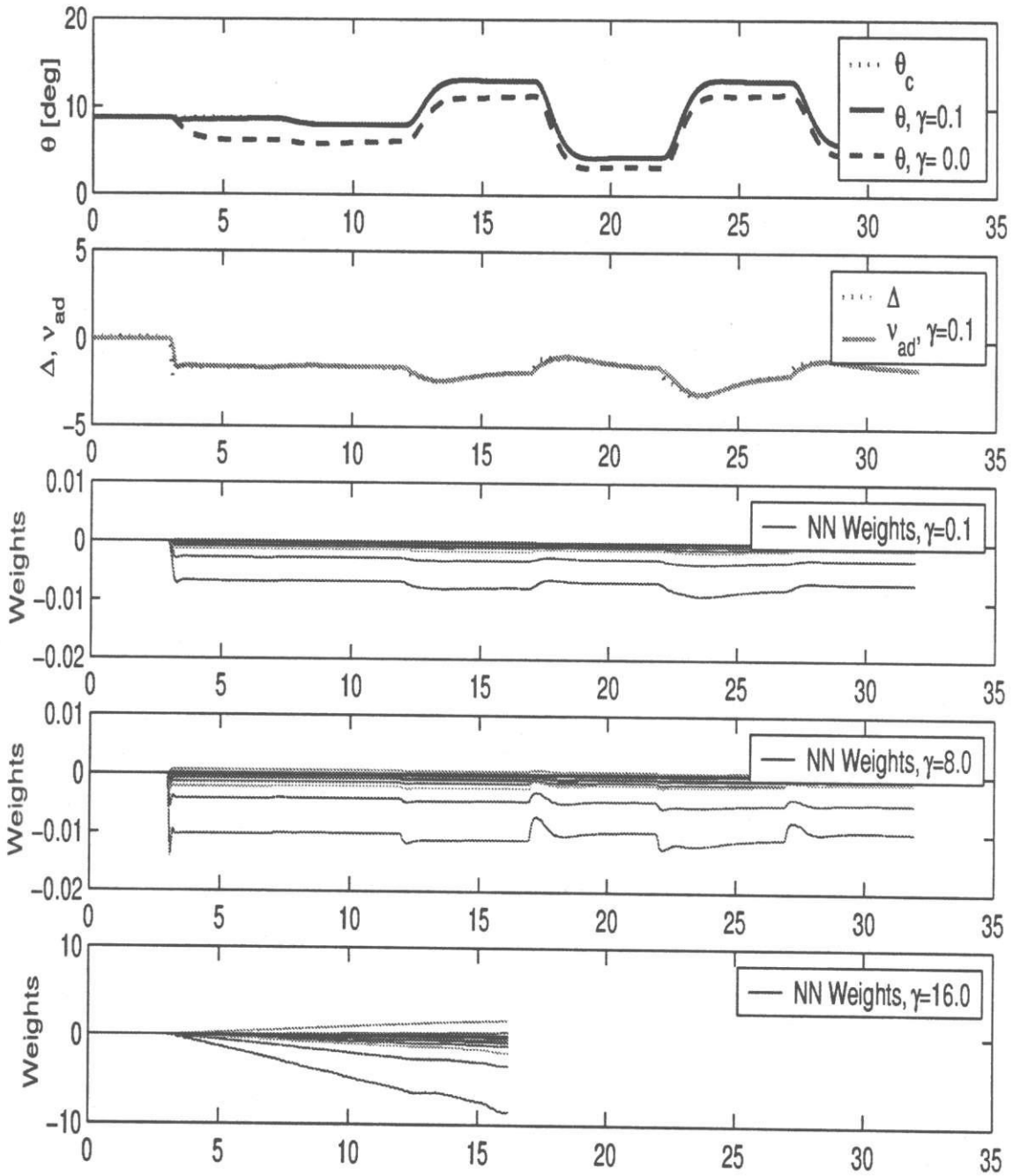


Figure 6: 150 Kts airplane mode,  $\widehat{M}_\delta = -1.0$ ,  $VN = (V_T - 30)/30$ ,  $\gamma = 0.1, 1.0, 8.0, 16.0$ .

## References

- [1] A.J. Calise and R.T. Rysdyk. Nonlinear adaptive flight control using neural networks. *IEEE Control Systems Magazine*, December 1998. Special Issue on Emerging Technologies.
- [2] R.T.Rysdyk and A.J.Calise. Adaptive model inversion flight control for tiltrotor aircraft. In *AIAA Journal of Guidance, Control, and Dynamics*, February 1998.
- [3] R.T.Rysdyk. *Adaptive Nonlinear Flight Control*. PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA, November 1998. PhD-thesis.
- [4] Rolf Rysdyk. *Software Implementation of the Linear Neural Network Augmented Model Inversion Controller for the XV-15*. Technical report, Georgia Tech, December 1998. Informal report , communication with NASA Ames Research Center, Vertical Motion Simulator Dept.
- [5] B.Etkin. *Dynamics of Atmospheric Flight*. John Wiley & Sons, New York, 1972.

# Appendix:

## Software Implementation of the Linear Neural Network Augmented Model Inversion Controller for the XV-15

### nomenclature

$A$	matrix of state derivatives
$B$	matrix of control derivatives
$K$	gain
$P$	positive definite solution to Lyapunov equation
$V$	weight matrix, from input to hidden layer
$W$	weight matrix, to output layer
$X$	aircraft states
$\bar{X}$	normalized aircraft states, linear neural network input
$Z$	weight matrix, combining $V$ and $W$
$a$	translational acceleration
$c_\phi$	$\cos\phi$
$e$	vector of tracking error
$\hat{f}$	approximate model of aircraft dynamics
$g$	gravitational constant
$n_1, n_2$	resp. number of inputs, and outputs of network
$p, q, r$	resp. roll, pitch, and yaw rate about body axes
$s_\phi$	$\sin\phi$
$u, v, w$	resp. velocity along body x-, y-, and z-axis
$\beta$	vector of basis functions
$\beta$	side slip angle <i>deg</i>
$\beta_M$	tilt-rotor mast angle <i>deg</i>
$\delta$	control surface deflection
$\gamma$	learning rate
$\lambda$	e-modification gain
$\nu$	pseudo control
$\omega$	vector of angular rates
$\omega_n$	second order system natural frequency
$\sigma$	sigmoidal <i>squashing</i> function
$\tau$	time constant
$\theta$	pitch angle
$\zeta$	filtered error signal or damping coefficient

## A Full Authority Implementation

The neural network (NN) augmented model inversion architecture is implemented assuming a full-authority SCAS. Fig 7 compares this implementation to the original SCAS structure.

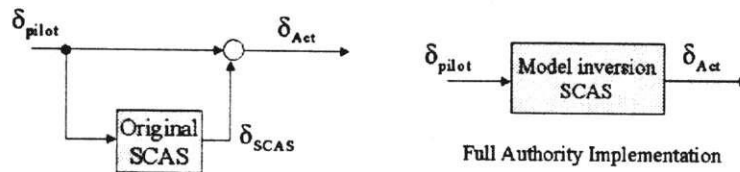


Figure 7: Original XV-15 and Full authority implementation.

## B Closed Loop Architecture

Fig 8 contains a diagram of the architecture used for implementation of ACAH control in the pitch channel. In Fig 8,

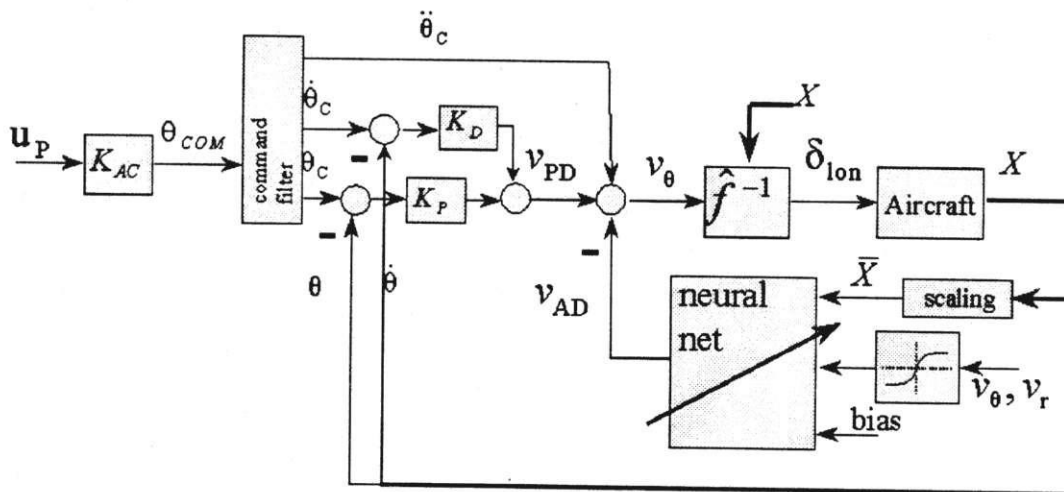


Figure 8: Adaptive NN augmented model inversion architecture, in the longitudinal channel, configured for ACAH.

$$v = \textit{pseudo-control, a desired (pitch) acceleration.} \quad (11)$$

$$\hat{f}^{-1} = \textit{an approximate inversion.} \quad (12)$$

$$\delta_{lon} = \textit{actuator-input in terms of "inches of stick deflection".} \quad (13)$$

Fig 9 contains a diagram of the architecture used for implementation of RCAH control in the roll channel. The yaw-channel is similar.

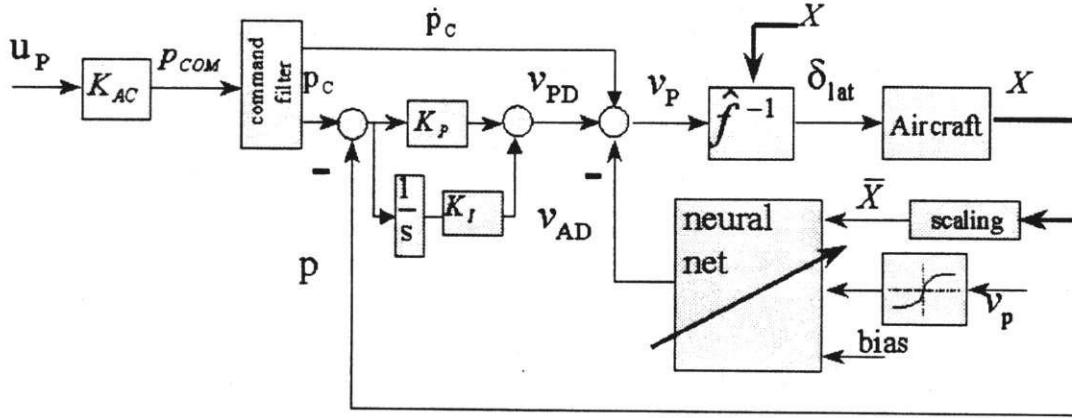


Figure 9: Adaptive NN augmented model inversion architecture, in the roll-channel, configured for RCAH.

## B.1 Stick Input: Roll, Pitch, and Yaw Command

A simple gain was used for  $K_{AC}$ , approximately similar to the original gains at the 30 Kts helicopter configuration.

$$K_{ACp} = 9 [\text{deg/sec/in}] = 0.16 [\text{rad/sec/in}] \quad (14)$$

$$K_{AC\theta} = -3 [\text{deg/in}] = -0.05 [\text{rad/in}] \quad (15)$$

$$K_{ACr} = -10 [\text{deg/sec/in}] = -0.17 [\text{rad/sec/in}] \quad (16)$$

If the model inversion is to be engaged in mid-flight; To avoid the transients and provide the attitude hold on activation, I used the following construction.

$$b_{rigging\theta} = \theta - K_{AC\theta}U_{P\theta} \quad (17)$$

$$\theta_{COM} = K_{AC\theta}U_{P\theta} + b_{rigging\theta} \quad (18)$$

Here (17) is used at initialization only, (18) is continuous. Furthermore, the command filters are initialized equal to the measured aircraft state at the moment of activation.

## B.2 Command Filter

The command filter is used to implement the desired handling qualities. For *Rate Command Attitude Hold* (RCAH) in the roll and yaw channels this was designed as in Fig (10). *Attitude Command Attitude Hold* is implemented as shown in Fig (11).

The filter input for each channel is resp.  $p_{COM}$ ,  $\theta_{COM}$ , and  $r_{COM}$ . The purpose of the command filter is to provide for smooth signals for the three channels, resp.

$$\begin{matrix} p_C & \text{and} & \dot{p}_C \\ \theta_C, & \dot{\theta}_C & \text{and} & \ddot{\theta}_C \\ r_C & \text{and} & \dot{r}_C \end{matrix} \quad (19)$$

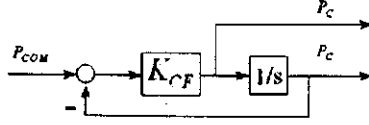


Figure 10: Command filter for RCAH in roll and yaw-channels.

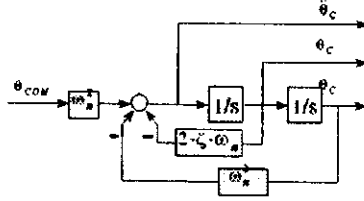


Figure 11: Command filter for ACAH in pitch channel.

The continuous time filter models for *Rate Command Attitude Hold* (RCAH) in channel p, and r are

$$A_p = -\frac{1}{\tau_p} \quad B_p = \frac{1}{\tau_p} \quad (20)$$

$$C_p = \begin{bmatrix} 1 \\ -\frac{1}{\tau_p} \end{bmatrix} \quad D_p = \begin{bmatrix} 0 \\ \frac{1}{\tau_p} \end{bmatrix} \quad (21)$$

With the roll-yaw coupling small, the time constant of the roll response is approximately the inverse of the bandwidth, so design

$$K_{cf} \geq \frac{1}{\tau_p} \approx \omega_{BW} \quad (22)$$

with

$$\tau_p = 0.5 \quad (23)$$

This provides the command filter for channel p, Fig 10, with a 5% settling time of approximately 1.5 seconds. The setup for the channel r is similar with a time constant

$$\tau_r = 0.25 \text{ sec} \quad (24)$$

*Attitude Command Attitude Hold* (ACAH) can be implemented as indicated in Fig 11. For the ACAH in the pitch channel

$$A_\theta = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ (\omega_n^2 R_3) & (2\zeta\omega_n R_3 - \omega_n^2) & (R_3 - 2\zeta\omega_n) \end{bmatrix} \quad B_\theta = \begin{bmatrix} 0 \\ 0 \\ -\omega_n^2 R_3 \end{bmatrix} \quad (25)$$

$$C_\theta = \text{Identity matrix} \quad D_\theta = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (26)$$

The dominant complex poles of the command filter, provide minimal overshoot

$$\zeta = 0.8 \quad (27)$$

and a 5% settling time of 1.5 seconds

$$\omega_n = 2.5 \text{ rad/sec} \quad (28)$$

The third pole is frequency separated from the dominant poles,

$$R_3 = -6.0 \quad (29)$$

The command filters were implemented in digital form through a zero-order hold approximation. The zero order hold is approximated by

$$A_d = I_n + A \times \Delta t + \frac{1}{2} \times A^2 \times \Delta t^2 \quad \text{and} \quad (30)$$

$$B_d = \left( I_n \Delta t + \frac{1}{2} A \Delta t^2 \right) B \quad (31)$$

where  $I_n$  is the identity matrix of appropriate dimension. The  $C$  and  $D$  matrices remain unaffected by the zero-order hold approximation. The filter output is then updated each integration step as

$$x = A_d x + B_d p_{COM} \quad (32)$$

$$y = C x + D p_{COM} \quad (33)$$

The output of the roll command filter

$$p_C = y(1) \quad (34)$$

$$\dot{p}_C = y(2) \quad (35)$$

The output of the yaw command filter is similar

$$r_C = y(1) \quad (36)$$

$$\dot{r}_C = y(2) \quad (37)$$

The output for the pitch command filter is

$$\theta_C = y(1) \quad (38)$$

$$\dot{\theta}_C = y(2) \quad (39)$$

$$\ddot{\theta}_C = y(3) \quad (40)$$

The signal  $\dot{\omega}_C$  is constructed from the command filter output as

$$\dot{\omega}_C = \begin{bmatrix} \dot{p}_C \\ \dot{\theta}_C \\ \dot{r}_C \end{bmatrix} \quad (41)$$

### B.3 Linear controller; PI and PD Tracking Error Dynamics

The tracking error dynamics are in part designed by

$$\nu_{PD} = \begin{bmatrix} \nu_{PD1} \\ \nu_{PD2} \\ \nu_{PD3} \end{bmatrix} \quad (42)$$

which is used to specify a desired settling on the tracking error. (Not to be confused with the desired handling qualities, which reside in the command filter.)

#### B.3.1 Roll channel (RCAH)

$$\nu_{PD1} = K_P \tilde{p} + K_I \int_{t_0}^t \tilde{p} d\tau \quad (43)$$

where

$$\tilde{p} = p_C - p \quad (44)$$

Here,  $p_C$  is command filter output, and  $p$  is the measured aircraft roll-rate.  $K_I$  and  $K_P$  were chosen so that the error dynamics settle in 0.5 seconds.

$$\zeta = 1.0 \quad (45)$$

$$\omega_n = 6.0 \quad (46)$$

$$K_P = 2\zeta\omega_n \quad (47)$$

$$K_I = \omega_n^2 \quad (48)$$

A similar representation is made for the yaw channel.

**Remark 2.1** The potential for integrator wind-up is an aspect to be addressed in the implementation for the simulator.

#### B.3.2 Pitch channel (ACAH)

$$\nu_{PD2} = K_P \tilde{\theta} + K_D \dot{\tilde{\theta}} \quad (49)$$

here

$$\tilde{\theta} = \theta_C - \theta \quad (50)$$

where  $\theta_C$  is the command filter output.  $K_P$  and  $K_D$  were chosen so that the error dynamics settle in 0.5 seconds

$$\zeta = 1.0 \quad (51)$$

$$\omega_n = 6.0 \quad (52)$$

$$K_D = 2\zeta\omega_n \quad (53)$$

$$K_P = \omega_n^2 \quad (54)$$

### B.3.3 Yaw channel (RCAH)

$$\nu_{PD3} = K_P \tilde{r} + K_I \int_{t_0}^t \tilde{r} d\tau \quad (55)$$

where

$$\tilde{r} = r_C - r \quad (56)$$

Here,  $r_C$  is command filter output or the output of automatic turn coordination, and  $r$  is the measured aircraft yaw-rate.

## B.4 Pseudo Control

The objective of this section is to construct the pseudo control, which forms the input to the actual inversion. The pseudo control is in the form of angular acceleration [ $rad/sec^2$ ]. The *pseudo control*,  $\nu$ , for the three channels is

$$\nu = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{bmatrix} \quad (57)$$

It is build up as

$$\nu = \nu_{PD} + \dot{\omega}_C - \nu_{AD} \quad (58)$$

where  $\omega_C$  was described in B.2,  $\nu_{ad}$  is the output of the NNs,

$$\nu_{ad} = \begin{bmatrix} \nu_{ad1} \\ \nu_{ad2} \\ \nu_{ad3} \end{bmatrix} \quad (59)$$

see section C, and  $\nu_{PD}$  is the output of a linear controller operating on an error signal, section B.3.

## B.5 Model Inversion

The object of this part is to calculate the actuator deflections with respect to the nominal actuator positions. The resulting actuator postions will form the output of the full-authority SCAS system.

The model inversion control is based on dynamics linearized about a nominal operating point, with the rotor dynamics residualized Eqn (63).  $\omega$  contains the rates about the body fixed axes [5]

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (60)$$

The architecture augments the controls of moments about the body axes, the controls of interest are

$$\delta = \begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{dir} \end{bmatrix} \quad (61)$$

The collective / throttle control position is treated as one of the relatively slow translational states, so

$$\mathbf{x}_1 = \begin{bmatrix} u \\ v \\ w \\ \delta_{col} \end{bmatrix} \quad (62)$$

A linear model of the XV-15, operating at a nominal operating point (I used 30 Kts at Helicopter Configuration, level flight at 1000' SA) is obtained as (for illustration only, for implementation see Eqn (110).)

$$\dot{\omega} = A_1 \mathbf{x}_1 + A_2 \omega + B \delta \quad (63)$$

where  $A_1$ ,  $A_2$  and  $B$  represent respectively the aerodynamic stability and control derivatives at the nominal operating point. The  $B$  matrix will be inverted in the calculation of the actuator deflections, Eqn (110).

The values that I used are

$$A_1 = \begin{bmatrix} 0.0 & -0.0074 & 0.0 & 0.0 \\ 0.0030 & 0.0 & -0.0024 & -0.0069 \\ 0.0 & 0.0090 & 0.0 & 0.0 \end{bmatrix} \quad (64)$$

$$A_2 = \begin{bmatrix} -0.6183 & 0.0 & 0.1673 \\ 0.0 & -0.7501 & 0.0 \\ 0.4488 & 0.0 & -0.2144 \end{bmatrix} \quad (65)$$

$$B^{-1} = \begin{bmatrix} 5.6748 & 0.0 & 1.4519 \\ 0.0 & -4.3894 & 0.0 \\ 1.6381 & 0.0 & 9.4878 \end{bmatrix} \quad (66)$$

Note that Eqn (66) gives  $B^{-1}$  and not  $B$ .

Use inversion of (63). This involves replacing the left-hand side of the equation with desired angular accelerations,

$$\dot{\omega}_D = \begin{bmatrix} \dot{p}_D \\ \dot{q}_D \\ \dot{r}_D \end{bmatrix} \quad (67)$$

where the elements of the desired acceleration,  $\dot{\omega}_D$ , for the combination of RCAH in channels p, and r, and ACAH in channel  $\theta$ , are

$$\begin{aligned}\dot{p}_D &= \nu_p \\ \dot{q}_D &= \frac{\nu_\theta}{c_\phi} + \nu_r t_\phi + pqt_\phi + pr + 2qrs_\phi t_\theta + q^2 s_\phi t_\phi t_\theta + r^2 c_\phi t_\theta \\ \dot{r}_D &= \nu_r\end{aligned}\quad (68)$$

where  $s_\phi$  is shorthand for  $\sin(\phi)$ , etc.

The actuator positions  $\delta$  relative to the nominal can be found as

$$\delta = \hat{B}^{-1}\{\dot{\omega}_D - \hat{A}_1 \mathbf{x}_1 - \hat{A}_2 \omega\} \quad (69)$$

## B.6 Actuator Input

The output of the full-authority SCAS system can now be formed as

$$\delta_{SCAS} = \delta_{30} - \delta_{neutral} + \delta \quad (70)$$

**Remark 2.2** With respect to trimming the simulator. In expression (110) the contribution of the NN enters through

$$\dot{\omega}_D = \dot{\omega}_D^{pseudo} + \dot{\omega}_D^{ad} \quad (71)$$

Therefore we can define  $\delta_{ad}$  as

$$\delta_{ad} = \hat{B}^{-1}\{\dot{\omega}_D^{ad}\} \quad (72)$$

where

$$\begin{aligned}\dot{p}_D &= \nu_{adp} \\ \dot{q}_D &= \frac{\nu_{ad\theta}}{c_\phi} + \nu_{adr} t_\phi + pqt_\phi + pr + 2qrs_\phi t_\theta + q^2 s_\phi t_\phi t_\theta + r^2 c_\phi t_\theta \\ \dot{r}_D &= \nu_{adr}\end{aligned}\quad (73)$$

Furthermore, at a particular trim operating point (for e.g. a simulator initial condition), indicated by an asterix '\*',

$$\dot{\omega}_D^{pseudo} = 0 \quad (74)$$

$$\omega = 0 \quad \text{however,} \quad (75)$$

$$\mathbf{x}_1 = \mathbf{x}_1^* - \mathbf{x}_{130} \quad (76)$$

The contribution of  $\delta$  in the SCAS output expression (109) is determined by

$$\delta = \hat{B}^{-1}\{\dot{\omega}_D - \hat{A}_1 x_1 - \hat{A}_2 \omega\} \quad (77)$$

$$= \hat{B}^{-1}\{\dot{\omega}_{D\ ad} - \hat{A}_1 x_1\} \quad (78)$$

$$= \delta_{ad} - \hat{B}^{-1} \hat{A}_1 (x_1^* - x_{130}) \quad (79)$$

The issue of initializing the actuator positions now becomes one of determining  $\delta_{ad}$  necessary for zero accelerations. This in turn means determination of the appropriate NN weights at any operating point.

**Remark 2.3** In the original code the SCAS output was

$$\begin{aligned} &XLTSAS \\ &XLNSAS \\ &XPDSAS \end{aligned} \quad (80)$$

Because of the full authority implementation, the coding in CONMIX.FOR has become

$$XLTT = ASAS = XLTSAS \quad (81)$$

and similar for the other channels.

## C Linear-in-the-Parameters Neural Network Structure

The definition of the NN consists of two parts.

1. the structure, and
2. the update law.

For a linear-in-the-parameters network the structure is described by

$$\nu_{AD} = \hat{W}' \beta(\bar{x}, \nu) = [w_1 \ w_2 \ w_3 \ \dots] \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n_1} \end{bmatrix} \quad (82)$$

where  $\hat{W}$  is an  $n_1 \times n_2$  *Weight* matrix initialized at all 0's, and  $\beta$  is made up of selected elements of the state vector, which is described next.

## D Creation of the Input Vector $\beta$

The NNs require the following scaled input signals

$$\begin{aligned}
 \bar{V} &= V/100 & V &\text{ equivalent total airspeed in } Kts \\
 \bar{w} &= w/100 & w &\text{ velocity along body z-axis in } ft/s \\
 \bar{\beta}_M &= \beta_M/90 & \beta_M &\text{ tiltrotor mast-angle in } deg \\
 \bar{a}_y &= a_y/32 & a_y &\text{ acceleration along body y-axis in } ft/s^2 \\
 \bar{\nu}_\theta &= \frac{1-e^{-\nu_\theta}}{1+e^{-\nu_\theta}} & & \\
 p, q, r & & &\text{ body angular rates } rad/s \\
 \phi, \theta & & &\text{ Euler angles } rad
 \end{aligned} \tag{83}$$

Note,  $V = \frac{\rho}{\rho_0} \sqrt{u^2 + v^2 + w^2}$  where  $\rho_0$  is air density at sea-level. Here, the pseudo control signals  $\nu$  are *squashed* by a sigmoidal function, in stead of just scaled.

$$\bar{\nu}_\theta = \frac{1 - e^{-\nu_\theta}}{1 + e^{-\nu_\theta}} \tag{84}$$

Similarly for  $\bar{\nu}_r$  and  $\bar{\nu}_p$ .

The input to each NN is organized into three seperate vectors, made up from the signals above.

### D.1 The Roll-Channel

The input vectors for the Roll-Channel are constructed from a 0.1 *bias* and the above signals as follows

$$\begin{aligned}
 C_1 &= [0.1, \bar{V}, \bar{V}^2] \\
 C_2 &= [0.1, \bar{\beta}_M, \bar{a}_y, \phi, p, \bar{\nu}_p] \\
 C_3 &= [0.1, \theta]
 \end{aligned} \tag{85}$$

### D.2 The Pitch-Channel

$$\begin{aligned}
 C_1 &= [0.1, \bar{V}, \bar{V}^2] \\
 C_2 &= [0.1, \bar{\beta}_M, \bar{w}, \theta, q, \bar{\nu}_\theta, \bar{\nu}_r] \\
 C_3 &= [0.1, \theta]
 \end{aligned} \tag{86}$$

### D.3 The Yaw-Channel

$$\begin{aligned}
 C_1 &= [0.1, \bar{V}, \bar{V}^2] \\
 C_2 &= [0.1, \bar{\beta}_M, \bar{a}_y, r, \bar{\nu}_r] \\
 C_3 &= [0.1, \phi]
 \end{aligned} \tag{87}$$

**Remark 4.1** To allow for easier implementation, it is possible to make the  $C_2$  vectors all of the same length, and setting the extra elements equal to 0, i.e.

$$\begin{aligned} C_2 \text{ in roll-channel} &= [0.1, \bar{\beta}_M, \bar{a}_y, \phi, p, \bar{v}_p, 0] \\ C_2 \text{ in yaw-channel} &= [0.1, \bar{\beta}_M, \bar{a}_y, r, \bar{v}_r, 0, 0] \end{aligned} \quad (88)$$

#### D.4 Vector of basisfunctions

The vector of basis functions  $\beta$  is composed of combinations of the elements of  $C_1$ ,  $C_2$ , and  $C_3$  by means of the kronecker product

$$\beta = \text{kron}(\text{kron}(C_1, C_2), C_3) \quad (89)$$

where,

$$\text{kron}(x, y) = [x_1 y_1 \ x_1 y_2 \ \cdots \ x_m y_n]^T \quad (90)$$

Therefore, this can be accomplished in two steps.

1. Create a fourth vector, where the subsub-index 'i' refers to the  $i^{\text{th}}$  element of that vector,

$$C_4 = \text{kron}(C_1, C_2) = [C_{1_1} C_{2_1} \ C_{1_1} C_{2_2} \ \cdots \ C_{1_2} C_{2_1} \ \cdots \ C_{1_3} C_{2_1} \ \cdots] \quad (91)$$

2. Then, using  $C_4$ , create the vector of basis functions  $\beta$  as

$$\beta = \text{kron}(C_4, C_3) = [C_{4_1} C_{3_1} \ C_{4_1} C_{3_2} \ \cdots \ C_{4_2} C_{3_1} \ \cdots \ C_{4_3} C_{3_1} \ \cdots] \quad (92)$$

The length of the vector  $\beta$  now determines the necessary length of the weight vector  $W$ . This is described next.

#### D.5 The weight vector, longitudinal channel

The weight vector is the adaptive part of the neural network. The weight vector is of the same size as  $\beta$ . It is initialized with all zeros.

$$W(t=0) = [0 \ 0 \ 0 \ \cdots \ 0]^T \quad (93)$$

Each integration step this vector is updated as

$$W_{\text{new}} = W_{\text{old}} + \Delta W \quad (94)$$

where

$$\Delta W = -\gamma \{ \zeta_\theta \beta + \lambda |\zeta_\theta| \widehat{W} \} \Delta t \quad (95)$$

with  $\gamma > 0$ , e.g.  $\gamma = 100$  and  $\lambda = 1$ ,  $\Delta t$  is the integration step, and where  $\zeta_\theta$  is a scalar filtered error term, that differs slightly for the different channels. In the pitch-channel,  $\zeta_\theta$  is constructed as

$$\zeta_\theta = P_{12}\tilde{\theta} + P_{22}\dot{\tilde{\theta}} \quad (96)$$

with

$$P_{12} = \frac{1}{2K_P} \quad (97)$$

$$P_{22} = \frac{1 + K_P}{2K_P K_D} \quad (98)$$

Where  $K_P$  and  $K_D$  are based on the linear controller,  $K_P = \omega_n^2 = 36.0$  and  $K_D = 2 * \zeta * \omega_n = 12.0$ . (Note:  $\zeta$  here refers to the damping ratio, not to be confused with the 'filtered error'  $\zeta_\theta$  in (96).)

## D.6 The neural net output

Finally, the neural network output signal at each integration step is constructed as

$$\nu_{AD} = \widehat{W}^T \beta = [w_1 \ w_2 \ w_3 \ \dots] \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n1} \end{bmatrix} = w_1\beta_1 + w_2\beta_2 + \dots \quad (99)$$

## D.7 The weight vector update for the lateral channels

Fig. (9) displays the linear controller as it is designed for *Rate Command Attitude Hold* (RCAH). The attitude retention is accomplished with an integrator. In the design of the weight update this means that the terms  $K_P$  and  $K_D$  in the longitudinal channel are now replaced by

$$\begin{aligned} K_P \text{ in pitch} &= K_I \text{ in roll/yaw} = \omega_n^2 \\ K_D \text{ in pitch} &= K_P \text{ in roll/yaw} = 2\zeta\omega_n \end{aligned} \quad (100)$$

where again  $\omega_n = 6$  and  $\zeta = 1$ , i.e.  $K_I = 36.0$  and  $K_P = 12.0$ . (Note:  $\zeta$  here refers to the damping ratio, not to be confused with the 'filtered error'  $\zeta_p$  in (102).)

The weight vector is of the same size as  $\beta$  for each channel. It is initialized with all zeros, and updated as

$$\Delta W = -\gamma_p \{ \zeta_p \beta_p + \lambda_p |\zeta_p| \widehat{W} \} \Delta t \quad (101)$$

with  $\gamma_p > 0$ , e.g.  $\gamma_p = 100$  and  $\lambda_p = 1$ ,  $\Delta t$  is the integration step, and where  $\zeta_p$  is the scalar filtered error term. In the roll-channel,  $\zeta_p$  is constructed as

$$\zeta_p = P_{12} \int_{t_0}^t \tilde{p} dt + P_{22} \tilde{p} \quad (102)$$

with

$$P_{12} = \frac{1}{2K_I} \quad (103)$$

$$P_{22} = \frac{1 + K_I}{2K_I K_P} \quad (104)$$

The yaw-channel RCAH is implemented similar to the roll-channel.

## D.8 The command generation for the yaw-channel: turn coordination

The automatic turn-coordination is based on Etkin [5], who defines a *coordinated turn* as:

1. a turn wherein the rate of change in heading is constant, and
2. the resultant of gravity and centrifugal force at the center of mass lies in the plane of symmetry.

A first order implementation is displayed in Fig 12, which is based on the rigid body equations of motion, expressed along the body-Y-axis. For the implementation of TC, only the command filter dynamics are considered, i.e. we assume perfect tracking of yaw rate command. The desired yaw rate can then be expressed as

$$r_{com} = \frac{1}{u} \{K_P(a_{Ycom} - a_Y) + w p + g s_\phi c_\theta\} \quad (105)$$

where

- $K_P$  is the gain of the linear controller in the roll-channel
  - $a_Y$  is the acceleration along the body-y-axis
  - $u, w$  are the velocities along the body-x, and z-axis, resp.
  - $g$  is the gravity constant
  - $s_\phi$  is short for  $\sin(\phi)$ , et-c.
- (106)

The dynamics due to  $\dot{v}$  are neglected as its value is usually not available from measurement. For the guidance outerloop we take the yaw rate to respond as directed by the command filter.

$$r = \frac{1}{\tau_r s + 1} r_{com} \quad (107)$$

Grouping the  $\dot{v}$  dynamics with the difference between measured and actual effects of roll-rate and gravity, as

$$\Delta_Y = \dot{v} - \left\{ \frac{\tau_r s}{\tau_r s + 1} \right\} (w p + g s_\phi c_\theta) \quad (108)$$

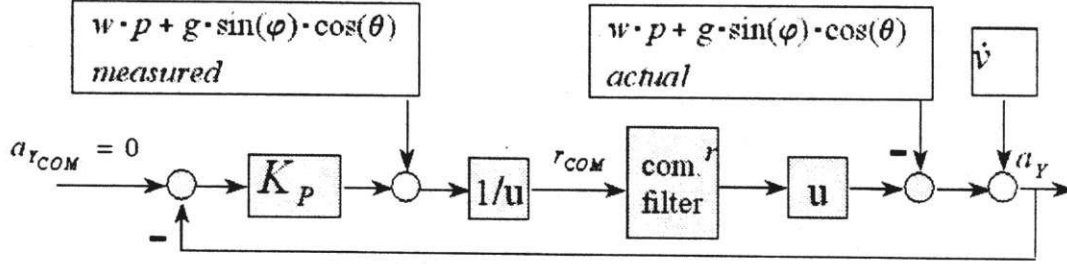


Figure 12: Turn coordination through suppression of lateral acceleration.

## E Initializing the Architecture

Let '\*' indicate a desired steady state operating point. Using a regular trim finding algorithm, obtain the desired state  $x_1^*$ , matching attitude  $\theta^*$  and matching control surface positions including  $\delta_{elev}^*$ . Furthermore, at initialization the weights of the NN are all zeros  $\hat{W} = 0$ , and therefore  $\nu_{ad} = 0$ .

The output of the full-authority SCAS system is formed as

$$\delta_{SCAS} = \delta_{30} - \delta_{neutral} + \delta \quad (109)$$

Here, the output of the model-inversion ' $\delta$ ' are actuator positions relative to a nominal and are calculated as

$$\delta = \begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \end{bmatrix} = \hat{B}^{-1} \{ \hat{\omega}_D - \hat{A}_1 x_1 - \hat{A}_2 \omega \} \quad (110)$$

The situation for trim in the longitudinal channel is indicated in Fig.(14).

Consider now the longitudinal channel

$$\delta_{SCAS}^* = \delta_{30} - \delta_{neutral} + \delta_{lon}^* \quad (111)$$

Given the state  $x_1^*$  and matching  $\theta^*$ , and  $\delta_{elev}^*$  obtain from Eqn.(110)

$$\delta^* = \begin{bmatrix} \delta_{lat}^* \\ \delta_{lon}^* \\ \delta_{ped}^* \end{bmatrix} = -\hat{B}^{-1} \{ \hat{A}_1 (x_1^* - x_1^{30}) \} \quad (112)$$



## F Stick Trim Position

Let  $^{**}$  indicate a desired steady state operating point. Using the original trim finding algorithm, obtain the desired state  $x_1^*$ , matching attitude  $\theta^*$  and matching control surface positions including  $\delta_{elev}^*$ , and the matching stick position  $U_T^*$ .

**Remark 6.1** The matching pair  $\delta_{elev}^*$  and  $U_T^*$  may be based on any type of FCS, including the original (mechanical) linkage. This is possible, since to the full authority augmentation this is considered a rigging offset. The full authority augmentation interprets all inputs as deflections from these trim values.

In Fig (14) the absolute stick position is represented by  $U_T$ . Therefore, let  $U_P$  represent the pilot deflection relative to the trim position. At trim, i.e.  $t = 0$  s we have

$$U_T(0) = U_T^* \quad (114)$$

$$U_P(0) = U_P^* = 0 \quad (115)$$

The situation for trim in the longitudinal channel is indicated in Fig.(14).

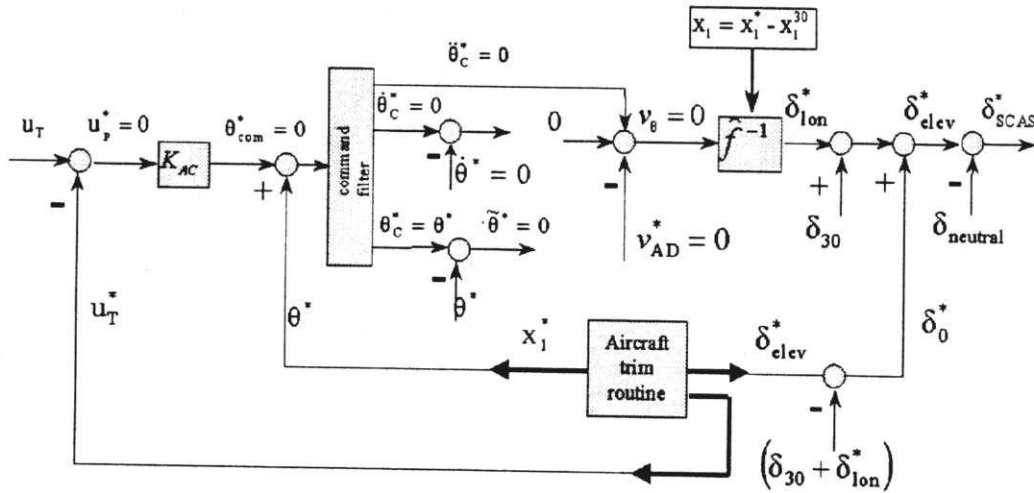


Figure 14: NN augmented model inversion architecture, at ACAH trim in the longitudinal channel.