Fault-Tolerant Nonlinear Adaptive Flight Control For the X-33

Final Report

1 September 2001 to 30 June 2003

Research Supported by:
NASA Marshall Space Flight Center
Huntsville, AL

Grant No. NAG-8-1638

Principal Investigator: Dr. Anthony J. Calise*
Principal Investigator: Dr. Eric N. Johnson**
Technical Advisor: Dr. John Hanson, NASA Marshall Space Flight Center
Graduate Research Assistant: Matthew Johnson

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

* Professor, (404) 894-7145, anthony.calise@ae.gatech.edu
** Professor, (404) 385-2519, eric.johnson@ae.gatech.edu

Contents

Abstract

Future reusable launch vehicles must be designed to fly a wide spectrum of missions and survive numerous types of failures. The X-33 Reusable Launch Vehicle Technology Demonstrator is used as a simulation platform for testing a neural network-based model-reference adaptive controller and compared to a gain scheduled controller over a wide matrix of test cases, including dispersions, modeling errors, and actuator failures. Actuator failures include engine out, aerodynamic actuator failures, and reaction control system failures. Modeling errors include mismodeling of the force and moment coefficients and the addition of approximate flexible dynamics. The resulting control system does not use gain tables and is not tuned for particular missions. Several test cases are examined in detail, particularly engine failure cases, which result in mission abort scenarios. Over the entire test matrix, the adaptive controller is able to perform slightly better than a gain-scheduled controller. The significance of this fact is that the adaptive approach offers an approach to launch vehicle flight control design that greatly simplifies the flight control system design process, and that ultimately may be adaptable to a wider range of failures and dispersions than are anticipated within the test matrix. Moreover, since for this class of vehicles the flight control system is essential in evaluating the impact of configuration changes in preliminary design, an approach that is adaptive to these design changes is highly desirable.

I. Introduction

The X-33 (Fig 1) was a proposed sub-orbital aerospace vehicle intended to demonstrate technologies necessary for future Reusable Launch Vehicles (RLVs). It included 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].

To achieve the cost benefits of an operational RLV, the amount of analysis and testing required per mission must 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, 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].

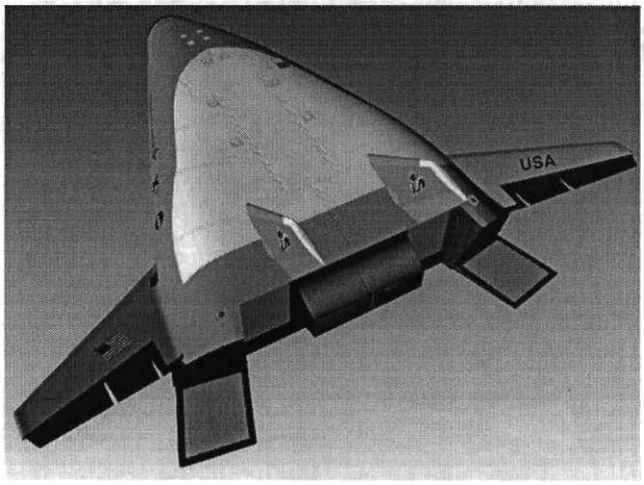

Figure 1. X-33 Reusable Launch Vehicle technology demonstrator.

Launch vehicle flight control is conventionally linearized about a series of operating points and then 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[4,5]. The use of NN-based adaptive flight control has been demonstrated in piloted simulation and flight test on the X-36 aircraft[6], in simulation and drop tests of the JDAM attack munition[7,8,9] and in a piloted simulation on a civil transport aircraft[10,11]. 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.

This approach has been implemented on the X-33[12,13] demonstrating adaptation to failures. Also, work has been done to introduce adaptation in the outer, guidance, loop[14]. In the previous work a trivial choice of feedback linearization was used based on an inertia matrix calculated based on an estimate of the fuel consumption, neglecting aerodynamic moments not due to the control effectors. In addition a method for addressing nonlinear system input characteristics, Pseudo Control Hedging (PCH), was introduced to facilitate correct adaptation in the presence of actuator position and rate limits, time delay, and input quantization.

The work described here is a continuation of the study described in Refs. 12 and 13. The adaptive controller has been improved for evaluation on a broader range of test cases, and adjusted so as to improve its performance in accordance with test criteria established by NASA Marshall Space Flight Center[15]. The test matrix includes nominal flight, Power Pack Out (PPO), Thrust Vector Control (TVC) failures and mismodeling, aerosurface failures, and reaction control system (RCS) failures. Dispersion cases are investigated for both nominal and abort situations. The complete test matrix is available in Appendix B. Across the test matrix the algorithm was scored on actuator deflection magnitude, duty cycle (a measure of control activity) peak body rates, dynamic pressure profile, steady state error, and how closely the vehicle follows the intended trajectory. The adaptive controller presented here was compared in Ref. 15 to a sliding mode controller[16], a trajectory linearization controller[17], and a reconfigurable allocator[18]. In addition, a hybrid direct-indirect adaptive controller[19] was tested in ascent only but not compared to the other controllers. NASA's tests are ongoing.

II. NN-Based State Feedback Adaptive Control

II.1 Control Architecture

Fig 2 is an illustration of Model Reference Adaptive Control (MRAC) [12,20] 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.

II.2 Model Reference Adaptive Control

For simplicity, consider the case of full model inversion, in which the plant dynamics are taken to be of the form

$$\begin{aligned} \dot{q} &= Q(q,\omega) \\ \dot{\omega} &= f(q,\omega,\delta) \\ \delta &= g(q,\omega,\delta_{cmd}) \end{aligned} \tag{1}$$

where $q \in \Re^4$ is a quaternion representing body attitude with respect to an inertial frame, $\omega \in \Re^3$ is the angular velocity of the body in the inertial frame expressed in the body frame, and $\delta, \delta_{cmd} \in \Re^m$ are actuator positions and actuator commands, respectively.

We introduce a pseudo-control input ν such that the dynamic relation between it and the system state is linear

$$\dot{\omega} = \nu \tag{2}$$

where

$$\nu = f(q,\omega,\delta) \tag{3}$$

The actual controls δ are obtained by inverting Eq. (3). Since the function $f(q,\omega,\delta)$ is usually not known exactly, an approximation is introduced

$$\nu = \hat{f}(q,\omega,\delta) \tag{4}$$

which results in a modeling error. The resulting dynamics can be written as

$$\dot{\omega} = \nu + \Delta(q,\omega,\delta) \tag{5}$$

where

$$\Delta(q,\omega,\delta) = f(q,\omega,\delta) - \hat{f}(q,\omega,\delta) \tag{6}$$

$\hat{f}$ is assumed to have a known inverse and obeys the control effectiveness sign condition

$$\text{sign}\left(\frac{\partial \hat{f}}{\partial \delta}\right) = \text{sign}\left(\frac{\partial f}{\partial \delta}\right) \tag{7}$$

The actuator command is constructed as*

$$\delta_{cmd} = \hat{f}^{-1}(x, \dot{x}, \nu) \tag{8}$$

The model error in Eq. (5) will be adaptively compensated using a neural network trained on-line.

As shown in Figure 2, the pseudo-control signal is constructed of four components

$$\nu = +\nu_{pd} - \nu_{ad} - \nu_r \tag{9}$$

where ν_{rm} is generated by the reference model, ν_{pd} is the output of the P-D compensator, ν_{ad} is the signal generated by the adaptive element introduced to compensate for the model inversion error, and ν_r is a robustifying term[5,21]

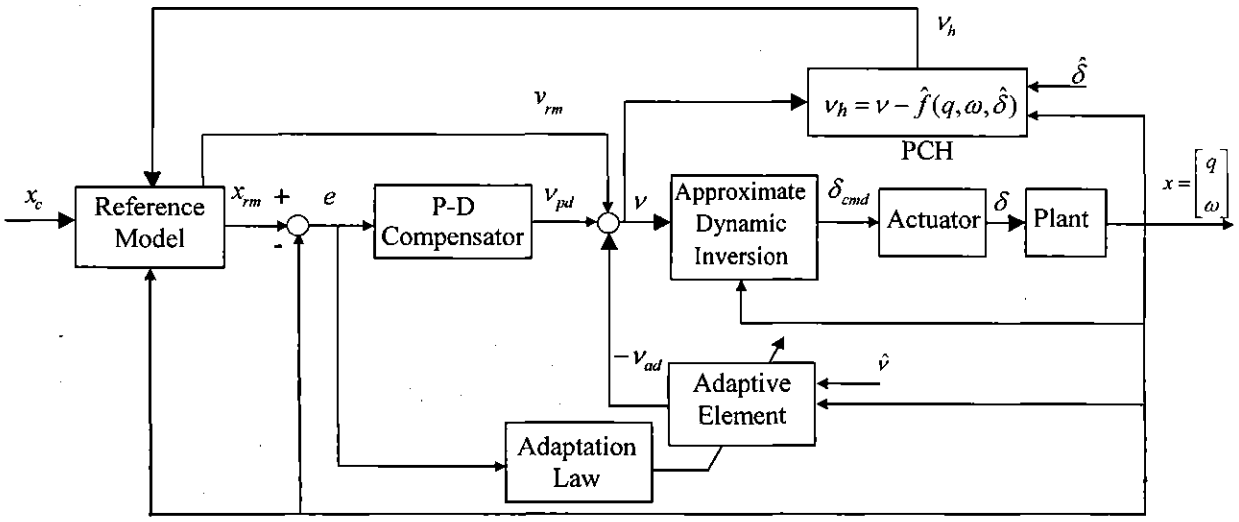

Figure 2. Implementation of PCH in NN adaptive control.

* For the case m>3 we assume here that a control allocation algorithm has been pre-defined. The particular allocation used in this implementation will be discussed in a later section.

II.3 Reference Model and Linear Controller Design

The hedge signal is

$$\nu_h = \hat{f}(q,\omega,\delta_{cmd}) - \hat{f}(q,\omega,\hat{\delta}) = \nu - \hat{\nu} \tag{10}$$

where $\hat{\delta}$ is an estimate of the actuator positions based on actuator models incorporating rate and position limits. The hedge signal is introduced into the reference model in the following way

$$\dot{q}_{rm} = Q\left[q, \omega_{rm} + (\Omega_{q_{rm}}^{-1}\Omega_q - I)\omega\right] \tag{11}$$

$$\dot{\omega}_{rm} = \nu_{crm}(q_{rm}, \omega_{rm}, q_c, \omega_c, \dot{\omega}_c) - \nu_h = \nu_{rm} \tag{12}$$

The reference model states are quaternion quantities. Details relating to the functional forms of $Q(*)$ and $\nu_{crm}(*)$ in the above equations will be provided later. The reference model is 7th order, and the states of the reference model represent a desired quaternion state q_{rm} and a desired angular rate vector ω_{rm}. Unlike the typical standard MRAC architecture, the plant states appear in the reference model, which is necessary in this case in order to arrive at an error equation in a form that is suitable for applying adaptive control. This will be further explained below.

Bounded external commands q_c and ω_c are provided by the guidance algorithm, where q_c is the commanded quaternion and ω_c is the commanded body angular rate vector.

The functional form of $Q(*)$ is taken from the expression for the quaternion rate[22]

$$\dot{q} = Q(q,\omega) = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} q = \frac{1}{2}\begin{bmatrix} -q_2 & -q_3 & -q_4 \\ q_1 & -q_4 & q_3 \\ q_4 & q_1 & -q_2 \\ -q_3 & q_2 & q_1 \end{bmatrix} \omega = \frac{1}{2}\Omega_q \omega \tag{13}$$

where

$$\Omega_q = \begin{bmatrix} -q_2 & -q_3 & -q_4 \\ +q_1 & -q_4 & +q_3 \\ +q_4 & +q_1 & -q_2 \\ -q_3 & +q_2 & +q_1 \end{bmatrix} \tag{14}$$

$$\Omega_q^{-1} = \begin{bmatrix} -q_2 & +q_1 & +q_4 & -q_3 \\ -q_3 & -q_4 & +q_1 & +q_2 \\ -q_4 & +q_3 & -q_2 & +q_1 \end{bmatrix} \tag{15}$$

such that

$$\Omega_q{}^{-1}\Omega_q = I \tag{16}$$

The reference model error vector is

$$e_{rm} = \begin{bmatrix} \tilde{Q}(q_c, q_{rm}) \\ \omega_c - \omega_{rm} \end{bmatrix} \tag{17}$$

Appendix A of [23] shows that the coordinates of the first term in the error vector can be expressed in terms of a quaternion error vector, $\tilde{Q}(q_c, q_{rm})$, in two ways:

$$\begin{aligned} \tilde{Q}(q_c, q_{rm}) &= -2sign\left(q_c{}^T q_{rm}\right)\Omega_{q_{rm}}{}^{-1} q_c \\ \tilde{Q}(q_c, q_{rm}) &= 2sign\left(q_{rm}{}^T q_c\right)\Omega_{q_c}{}^{-1} q \end{aligned} \tag{18}$$

Differentiating, we have

$$\dot{e}_{rm} = \begin{bmatrix} \dot{\tilde{Q}}(q_c, q_{rm}) \\ \dot{\omega}_c - \dot{\omega}_{rm} \end{bmatrix} = \begin{bmatrix} 2\Omega_{q_{rm}}{}^{-1}\dot{q}_c - 2\Omega_{q_c}{}^{-1}\dot{q}_{rm} \\ \dot{\omega}_c - \nu_{crm}(q_{rm}, \omega_{rm}, q_c, \omega_c) - \nu_h \end{bmatrix} \tag{19}$$

The reference model tracking error is assumed to be small, and thus $q_c \approx q_{rm}$, implying that $\Omega_{q_c}{}^{-1}\Omega_{q_{rm}} - I \approx 0$, and thus so long as the attitude tracking error does not reach a magnitude of 180°, the first element becomes $\omega_c - \omega_{rm}$.

The function ν_{crm} is chosen as

$$\nu_{crm} = \begin{bmatrix} K_{pc} & K_{dc} \end{bmatrix} e_{rm} \tag{20}$$

where $K_{dc} > 0, \in \Re^{3\times 3}$ and $K_{pc} > 0, \in \Re^{3\times 3}$ are diagonal matrices. This ensures that

$$A_{rm} = \begin{bmatrix} 0 & I \\ -K_{pc} & -K_{dc} \end{bmatrix} \tag{21}$$

is Hurwitz. The reference model tracking dynamics then reduce to

$$\dot{e}_{rm} = A_{rm}\, e_{rm} + B_{rm}\left(\dot{\omega}_c - \nu_h\right) \tag{22}$$

where

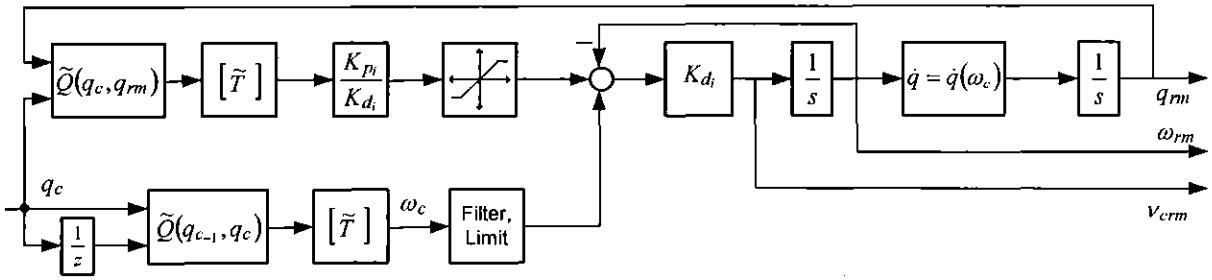

Figure 3. Reference Model.

$$B_{rm} = [0 \quad I]^T \tag{23}$$

Figure 3 shows the reference model architecture, where $\tilde{T}$ is the transform from quaternions to body attitude. Back-differentiation is used to calculate a feed forward rate command

We write the tracking error dynamics as

$$e = \begin{bmatrix} \tilde{Q}(q_{rm,q}) \\ \omega_{rm} - \omega \end{bmatrix} \tag{24}$$

Differentiating Eq. (17), we get

$$\dot{e} = \begin{bmatrix} \dot{\tilde{Q}}(q_{rm}, q) \\ \dot{\omega}_{rm} - \dot{\omega} \end{bmatrix} = \begin{bmatrix} 2\Omega_q{}^{-1}\dot{q}_{rm} - 2\Omega_{q_{rm}}{}^{-1}\dot{q} \\ v_{crm}(q_{rm}, \omega_{rm}, q_c, \omega_c, \dot{\omega}_c) - v_h - f(q, \omega, \delta) \end{bmatrix} \tag{25}$$

Under a similar set of assumptions, the first element becomes $\omega_{rm} - \omega$ as in Eq. (19). Following a procedure similar to that outlined above and applying Eqs. (1), (6), (9), (10), (11), (12), the second element simplifies to yield

$$\dot{e} = \begin{bmatrix} \omega_{rm} - \omega \\ -v_{pd} + v_{ad} + v_r - \Delta \end{bmatrix} \tag{26}$$

Choosing

$$v_{pd} = [K_p \quad K_d]e \tag{27}$$

the error dynamics reduce to a standard form for the application of adaptive control.[24]

$$\dot{e}_{rm} = Ae_{rm} + B[v_{ad} + v_r - \Delta] \tag{28}$$

where

$$A_{rm} = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} \tag{29}$$

and $B = B_{rm}$.

Figure 4 shows the manner in which this is computed.

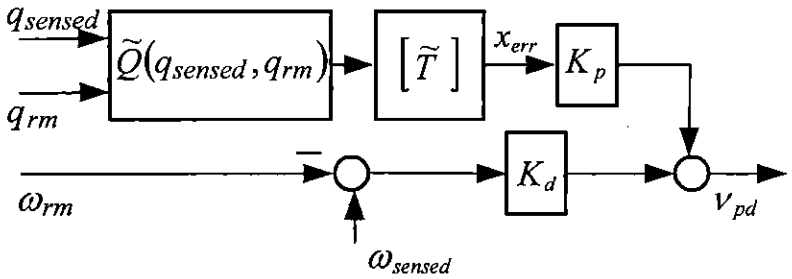

Figure 4. Proportional-Derivative Controller.

II.4 Neural Network for Inversion Error Compensation

A nonlinear single hidden-layer (SHL) NN is used to compensate for the model inversion error.[25,26] For an input vector $\bar{x}$, the output of the SHL NN is given by

$$\nu_{ad} = W^T \sigma\left(V^T \bar{x}\right) \tag{30}$$

where V and W are the input and output weighting matrices, respectively, and σ is a vector of sigmoid activation functions. The weights are adapted according to the following equations[7,22]:

$$\dot{W} = -\left[\left(\sigma - \sigma' V^T \bar{x}\right)\eta + \kappa \|e\| W\right]\Gamma_W \tag{31}$$

$$\dot{V} = -\Gamma_V \left[\bar{x}\eta W^T \sigma' + \kappa \|e\| V\right] \tag{32}$$

where Γ_W and Γ_V are the positive definite learning rate matrices, σ' is the partial derivative of the sigmoids σ with respect to the NN inputs $\bar{x}$, and κ is the e-modification parameter. η is defined by

$$\eta = e^T PB \tag{33}$$

Here, $P \succ 0$ is a positive definite solution of the Lyapunov equation

$$A^T P + PA + Q = 0 \tag{34}$$

for any positive definite $Q \succ 0$. A and B in the above equations are the tracking error dynamics matrices defined in Eqs. (21) and (23). The robustifying term in Eq. (9) is

$$\nu_r = -\left(\|Z\| + \bar{Z}\right) K_r \eta^T \tag{35}$$

where

$$Z = \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \tag{36}$$

$K_r \in \Re^{3\times 3}$ diagonal, $K_r < 0$, and $\bar{Z}$ is such that $\|Z^*\| \le \bar{Z}$, where Z^* denotes an unknown set of ideal weights.

III. Pseudo Control Hedging

III.1 Hedge Calculation with Redundant Actuation

Because the X-33 has redundant actuation, care must be taken when calculated the hedging signal. Consider the following affine system:

$$\begin{aligned} \dot{x} &= f(x) + B(x)\delta \\ y &= Cx \end{aligned} \tag{37}$$

where x is the state vector, u is the vector of control variables and y represents the *regulated* output variables. It is assumed that $\dim\{\delta\} > \dim\{y\}$. Since there are more control variables than there are regulated output variables, a control allocation matrix is often introduced that relates the control demand associated with each output variable to the actual controls. Letting δ_e denote the *effective* control demand, then

$$\delta = T_a \delta_e \tag{38}$$

To compute the inversion, differentiate the output once and use equations (37) and (38) to obtain

$$\dot{y} = C\dot{x} = C[f(x) + BT_a\delta_e] = \nu \tag{39}$$

where ν is the required pseudo control. Assuming that $C\hat{B}(x)T_a$ is invertible for all x, then the *effective* control solution reduces to

$$\delta_e = (C\hat{B}(x)T_a)^{-1}(\nu - C\hat{f}(x)) \tag{40}$$

where $\hat{B}(x)$ and $\hat{f}(x)$ denote estimates of $B(x)$ and $f(x)$. Substitution of (40) into (38) provides the actual control that is to be applied to the plant. The PCH signal is next introduced as an additional input into the reference model, forcing it to "move back". If the reference model update without PCH was of the form

$$\ddot{x}_{rm} = f_{rm}(x_{rm}, \dot{x}_{rm}, x_c) \tag{41}$$

where x_c is the external command signal, then the reference model update with PCH becomes

$$\ddot{x}_{rm} = f_{rm}(x_{rm}, \dot{x}_{rm}, x_c) - \nu_h \tag{42}$$

The instantaneous pseudo-control output of the reference model that is used as an input to the linearized plant model is not changed by the use of PCH and remains

$$\nu_{rm} = f_{rm}(x_{rm}, \dot{x}_{rm}, x_c) \tag{43}$$

Hence, the effect of the PCH signal on the pseudo-control is introduced only through the reference model dynamics. This results from the stability analysis of NN based adaptive control with PCH, detailed in Ref. 18.

When implementing PCH, (38) is viewed as defining the commanded control

$$\delta_c = T_a \delta_e \tag{44}$$

which may be different from the actual control displacements, due to actuator position and rate limits, and other effects that one might choose to model. For example, with PCH one may also take into account the dynamics of the actuator as well, although if the bandwidth of an actuator is much greater than the design bandwidth set by the command filters, this effect can be ignored. The hedging signal is computed according to the following equation

$$\nu_h = \nu - C[\hat{f}(x) + \hat{B}(x)\hat{\delta}] \tag{45}$$

where the elements of $\hat{\delta}$ denote estimates of the actual control positions derived by processing each element of the commanded control obtained from Equation (44) through a model for the corresponding actuator.

III.2 Inverse model

The form of the inverse model $\hat{f}^{-1}$ is shown in Figure 5. The pseudo control, ν, is transformed from desired angular acceleration to desired torque, T, by taking the pseudo-inverse of the actuator effectiveness matrix and multiplication by the approximate inertia. The desired torques are first allocated to the aerodynamic actuators, then daisy-chained to the RCS jet selection allocator. The allocation is daisy-chained in order to preserve RCS fuel if possible. Figure 6 and 7 show the aerodynamic and RCS allocations, respectively.

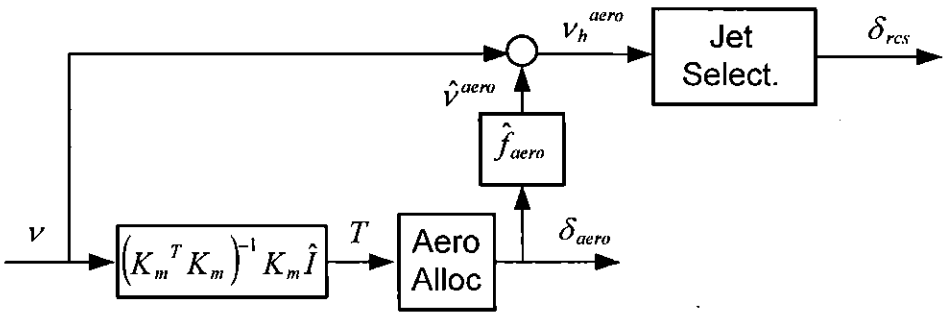

Figure 5. Inverse model dynamics.

Figure 6 shows that as commanded moments enter the aerodynamic allocator, axis priority is given to yaw over roll. First both yaw and roll are subjected to estimated rate limits a and b, based on an estimated yaw and roll rate limits calculated from the flap rate limit and the flap effectiveness

in yaw and roll. The yaw moment is then magnitude-limited by e, which is found from the maximum flap deflection allocated to yaw and the flap effectiveness in yaw, combined with the maximum elevon deflection allocated to yaw and the elevon effectiveness in yaw. From these commands, flap and elevon 'amounts used' are calculated and sent into the two roll moment magnitude limiters, which have magnitudes c and d. These magnitudes are calculated in a similar manner, based on the desired moments and the actuator effectiveness while accounting for the extent to which each actuator has been used for yaw control.

The commands in the three axes are multiplied by the actuator effectiveness matrix to get the desired actuator deflections and commands for each actuator are position and rate limited.

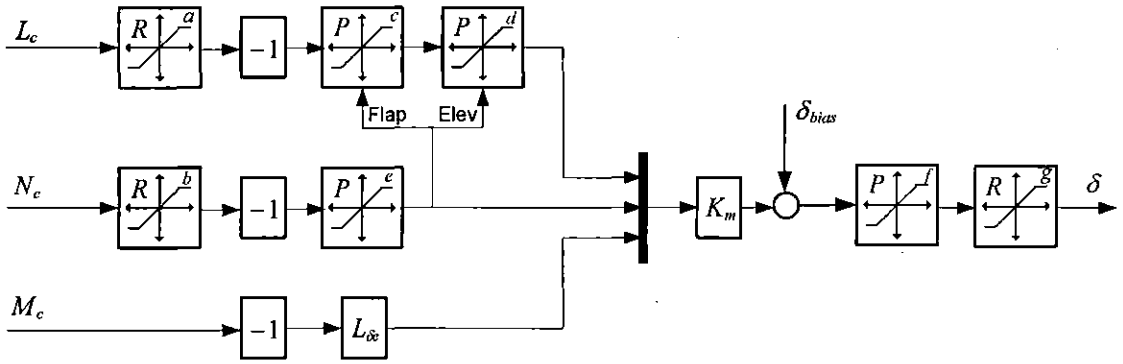

Figure 6. Allocation in the Aerodynamic Actuators.

The RCS allocation for entry is shown in Figure 7. The hedge from the aerodynamic actuators is multiplied by the inertia matrix to determine the torque deficit that we wish to eliminate with RCS. Since there are only eight jets, this torque deficit is compared with every possible combination of firings, and the errors in torque along with the number of jets fired is used in the calculation of the cost function. Two versions of the cost function are used:

$$\begin{aligned} Cost &= (T_c - T_a)k_1 + (\#\text{Jets})^2 \cdot k_2 \\ Cost &= (T_c - T_a)k_1 + \#\text{Jets} \cdot \text{k}_3 + \Delta RCS \cdot k_4 \end{aligned} \tag{46}$$

The first version is engaged if there are no known failures in the system. In this case, the cost function heavily penalizes using too many jets at once in order to reduce fuel utilization. In the second case, when there are known failures, the number of jets are only penalized linearly, and the number of firing changes (ΔRCS) is penalized to prevent the jets from switching on and off excessively. The amount of torque achieved from the RCS system is added to the aerodynamic hedge signal and transformed into stability axes to get the complete hedge signal for both the aerodynamic and RCS actuators.

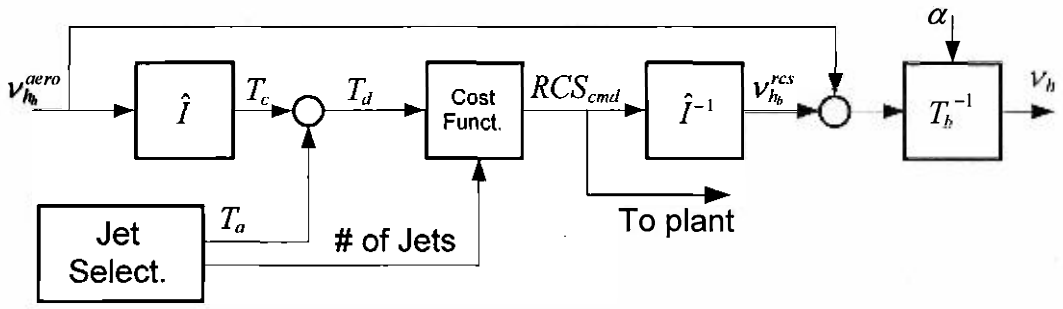

Figure 7. Hedge Calculation: Deficit in Aerodynamic Actuation + Deficit in RCS Actuation.

IV. Results

The flight control design is evaluated from launch to the beginning of the Terminal Area Energy Management (TAEM) phase. Missions being with vertical launch and achieve peak Mach numbers of approximately 8, peak altitudes of 180,000 feet, and dynamic pressures that range from 20-400 psf. During ascent, vehicle mass decreases by approximately a factor of 3, and vehicle inertia by a factor of 2. The controller was tested over a wide matrix of cases, including single tests of two nominal (no failure) cases, two PPO cases, four TVC failure cases, 17 aerosurface failures with different surfaces and different time of failure, four RCS failures, seven occurrences of unknown modeling errors, and ten dispersion tests (different random seed and season of the year) with both failures and the nominal cases.

IV.1 Ascent Flight Control

The adaptive flight control design illustrated in Figures 2-6 was utilized for ascent flight control. Nominal inversion consisted of multiplying desired angular acceleration by an estimate of vehicle inertia, and utilizing the control allocation system shown in Figure 6. NN inputs were angle-of-attack, side-slip angle, bank angle, sensed vehicle angular rate, and estimated pseudo-control ($\hat{\nu} = \nu - \nu_h$). This provides the 'best' estimate for the correct position of the actuators. Four hidden activation functions were used; learning rates on W were unity for all axes and learning rates for V were 200 for all inputs. K_p and K_d were chosen based on a maximum natural frequency of 1.0 $^{rad}/_{sec}$ 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, but was not given knowledge of actuation failures when they occurred. Knowledge of an actuation failure was used in the RCS allocation, as Eq. (46) shows.

IV.2 Entry Flight Control

At the beginning of the entry phase, the values of the NN parameters and weight matrices are maintained from the ascent phase. However, a slower linear response was specified in recognition of a reduction in available control power. K_p and K_d were chosen based on a natural frequency of 0.7, 0.8, and 0.8 $^{rad}/_{sec}$ for stability-axis roll, pitch, and stability-axis yaw axes respectively, and a damping ratio of 0.9. Moreover, the natural frequencies are linearly reduced to a minimum of 70% if the dynamic pressure falls below 50 psf. The guidance command during entry changes for attitude command to angle of attack and bank angle command. These commands were 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.

IV.3 Test Case 1

Figures 8-15 show a comparison of the performance for the adaptive controller and the baseline PID controller designed by NASA (NPID). Attitude error angles in the ascent phase are shown in Figure 8, and the differential throttle and aerodynamic actuator time histories are shown in Figures 9 and 10, respectively. For this baseline case, the adaptive controller seems to perform slightly better at regulating attitude errors. The primary difference between the control time histories for the two controllers seems to be that the adaptive controller relies more on aerodynamic actuation than differential throttle, and that the adaptive controller has significant actuation near t=0. Since the dynamic pressure is very low at this time, the adaptive controller should have been designed to avoid this. Future work will take this into account.

Entry tracking is shown in Figures 11-13. Results for the two controllers are similar, with the NPID appearing to do slightly better in regulating the sideslip angle command. Sideslip regulation in the adaptive architecture needs to be improved, as the vehicle is very sensitive to small sideslip variations in entry, especially at high dynamic pressures, where a 1° sideslip angle can cause departure. Figures 14 and 15 show the aerodynamic actuator and RCS time histories. In entry, the rudders are in the wake of the vehicle, and have little to no authority. The NPID commands zero for most of the duration of entry, while the adaptive controller commands saturation. For the rudders in entry, differences in the commands are essentially moot. The adaptive controller saturates the differential elevon just before 250 seconds, and it remains saturated for nearly 50 seconds. Meanwhile, the NPID design is using more RCS in the roll channel. The two controllers are using different actuators to track the roll command that occurs at about 240 seconds. The adaptive architecture is structured to use the aerodynamic actuators first in order to conserve RCS fuel. The aerodynamic actuator time histories for the adaptive control system are noisier than those of the NPID controller.

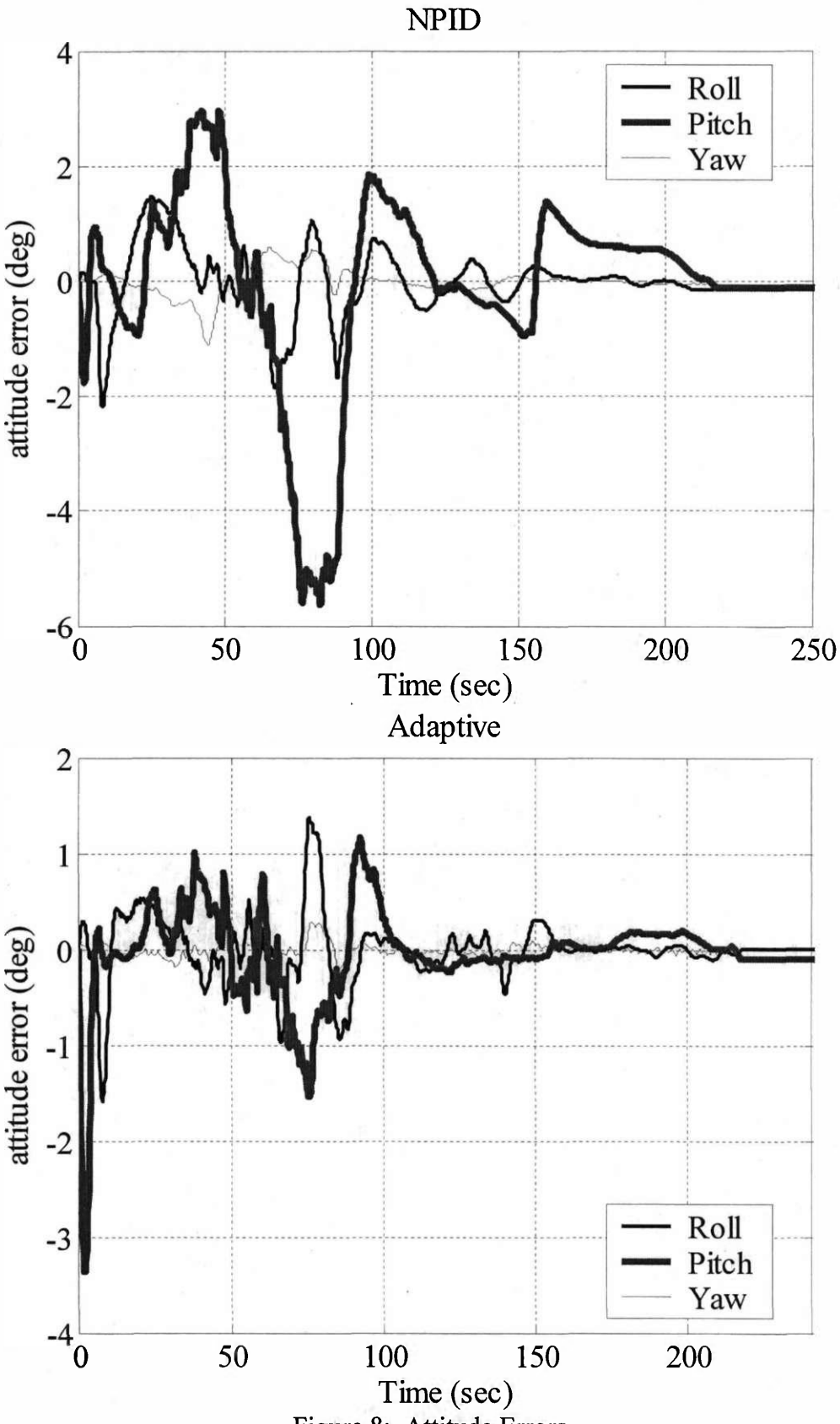

Figure 8: Attitude Errors

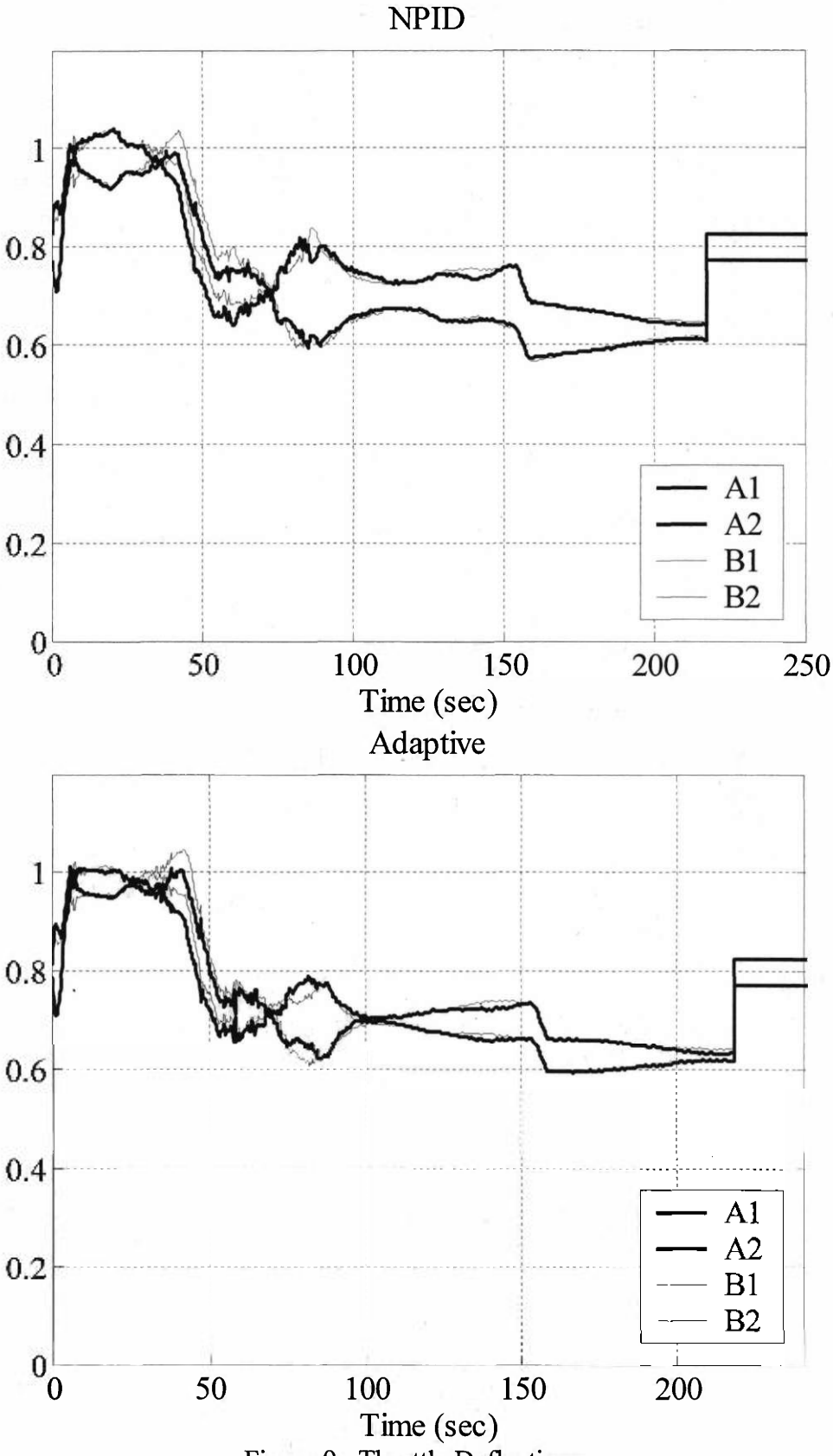

Figure 9: Throttle Deflections

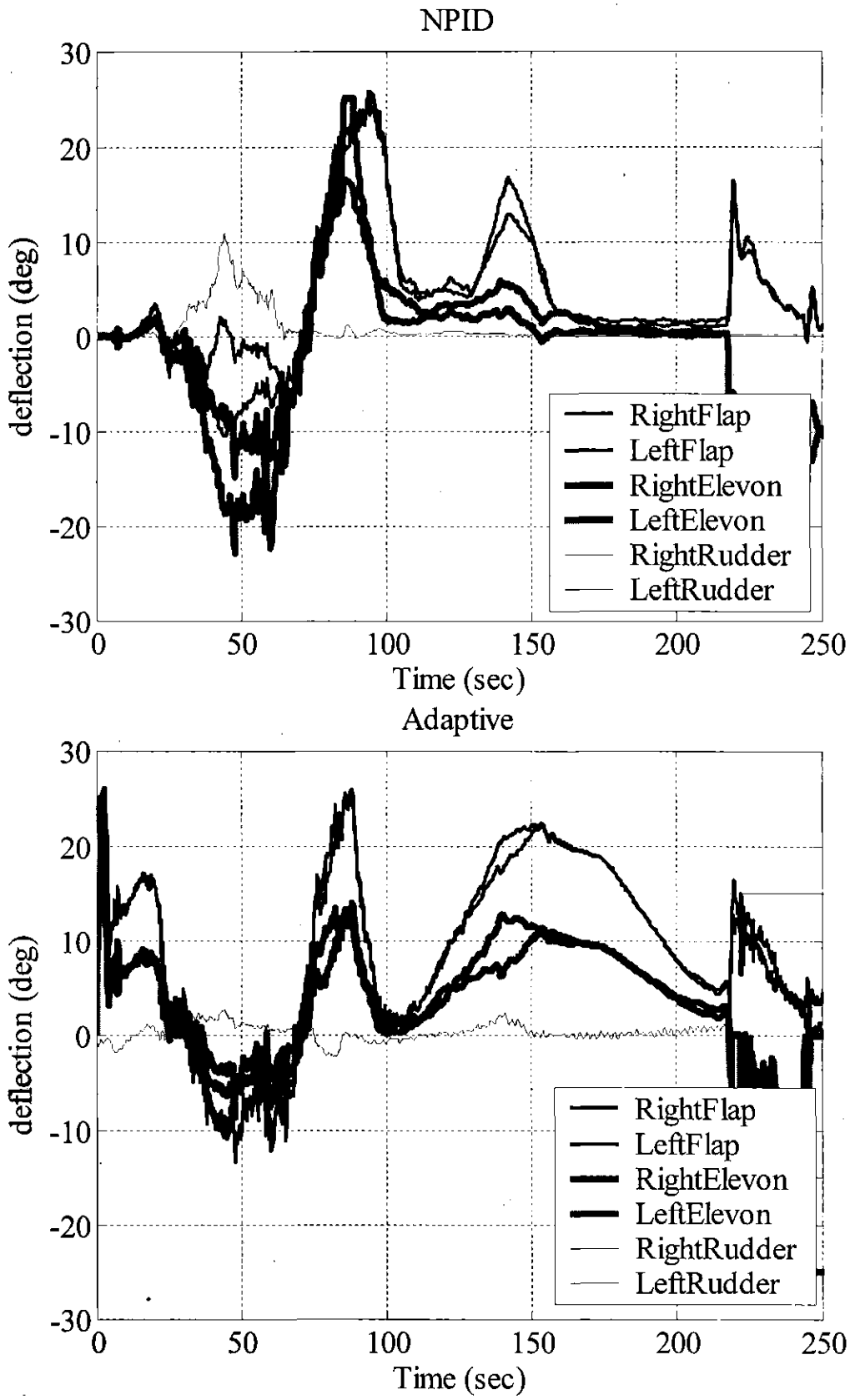

Figure 10: Actuator Commands in Ascent

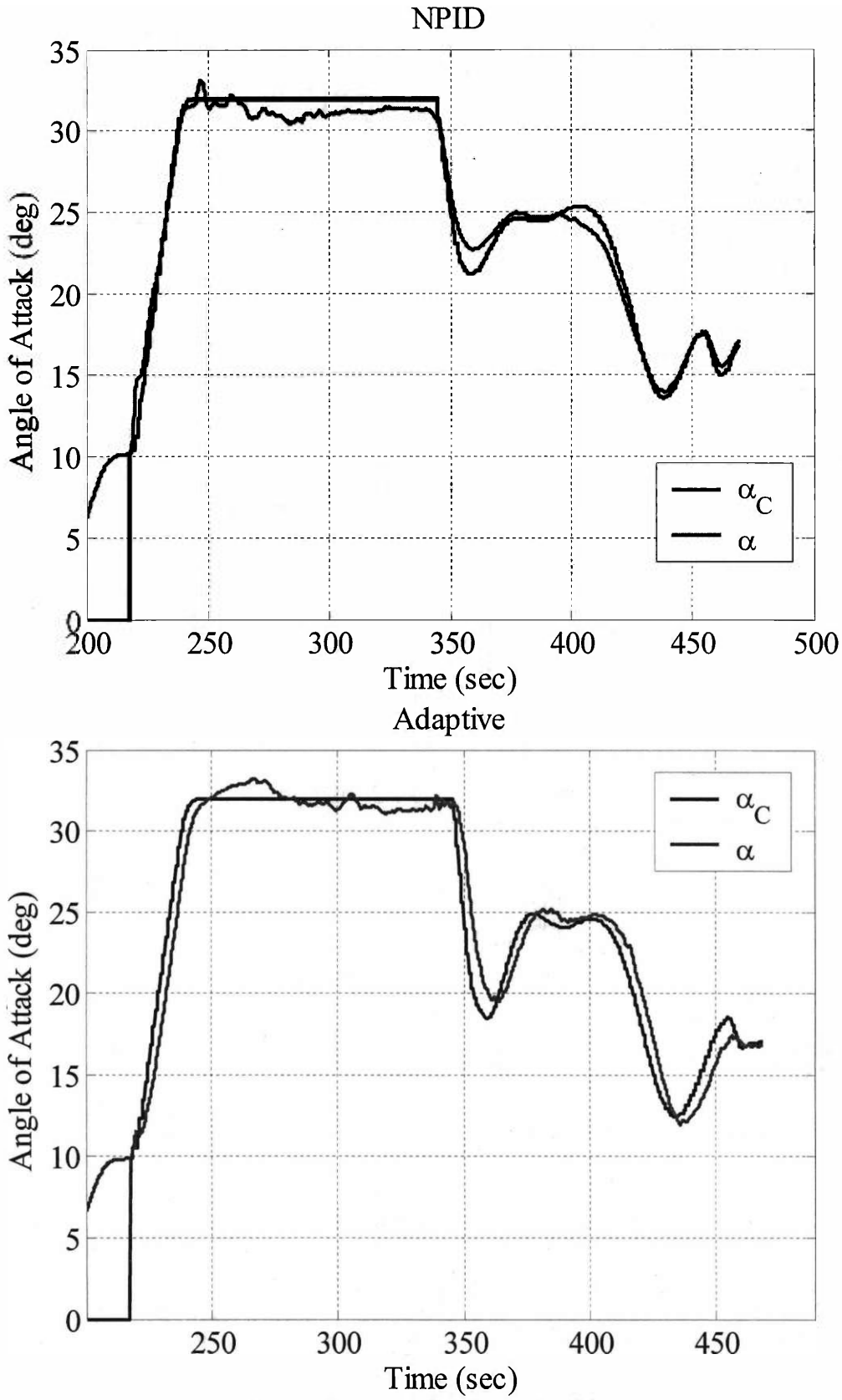

Figure 11: Alpha response in Entry

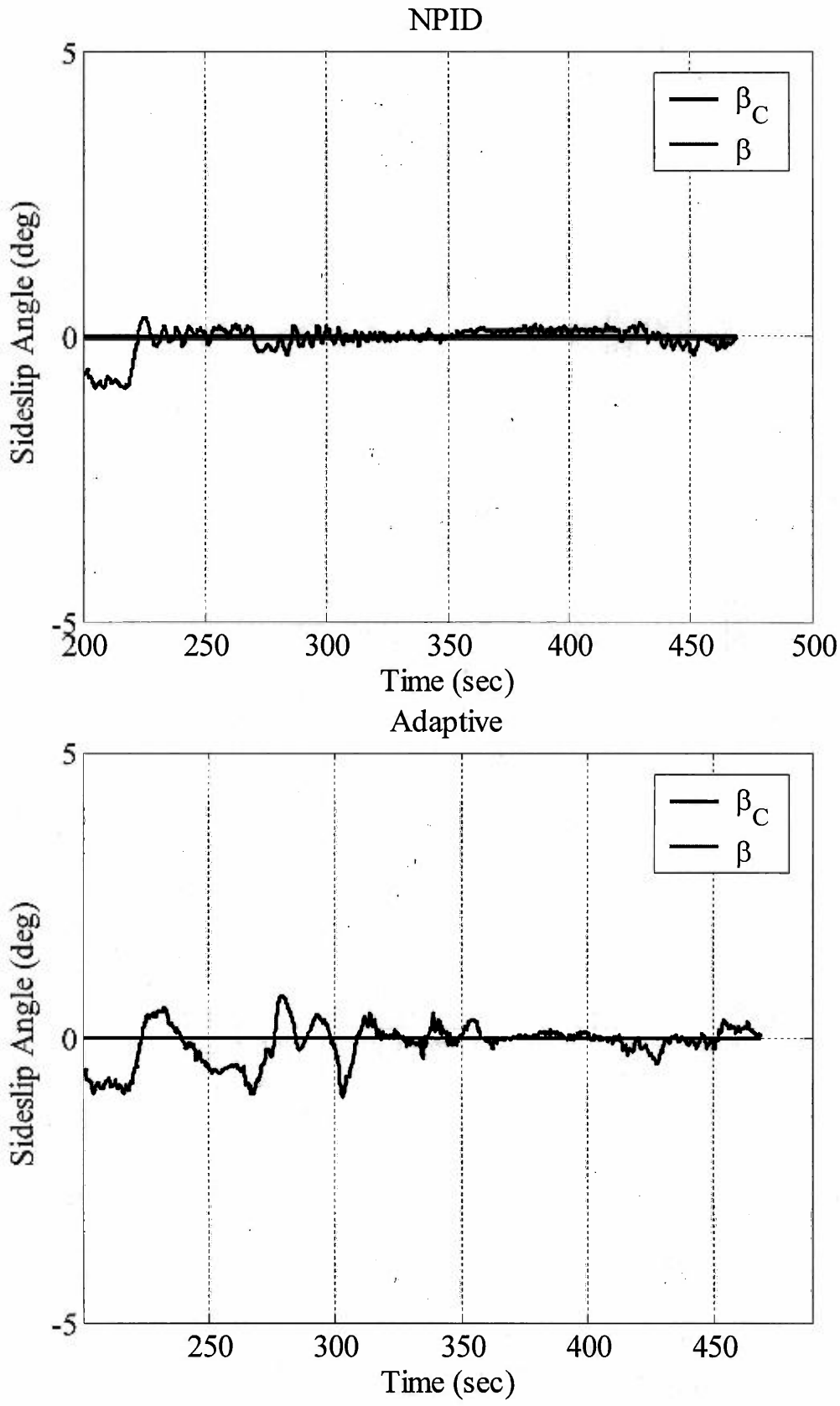

Figure 12: Beta response in Entry

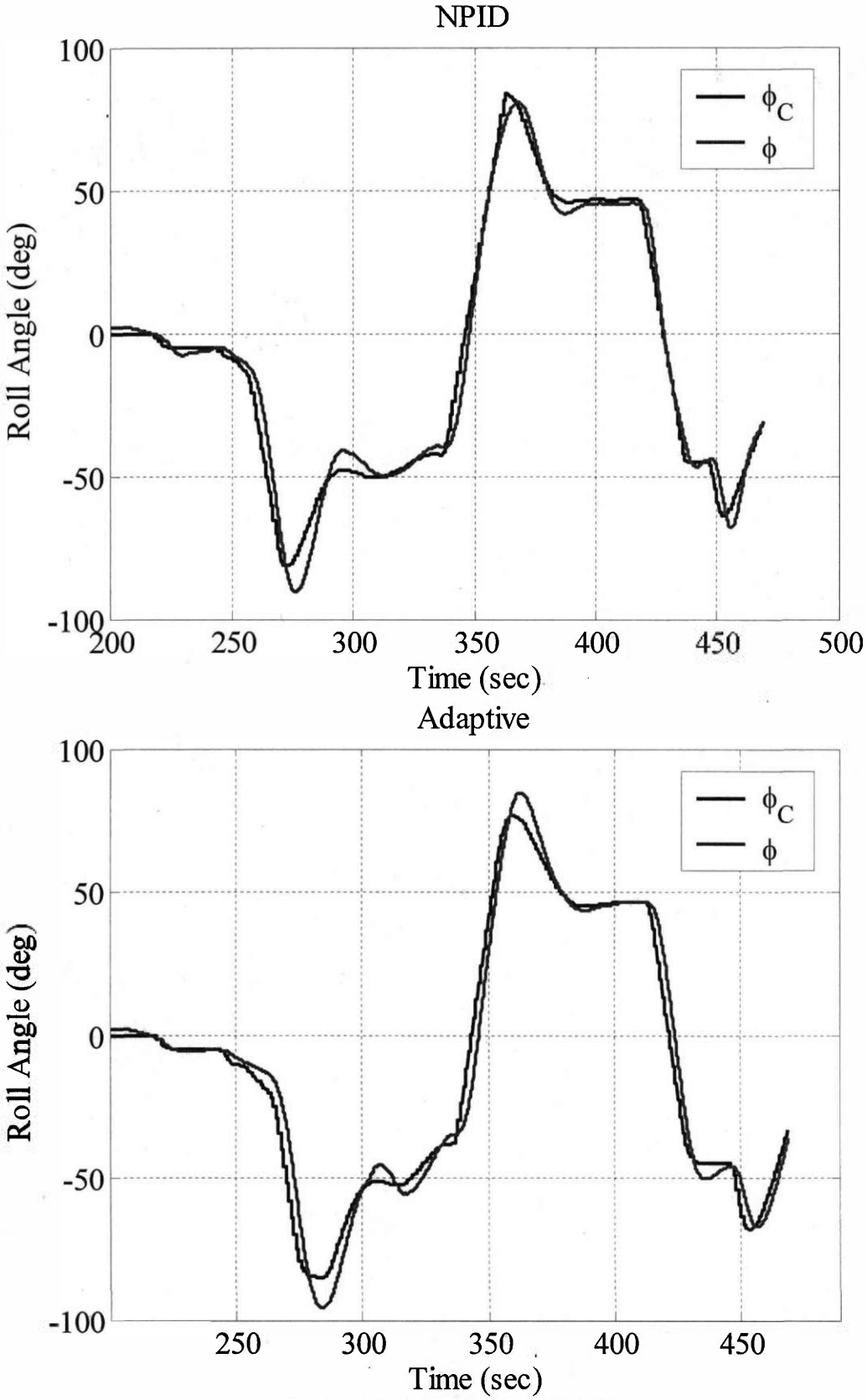

Figure 13: Phi response in Entry

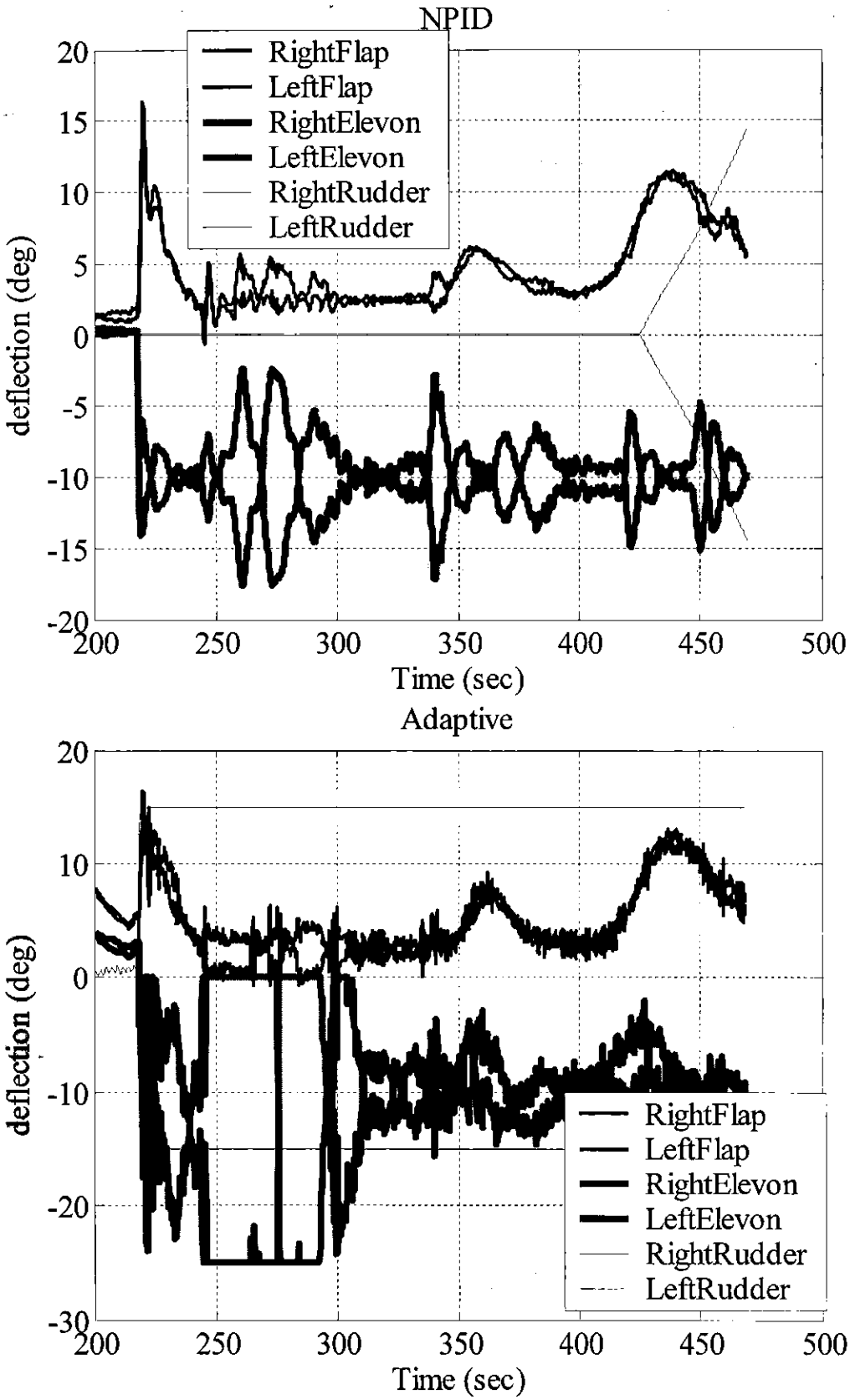

Figure 14: Actuator Commands in Entry

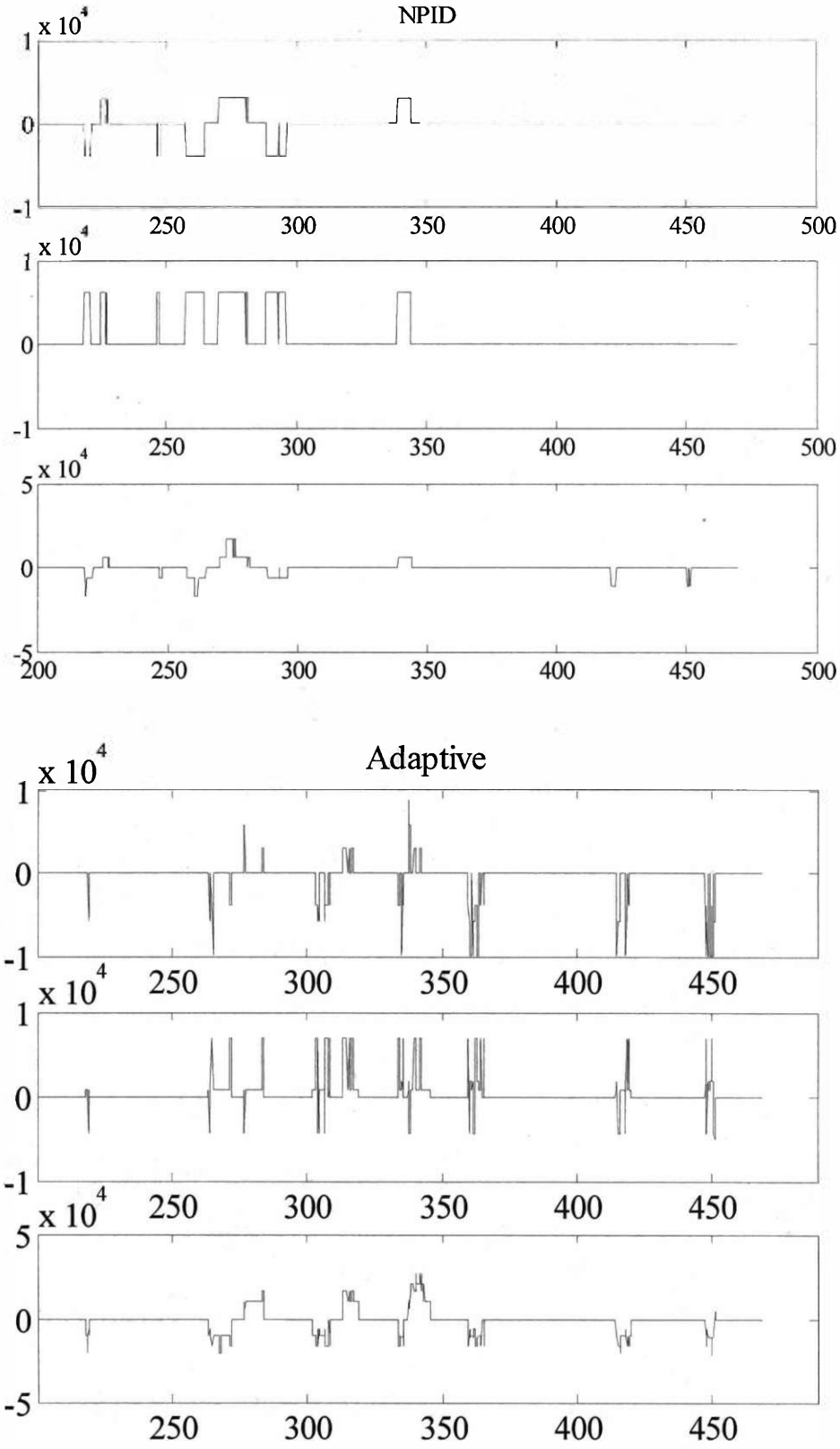

Figure 15: RCS Moments in Entry

IV.4 Test Case 44

Test case 44 is a dispersion case where there is a Power Pack Out (PPO) failure at 38 seconds, which results in an abort of the mission to an alternate landing site. Figures 16-19 show attitude errors, differential throttle commands, actuator time histories, and dynamic pressures for seed 46 of test case 44. For this case, neither the adaptive algorithm nor the NASA PID (NPID) survives past the ascent phase for any of the seeds examined. The figures show consistency in the performance of the first adaptive algorithm, which fails to track in pitch after the PPO, and quits running at about 120-140 seconds. The flaps typically saturate for a short period while attempting to track, while the elevons never saturate. The performance of the NPID at first glance roughly matches that of the adaptive algorithm. An examination of nine individual seeds (45-53) show that simply tabulating a score of whether the adaptive algorithm or the NPID lasts longer shows about three ties (seeds 49, 50, 51) three adaptive wins (seeds 45, 46, 52) and three NPID wins (seeds 47, 48, 53). Because the performance of the adaptive system is quite consistent, more can be learned by examining how the NPID performed. In cases where the NPID performs worse than the adaptive system, if primarily does so because it spins out of control early. In cases where it performs better, it does so by saturating both the elevons and flaps, and the algorithm manages to begin to regulate the pitch error. Towards the end of ascent, then, the vehicle spins out of control.

Judging by the 'most successful' results from the NPID, it seemed that the primary reason why the adaptive algorithm was doing such a poor tracking job was that it was failing to saturate the elevons to drive down the pitch error. Four changes were made to tune the ascent architecture. The first can be seen in Figure 3. A limit is set on the commanded angular rate in the reference model This appears in the limit block and the block called Filter Limit, which is simply a magnitude limited first-order filter. This limit was previously set for each axis at [0.05 0.05 0.05] rad/sec. This value was increased to [0.10 0.20 0.10] rad/sec. In addition, the bandwidth of the linear controller and reference models were increased from [0.8 0.8 0.8] rad/sec to [1.0 1.0 1.0] rad/sec. Finally, the values in the allocation matrix of differential thrust to pitch and flap to pitch were both cut approximately in half. Results after making these changes are labeled as Adaptive2 and are shown in Figures 16-19.

The performance of the control system after the changes shows some definite improvement. For instance, after the initial pitch error at the point where the abort trajectory is implemented, the new system successfully returns to zero quickly in nearly every case. The actuator time histories reveal that the algorithm is now saturating both the elevons as well as the flaps in order to remove the pitch error. Comparison of the two results shows that only for the seed 48 case does the previous adaptive control system emerge the winner, by the 'time elapsed until failure' criterion. The two systems roughly tie for case 45, 49, 50, 51, 53, and for the majority of those the newer version lasts slightly longer. For cases 46, 47, and 52, Adaptive2 survives the entire ascent phase, and reaches transition before the algorithm fails. The NPID algorithm does not survive the ascent phase in any of the seeds examined.

Full saturation of the aerodynamic surfaces is achieved for the period of ascent for which the vehicle fails to track the pitch attitude command. It seems unlikely that better tracking can be achieved for this case absent a new trajectory, which may require aborting to a different landing site or an overhaul of the guidance algorithm.

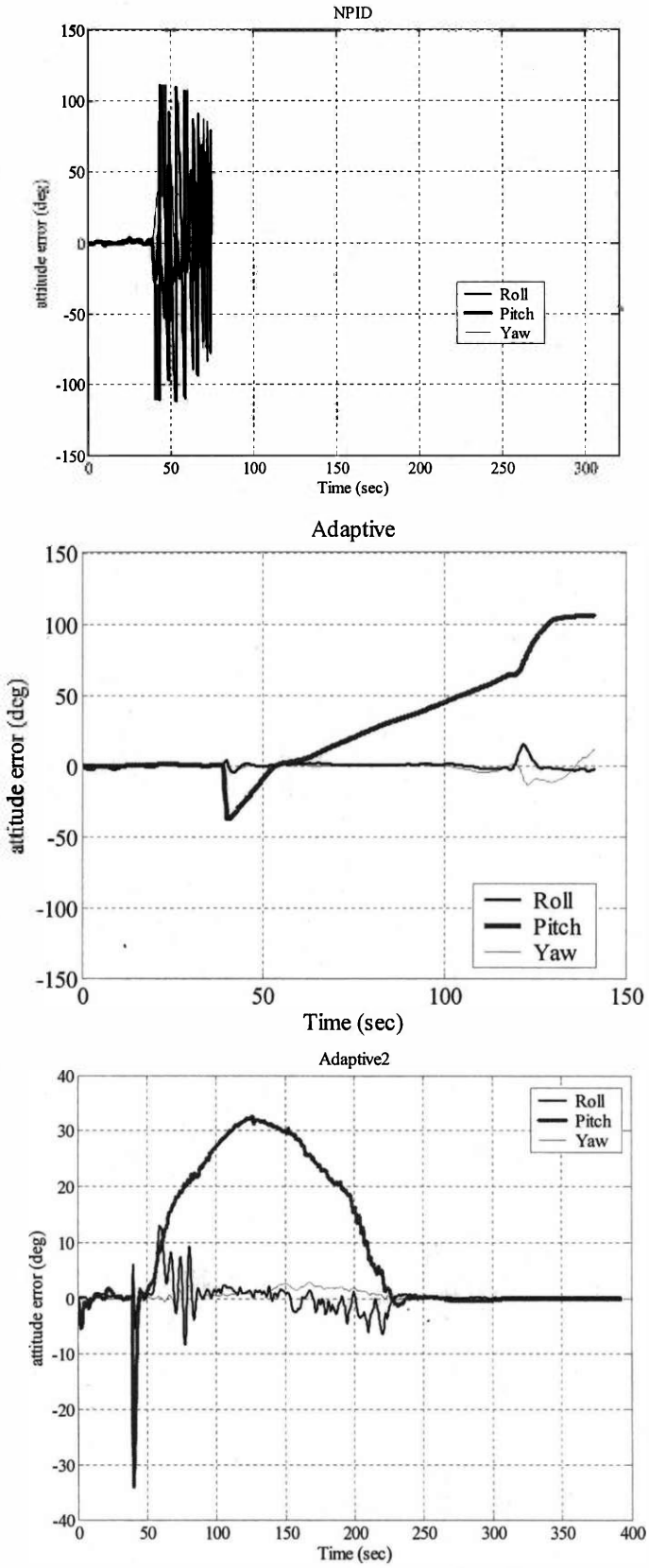

Figure 16. Attitude Errors (Case 44, Seed 46)

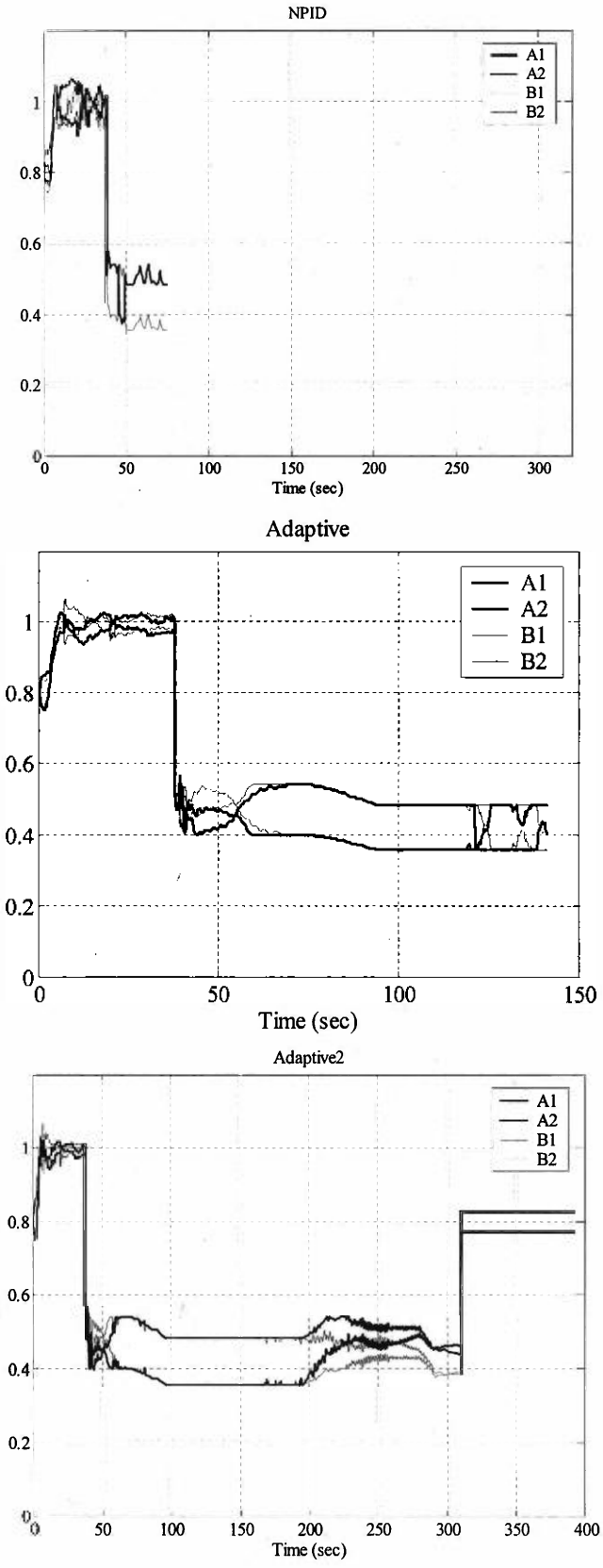

Figure 17. Throttle Deflections (Case 44, Seed 46)

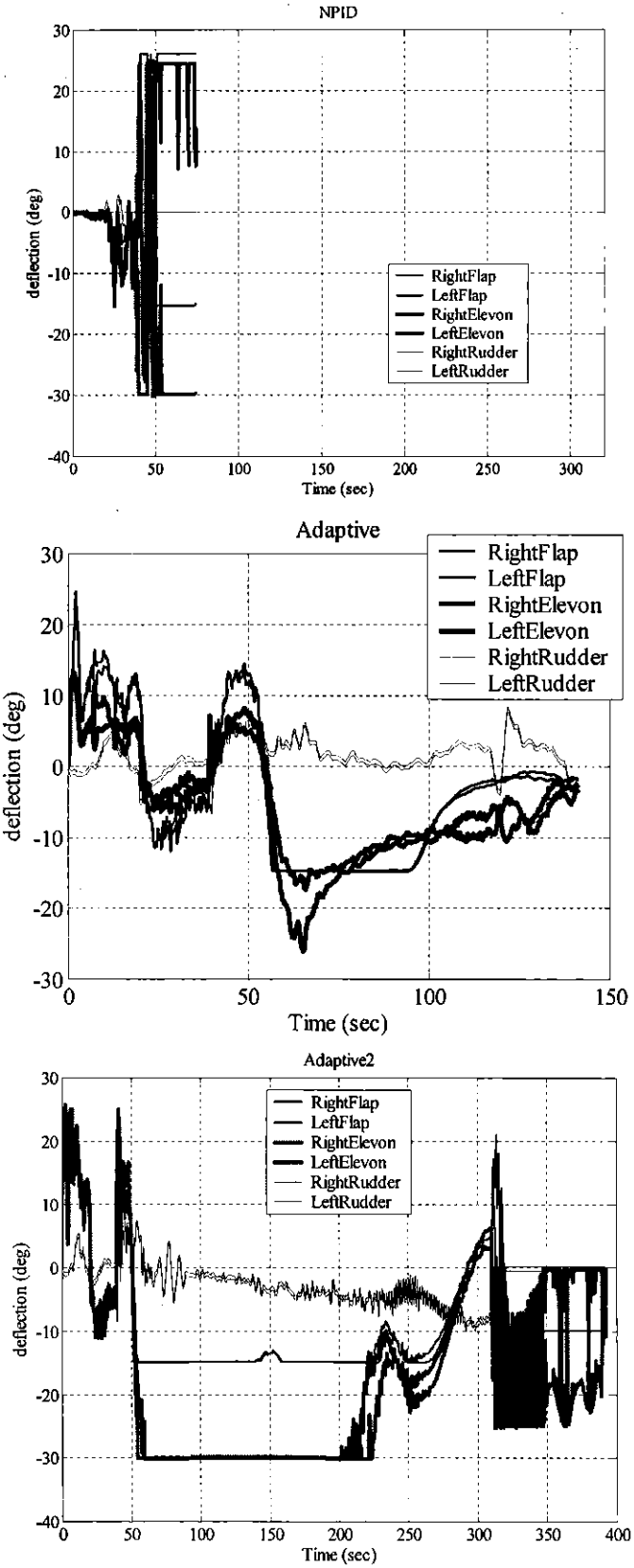

Figure 18. Actuator Commands (Case 44, Seed 46)

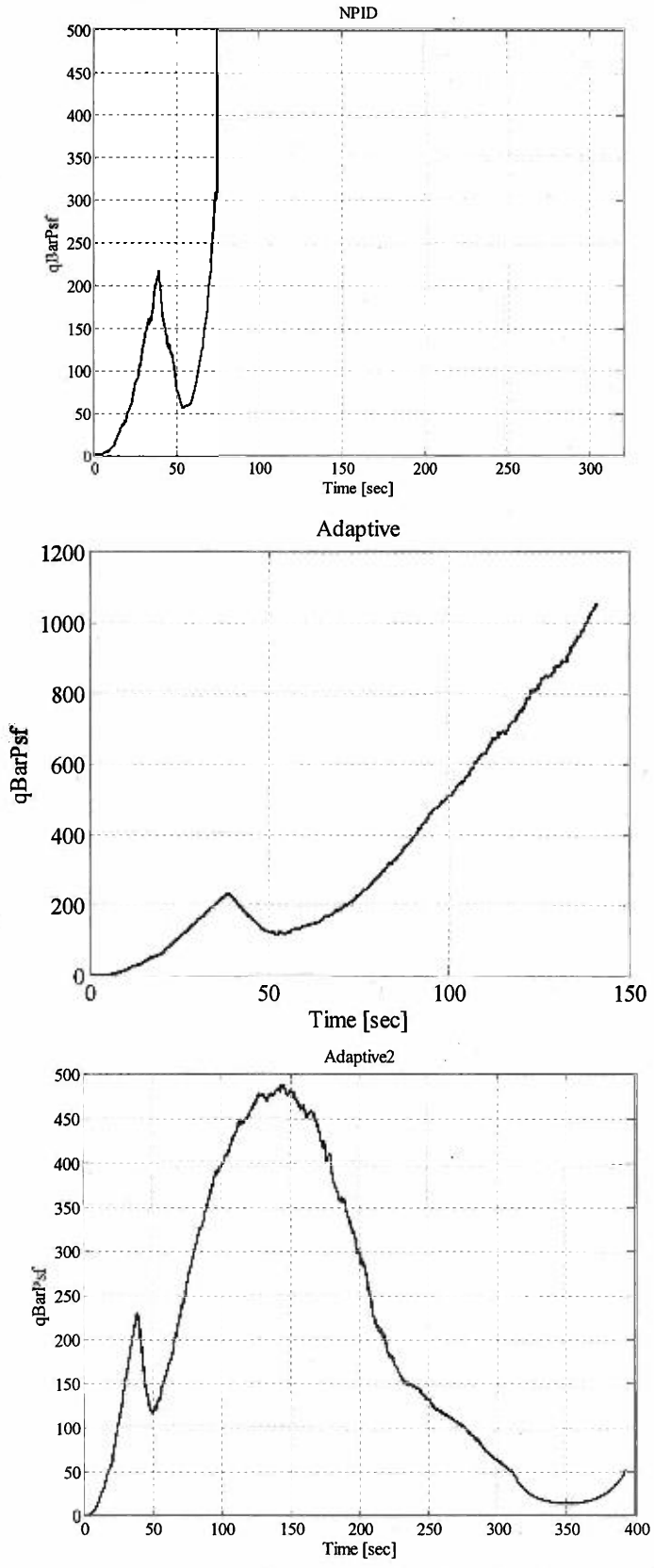

Figure 19. Dynamic Pressure (Case 44, Seed 46)

IV.5 Test Case 43

Test case 43 is a dispersion case where there is a PPO failure at 40 seconds, which results in an abort of the mission to an alternate landing site. Table 2 shows the pass/fail results for 9 seeds for both the adaptive and the NPID algorithms. The three TAEM criteria included altitude, heading angle, and range. The TAEM criteria results and the total time elapsed in the run are shown. There does not appear to be a clear winner according to these criteria, so a closer look was taken at some individual test case runs. The number of figures required to look closely at multiple runs for both ascent and entry is large. So, for this report, we will focus on seed 44 of test case 43. The comparison shown in Table 2 is for the adaptive algorithm prior to the changes discussed in Section IV.4. As before, the NPID algorithm tracks somewhat better in ascent, but the most noticeable differences are in entry. Neither the angle of attack, sideslip angle, nor roll angle commands are tracked well by the Adaptive controller, especially at low dynamic pressures. Examining the actuator command plots can explain why. At low dynamic pressures, the NPID algorithm uses approximately 25 degrees of differential elevon, which results in a large amount of roll authority. The adaptive algorithm has very little differential elevon usage, partly because there is an artificial limit (shown in Figure) used for allocation, and partly because the gain in that channel is small.

It had been suspected earlier that these limits might need to be relaxed, but early efforts to do so had resulted in many more instances of failures in a dispersion run. A second attempt to do so was made, during which the nature of the failure was more closely examined. It became clear that the large amount of differential actuator (elevon and flap) was resulting in departure at high dynamic pressures while showing the expected good tracking results at the low dynamic pressures. Thus the allocation scheme presented in Figure 20 was implemented, resulting in a set of large differential limits at low dynamic pressures and a set of small differential limits at high dynamic pressures.

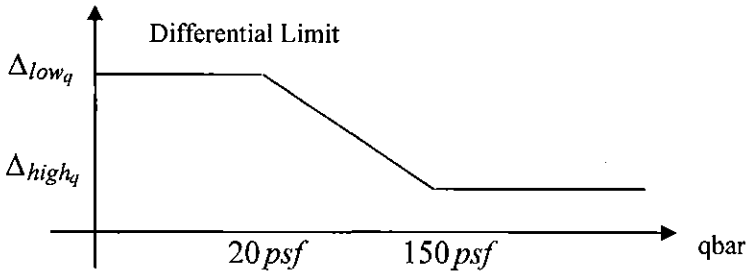

Figure 20. Typical allocation scheduling diagram

Table 1 shows the values chosen for the differential limits on the elevons and flaps for the lateral modes after tuning the algorithm.

Table 1. Differential Limits on the Actuators

Actuator	Mode	Δ_{low_q}	Δ_{high_q}
Elevon	Roll	20°	10°
Elevon	Yaw	15°	7.5°
Flap	Roll	5°	2.5°
Flap	Yaw	4°	2.0°

Table 3 shows the pass/fail results for the same 9 seeds for both before and after implementing the changes to the ascent allocation scheme, and the allocation schedule of Figure 20. It can be seen that with the changes, the algorithm passes more of the cases and gets nearer to passing the cases when it fails, yielding partial scores. Figures 21-30 also show the results for the three architectures on test case 43.

Table 2. Case 43 results for Adaptive and NPID

Pass/Fail Criteria:

Alt 96,000±7000ft Heading ±10° Range 30±7nm

Seed 44	Adaptive (Pass)	NPID (Fail)
Alt (ft.)	95,300	83,900
Heading (deg)	7.9	12.56
Range (nm)	28.6	70.7
Time (sec)	518	447

Seed 45	Adaptive (Fail)	NPID (Fail)
Alt (ft.)	103,800	103,000
Heading (deg)	-2.4	-1.8
Range (nm)	61.9	40.8
Time (sec)	495	519

Seed 46	Adaptive (Fail)	NPID (Pass)
Alt (ft.)	109,000	95,900
Heading (deg)	1.2	5.4
Range (nm)	44.7	26.7
Time (sec)	512	528

Seed 47	Adaptive (Pass)	NPID (Fail)
Alt (ft.)	101,200	102,200
Heading (deg)	-.8	-0.8
Range (nm)	31.9	37.8
Time (sec)	523	519.3

Seed 48	Adaptive (Pass)	NPID (Pass)
Alt (ft.)	101,900	96,400
Heading (deg)	-1.0	5.4
Range (nm)	32.7	29.8
Time (sec)	520	513

Seed 49	Adaptive (Pass)	NPID (Fail)
Alt (ft.)	96,800	100,000
Heading (deg)	4.6	13.2
Range (nm)	28.9	33.4
Time (sec)	516	506

Seed 50	Adaptive (Pass)	NPID (Pass)
Alt (ft.)	102,200	100,700
Heading (deg)	-1.2	-0.7
Range (nm)	61.2	33.5
Time (sec)	490	519

Seed 51	Adaptive (Fail)	NPID (Fail)
Alt (ft.)	93,200	87,400
Heading (deg)	-2.5	19.1
Range (nm)	91.9	71.7
Time (sec)	442	438

Seed 52	Adaptive (Fail)	NPID (Pass)
Alt (ft.)	83,200	96,400
Heading (deg)	-13.8	5.6
Range (nm)	103.4	27.1
Time (sec)	432	529

Table 3. Case 43 results for Adaptive and Adaptive2

Pass/Fail Criteria:
Alt 96,000±7000ft Heading ±10° Range 30±7nm

Seed 44	Adaptive (Pass)	Adaptive2 (Pass)
Alt (ft.)	95,300	95979
Heading (deg)	7.9	5.4
Range (nm)	28.6	27.9
Time (sec)	518	521
Seed 45	**Adaptive (Fail)**	**Adaptive2 (Fail)**
Alt (ft.)	103,800	103,800
Heading (deg)	-2.4	-2.4
Range (nm)	61.9	51.4
Time (sec)	495	506
Seed 46	**Adaptive (Fail)**	**Adaptive 2 (Pass)**
Alt (ft.)	109,000	102,700
Heading (deg)	1.2	-0.2
Range (nm)	44.7	28.0
Time (sec)	512	533
Seed 47	**Adaptive (Pass)**	**Adaptive 2 (Pass)**
Alt (ft.)	101,200	102,300
Heading (deg)	-.8	-0.8
Range (nm)	31.9	34.2
Time (sec)	523	522
Seed 48	**Adaptive (Pass)**	**Adaptive 2 (Pass)**
Alt (ft.)	101,900	101,100
Heading (deg)	-1.0	-0.3
Range (nm)	32.7	30.6
Time (sec)	520	522
Seed 49	**Adaptive (Pass)**	**Adaptive 2 (Pass)**
Alt (ft.)	96,800	94,700
Heading (deg)	4.6	4.4
Range (nm)	28.9	28.4
Time (sec)	516	514
Seed 50	**Adaptive (Fail)**	**Adaptive 2 (Fail)**
Alt (ft.)	102,200	103,300
Heading (deg)	-1.2	-1.3
Range (nm)	61.2	46.3
Time (sec)	490	504
Seed 51	**Adaptive (Fail)**	**Adaptive 2 (Fail)**
Alt (ft.)	93,200	102,000
Heading (deg)	-2.5	-0.1
Range (nm)	91.9	58.1
Time (sec)	442	496
Seed 52	**Adaptive (Fail)**	**Adaptive 2 (Fail)**
Alt (ft.)	83,200	104,300
Heading (deg)	-13.8	-0.2
Range (nm)	103.4	37.6
Time (sec)	432	520

It's clear that tracking is improved both in ascent as well as entry when employing the Adaptive2 design. Figures 24 - 26 show that the large oscillations that appear in the Adaptive entry response at low dynamic pressures are now mostly eliminated in the Adaptive2 response. The NPID baseline fails 15/200 seeds for Test Case 43. Adaptive and Adaptive2 failed 111/200 and 34/100 seeds, respectively.

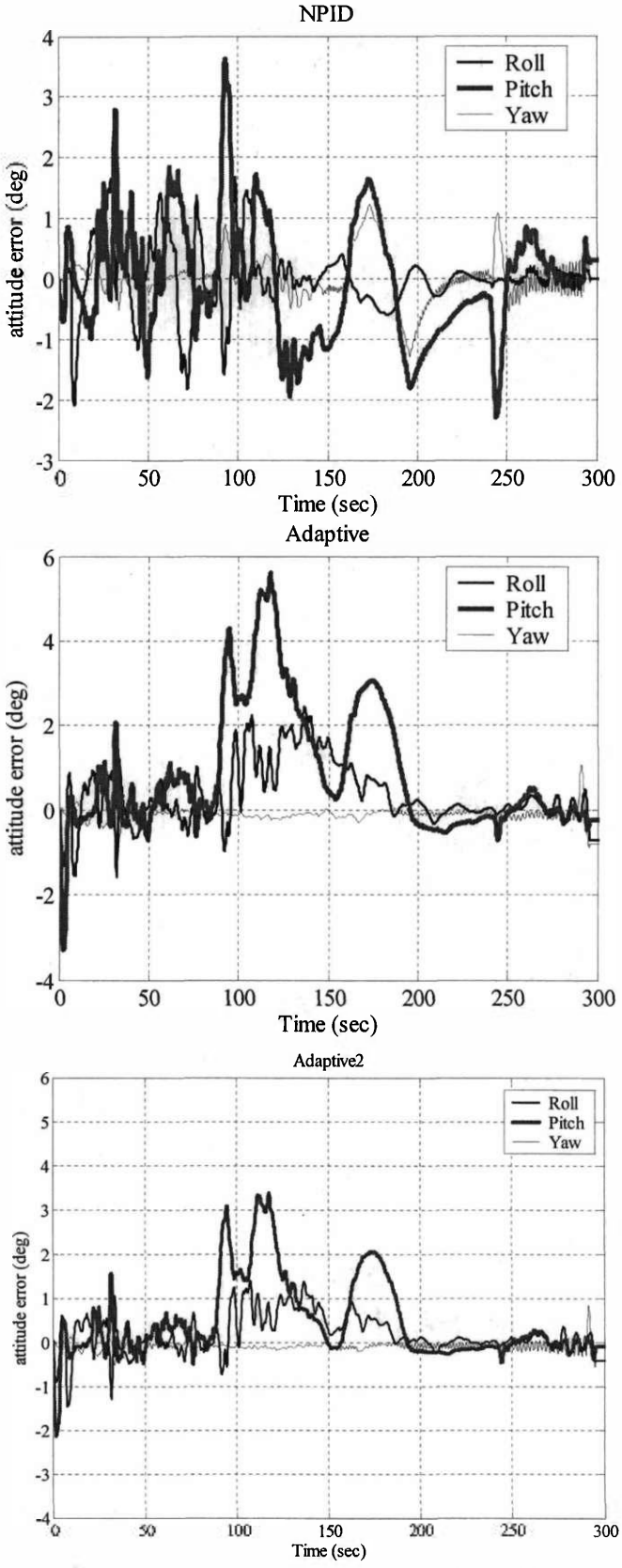

Figure 21. Attitude Errors in Ascent (Case 43, Seed 44)

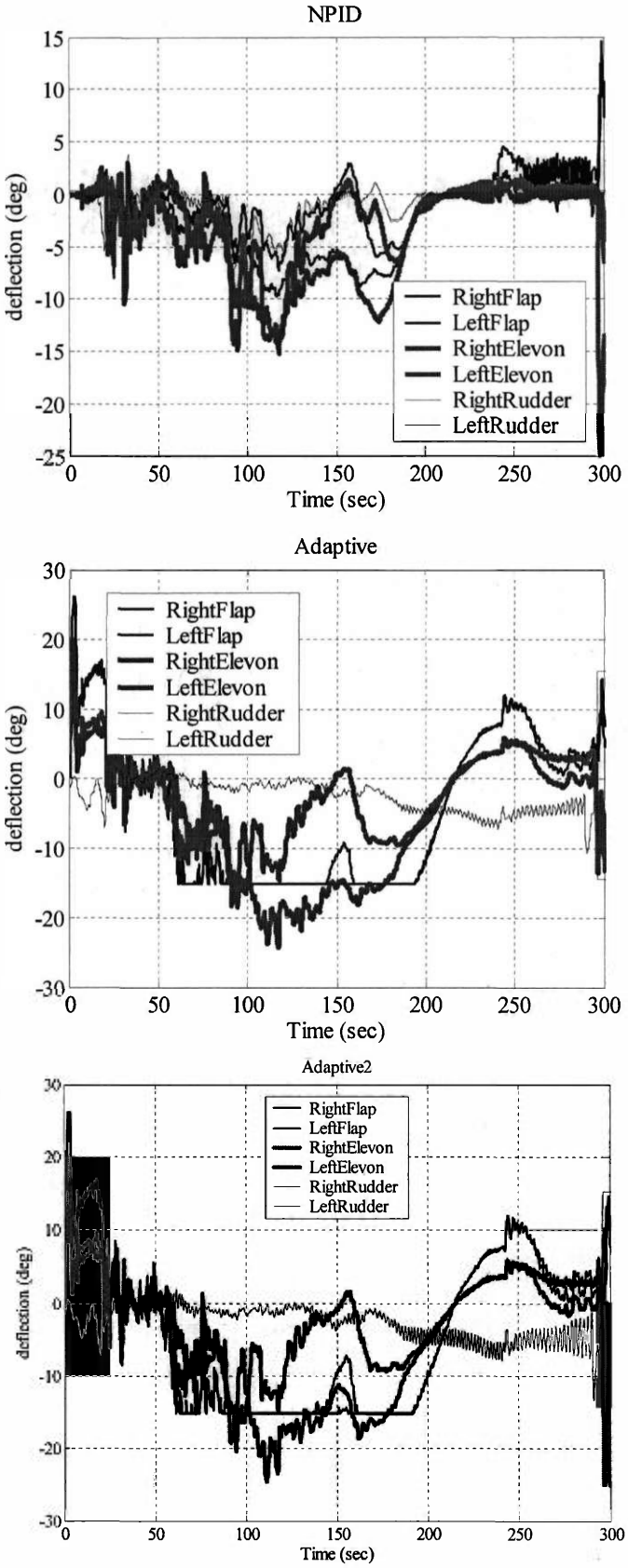

Figure 22. Actuator Commands in Ascent (Case 43, Seed 44)

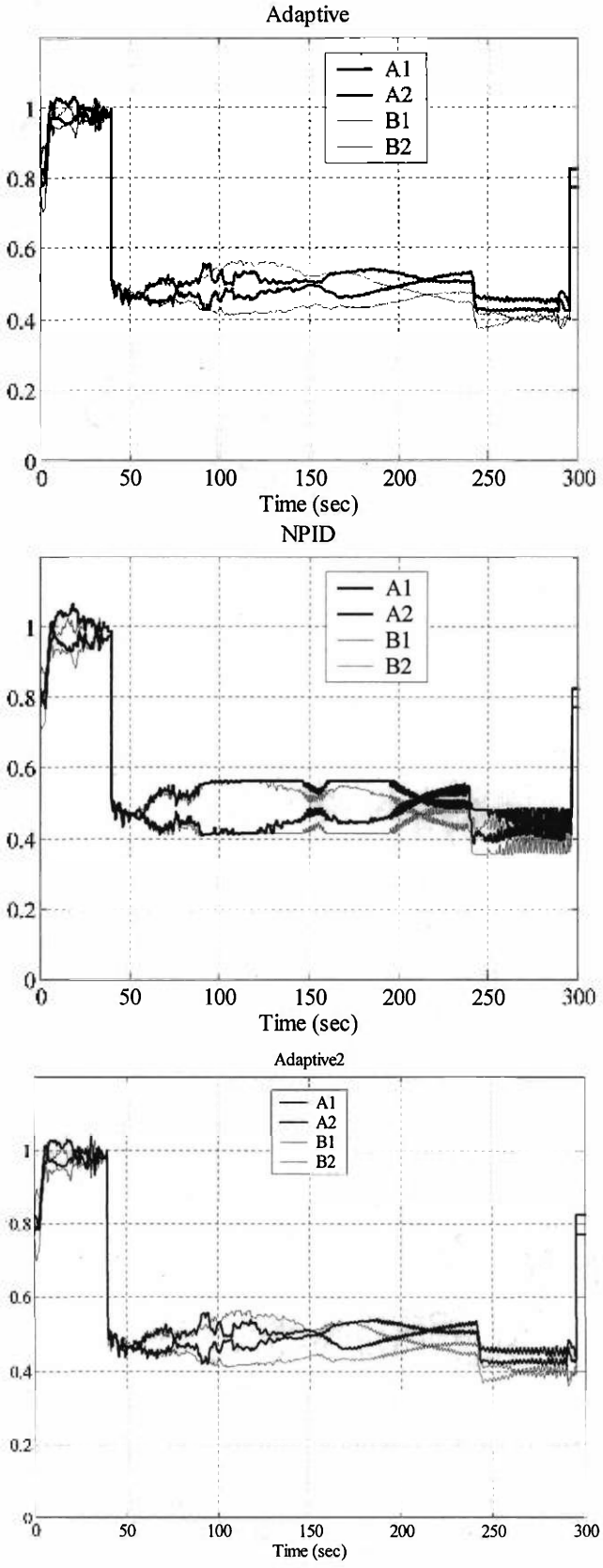

Figure 23. Throttle Commands in Ascent (Case 43, Seed 44)

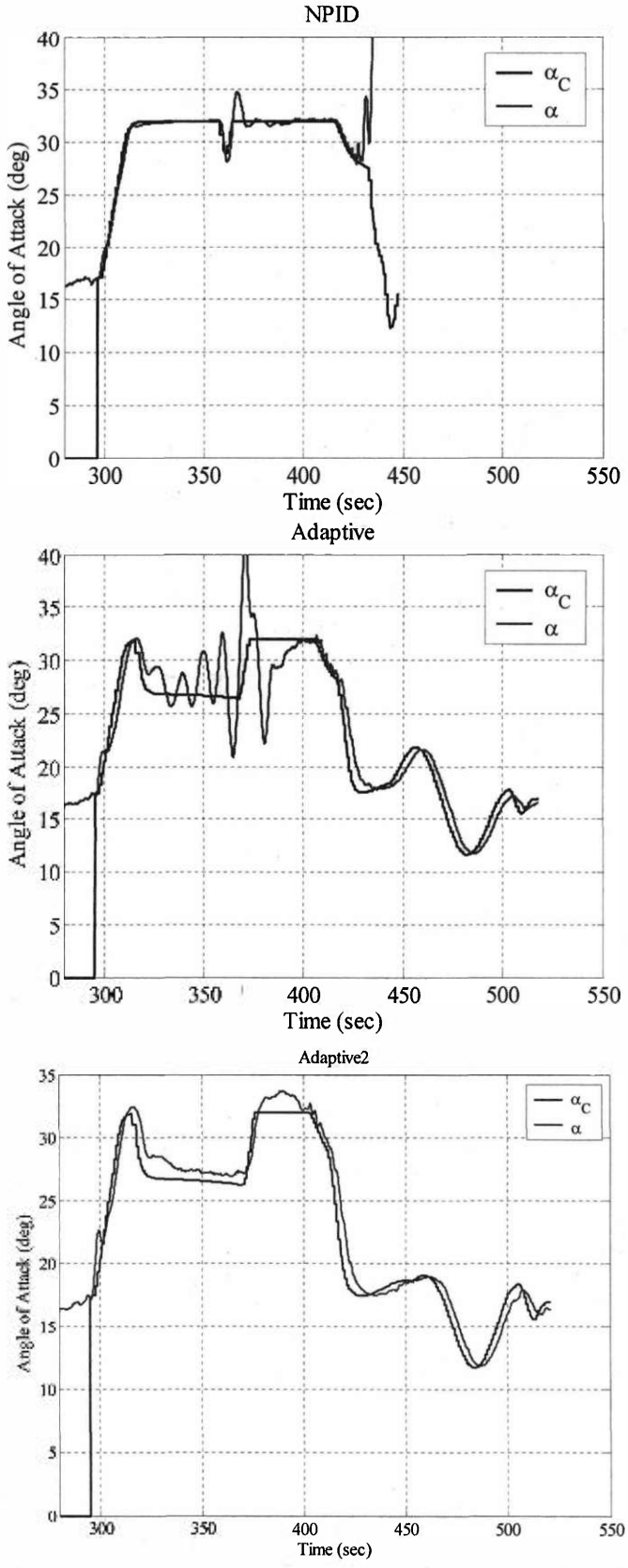

Figure 24. Alpha response in Entry (Case 43, Seed 44)

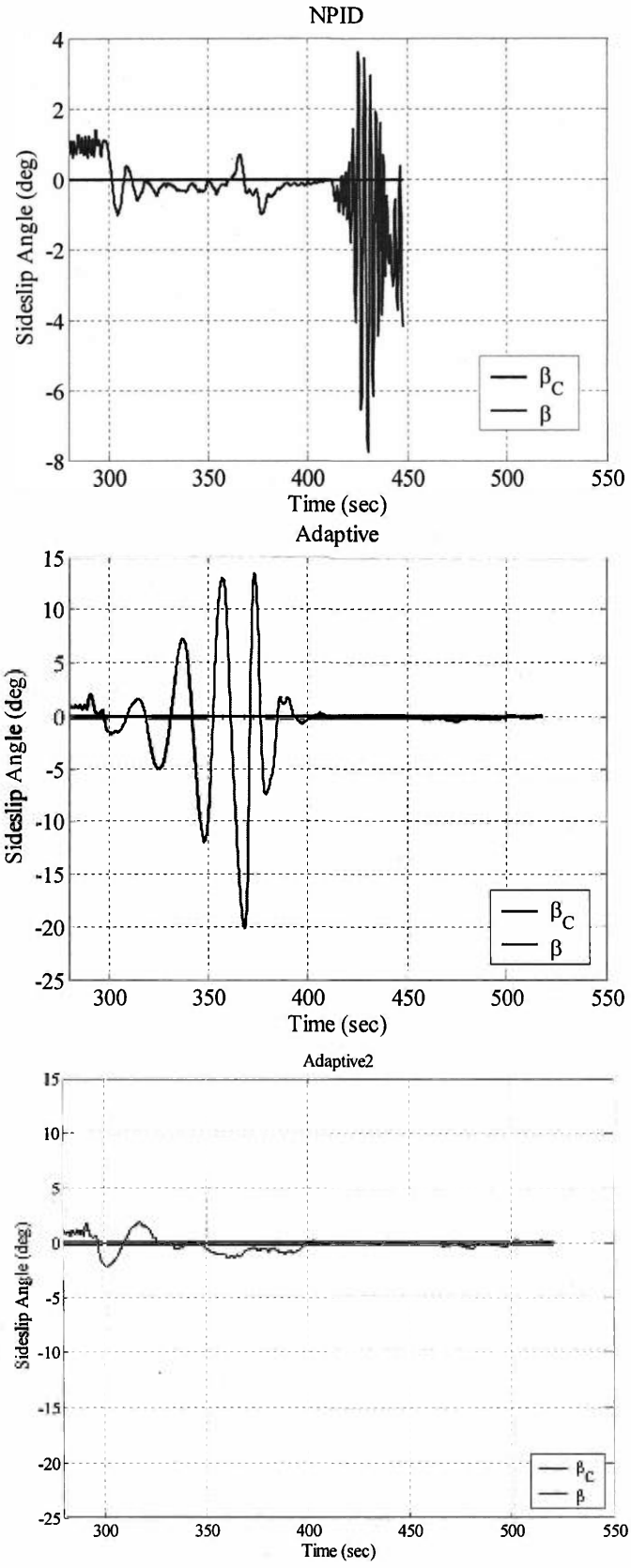

Figure 25. Beta response in Entry (Case 43, Seed 44)

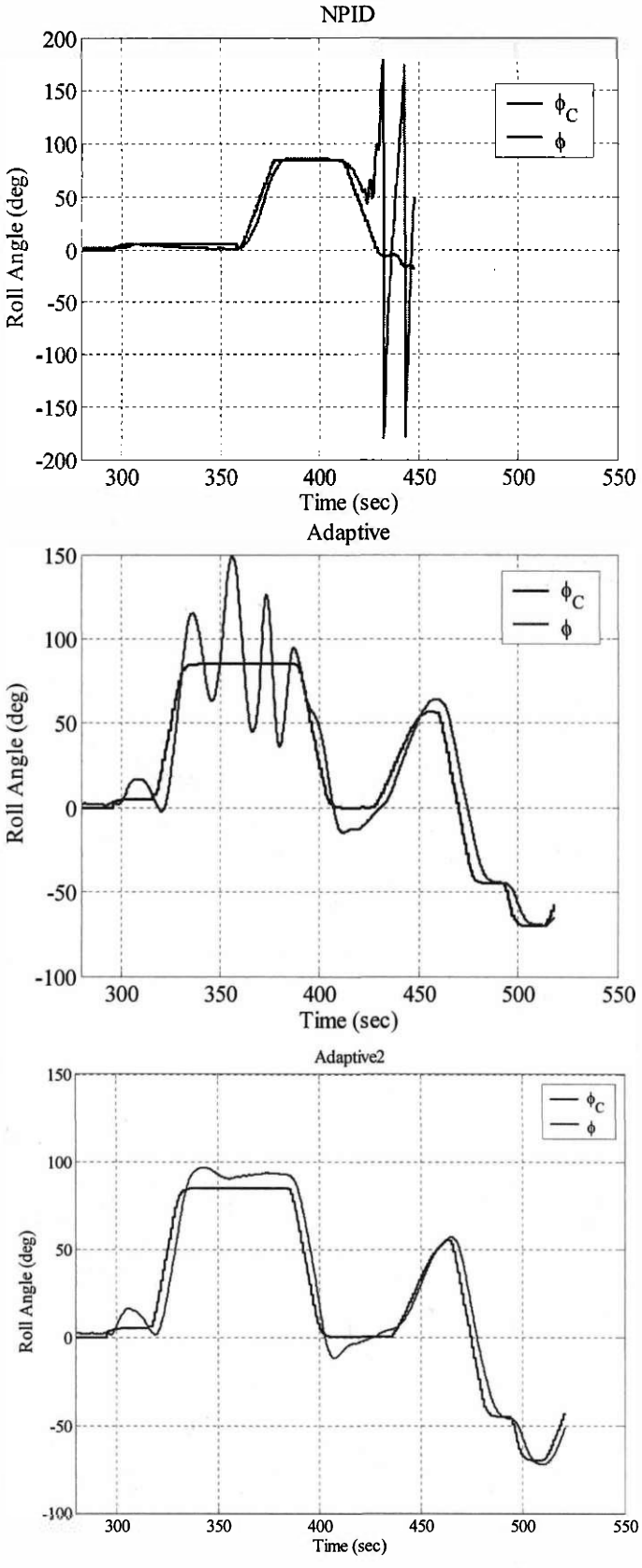

Figure 26. Phi response in Entry (Case 43, Seed 44)

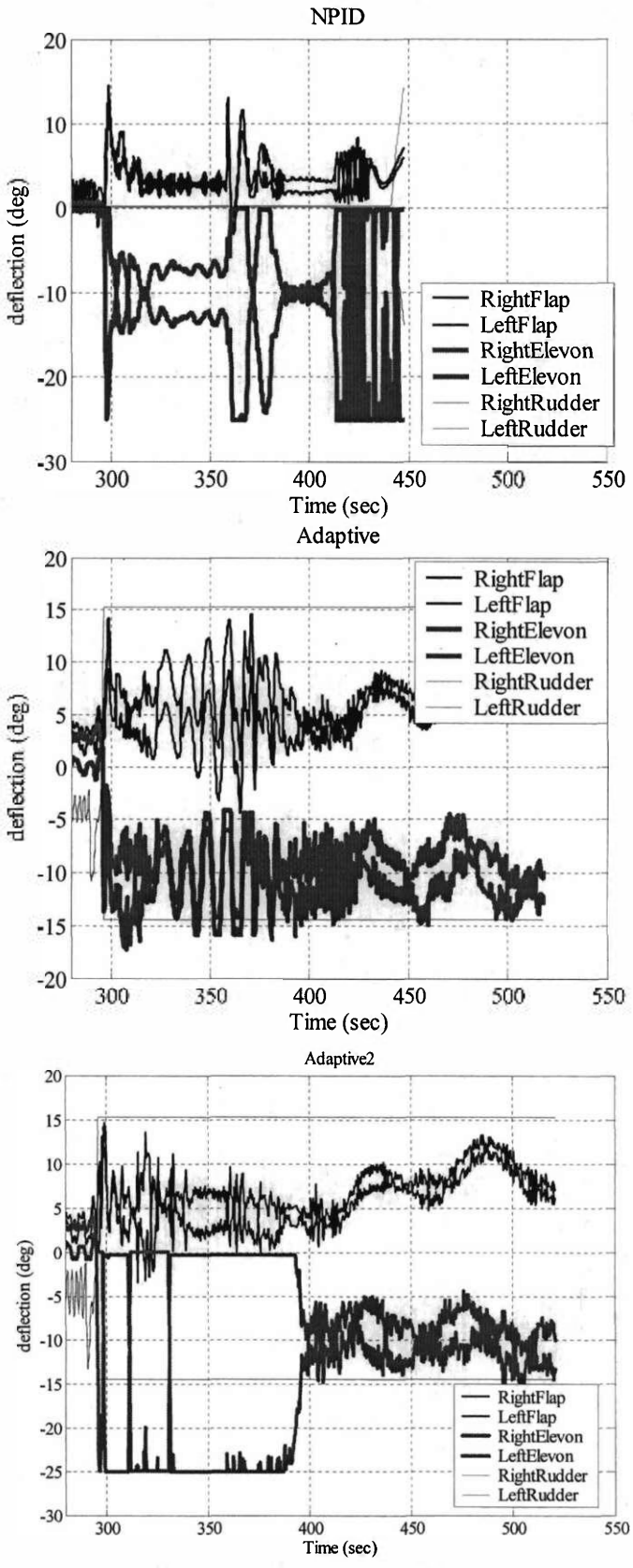

Figure 27. Actuator Commands in Entry (Case 43, Seed 44)

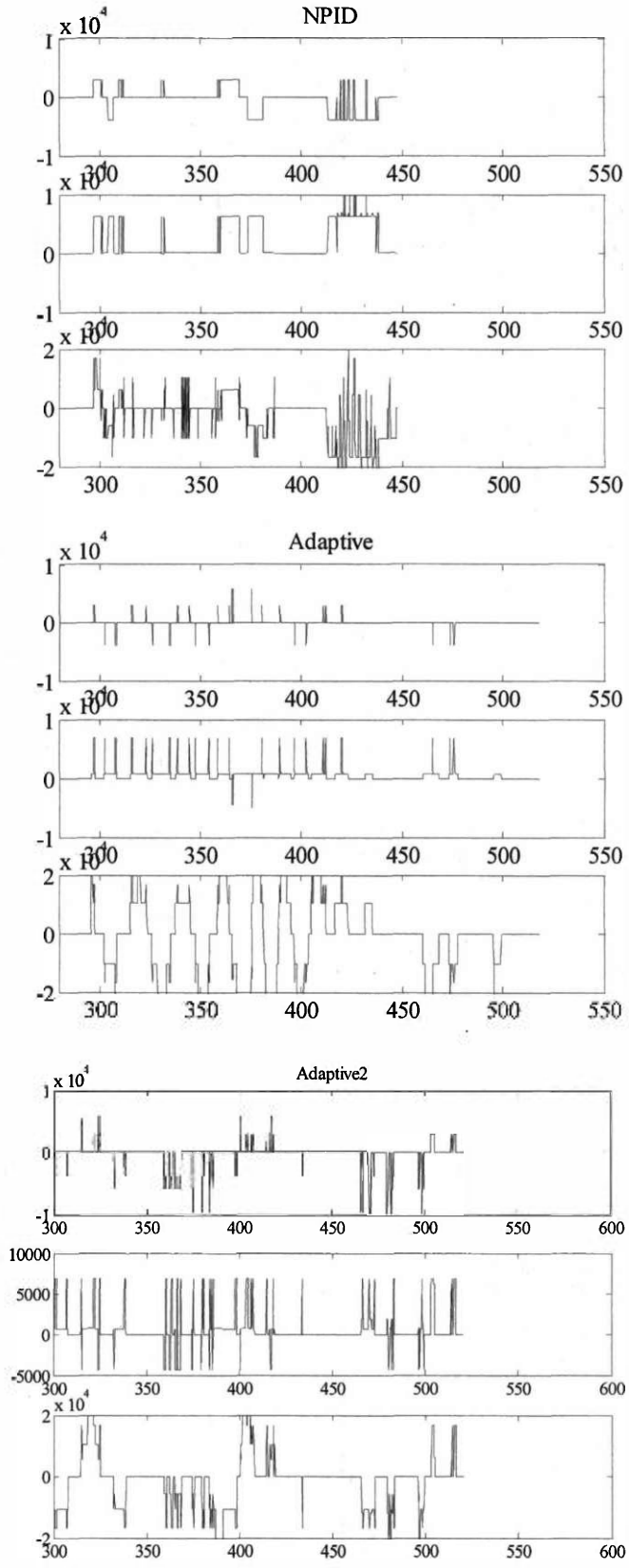

Figure 28. RCS Moments in Entry (Case 43, Seed 44)

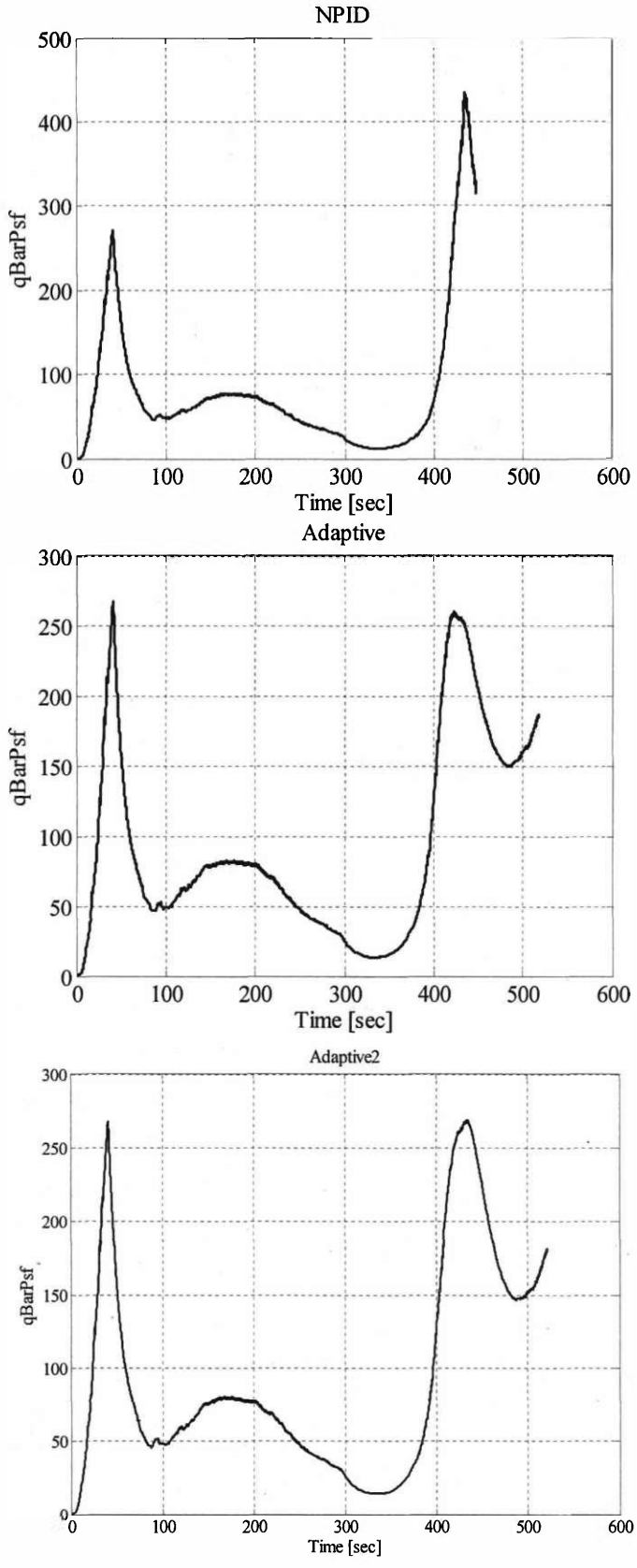

Figure 29. Dynamic Pressure (Case 43, Seed 44)

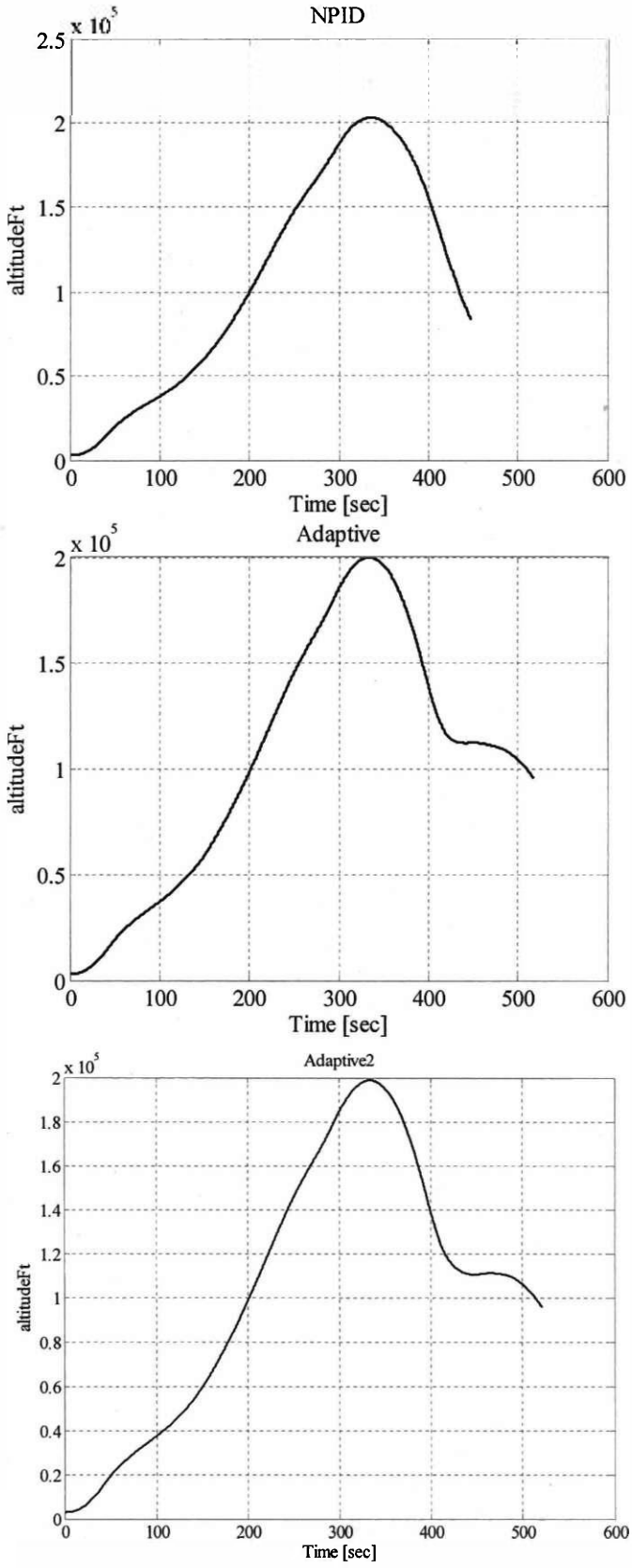

Figure 30. Altitude (Case 43, Seed 44)

IV.6 All Test Cases

Once the changes were made to try to improve test cases 43 and 44, the entire algorithm was run through all the tests again to verify that none of the changes had significantly harmed our scores on the other tests. Figure 31 shows results for the NPID algorithm (labeled NASA PID) and the two versions of the adaptive algorithm (labeled Adaptive and Adaptive 2). Overall it appears that the differences in scoring are small. Test cases 12, 13, 41 and 43 improved, and test cases 2, 14, 23, 25, and 36 slightly worsened between the previous adaptive algorithm and Adaptive2.

The NPID controller scores better on nearly every test for which both it and the adaptive controllers pass. The primary reasons for its higher scores are that it tracks the commands more closely and does so using less actuator energy, mostly because the NPID controls aren't as active. Over the entire test matrix, the adaptive control system scores better than the NPID because it passes more of the tests, reaching the TAEM conditions in cases where the NPID does not.

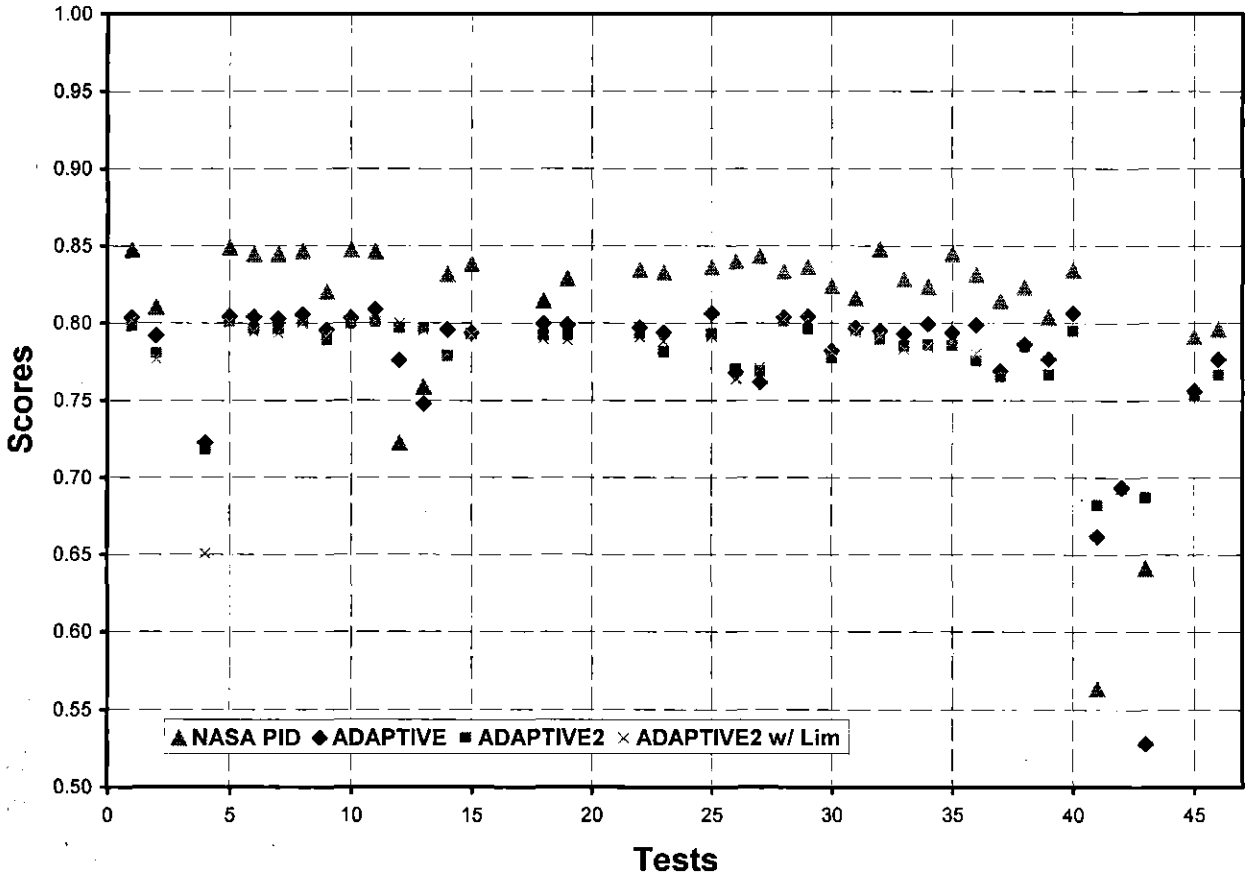

Figure 31. Comparison of test results for each version of the controller.

Table 4 shows a history of test results since November 2002. The test criteria changed in May 2003, which is why the baseline algorithm's scores changed at that time. The May 2003 changes were unfavorable to our algorithm. While we decreased the number of failed tests, the overall score still decreased due to the introduction of more rigorous testing criteria. After implementing the changes of IV.4 and IV.5, the algorithm was re-scored according to NASA's criteria, and the overall score increased due to passing one test more than in May, test case 26.

Table 4. **Final Test Scores**

	November 2002		March 2003		May 2003		November 2003	
	Score	Tests Failed	Score	Tests Failed	Score	Tests Failed	Score	Tests Failed
Baseline	59.9%	12	59.9%	12	64.5%	12	64.5%	12
Adaptive	65.8%	9	66.0%	9	63.6%	7	66.4%	6

V. Conclusion and Future Work

This report details the design of a neural network based adaptive flight controller for NASA's X-33 launch vehicle model. Of particular interest is the design of the reference model, and the implementation of hedging to protect the adaptive process from the effects of actuator limits during a failure. Examples are given to illustrate the manner in which the design can be improved for a several failure scenarios, by modifying aerodynamic control allocation in entry and relaxing limits in the reference model. Improvement was shown primarily in the power pack out cases. We believe that further improvements are possible, both by refinement of the adaptive control algorithm and by implementing changes in the guidance algorithm.

In the follow-on work, theoretical analysis will be done with regard to adaptation to bang-bang actuation such as that present in the reaction control system. Also, the proof of tracking error boundedness needs to be improved. That is, while the tracking error signals are bounded, the hedge signal may modify the reference model to such a degree that the reference command diverges from the desired command. This behavior was observed when an unachievable command is given. An alternative is to modify the command to ensure that the vehicle is never commanded to follow a trajectory for which there is insufficient control authority. In summary, we plan to:

1. Address deficiencies that have been identified during testing at NASA.
2. Extend our adaptive flight control approach to include TAEM and A/L phases.
3. Exploration of theoretical aspects including hedging in the presence of dynamic actuator nonlinearities and proof of absolute stability.
4. Preparation of software for delivery and independent evaluation at NASA Marshall. This will, if necessary, include updating all existing codes to a new version of MAVERIC.
5. Support of evaluation efforts at NASA Marshall

VI. 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. Calise A. J., Lee, S., and Sharma, M. "Development of a Reconfigurable Flight Control Law for Tailless Aircraft", *AIAA Journal of Guidance, Control, and Dynamics*, Vol 24 No.5, 2001, pp. 896-902.

6. 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.

7. Calise A. J., Sharma, M., and Lee, S. "Adaptive Autopilot Design for Guided Munitions", *AIAA Journal of Guidance, Control, and Dynamics*, Vol 23 No. 5, 2000.

8. Sharma, M., Calise A. J., and Corban, J. E. "Application of an Adaptive Autopilot Design to a Family of Guided Munitions", *AIAA Guidance, Navigation and Control Conference, Denver,* CO, August 2000.

9. Corban, E.J., Burkemper, B., Holt, K. Evers, J. Calise, A.J. Sharma, M. "Flight test of an adaptive autopilot for a precision guided munition" *AIAA Missile Sciences Conference*, Monterey, CA, 2002.

10. Idan, M., Johnson, M., and Calise, A. "A Hierachical Approach to Adaptive Control for Improved Flight Safety" *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 6, 2002.

11. Kaneshige, J., and Gundy-Burlet, K. "Integrated neural Flight and Propulsion Control System" *AIAA Guidance, Navigation, and Control Conference*, 2001.

12. Johnson, E. N., Calise, A. J., "Limited Authority Adaptive Flight Control for Reusable Launch Vehicles" *Journal of Guidance, Control and Dynamics*, Vol 26, No. 6, Nov-Dec 2003, pp 906-913.

13. Johnson, M., Calise, A., Johnson, E., "Evaluation of an Adaptive Method For Launch Vehicle Flight Control" *AIAA Guidance, Navigation, and Control Conference*, 2003.

14. Johnson, E., Calise, A., Corban, J.E., A Six Degree-of-Freedom Adaptive Flight Control Architecture for Trajectory Following. *AIAA Guidance, Navigation, and Control Conference*, 2002.

15. Hanson, J., Hall, C., Mulqueen, J., Advanced Guidance and Control for Hypersonics and Space Access. *JANNAF Interagency Propulsion Committee Meeting*, 2003.

16. Shtessel, Y., Zhu, J., and Daniels, D., Reusable Launch Vehicle Attitude Control using a Time-Varying Sliding Mode Control Technique. *AIAA Guidance, Navigation, and Control Conference*, 2002.

17. Zhu, J., Lawrence, D., Fisher, J., Shtessel, Y., Hodel, A.S., and Lu, P. Direct Fault Tolerant RLV Attitude Control – A Singular Perturbation Approach. *AIAA Guidance, Navigation, and Control Conference*, 2002.

18. Hodel, A. S., Callahan, R. Autonomous Reconfigurable Control Allocation (ARCA) for Reusable Launch Vehicles. AIAA Guidance, Navigation, and Control Conference, 2002.

19. Doman, D., Leggett, D., Ngo, A., Saliers, M., and Pachter, M., Development of a Hybrid Direct-Indirect Adaptive Control System for the X-33. AIAA Guidance, Navigation, and Control Conference, 2002.

20. Slotine, J., and Li, W. Applied Nonlinear Control. Prentice Hall, 1991.

21. Lewis, F., Yesildirek, A., and Liu, K., "Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance", *IEEE Transactions on Neural Networks*, Vol. 7 No. 2 March 1996 pp. 388-399.

22. Stevens, B. and Lewis, F., *Aircraft Control and Simulation*, John Wiley & Sons, 1992.

23. Johnson, E. *Limited Authority Adaptive Flight Control*, PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA, December 2000.

24. Narendra, K. and Annaswamy, A., *Stable Adaptive Systems*, Prentice Hall, 1989.

25. Funahashi, K., "On the Approximate Realization of Continuous Mappings by Neural Networks", Neural Networks, Vol. 2, 1989.

26. Hornik, K., Stinchcombe, M. and White, H., "Multilayer Feedforward Networks are Universal Approximators", Neural Networks, Vol. 2, 1989.

Appendix A
Neural Network-Based Adaptive Control

A.1 Approximate System Linearization

One of the common methods for controlling nonlinear dynamical systems is based on approximate feedback linearization[A1]. The form that is employed in each control channel depends on the relative degree of the controlled variable. To simplify our discussion, we assume that the system has full relative degree, where each regulated variable (element of the state vector x) has a relative degree of two. This means that each regulated variable has to be differentiated twice before a control variable appears explicitly

$$\ddot{x} = f(x, \dot{x}, u) \tag{A1}$$

A system has *full* relative degree when the sum of the relative degree of each all the regulated variables equation the degree of the system. This requires that for the system (A1) $x, u \in \Re^n$. An example is the case of attitude control, $x, u \in \Re^3$, where the elements of x correspond to the roll, pitch and yaw angles, and u are the torques commanded about each axis. A variant of this form arises in which angular rate is regulated. Here, the equation of motion for that degree of freedom is expressed in first-order form. A pseudo-control ν is defined such that the dynamic relation between it and the system state is linear

$$\ddot{x} = \nu \tag{A2}$$

where

$$\nu = f(x, \dot{x}, u) \tag{A3}$$

Ideally, the actual controls (u) are obtained by inverting Eq. (A3). Since the function $f(x, \dot{x}, u)$ is not known exactly, an approximation is used

$$\nu = \hat{f}(x, \dot{x}, u) \tag{A4}$$

which results in

$$\ddot{x} = \nu + \Delta(x, \dot{x}, u) \tag{A5}$$

where the modeling error is represented by

$$\Delta(x, \dot{x}, u) = f(x, \dot{x}, u) - \hat{f}(x, \dot{x}, u) \tag{A6}$$

The approximation, $\hat{f}$, must be chosen such that an inverse with respect to u exists. Consequently, the torque or effective actuator displacement in each axis is constructed as

$$u_{cmd} = \hat{f}^{-1}(x, \dot{x}, \nu) \tag{A7}$$

An adaptive process can be used to augment the inverting solution so as to cancel the modeling error represented in (A6). As shown in Figure A.1, the pseudo-control signal in (A5) is constructed of three components

$$\nu = \nu_{rm} + \nu_{lc} - \nu_{ad} \tag{A8}$$

where ν_{rm} is the pseudo-control component generated by the reference model, ν_{lc} is the output of the linear controller, and ν_{ad} is generated by the adaptive element introduced to compensate for the model inversion error. In the case of perfect actuation $(u = u_{cmd})$ and perfect adaptation $(\nu_{ad} = \Delta)$, the commanded pseudo-control signal ν equals $\ddot{x}$, the acceleration of the reference model state.

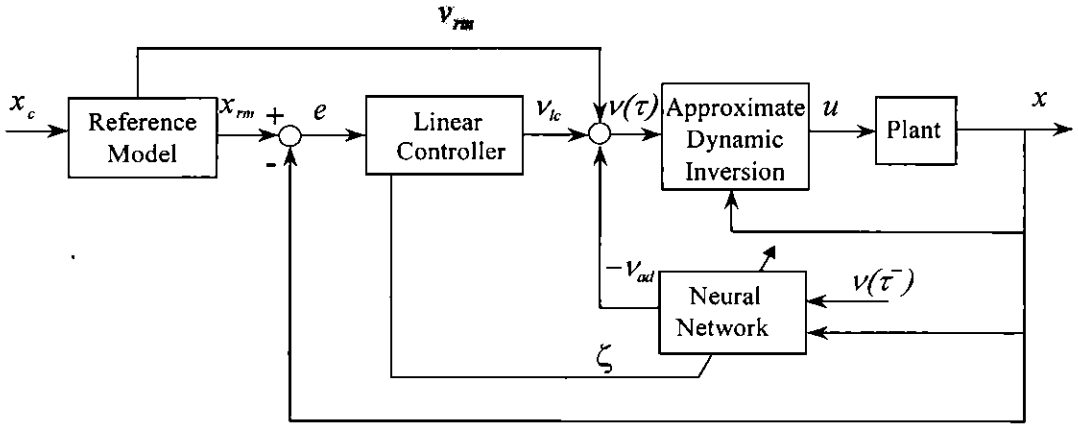

Figure A.1. NN-augmented adaptive control architecture.

Eqs. (A6-A8) imply that the model inversion error is a function of the pseudo-control ν and consequently of the NN output ν_{ad}. This explains why $\nu(\tau^-)$ is depicted as one of the necessary inputs to the NN. A delayed value is used to avoid an algebraic loop. To guarantee existence and uniqueness of a fixed-point solution for ν_{ad} such that $\nu_{ad} = \Delta$, it is assumed that the map $\nu_{ad} \mapsto \Delta$ is a contraction. It is shown in [A3] that this is equivalent to the following two requirements on $\hat{f}$:

$$\text{sign}\left(\frac{\partial \hat{f}}{\partial \delta}\right) = \text{sign}\left(\frac{\partial f}{\partial \delta}\right) \tag{A13}$$

$$\left|\frac{\partial \hat{f}}{\partial \delta}\right| > \left|\frac{\partial f}{\partial \delta}\right| / 2 > 0 \tag{A14}$$

These conditions imply that there no unknown reversals in control effectiveness, and that the true control effectiveness is not underestimated by more that a factor of 2.

A.2 Linear Controller Design

A linear controller is designed for each degree of freedom assuming perfect inversion $\left(\hat{f} = f\right)$. In the SISO case, if the controlled variable has relative degree two (as illustrated in the preceding section), the inverted system and the state tracking error dynamics associated with the inverted plant have two poles at the origin. A linear controller is designed so that the error dynamics are stabilized. In the state feedback case, this can be achieved using a standard proportional-derivative (PD) controller, although additional integral action can be incorporated to improve steady state performance. In general, the linear controller can be designed using any technique as long as the linearized closed-loop system is stable.

For a second order system,

$$\nu_{lc} = \begin{bmatrix} K_P & K_D \end{bmatrix} e \tag{A9}$$

where the tracking error vector is defined by

$$e = \begin{bmatrix} x_{rm} - x \\ \dot{x}_{rm} - \dot{x} \end{bmatrix} \tag{A10}$$

The controller gains are chosen so that the tracking error dynamics given by

$$\dot{e} = Ae + B\left(\nu_{ad} - \Delta\right) \tag{A11}$$

$$A = \begin{bmatrix} 0 & I \\ -K_P & -K_D \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{A12}$$

are stable, i.e., the eigenvalues of A are prescribed. It is evident from Eq. (A11) that the role of the adaptive component, ν_{ad}, is to cancel Δ. This approach extends directly to the MIMO case.

The closed-loop system made up of the linear controller and inverted plant is driven by the output of an at least 2nd order reference model. The reference model is hedged in the presence of saturation or failure using the pseudo-control hedging methodology, presented next.

A.3 Pseudo-Control Hedging (PCH)

PCH introduces a modification to NN-based model reference adaptive flight control. It is used to address NN adaptation difficulties arising from various actuation nonlinearities, including actuator position and/or rate saturation, discrete (magnitude quantized) control, time delays and

actuator dynamics[A4]. NN training difficulties occur when unmodeled actuator characteristics are encountered. For example, the NN adaptive element will attempt to adapt to these nonlinearities, even when it is impossible to do so. The goal of PCH to prevent the adaptive element from attempting to adapt to these characteristics, while not affecting NN adaptation to other sources of inversion error. During periods when a control is position saturated, PCH allows the NN to correctly estimate Δ. Conceptually, PCH "moves the reference model backwards" by an estimate of the amount the controlled system did not move due to selected actuator characteristics (such a position and rate limits, time delays, etc). The reference model is limited or hedged according to an estimate of the difference between the commanded and actually achieved pseudo-control

To briefly review the PCH concept, consider the case of full model inversion, in which the plant dynamics are described by (A1). The pseudo-control signal defined in (A4) represents the desired acceleration, while the actuator commands are given by (A7). The dynamic inversion element is designed assuming perfect actuation. Hence, during periods when the actuator is limited (or if the actuator is discrete rather than continuous) $u_{cmd} \neq u$. The pseudo-control hedge signal ν_h is defined as the difference between the commanded pseudo-control input and the actually achieved pseudo-control. To compute this difference, a measurement or an estimate of the actuator position $(\hat{u})$ is required. This estimate is then used to compute the pseudo-control hedge as

$$\nu_h = \hat{f}(x, \dot{x}, u_{cmd}) - \hat{f}(x, \dot{x}, \hat{u}) = \nu - \hat{f}(x, \dot{x}, \hat{u}) \tag{A15}$$

Figure A.2 illustrates the manner in which ν_h can be estimated for an actuator that is position and rate limited.

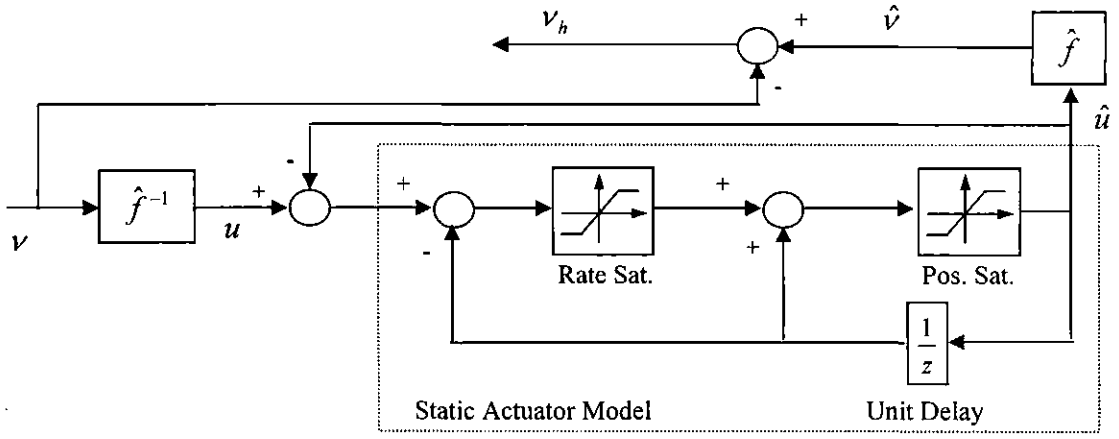

Figure A.2. Calculation of the Hedge Signal.

The PCH signal is introduced as an additional input into the reference model, forcing it to "move back". If the reference model dynamics *without* PCH have the form

$$\ddot{x}_{rm} = f_{rm}(x_{rm}, \dot{x}_{rm}, x_c) \tag{A16}$$

where x_c is the external command signal, then the reference model update *with* PCH is set to

$$\ddot{x}_{rm} = f_{rm}\left(x_{rm}, \dot{x}_{rm}, x_c\right) - \nu_h \tag{A17}$$

This is illustrated in Figure A.3 for a linear 2nd or reference model. A theoretical justification for the PCH approach is provided in [A5].

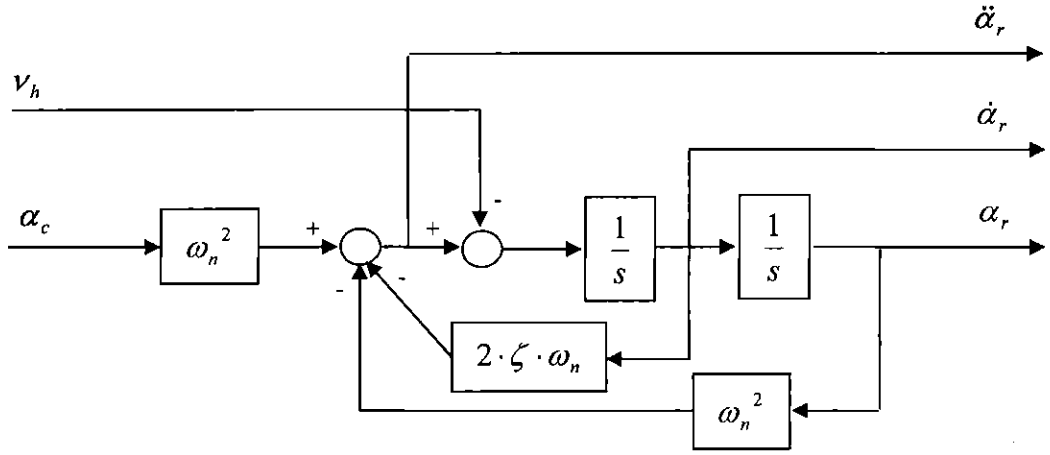

Figure A.3. Hedged Second Relative Degree Reference Model.

A.4 The Neural Network

A nonlinear single hidden layer (SHL) NN is used to compensate for inversion error. The SHL NN was chosen because of its universal approximation property[A6, A7], and because its effectiveness in relation to linearly-parametered adaptive control (including linearly parameterized NNs) has been demonstrated for flight control applications[A2].

For an input vector $\bar{x}$, which is constructed from the measured states, the reference model outputs and the pseudo-control signal, the output of the SHL NN is given by

$$\nu_{ad} = \nu_{ad0} + \nu_r \tag{A19}$$

where ν_{ad0} is defined as in (A20) and ν_r is a robustifying term defined below. The first term in (A19) is the main output signal from the NN

$$\nu_{ad0} = W^T \sigma\left(V^T \bar{x}\right) \tag{A20}$$

where V and W are the input and output weighting matrices, respectively, and σ is a vector of sigmoid activation functions. The NN weights are adapted in real time using the following NN weights training rules:[A5, A8]

$$\dot{W} = -\left[\left(\sigma - \sigma' V^T \bar{x}\right)\eta + \kappa \|e\| W\right]\Gamma_W \tag{A21}$$

$$\dot{V} = -\Gamma_V \left[\bar{x}\eta W^T \sigma' + \kappa \|e\| V\right] \tag{A22}$$

where Γ_W and Γ_V are positive definite learning rate matrices (typically chosen as a scalar times an identity matrix), σ' is the partial derivative of the sigmoids σ with respect to the NN inputs $\bar{x}$, and κ is the known as e-modification gain for classical adaptive control theory. The training signal η is defined by

$$\eta = e^T PB \tag{A23}$$

where $P \succ 0$ is the unique positive definite solution of the Lyapunov equation

$$A^T P + PA + Q = 0 \tag{A24}$$

for any $Q \succ 0$. A and B in the above equations are the tracking error dynamics matrices defined in Eq. (A12).

To determine the robustifying terms, define

$$Z = \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \tag{A25}$$

Let $\|\cdot\|$ denote Frobenius norm, and assume $\|Z\| \le \bar{Z}$. The robustifying terms in (A19) are

$$\nu_r = -\left(\|Z\| + \bar{Z}\right) K_r \eta^T \tag{A26}$$

$K_r \in \Re^{3\times 3}$ diagonal, $K_r < 0$, and $\bar{Z}$ is such that $\|Z^*\| \le \bar{Z}$, where Z^* denotes an unknown set of ideal weights.

A.5 References

A1. Isodori, A., "Nonlinear Control Systems". Springer Verlag, Berlin, 1989.

A2. Calise A. J., Lee, S. and Sharma, M. "Development of a Reconfigurable Flight Control Law for the X-36 Tailless Fighter Aircraft" *AIAA Journal of Guidance, Navigation, and Control*, Vol 24, No.5, 2001, pp.896-902

A3. Calise, A. J., Hovakimyan, N. and Idan, M. "Adaptive Output Feedback Control of Nonlinear Systems Using Neural Networks" *Automatica*, August, 2001.

A4. Hovakimyan, N., Kim, N., Calise, A. J., Prasad, J.V.R. and Corban, J. Eric. "Adaptive Output Feedback for High-Bandwidth Control of an Unmanned Helicopter" In *AIAA Guidance, Navigation and Control Conference*, 2001

A5. Johnson, E., *Limited Authority Adaptive Flight Control.* Ph.D. Thesis, Georgia Institute of Technology, 2000.

A6. Funahashi, K., "On the Approximate Realization of Continuous Mappings by Neural Networks", *Neural Networks,* Vol. 2, 1989.

A7. Hornik, K., Stinchcombe, M. and White, H., "Multilayer Feedforward Networks are Universal Approximators", *Neural Networks,* Vol. 2, 1989.

A8. Johnson, E. and Calise, A. J., "Neural Network Adaptive Control of Systems with Input Saturation", In *American Controls Conference,* June, 2001.

A9. N. Hovakimyan, F. Nardi, A. Calise, "Adaptive Output Feedback Control of Uncertain Systems using Single Hidden Layer Neural Networks", accepted to *IEEE Transactions on Neural Networks*

Appendix B
Test Case Matrix

Nominal Flight:
FC1. Michael 10a1 nominal
FC2. Michael 10d1 nominal

All remaining tests use the Michael 10a1 reference profile as a starting point.

Engine Failure:
FC3. 36 sec PPO (early abort to Ibex)
FC4. 50 sec PPO (early abort to Michael)

Thrust Vector Control Failure:
FC5. TVC command bias on Engine A Roll/Pitch TVC commands of +0.5%
FC6. TVC command bias on Engine B of Roll/Pitch TVC commands of –1.0%
FC7. TVC command bias on Yaw TVC commands of +1.0%
FC8. +3-sigma Fz, My on Engine A, -3-sigma Fz, My on Engine B

Aerosurface Failures:
FC 9. Right inboard elevon fails to +10 deg at 50 seconds into flight for 30 seconds
FC 10. Left outboard elevon fails to –15 deg at 275 seconds into flight for 45 seconds
FC 11. Right flap fails to +2 deg at 150 seconds into flight for 20 seconds
FC 12. Right flap fails to +2 deg at 300 seconds into flight for 20 seconds
FC 13. Right rudder fails to –30 deg at 30 seconds into flight for remainder of flight
FC 14. Left inboard elevon fails to null at 35 seconds into flight
FC 15. Right outboard elevon fails to null at 250 seconds into flight
FC 16. Right flap fails to null at 20 seconds into flight
FC 17. Left flap fails to null at 215 seconds into flight.
FC 18. Right outboard elevon jams at 58 seconds
FC 19. Left inboard elevon jams at 208 seconds into flight
FC 20. Right flap jams at 170 seconds into flight
FC 21. Left flap jams at 280 seconds into flight
The aerosurface failures tests below occur at t=0:
FC 22. Right inboard elevon fails to "trailing in the breeze" (coefficients set to zero)
FC 23. Left outboard elevon fails to "trailing in the breeze" (coefficients set to zero)
FC 24. Right flap fails to "trailing in the breeze" (coefficients set to zero)
FC 25. Right rudder fails to "trailing in the breeze" (coefficients set to zero)

Reaction Control System Failure:
FC26. Fail 1&10 at MECO (loss of pure yaw capability)
FC27: Fail 5&9 at MECO (loss of pure yaw capability)
FC28: Fail 4 at MECO (loss of yawroll capability)
FC29: Fail 8 at MECO (loss of yawroll capability)

Modeling Errors:
FC30. +3-sigma Cm, +3-sigma C_L, + 3-sigma C_D (pitch moment, lift, and drag coefficients)
FC31. –4-sigma Cm, -4-sigma C_L, -4-sigma C_D
FC32. -3-sigma CY, -3-sigma Cl, -3-sigma Cn (side force, roll moment, and yaw moment coefficients)
FC33. –3-sigma Cm (body flap), +3-sigma CY,Cl,Cn (body flap)
FC34. +3-sigma CY,Cl,Cn (elevons)
FC35. +4-sigma jet effect increments
FC36. +3-sigma adverse yaw moment increments on elevons & body flaps

Dispersion Tests:
FC37. Michael 10a1, 100 dispersion Monte Carlo set
FC38. Michael 10a1, 200 dispersion Monte Carlo set, different seed and season
FC39. Michael 10d1, 100 dispersion Monte Carlo set
FC40. Michael 10d1, 200 dispersion Monte Carlo set, different seed and season
FC41. Michael 10a1, 60 sec PPO, 100 dispersion Monte Carlo set
FC42. Michael 10a1, 112 sec PPO, 100 dispersion Monte Carlo set
FC43. Michael 10a1, 40 sec PPO, 200 dispersion Monte Carlo set, different seed
FC44. Michael 10d1, 38 sec PPO, 200 dispersion Monte Carlo set, different seed
FC45. Michael 10a1, 100 dispersion Monte Carlo set with flex filter in attitude and rate error loops.
FC46. Michael 10d1, 100 dispersion Monte Carlo set with flex filter in attitude and rate error loops.