# Pseudo-Control Hedging: A New Method For Adaptive Control

**Article** · September 2001

Source: CiteSeer



Eric N. Johnson
Pennsylvania State University


Anthony J. Calise
Georgia Institute of Technology

# PSEUDO-CONTROL HEDGING:
# A NEW METHOD FOR ADAPTIVE CONTROL

Eric N. Johnson and Anthony J. Calise

*School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332*

Advances in Navigation Guidance and Control Technology Workshop

November 1-2, 2000

Redstone Arsenal, Alabama

## Abstract

In the application of adaptive flight control, significant issues arise due to vehicle input characteristics such as actuator position limits, actuator position rate limits, and linear input dynamics. The concept of modifying a reference model to prevent an adaptation law from "seeing" and adapting-to these system characteristics is introduced. A specific adaptive control method based on this concept, termed Pseudo-Control Hedging, is introduced that accomplishes this for any Model Reference Adaptive Controller that includes approximate feedback linearization. This method enables continued adaptation while the plant input is saturated. Acceptance and flight certification of an online Neural Network adaptive control law for the X-33 Reusable Launch Vehicle technology demonstrator is discussed as motivation for this work. Simulation results applying the method to the X-33 are described.

## Nomenclature

| | |
|---|---|
| $A,B$ | Error dynamics matrices |
| $e$ | Model tracking error |
| $f(\cdot)$ | System dynamics |
| $I$ | Identity matrix |
| $K$ | Diagonal gain matrix |
| $n$ | Number of states or neural network nodes |
| $P,Q$ | Positive definite matrices, Lyapunov equation |
| $q$ | Quaternion |
| $r$ | Modified model tracking error |
| $W,V$ | Neural network input, output weights |
| $x$ | System states |

| | |
|---|---|
| $Z$ | Neural network weights |
| $\delta$ | Plant inputs |
| $\Delta(\cdot)$ | Model error |
| $\Gamma$ | Diagonal matrix containing neural network learning rates |
| $\varepsilon$ | Model error reconstruction error |
| $\nu$ | Pseudo-control |
| $\omega$ | Angular rate |
| $\sigma(\cdot)$ | Sigmoidal function |
| $\xi(\cdot)$ | Quaternion error angles function |

**Subscripts**

| | |
|---|---|
| $ad$ | Adaptive control |
| $c, g$ | Commanded/guidance (outer loop) |
| $cmd$ | Actuator control (input to actuator) |
| $h$ | Hedge |
| $pd, p, d$ | Proportion-derivative linear control |
| $r$ | Robustifying |
| $rm$ | Reference model |

# Introduction

The X-33 is a sub-orbital aerospace vehicle intended to demonstrate technologies necessary for future Reusable Launch Vehicles (RLVs), shown in Fig 1. It includes several key features of Lockheed Martin's proposed VentureStar RLV, including the linear aerospike engine, vertical take-off, horizontal landing, heat dissipation system, and aerodynamic configuration[1].

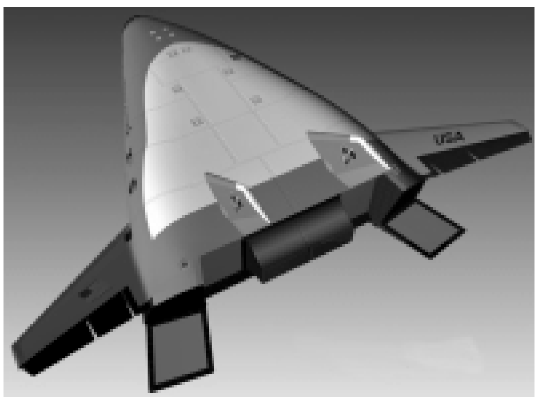

*Figure 1 – X-33 Reusable Launch Vehicle (RLV) technology demonstrator*

To achieve the cost benefits of an operational RLV, the amount of analysis and testing required per mission needs to be reduced over that performed for the partially re-usable Space Shuttle. A goal for future RLV flight control is to design and test the flight control system to operate within a prescribed flight envelope and loading margin, and requiring only payload/fuel parameters and "route" to be specified for a given mission. It has been estimated that this level of improvement would save three man-years of labor per RLV mission[2].

Launch vehicle flight control is conventionally linearized about a series of operating points and gain-scheduled. These operating points normally include a range of either Mach number, velocity, altitude, time, or some other parameter used to determine vehicle progress with respect to a nominal trajectory. Separate sets of gain tables are often included for abort cases and failure cases, as is the case with the current X-33 design[3].

Gain scheduling is a very powerful and successful method, but has a distinct drawback for the RLV: the number of required gains to be designed and scheduled becomes very large. If one also imposes the design constraint that these gains must allow for a range of possible missions, payloads, and anticipated failure modes, then the number of required gains can become prohibitive.

In recent years, several theoretical developments have given rise to the use of Neural Networks (NN) that learn/adapt online for nonlinear systems[4]. The use of NN adaptive flight control has been demonstrated in piloted simulation and flight test on the X-36 aircraft[5]. These tests included failures to the flight control system that necessitated adaptation. The fact that this architecture enables adaptation to an arbitrary nonlinear non-affine plant in real-time makes it an attractive candidate to replace RLV gain tables. This approach has the additional benefit that recovery from a class of vehicle component failures can be shown.

One important characteristic of the work described here includes a trivial choice for the feedback linearization design. In the X-33 application described, an inverse transformation is used based only on the estimated inertia matrix and existing, NASA Marshall Space Flight Center developed, control allocation systems[1]. Therefore, aerodynamic moments are neglected in realizing this approximate inversion – except those due to the controls themselves. The neglected effects are large, and vary dramatically during the flight, greatly affecting stability and trim. While this represents the extreme to which one might choose to rely on adaptation, it may be desirable from the perspective of safety and utility. The resulting controller could then be used for the greatest variety of unanticipated failures, abort trajectories, and configuration changes, since it is less reliant on a-priori assumptions.

A method for addressing system input characteristics is also introduced in this paper, which has the potential to allow this class of adaptive control systems to operate properly in the presence of a wide class of plant input characteristics: actuator position limits, actuator rate limits, time delay, and input quantization. The method is termed Pseudo-Control Hedging (PCH), and is motivated further in the following section. A specific design for NN adaptive flight control is also given, and results applying it to the X-33 are described.

## Motivation for Pseudo-Control Hedging

### Saturation

Input saturation and input rate saturation present a significant problem for adaptive control, perhaps even more so than for non-adaptive control. Saturation violates the assumption of affinity in control, which is common in the literature. It also violates the assumption that the sign of the effect of the control is known/non-zero, since the effect of additional control input is zero once saturation is entered. Saturation also implies controllability and invertability issues during saturation. This temporary loss of control effectiveness violates necessary conditions for feedback linearization.

Avoiding saturation is important for systems where the open-loop plant is highly unstable, but introduces conservatism otherwise. One approach is to avoid saturation by command adjustment, which has been demonstrated in an adaptive control setting[6]. There are also results that bound the feedback control by some form of a squashing function where adaptation is effectively slowed as saturation is approached[7].

Current approaches to input saturation used specifically for NN adaptive flight control include reducing adaptation rate, as suggested by the theory, to the point of stopping completely once the control is saturated. *However, adaptation during saturation is important.* For X-33, one or more control surfaces are often saturated for a significant period of time during normal flight. This is particularly true for the aerodynamic surfaces during portions of entry.

For adaptive control in general, modified adaptive laws have been proposed for linear plants that can be designed to exhibit the property that the adaptive law does not "see" an input saturation characteristic[8]. Such methods can be fit into the general category of error signal modification in Model Reference Adaptive Control (MRAC)[9]. PCH is related, in that it also relies on modifying the error signal. However, in PCH the modification is to the reference model (rather than the error signal directly or the adaptive law), not limited to saturation, linear plants, or linear reference models.

<u>Selected Linear Input Dynamics</u>

NN adaptive flight control often requires that input (i.e. actuator) dynamics are known or negligible. Improved robustness to unknown input dynamics utilizing more general adaptive control techniques has been shown[10].

However, as with saturation, it is often not advisable to cancel certain linear input dynamics. For one, actuators may be driven excessively if their associated linear input dynamics are inverted. A second example is a notch filter used to prevent exciting a structural mode. Unfortunately, to regard these input dynamics as unknown *or* to include them in the dynamic inversion element would result in attempted cancellation. Ideally, the control system designer would like to prevent the adaptive element from attempting to cancel selected input dynamics.

<u>Quantized Control</u>

When the input is highly quantized, or simply "bang-zero-bang" in the most extreme case, adaptive control theory is challenged in many of the same ways it was challenged in the input saturation case. For re-entry, the space shuttle orbiter and the X-33 use a combination of continuous aerodynamic controls and "bang-zero-bang" Reaction Control System (RCS) thrusters[3]. Linear controllers can be designed to utilize a combination of continuous and quantized control, or even for exclusively quantized control. However, nonlinear adaptive control methods are challenged by quantized control. Affinity and knowledge of the sign of control (as a partial derivative) are both violated as in the case of saturation.

Flight Certification of Adaptive Controllers

Any flight certification authority should address many issues, but two that are particularly troublesome for adaptive controllers are:

1. *Is it possible for the adaptive controller to cause harm to the vehicle?* This is a difficult issue associated with adaptive controllers, since it is inherently difficult to show that the controller will rarely "learn incorrectly" under reasonable assumptions. The assumptions addressed in this work relate to system input characteristics.
2. *Can the adaptive element recover from a failure in adaptation?* If, for any reason, the adaptive element has learned incorrectly to an extreme level, the adaptive controller should be able to recover. An extreme level of incorrect learning might be characterized by commanding saturated control deflections when only small deflections are appropriate. This type of recovery is enabled by correct adaptation during input saturation.

Adaptation While Not in Direct Control

The flight control system designer would like to allow adaptive controllers to adapt while not in complete control of the plant. This has several uses:

- Pre-adaptation of adaptive element before handing over control to an adaptive controller has at least two uses. First, this has the potential to reduce transients at hand-over to an adaptive controller. Second, flight testing could include getting useful results from "running" the adaptive flight control system before actually giving it control of the aircraft as an incremental test.
- Partial control would allow adaptive element to maintain control over "part" (e.g. an axis, or a subset of available actuators) of the system dynamics while another system controls the remainder of the system.

# Pseudo-Control Hedging

In this section, a modification to previous adaptive control work is introduced that addresses the issues raised:

- Protecting the adaptive element from problems (i.e. internal stability) associated with input saturation
- Protecting the adaptive element from problems associated with input rate saturation
- Protecting the adaptive element from problems associated with quantized control
- Preventing the attempted adaptation to selected known input dynamics
- Allowing recovery from temporarily faulty adaptation
- Allowing "correct" adaptation while in partial or no control of the plant

## Description of Pseudo-Control Hedging

The method introduced here is termed Pseudo-Control Hedging (PCH). The purpose of the method is to prevent the adaptive element of an adaptive control system from trying to adapt to a class of system input characteristics (characteristics of plant or of the controller). To do this, the adaptive law is prevented from "seeing" the system characteristic.

A plain-language conceptual description of the method is: *The reference model is moved backwards (hedged) by an estimate of the amount the plant did not move due to system characteristics the control designer does not want the adaptive control element to 'know' about.* To formalize the language in the context of an adaptive control law involving dynamic inversion, "movement" should be replaced by some system signal. Preventing the adaptive element from 'knowing' about a system characteristic means to prevent that adaptive element from seeing the system characteristic as model tracking error.

Fig 2 is an illustration of conventional MRAC[11] with the addition of PCH compensation. The PCH compensator is designed to modify the response of the reference model in such a way as to prevent the adaptation law from seeing the effect of certain controller or plant system characteristics.

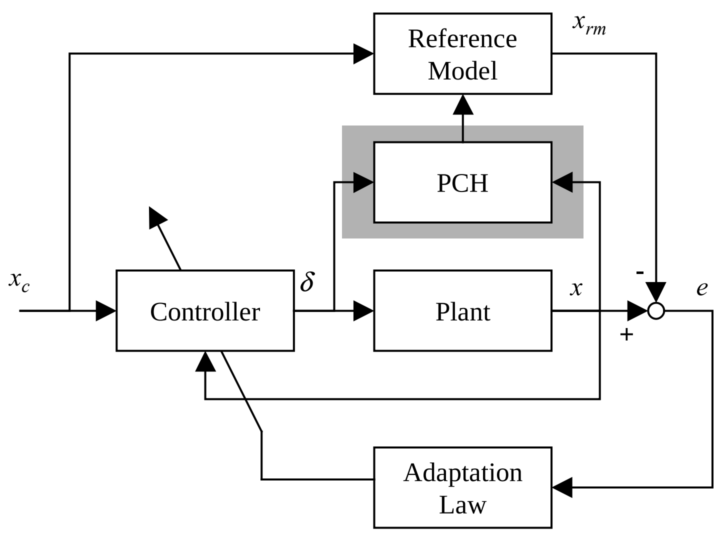


*Figure 2 – Model Reference Adaptive Control (MRAC) with Pseudo-Control Hedging (PCH) compensation*

The more specific case of PCH applied to a NN adaptive control architecture is illustrated in Fig 3. A difference is that the reference model appears within the feedforward path. The adaptive element is any compensator attempting to correct for errors in the approximate dynamic inversion. This could be as simple as an integrator or something more powerful such as an on-line NN.

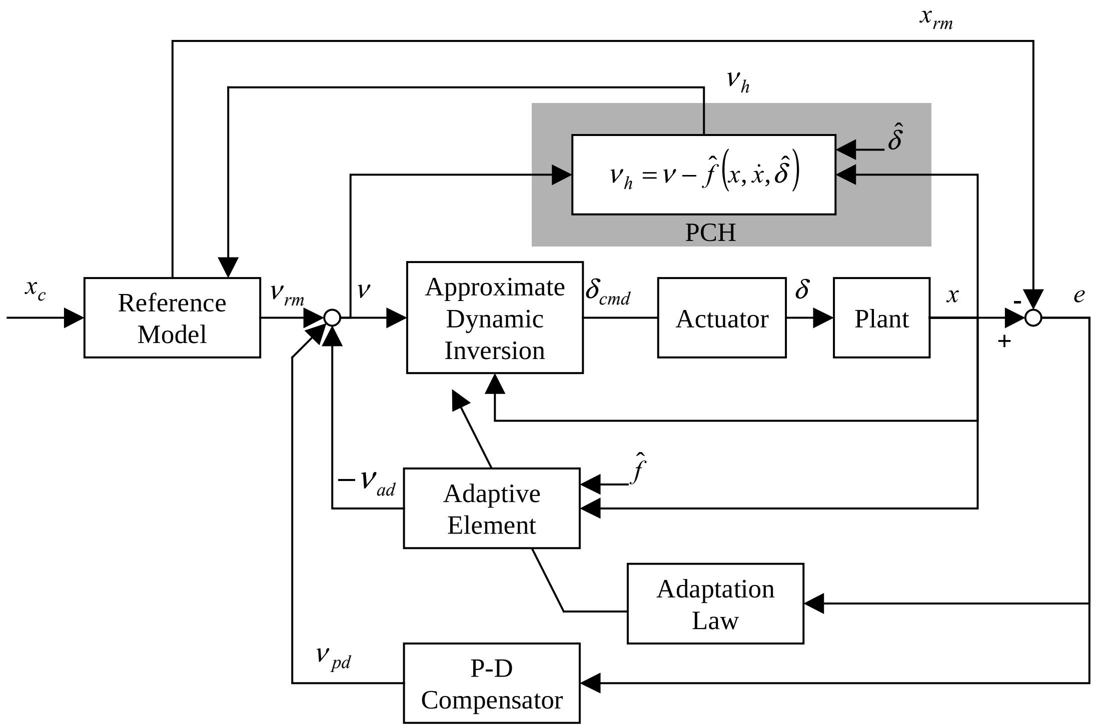


*Figure 3 – MRAC including an approximate dynamic inversion with PCH; The PCH component utilizes an estimate of actuator position based on actuator command and possibly vehicle state*

The design of the PCH compensator for the controller architecture illustrated in Fig 3 is now described. For simplicity, consider the case of full model inversion, in which the plant dynamics are taken to be of the form

$$\ddot{x} = f(x, \dot{x}, \delta) \tag{1}$$

where $x, \dot{x}, \delta \in \Re^n$. An approximate dynamic inversion element is developed to determine actuator commands of the form

$$\delta_{cmd} = \hat{f}^{-1}(x, \dot{x}, \nu) \tag{2}$$

where $\nu$ is the pseudo-control signal, and represents a desired $\dot{x}$ that is expected to be approximately achieved by $\delta_{cmd}$. That is, this dynamic inversion element is designed without consideration of the actuator model. This command ($\delta_{cmd}$) will not equal actual control ($\delta$) due to the actuator. To get a pseudo-control hedge ($\nu_h$), an estimated actuator position ($\hat{\delta}$) is determined based on a model or a measurement. In cases where the actuator position is measured, it is regarded as known ($\hat{\delta} = \delta$). This estimate is then used to get the difference between commanded pseudo-control and the achieved pseudo-control

$$\nu_h = \hat{f}(x, \dot{x}, \delta_{cmd}) - \hat{f}(x, \dot{x}, \hat{\delta}) \tag{3}$$

$$= \nu - \hat{f}(x, \dot{x}, \hat{\delta}) \tag{4}$$

With the addition of PCH, the reference model shown in Fig 8 has a new input, $\nu_h$. As introduced earlier, the pseudo-control hedge is to be subtracted from the reference model state update. For example, if the (stable) reference model dynamics *without PCH* was of the form

$$\ddot{x}_{rm} = f_{rm}(x_{rm}, \dot{x}_{rm}, x_c), \tag{5}$$

where $x_c$ is the command signal, then the reference model dynamics *with PCH* becomes

$$\ddot{x}_{rm} = f_{rm}(x_{rm}, \dot{x}_{rm}, x_c) - \nu_h. \tag{6}$$

The instantaneous pseudo-control output of the reference model (if used) is not changed by the use of PCH, and remains

$$\nu_{rm} = f_{rm}(x_{rm}, \dot{x}_{rm}, x_c). \tag{7}$$

In other words, the PCH signal $\nu_h$ affects reference model output $\nu_{rm}$ only through changes in reference model state.

The following sections go deeper into the theory and discuss the limitations of the method. There are two fundamental changes associated with PCH that effect existing boundedness theorems for the NN architecture[12,13]. Firstly, there is a change to the model tracking error dynamics ($e$), which is the basis for adaptation laws in this earlier work, discussed below. Secondly, the reference model cannot be presumed to remain stable in general, also discussed below.

Model Tracking Error Dynamics

The complete pseudo-control signal for the system introduced in Fig 8 and described in more detail in the previous section is

$$\nu = \nu_{rm} + \nu_{pd} - \nu_{ad} + \nu_r \tag{8}$$

where the reference model signal $\nu_{rm}$ is given by Eqn (7), the PD compensator output $\nu_{pd}$ is a Proportional-Derivative (PD) compensator acting on tracking error

$$\nu_{pd} = \begin{bmatrix} K_d & K_p \end{bmatrix} e, \tag{9}$$

where $K_d$ and $K_p$ are diagonal matrices containing desired second-order linear error dynamics and model tracking error is expressed as

$$e = \begin{bmatrix} \dot{x}_{rm} - \dot{x} \\ x_{rm} - x \end{bmatrix}. \tag{10}$$

The adaptation signal $(-\nu_{ad} + \nu_r)$ is the output of the adaptive element, where the so-called robustifying term $\nu_r$ is dropped in the remainder of this section for clarity. The model tracking error dynamics are now found by differentiating Eqn (10) and utilizing the previous Eqns:

$$\dot{e} = Ae + B\left[\nu_{ad}\left(x, \dot{x}, \hat{\delta}\right) - f(x, \dot{x}, \delta) + \hat{f}\left(x, \dot{x}, \hat{\delta}\right)\right] \tag{11}$$

where

$$A = \begin{bmatrix} -K_d & -K_p \\ I & 0 \end{bmatrix} \tag{12}$$

$$B = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{13}$$

Model error to be compensated for by $\nu_{ad}$ is defined as

$$\Delta\left(x,\dot{x},\delta,\hat{\delta}\right) = f\left(x,\dot{x},\delta\right) - \hat{f}\left(x,\dot{x},\hat{\delta}\right) \tag{14}$$

If one assumes that $\delta$ is exactly known ($\hat{\delta} = \delta$), it follows from Eqn (11) that

$$\dot{e} = Ae + B\left[\nu_{ad}\left(x,\dot{x},\hat{\delta}\right) - \Delta\left(x,\dot{x},\hat{\delta}\right)\right] \tag{15}$$

Eqn (15) is of the same form as the model tracking error dynamics seen in previous work. As a result, the boundedness of a NN adaptation law given by earlier results can be used with some modification.

**Remark 1:** If instead one makes the less restrictive assumption that the realization of $\hat{\delta}$ does not produce any additional dynamics, i.e. contains no internal states, then

$$\dot{e} = Ae + B\left[\nu_{ad}\left(x,\dot{x},\hat{\delta}\right) - \Delta'\left(x,\dot{x},\hat{\delta}\right)\right] \tag{16}$$

where

$$\Delta'\left(x,\dot{x},\hat{\delta}\right) = \Delta\left(x,\dot{x},\delta,\hat{\delta}\right) \tag{17}$$

appears as model error to the adaptive law, and previous results will apply with the same qualifications.

**Remark 2:** When the realization of $\hat{\delta}$ does contain additional dynamics, these dynamics will appear as unmodeled input dynamics to the adaptive law. Previous results that improve robustness to unmodeled input dynamics[10] can be applied to address a residual model error ($\varepsilon'$), which comes about when Eqns (11) and (17) are applied to put tracking error dynamics in the following form:

$$\dot{e} = Ae + B\left[\nu_{ad}\left(x,\dot{x},\hat{\delta}\right) - \Delta'\left(x,\dot{x},\hat{\delta}\right) + \varepsilon'(t)\right] \tag{18}$$

<u>Stability of the Reference Model</u>

A significant difference to previous MRAC work due to PCH is that the reference model is not necessarily stable. This occurs because the assumptions made up to this point allow the adaptive element to continue to function when the actual control signal has been replaced by *any* arbitrary signal. This completely arbitrary signal does not necessarily stabilize the plant.

However, stability and tracking are still of interest for closed-loop control. System characteristics to be removed from the adaptation must be limited to items that are a function of the commanded control, such as saturation, quantized control, linear input dynamics, and latency. This class of system characteristics will be referred to in this section as an actuator model. In general, this actuator model could also be a function of plant state.

System response for ( $\hat{\delta} = \delta$ ) is now

$$\ddot{x} = \hat{f}\left(x, \dot{x}, \hat{\delta}\right) + \Delta\left(x, \dot{x}, \hat{\delta}\right) \tag{19}$$

When the actuator is ideal, one obtains

$$\ddot{x} = \hat{f}\left(x, \dot{x}, \delta_{cmd}\right) + \Delta\left(x, \dot{x}, \hat{\delta}\right) \tag{20}$$

**Remark 3:** When the actuator is "ideal" and the actual position and the commanded position are equal, the addition of PCH has no effect on any system signal.

**Remark 4:** When the actuator position and command differ, the adaptation occurs as though the command had corresponded to the actual. Also, the system response is as close to the command as was permitted by the actuator model.

Previous results suggest that interactions between the adaptive element and the actuator model are prevented by PCH. However, there can clearly be interactions between the actuator model and the reference model plus linear compensation (PD control) that can cause a detriment to stability and tracking performance. It is through selection of desired dynamics that the control system designer should address the actuator, utilizing any of a large number of methods from non-adaptive control, independent of the adaptive law. This is the desired condition, because the limitations of the actuators are normally a primary driver in selection of desired system dynamics.

## A Pseudo-Control Hedging Architecture

In this section, PCH is specialized to the case of NN adaptive control.

Neural Network

Single Hidden Layer (SHL) Perceptron NNs are universal approximators[14] in that they can approximate any smooth nonlinear function to within arbitrary accuracy, given a sufficient number of hidden layer neurons and input information. Fig 4 shows the structure of a SHL NN.

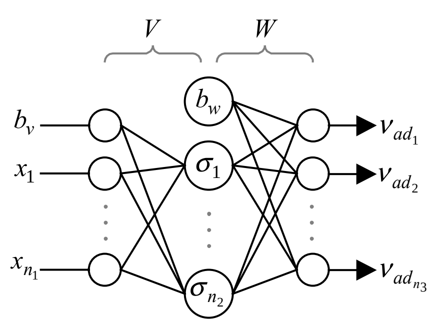


*Figure 4 – The Single Hidden Layer (SHL) Perceptron Neural Network*

The following definitions are convenient for further analysis[15]. The input-output map can be expressed as

$$v_{ad_k} = b_w \theta_{w,k} + \sum_{j=1}^{n_2} w_{j,k} \sigma_j \tag{21}$$

where $k = 1, \cdots, n_3$ and

$$\sigma_j = \sigma\left( b_v \theta_{v,j} + \sum_{i=1}^{n_1} v_{i,j} x_i \right) \tag{22}$$

Here $n_1$, $n_2$, and $n_3$ are the number of input nodes, hidden layer nodes, and outputs respectively. The scalar function $\sigma$ is a sigmoidal activation function that represents the 'firing' characteristics of the neuron, e.g.

$$\sigma(z) = \frac{1}{1 + e^{-az}} \tag{23}$$

The factor $a$ is known as the activation potential, and is normally a distinct value for each neuron. For convenience define the two weight matrices

$$V = \begin{bmatrix} \theta_{v,1} & \cdots & \theta_{v,n_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_1,1} & \cdots & v_{n_1,n_2} \end{bmatrix} \tag{24}$$

$$W = \begin{bmatrix} \theta_{w,1} & \cdots & \theta_{w,n_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{bmatrix} \tag{25}$$

and define a new sigmoid vector

$$\bar{\sigma}(z) = \begin{bmatrix} b_w & \sigma(z_1) & \sigma(z_2) & \cdots & \sigma(z_{n_1}) \end{bmatrix}^T \tag{26}$$

where $b_w \geq 0$ allows for the threshold $\theta_w$ to be included in the weight matrix $W$. Define

$$\bar{x} = \begin{bmatrix} b_v & x_1 & x_2 & \cdots & x_{n_1} \end{bmatrix}^T \tag{27}$$

$b_v \geq 0$ is an input bias that allows for the threshold $\theta_v$ to be included in the weight matrix $V$. With the above definitions, the input-output map of the SHL NN in the controller architecture can be written in matrix form as

$$v_{ad} = W^T \bar{\sigma}\left(V^T \bar{x}\right) \tag{28}$$

<u>Guaranteed Boundedness</u>

Consider a SHL perceptron approximation of the nonlinear function $\Delta$, introduced in Eqn (15), over a domain $D$ of $\bar{x}$. There exists a set of ideal weights $\{W^*, V^*\}$ that bring the output of the

NN to with an $\varepsilon$-neighborhood of the error $\Delta(x, \dot{x}, \delta) = \Delta(\bar{x})$. This $\varepsilon$-neighborhood is bounded by $\bar{\varepsilon}$, defined by

$$\bar{\varepsilon} = \sup_{\bar{x}} \left\| W^T \bar{\sigma}\left(V^T \bar{x}\right) - \Delta(\bar{x}) \right\| \tag{29}$$

The universal approximation theorem implies that $\bar{\varepsilon}$ can be made arbitrarily small given enough hidden layer neurons. The matrices $W^*$ and $V^*$ can be defined as the values that minimize $\bar{\varepsilon}$. These values are not necessarily unique.

The NN outputs are represented by $v_{ad}$ where $W$ and $V$ are estimates of the ideal weights. Define

$$Z = \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \tag{30}$$

and let $\|\cdot\|$ imply the Frobenius norm.

**Assumption 1:** *The norm of the ideal NN weights is bounded by a known positive value*

$$\left\| Z^* \right\| \leq \bar{Z} \tag{31}$$

Define the derivative of the sigmoids as

$$\bar{\sigma}_z(z) = \frac{\partial \bar{\sigma}(z)}{\partial z} = \begin{bmatrix} 0 & \cdots & 0 \\ \frac{\partial \sigma(z_1)}{\partial z_1} & & 0 \\ & \ddots & \\ 0 & & \frac{\partial \sigma(z_{n_2})}{\partial z_{n_2}} \end{bmatrix} \tag{32}$$

From the tracking error dynamics described previously, Eqn (15), define the vector

$$r = \left(e^T P B\right)^T \tag{33}$$

Where $P \in \Re^{2n \times 2n}$ is the positive definite solution to the Lyapunov Equation $A^T P + PA + Q = 0$. Where a reasonable positive definite choice for Q is

$$Q = \begin{bmatrix} K_d K_p & 0 \\ 0 & K_d K_p^2 \end{bmatrix} \frac{1}{\frac{1}{4} n_2 + b_w^2} \tag{34}$$

The robustifying signal is chosen to be

$$v_r = -\left[K_{r0} + K_{r1}\left(\|Z\| + \bar{Z}\right)\right] r \tag{35}$$

with $K_{r0}, K_{r1} > 0, \in \Re^{n \times n}$.

The following theorem guarantees uniform ultimate boundedness of tracking errors, weights, and plant states. With a non-ideal actuator, one must also apply Assumption 2.

**Assumption 2:** *Reference model signals remain bounded.*

**Theorem 1:** *Consider the feedback linearizable system given by Eqn (1), where* $\mathrm{sign}\left(\frac{\partial f_i}{\partial \delta_j}\right)$ *known* $\forall i, j = 1,2,3$. *The augmented feedback control law given by Eqn (2), with* $\nu$ *defined by Eqns (4), (5), (7), (8), (9), (28), and (35), where* $\dot{W}$ *and* $\dot{V}$ *satisfy*

$$\dot{W} = -\Gamma_w \left\{ \left( \bar{\sigma} - \bar{\sigma}_z V^T \bar{x} \right) r^T + \lambda \| r \| W \right\} \tag{36}$$

$$\dot{V} = -\left\{ \bar{x} \left( r^T W^T \bar{\sigma}_z \right) + \lambda \| r \| V \right\} \Gamma_v \tag{37}$$

*with* $\Gamma_w, \Gamma_v > 0$ *and* $\lambda > 0$, *guarantees that all signals in the closed loop system remain bounded.*

Proof: see Appendix; see also Ref 16 and Ref 10.

# Quaternion-Based NN Adaptive Flight Control Architecture

A quaternion-based adaptive flight control architecture is illustrated in Fig 5. The flight control system determines a desired angular acceleration, or pseudo-control, which forms the input to a nominal dynamic inversion. The nominal dynamic inversion converts these desired angular accelerations into the required control torque commands and then actuator commands (utilizing a control allocator).

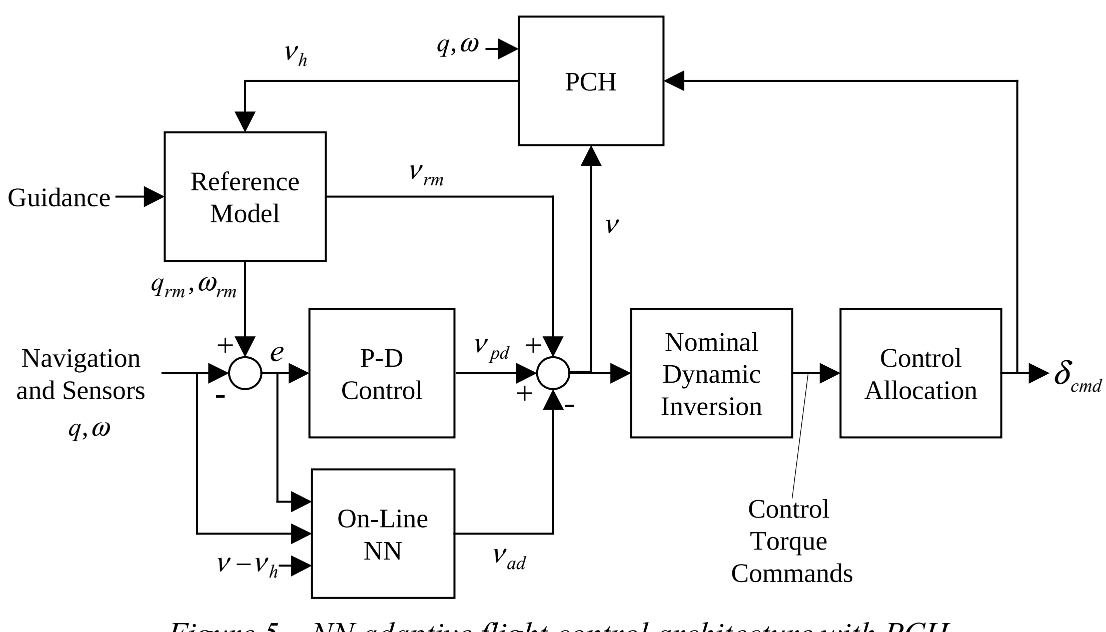


*Figure 5 – NN adaptive flight control architecture with PCH*

Desired Dynamics

The reference model is illustrated in Fig 6. A guidance attitude command is provided as a quaternion stored as a vector, $q_g \in \Re^4$, with an associated angular rate command vector, $\omega_g \in \Re^3$. Nominally, this reference model gives a second order response to changes in the guidance command and with quaternion error angles normally calculated given two quaternions with the function

$$\xi(q,r) = -2\,\mathrm{sign}(q_1 r_1 + q_2 r_2 + q_3 r_3 + q_4 r_4)\times \begin{bmatrix} -q_1 r_2 + q_2 r_1 - q_3 r_4 + q_4 r_3 \\ -q_1 r_3 + q_2 r_4 + q_3 r_1 - q_4 r_2 \\ -q_1 r_4 - q_2 r_3 + q_3 r_2 + q_4 r_1 \end{bmatrix} \tag{38}$$

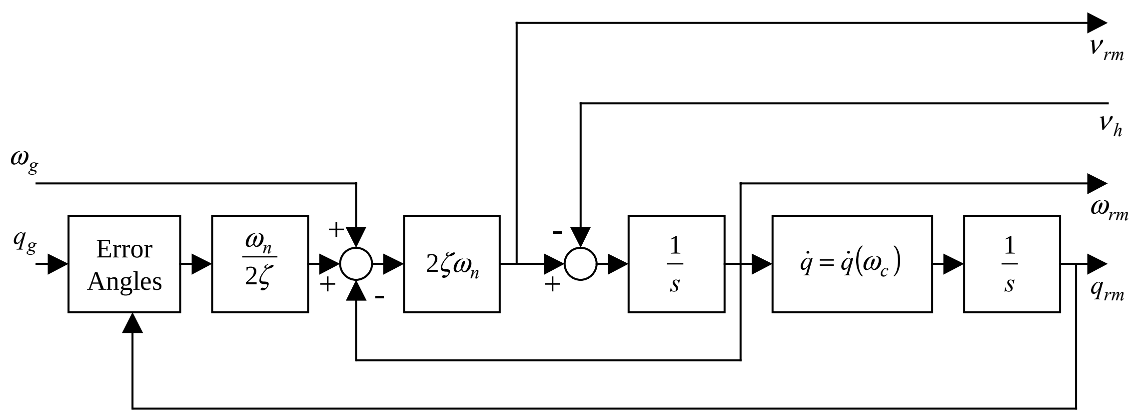

*Figure 6 – Quaternion-based attitude reference model with PCH*

for which $\nu_{rm}$ has the analytic form

$$\nu_{rm} = K_d(\omega_g - \omega_{rm}) + K_p\xi(q_g, q_{rm}) \tag{39}$$

and the reference model dynamics are of the form

$$\dot{\omega}_{rm} = K_d(\omega_g - \omega_{rm}) + K_p\xi(q_g, q_{rm}) - \nu_h \tag{40}$$

PD gains are applied to the tracking error in a manner similar to that used for reference model dynamics

$$\nu_{pd} = M\{K_d(\omega_{rm} - \omega) + K_p\xi(q_{rm}, q)\} \tag{41}$$

where $M$ is chosen to be identity for X-33. It should given larger values when the reference model dynamics are intended to be the dominant, lower frequency, response.

Nominal Inversion and Pseudo Control

The pseudo control is selected as

$$\nu = \nu_{rm} + \nu_{pd} + \nu_r - \nu_{ad} \tag{42}$$

Here, $\nu_{ad}$ is the NN output, and $\nu_r$ is the robustifying signal defined previously.

Vehicle angular acceleration can be modeled by

$$\dot{\omega} = f(x, \omega, \delta) \tag{43}$$

where $\omega \in \Re^3$ represents the angular rate of the vehicle, $\delta \in \Re^m$ represents the control effectors, and $x$ represents other vehicle states that angular acceleration depends upon, with $m > 3$. The approximately feedback linearizing control law is

$$\delta = \hat{f}^{-1}(x, \omega, \nu) \tag{44}$$

# X-33 Results

This flight control architecture has been tested in the Marshall Aerospace Vehicle Representation in C (MAVERIC)[17], which is the primary guidance and control simulation tool for X-33. This work has included flight control design from launch to the beginning of the Terminal Area Energy Management (TAEM) phase. Missions include vertical launch and peak Mach numbers of approximately 8, altitudes of 180,000 feet, and dynamic pressures of 500 KEAS. During ascent, vehicle mass drops by approximately a factor of 3, and vehicle inertia by a factor of 2.

## Ascent Flight Control

The flight control design illustrated in Fig 5 was utilized without modification for ascent flight control. Nominal inversion consisted of multiplying desired angular acceleration by an estimate of vehicle inertia, and utilizing a fixed-gain version of the baseline ascent control allocation system[1]. NN inputs were angle-of-attack, side-slip angle, bank angle, sensed vehicle angular rate, and estimated pseudo-control ($\hat{\nu} = \nu - \nu_h$). Four middle layer neurons were used; learning rates on $W$ were unity for all axes and learning rates for $V$ were 20 for all inputs. $K_p$ and $K_d$ were chosen based on a natural frequency of 1.0, 1.5, and 1.5 ${}^{rad}\!/_{\sec^2}$ for the roll, pitch, and yaw axes respectively and a damping ratio of 0.7.

The aerodynamic surface actuator position and rate limits are included in the PCH, as is the position and rate limits of the main engine thrust vectoring. PCH also had knowledge of the axis priority logic within the control allocation system.

### Nominal Ascent Results

Results of this NN adaptive flight control architecture running on the MAVERIC version 7.5.1 X-33 simulator are now discussed. In all cases, the mission designation is 10a1. Attitude error angles for a nominal ascent phase are shown in Fig 7. This error is between the guidance command and the vehicle state. Ascent phase ends at Main Engine Cut-Off (MECO) at 212 seconds for this flight.

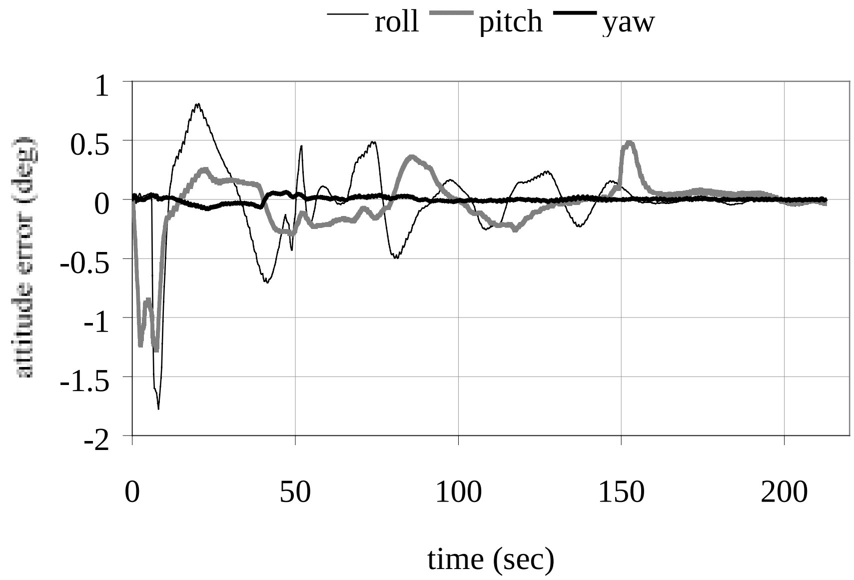


*Figure 7 – Attitude Error Angles for Nominal X-33 Ascent Phase*

<u>PPO Failure Results</u>

Attitude error angles for ascent phase with a Power-Pack Out (PPO) occurring at 60 seconds is shown in Fig 8. In this failure mode, a turbopump failure reduced total thrust and available control power. The adaptive flight control system utilized no knowledge of the failure, and was forced to adapt. However, the control allocation logic did utilize knowledge of the failure for this case.

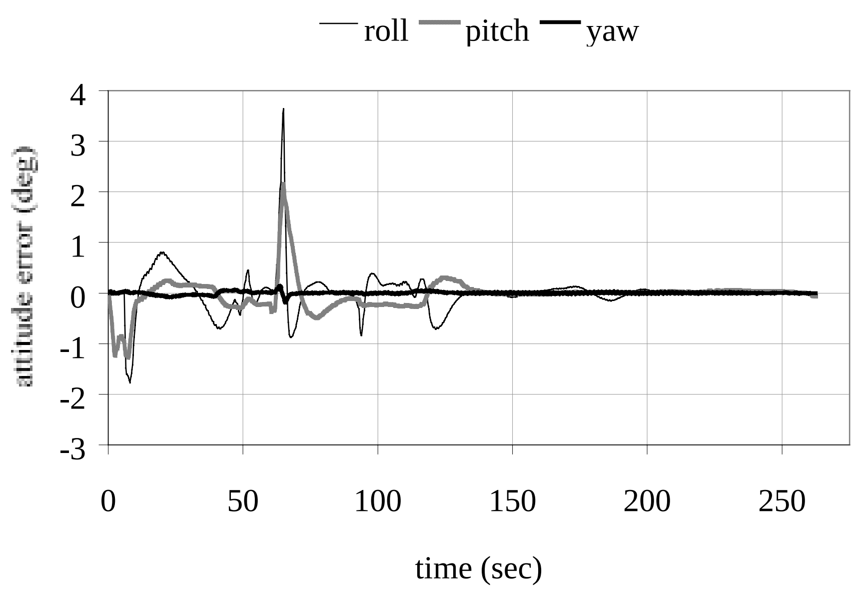


*Figure 8 – Attitude error angles for X-33 ascent, Power-Pack Out at 60 seconds*

Multiple Actuator Hard-overs Results

In order to illustrate some of the properties of this implementation of PCH, a failure case is chosen where it is temporarily not possible to maintain reasonable model tracking. Near maximum dynamic pressure, half of the aerodynamic surfaces go hard-over in a pro-left turn (right rudder full trailing edge right, right elevons full trailing edge down, right body flap full trailing edge down). This is severe enough that insufficient control power remains to maintain the desired command until dynamic pressure drops. The vehicle does three rolls to the left before the dynamic pressure drops enough to enable vehicle control. Attitude error angles are shown in Fig 9. The control system had no knowledge of failures; this includes the control allocation and the PCH computation components.

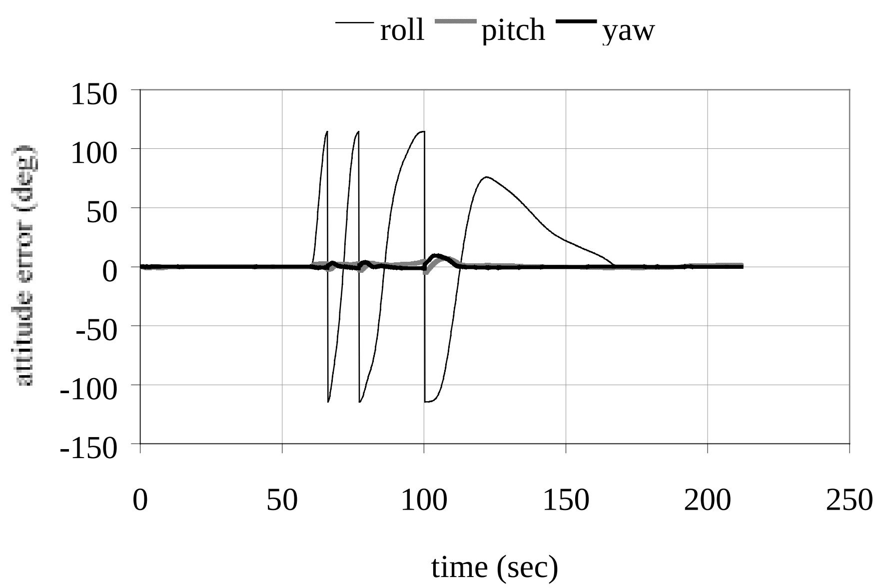


*Figure 9 – Attitude error angles for X-33 Ascent, multiple actuator hard-overs at 60 seconds, insufficient control power remains to maintain desired command for approximately 60 seconds*

To illustrate the effect of PCH, the estimated model error $\nu_{ad}$, the roll hedge $\nu_h$, and the desired pseudo-control $\nu$ are shown for the roll axis in Fig 10. The model error $\nu_{ad}$ tracks, among other phenomena, the trim needed due to the hard-overs. The magnitude of the hedge indicates how much less authority the control system is getting than what it desires. The behavior of $\nu_{ad}$ illustrates that the NN is well behaved in the presence of considerable actuator saturation (as indicated by the presence of a non-zero $\nu_h$ in this case). The remaining aerodynamic surface actuator commands are shown in Fig 11. During portions of this trajectory the remaining actuators are at absolute limits; in other portions they are experiencing axis priority limits.

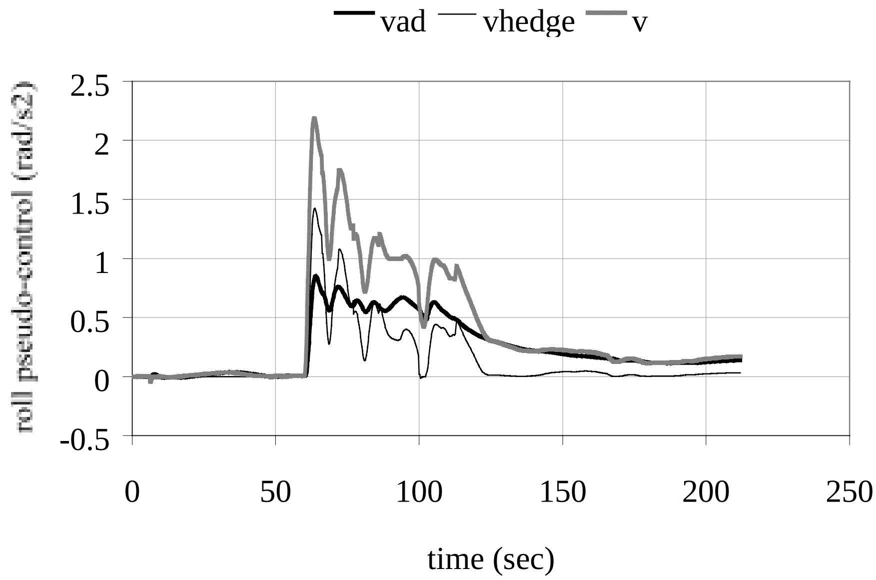


*Figure 10 – Roll pseudo-control signals; showing satisfactory behavior of NN output, and the large magnitude of the hedge signal for multiple actuator hard-overs occurs at 60 seconds*

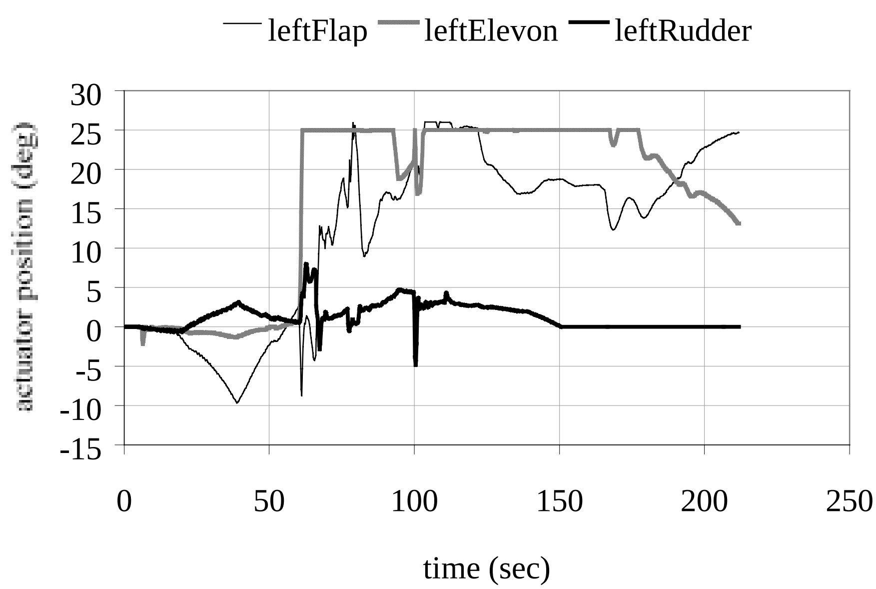


*Figure 11 – Actual actuator positions for non-failed aerodynamic surface actuators*

Although Assumption 2 was violated for approximately 60 seconds, the NN and tracking error behaved properly as suggested by the theory, and the system recovered quickly once sufficient control power was regained.

## Entry Flight Control

At the beginning of entry phase, all NN parameters, inputs, and weight matrices are maintained from the ascent phase. However, slower linear response was specified to correspond to a reduction in available control power. $K_p$ and $K_d$ were chosen based on a natural frequency of 0.5, 0.8, and 0.7 $^{rad}/_{\sec^2}$ for stability-axis roll, pitch, and stability-axis yaw axes respectively and a damping ratio of 0.7 for pitch and yaw, 1.0 for roll. Also, the formulation of guidance command differs, being an angle of attack and angle of bank command, rather than an attitude command. This was converted into an attitude command by finding the attitude that corresponds to the specified guidance command, assuming vehicle velocity with respect to the air-mass was fixed. Once again, the nominal inversion consisted of multiplying desired angular acceleration by an estimate of vehicle inertia, and utilizing the baseline entry control allocation system[1]. This includes RCS jet selection.

The aerodynamic surface actuator position and rate limits were included in the PCH, as was the RCS quantization. These limits included axis priority logic within the control allocation system.

Nominal Entry Results

Angle of attack and bank angle for nominal transition and entry are shown in Figs 12 and 13 respectively. Performance is satisfactory without the need for gain scheduling.

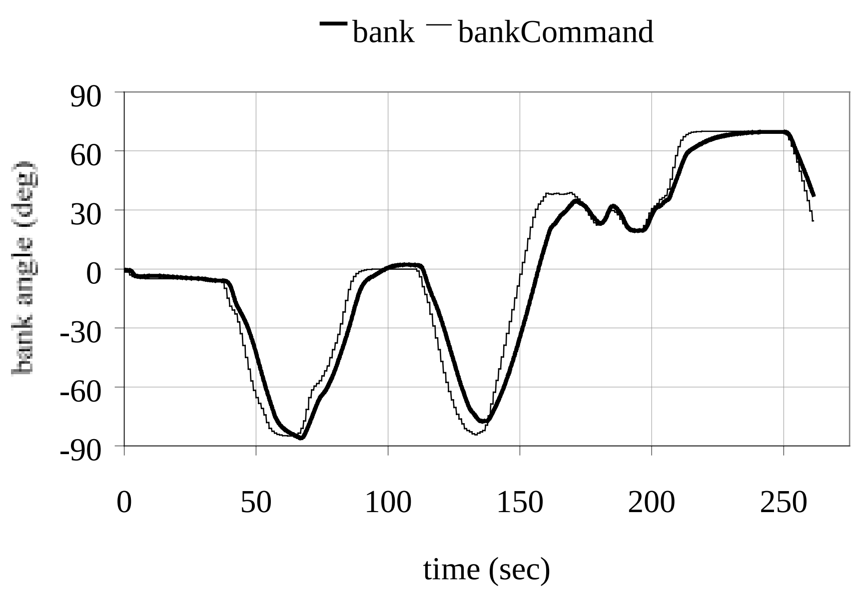


*Figure 12 – Command and actual angle of attack for nominal X-33 entry phase*

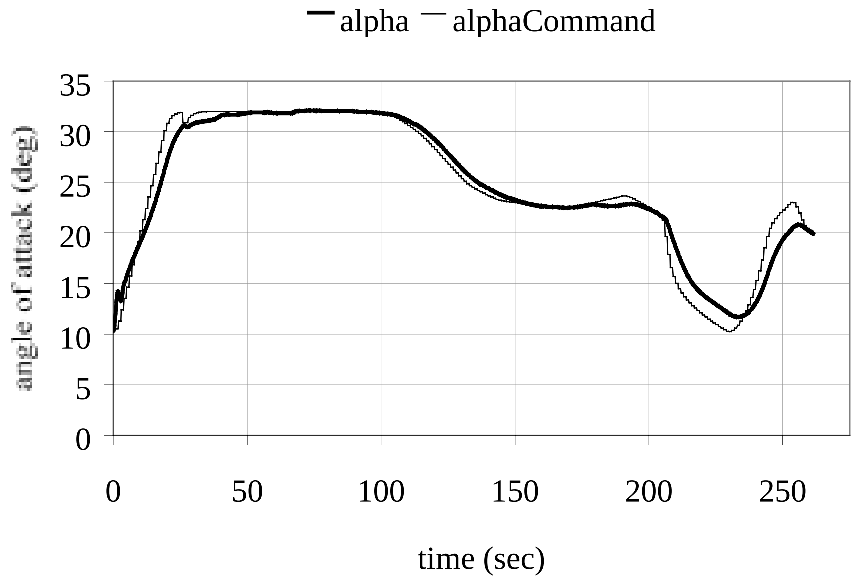


*Figure 13 – Command and actual bank angle for nominal X-33 entry phase*

# Conclusions

Work to date has demonstrated a potential for NN-based adaptive control to reduce RLV flight control tuning requirements and to enhance flight control system performance with respect to failure scenarios.

The Pseudo-Control Hedging (PCH) result was critical in achieving the performance demonstrated. The implications of PCH include the ability to protect the adaptive element from problems (faulty adaptation) associated with input position and rate saturation, as well as quantized control. It also allows the designer to prevent attempted adaptation to other selected input dynamics. PCH also enables correct adaptation while in partial or no control of the plant.

The overall result is that flight certification of this class of adaptive control becomes considerably more practical.

# Acknowledgements

NASA Marshall Space Flight Center supported this research, grant number NAG 3-1638.

# Appendix

**Proof of Theorem 1:**[18] Tracking error dynamics, Eqn 14, with the robustifying term included can be written

$$\dot{e} = Ae + B\{\nu_{ad} - \nu_r - \Delta\} \quad \text{(A-1)}$$

$$\dot{e} = Ae + B\{W^T \overline{\sigma}(V^T \overline{x}) - W^{*T} \overline{\sigma}(V^{*T} \overline{x}) + \varepsilon + \nu_r\} \quad \text{(A-2)}$$

where and $\varepsilon$ is the instantaneous residual network approximation error. Utilizing a Taylor-series expansion for the sigmoids with respect to $\hat{V}$, this can be rewritten as

$$\dot{e} = Ae + B\begin{Bmatrix} W^T\left[\overline{\sigma}(V^T\overline{x}) - \overline{\sigma}_z(V^T\overline{x})V^T\overline{x}\right] - \\ W^T\overline{\sigma}_z(V^T\overline{x})\tilde{V}^T\overline{x} + \varepsilon + w + \nu_r \end{Bmatrix} \quad \text{(A-3)}$$

where

$$w = W^{*T} O(\tilde{V}^T\overline{x})^2 + \tilde{W}^T\overline{\sigma}_z(V^T\overline{x})V^{*T}\overline{x} \quad \text{(A-4)}$$

and $\tilde{W} = W - W^*$, $\tilde{V} = V - V^*$. The term $O(\tilde{V}^T\overline{x})^2$ represents 2[nd] and higher terms of the Taylor-series expansion. An upper bound on the norm of $w$ can be written as

$$\|w\| \le c_0 + c_1\|Z\| + c_2\|Z\|\|r\| + c_3\|Z\|^2 \quad \text{(A-5)}$$

where $c_i, i = 0,1,2,3$ depends on the size of the NN and the assumed over-bound on the weights, $\overline{Z}$. Define the Lyapunov function

$$L(e,V,W) = \tfrac{1}{2}\left[e^T Pe + tr\left(\tilde{V}\Gamma_v^{-1}\tilde{V}^T\right) + tr\left(\tilde{W}^T\Gamma_w^{-1}\tilde{W}\right)\right] \quad \text{(A-6)}$$

The time derivative of this function can be expressed as

$$\dot{L} = -\|e\|^2 + r^T(\varepsilon + w + \nu_r) + \lambda\|r\| tr\left(\tilde{Z}^T Z\right) \quad \text{(A-7)}$$

By utilizing the triangle inequality and including the definition of $\nu_r$, one obtains

$$\begin{aligned} \dot{L} \le &-\|e\|^2 + \|r\|(\overline{\varepsilon} + \|w\|) + \lambda\|r\|\|\tilde{Z}\|^2 + \lambda\|r\|\|\tilde{Z}\|\overline{Z} \\ &-K_{r0}\|r\|^2 - K_{r1}\|r\|^2(\|Z\| + \overline{Z}) \end{aligned} \quad \text{(A-8)}$$

This can now be expressed as

$$\dot{L} \le -\|e\|^2 - a_0\|r\|^2 + a_1\|r\| \quad \text{(A-9)}$$

where

$$a_0 = K_{r0} + 2K_{r1}\overline{Z} + (K_{r1} - c_2)\|\tilde{Z}\| \quad \text{(A-10)}$$

$$a_1 = (c_3 - \lambda)\|\tilde{Z}\|^2 + (\lambda\bar{Z} - c_1)\|\tilde{Z}\| + \bar{\varepsilon} + c_0 \tag{A-11}$$

So $a_0$ is greater than zero when $K_{r1} > c_2$. This condition is sufficient to guarantee that $\dot{L}$ is negative semidefinite when $\|r\| \ge |a_1/a_0|$. Therefore, via LaSalle theorem, $e$ and $Z$ remain bounded. When Assumption 2 is also enforced, all system signals are bounded.

# References


[1]Hanson, J., Coughlin, D., Dukeman, G., Mulqueen, J., and McCarter, J. Ascent, Transition, Entry, and Abort Guidance Algorithm Design for X-33 Vehicle. *AIAA Guidance, Navigation, and Control Conference*, 1998.

[2]Hanson, J. of NASA MSFC. Personal Communication. April 3, 2000.

[3]Hall, C., Gallaher, M, and Hendrix, N. X-33 Attitude Control System Design for Ascent, Transition, and Entry Flight Regimes. *AIAA Guidance, Navigation, and Control Conference*, 1998.

[4]Calise, A., and Rysdyk, R. Nonlinear Adaptive Flight Control Using Neural Networks. *Control Systems Magazine*, December 1998.

[5]Brinker, J., and Wise, K. Flight Testing of a Reconfigurable Flight Control Law on the X-36 Tailless Fighter Aircraft. *AIAA Guidance, Navigation, and Control Conference*, 2000.

[6]Jiang, T. *Combined Model- and Rule-based Controller Synthesis With Application to Helicopter Flight Control*. Ph.D. Thesis, Georgia Institute of Technology, 1998.

[7]Rovithakis, G. Nonlinear Adaptive Control in the Presence of Unmodelled[sic] Dynamics using Neural Networks. *Proceedings of the Conference on Decision and Control*, 1999.

[8]Wang, H. and Sun, J., Modified Model Reference Adaptive Control with Saturated Inputs, *Proceedings of the 31st Conference on Decision and Control*, IEEE, 1992.

[9]Monopoli, R., Adaptive Control for Systems with Hard Saturation, *Proceedings of the 1975 IEEE Conference on Decision & Control*, IEEE, 1975.

[10]Rysdyk, R. *Adaptive Nonlinear Flight Control*. Ph.D. Thesis, Georgia Institute of Technology, 1998.

[11]Slotine, J., and Li, W. *Applied Nonlinear Control*. Prentice Hall, 1991.

[12]Kim, B., and Calise, A. Nonlinear Flight Control Using Neural Networks. *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997.

[13]McFarland, M. *Adaptive Nonlinear Control of Missiles Using Neural Networks*. Ph.D. Thesis, Georgia Institute of Technology, 1997.

[14]Hornik, K., Stinchombe, M., and White, H., Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, Vol. 2, 1989.

[15]Lewis, F., Jagannathan, S., and Yesildirek, A., *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor and Fancis, London, 1998.

[16]McFarland, M., and Calise, A.  Multilayer Neural Networks and Adatpive Nonlinear Control of Agaile Anti-Air Missles.  *AIAA Guidance, Navigation, and Control Conference*, 1997.

[17]McCarter, J.  *Maveric Users Guide*.  NASA/MSFC/TD54, 1999.

[18]Calise, A., Lee, S., and Sharma, M.  Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft.  *AIAA Guidance, Navigation, and Control Conference*, 1998.