

FURTHER EVALUATION OF AN ADAPTIVE METHOD FOR LAUNCH VEHICLE FLIGHT CONTROL

Matthew D. Johnson^{*}, Anthony J. Calise[†], Eric N. Johnson[‡]

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

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 comparing it to a gain scheduled controller in the case of a mission undergoing an engine-out abort scenario. The adaptive controller is able to perform nearly as well as a gain-scheduled controller, and suggestions are made for further improvements.

I. Introduction

THE X-33 (Figure 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 configuration¹.

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 mission².

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 design³.

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.

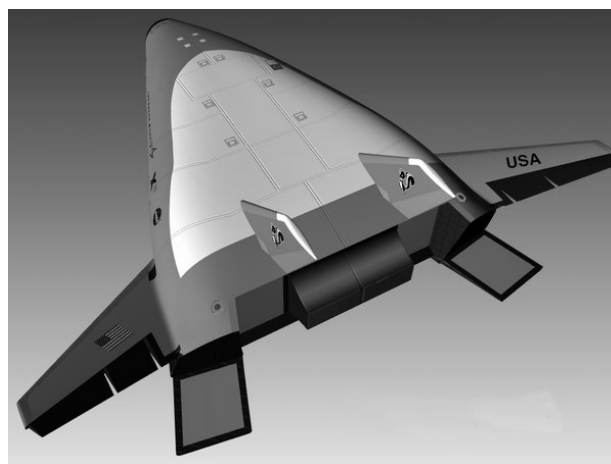


Figure 1. X-33 Reusable Launch Vehicle technology demonstrator.

^{*} Graduate Research Assistant. Student Member AIAA.

[†] Professor. Fellow AIAA.

[‡] Lockheed Martin Assistant Professor of Avionics Integration. Member AIAA.

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 aircraft⁶, 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, loop¹⁴. 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 Center¹⁵. 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. 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 controller¹⁶, a trajectory linearization controller¹⁷, and a reconfigurable allocator¹⁸. In addition, a hybrid direct-indirect adaptive controller¹⁹ was tested in ascent only but not compared to the other controllers. NASA's tests are ongoing.

This paper details the design of the adaptive controller, and compares its performance to NASA's baseline controller. Recent efforts to improve the results relative to NASA's criteria are also documented.

II. Control Architecture

Figure 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.

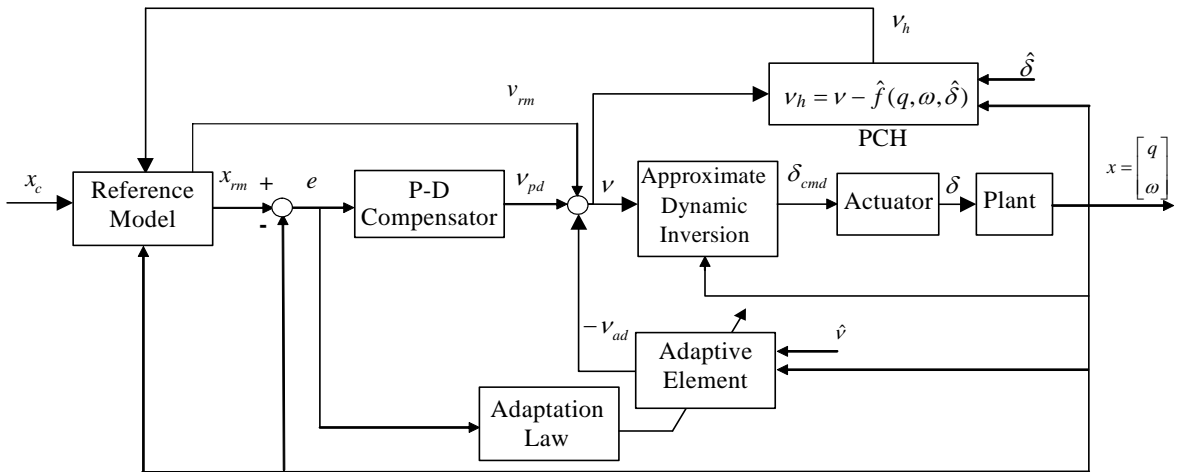


Figure 2. MRAC including an approximate dynamic inversion with PCH.

A. 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 $q \in \mathfrak{R}^4$ is a quaternion representing body attitude with respect to an inertial frame, $\omega \in \mathfrak{R}^3$ is the angular velocity of the body in the inertial frame expressed in the body frame, and $\delta, \delta_{cmd} \in \mathfrak{R}^m$ 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[§]

$$\delta_{cmd} = \hat{f}^{-1}(q, \omega, \delta)\tag{8}$$

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

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

For the case $m > 3$ we assume that a control allocation algorithm has been pre-defined.

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

B. Reference Model and Linear Controller Design

The hedge signal is

$$v_h = \hat{f}(q, \omega, \delta_{cmd}) - \hat{f}(q, \omega, \hat{\delta}) = v - \hat{v} \quad (10)$$

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

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

$$\dot{\omega}_{rm} = v_{crm}(q_{rm}, \omega_{rm}, q_c, \omega_c, \dot{\omega}_c) - v_h = v_{rm} \quad (12)$$

The reference model states are quaternion quantities. Details relating to the functional forms of $Q(*)$ and $v_{crm}(*)$ 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 rate²²

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

where

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

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

such that

$$\Omega_q^{-1} \Omega_q = I \quad (16)$$

The reference model error vector is

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

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:

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

Differentiating, we have

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

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 v_{crm} is chosen as

$$v_{crm} = [K_{pc} \quad K_{dc}] e_{rm} \quad (20)$$

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

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

is Hurwitz. The reference model tracking dynamics then reduce to

$$\dot{e}_{rm} = A_{rm} e_{rm} + B_{rm} (\dot{\omega}_c - v_h) \quad (22)$$

where

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

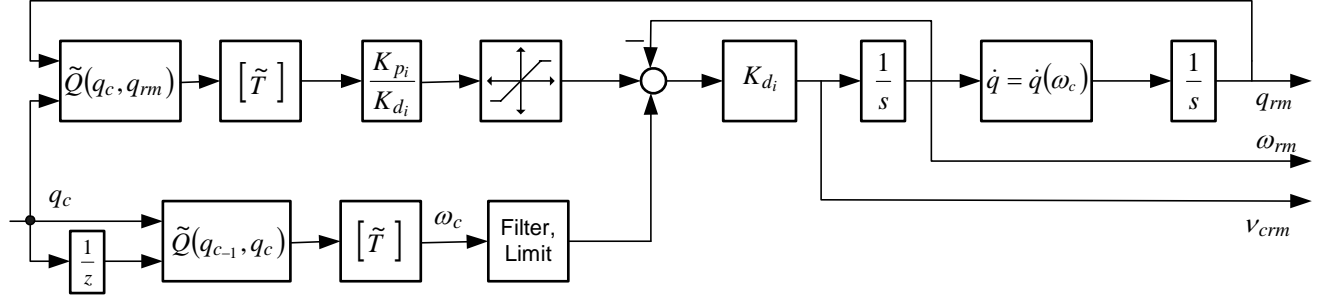


Figure 3. Reference Model.

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

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

Differentiating Eq. (24), we get

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

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

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

Choosing

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

the error dynamics reduce to a standard form for the application of adaptive control.²³

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

where

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

and $B = B_{rm}$.

Figure 4 shows the manner in which this is computed.

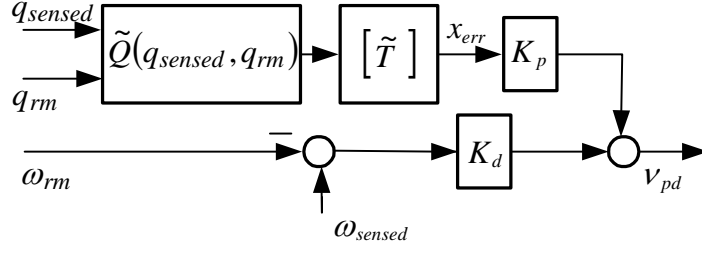


Figure 4. Proportional-Derivative Controller.

C. Neural Network for Inversion Error Compensation

A nonlinear single hidden-layer (SHL) NN is used to compensate for the model inversion error.^{24,25} For an input vector \bar{x} , the output of the SHL NN is given by

$$v_{ad} = W^T \sigma(V^T \bar{x}) \quad (30)$$

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}:

$$\dot{W} = -\left[(\sigma - \sigma V^T \bar{x}) \eta + \kappa \|e\| W \right] \Gamma_W \quad (31)$$

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

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

$$\eta = e^T P B \quad (33)$$

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

$$A^T P + P A + Q = 0 \quad (34)$$

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

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

where

$$Z = \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \quad (36)$$

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

D. Inverse Model and Hedge Calculation

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 matrix. 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 Figure 7 show the aerodynamic and RCS allocations, respectively.

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.

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:

$$\begin{aligned} Cost &= (T_c - T_a) k_1 + (\#Jets)^2 \cdot k_2 \\ Cost &= (T_c - T_a) k_1 + \#Jets \cdot k_3 + \Delta RCS \cdot k_4 \end{aligned} \quad (37)$$

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.

III. Simulation Results and Discussion

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.

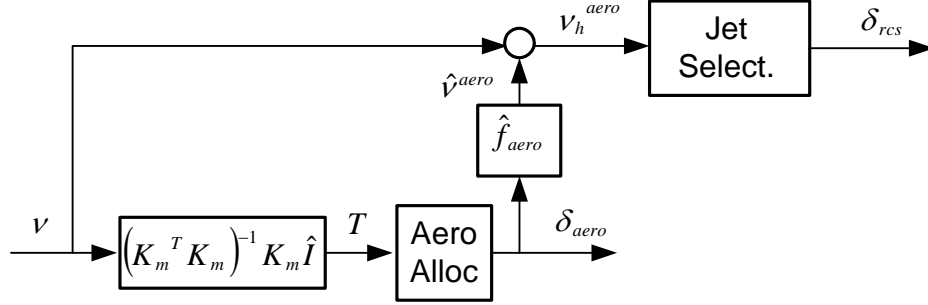


Figure 5. Inverse model dynamics.

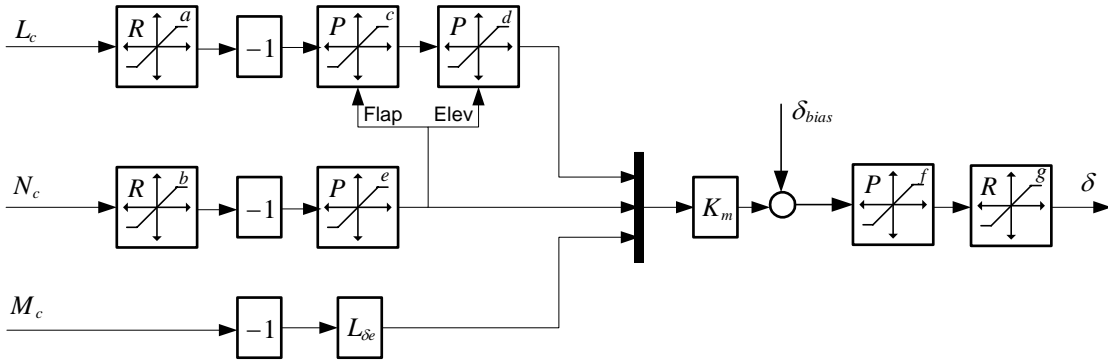


Figure 6. Allocation in the Aerodynamic Actuators.

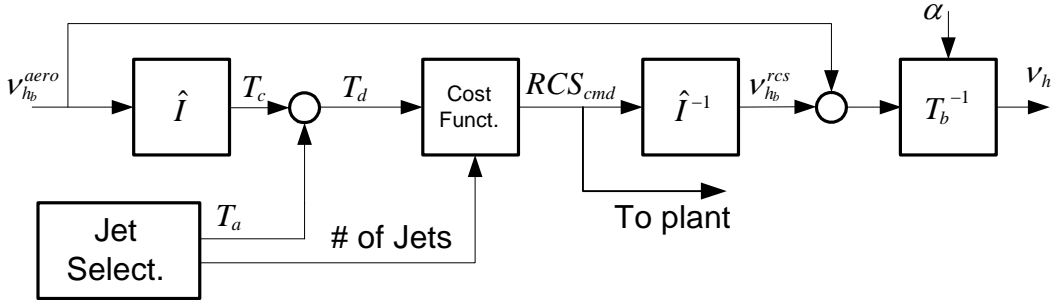


Figure 7. Hedge Calculation: Deficit in Aerodynamic Actuation + Deficit in RCS Actuation.

A. 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 ($\hat{v} = v - v_h$). 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. (37) shows.

B. 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.

C. Power Pack Out at 40 Seconds Case

First we examine a particular failure case where one of the rocket motor power packs fails at 40 seconds. This results in reduced thrust and requires the vehicle to abort the mission and follow a new trajectory to a new landing site. A Monte Carlo examination of this case shows that the baseline scores better because the NPID baseline succeeds for 185/200 seeds, and the adaptive architecture discussed above succeeded in only 89/200 seeds.

Examination of the performance of the adaptive algorithm showed that it had difficulty tracking the angle of attack, roll, and sideslip angle commands in entry. At low dynamic pressures, the NPID algorithm responded to decreased aerodynamic effectiveness by using approximately 25 degrees of differential elevon, which resulted in a large amount of roll authority. In the adaptive case, software limits to the commanded roll moment (shown in Figure 6 as ‘P’) have been applied to assure sufficient authority in the pitch channel for trim as the dynamic pressure rises. To meet both of these concerns over the entry profile, the allocation scheme presented in Table 1 and Figure 8 was implemented, resulting in a set of large differential actuator limits at low dynamic pressures and a set of small differential actuator limits at high dynamic pressures.

Figures 9-12 show results for seed 44 of the dispersion test for both the NPID case and the final version of the adaptive algorithm. after implementing this allocation change during entry. The NPID fails in this case, departing from the commanded trajectory and spinning out of control. The adaptive algorithm tracks the commands until reaching the required TAEM conditions. The control time history shows that immediately after the transition to entry, the adaptive algorithm saturates the differential elevon command in order to maintain roll tracking. Figure 12 shows the time history of the RCS actuation moment. After implementation of these changes, the adaptive algorithm succeeds in 166/200 of the individual runs.

Table 1. Differential Limits on the Actuators

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

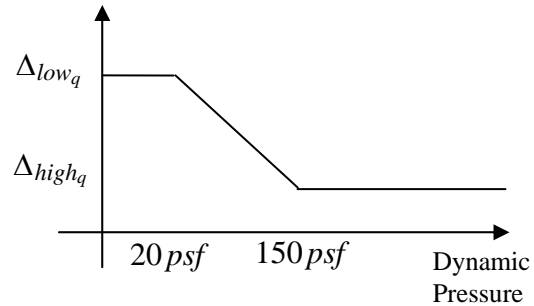


Figure 8. Typical allocation scheduling diagram.

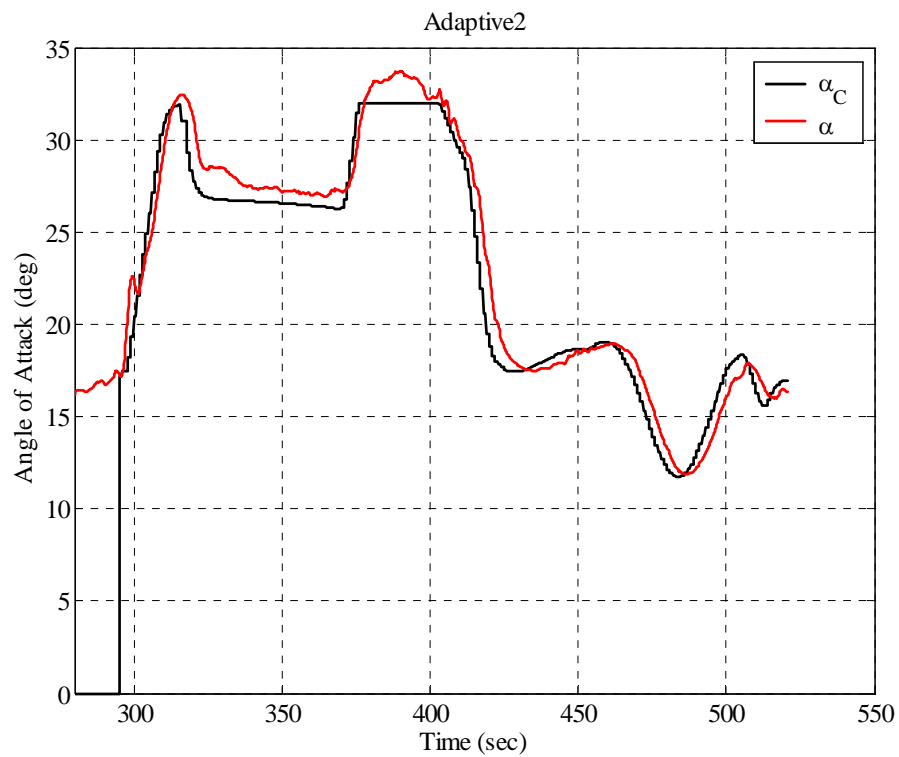
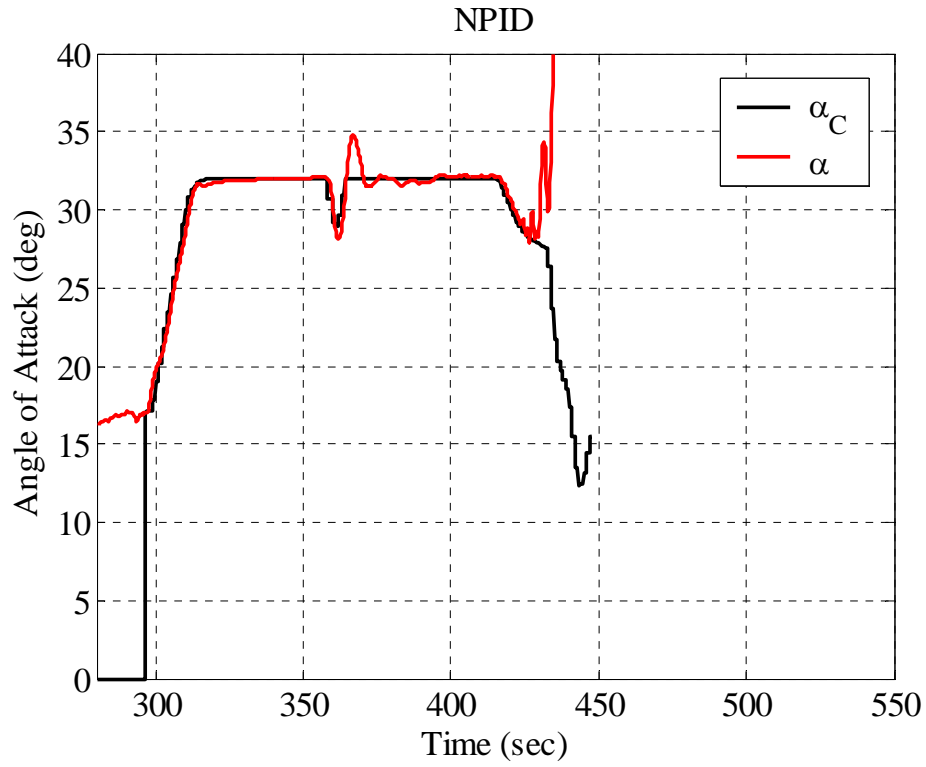


Figure 9. Alpha response in Entry.

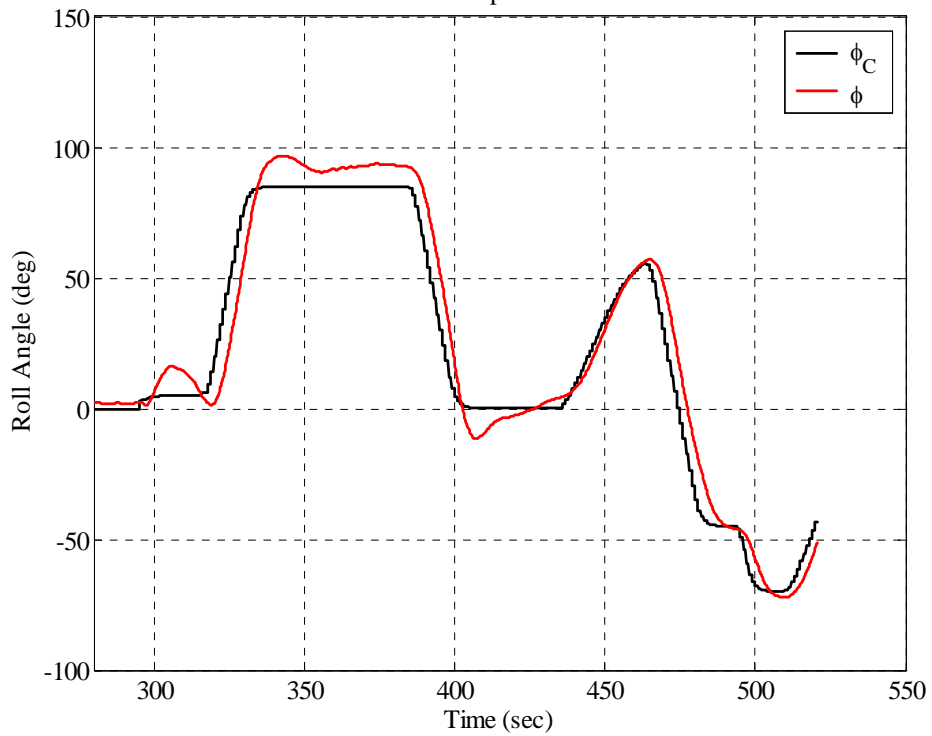
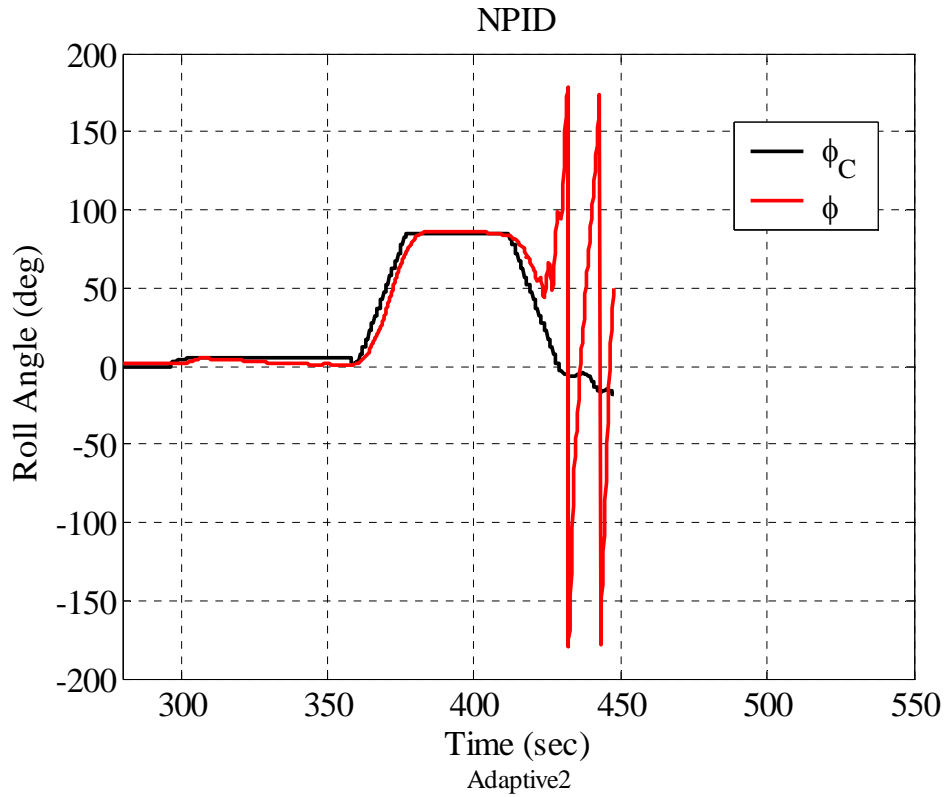


Figure 10. Phi response in Entry.

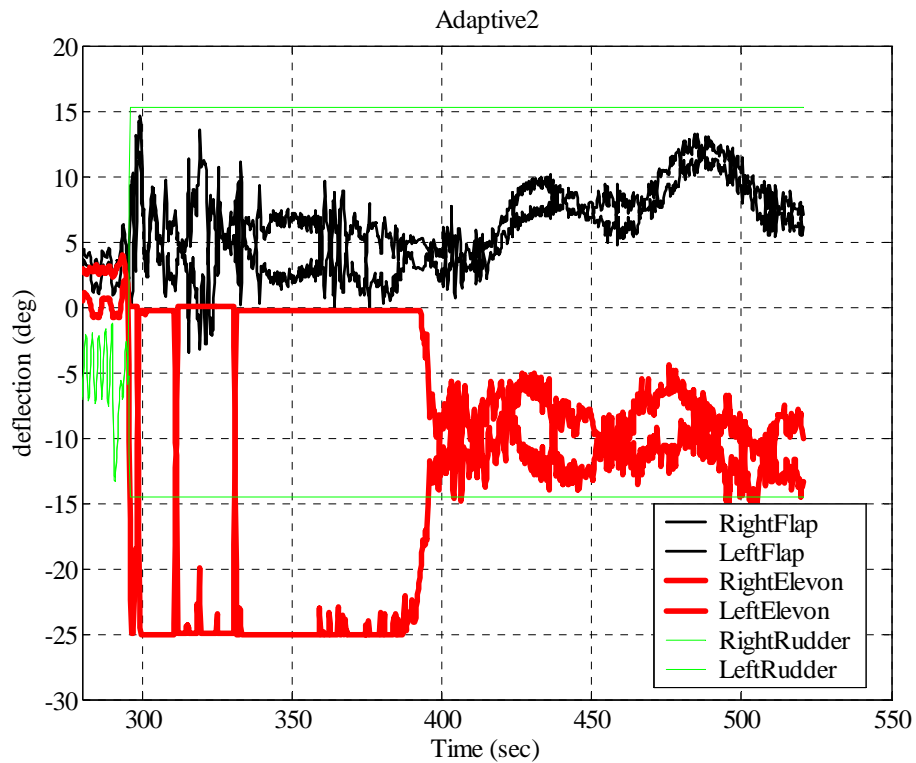
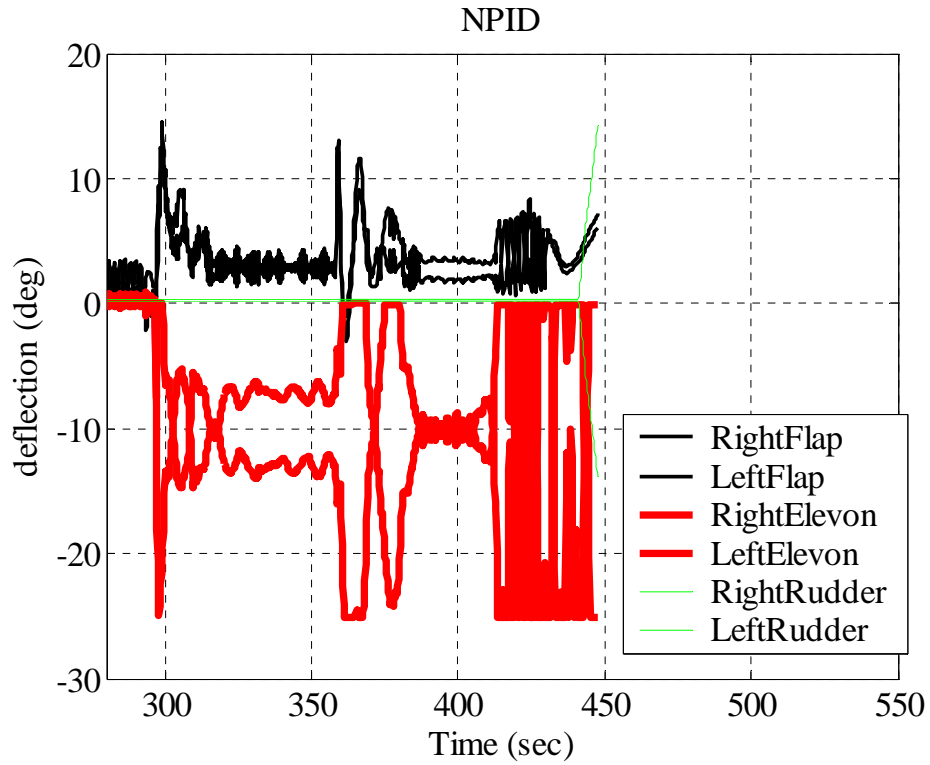


Figure 11. Actuator Commands in Entry.

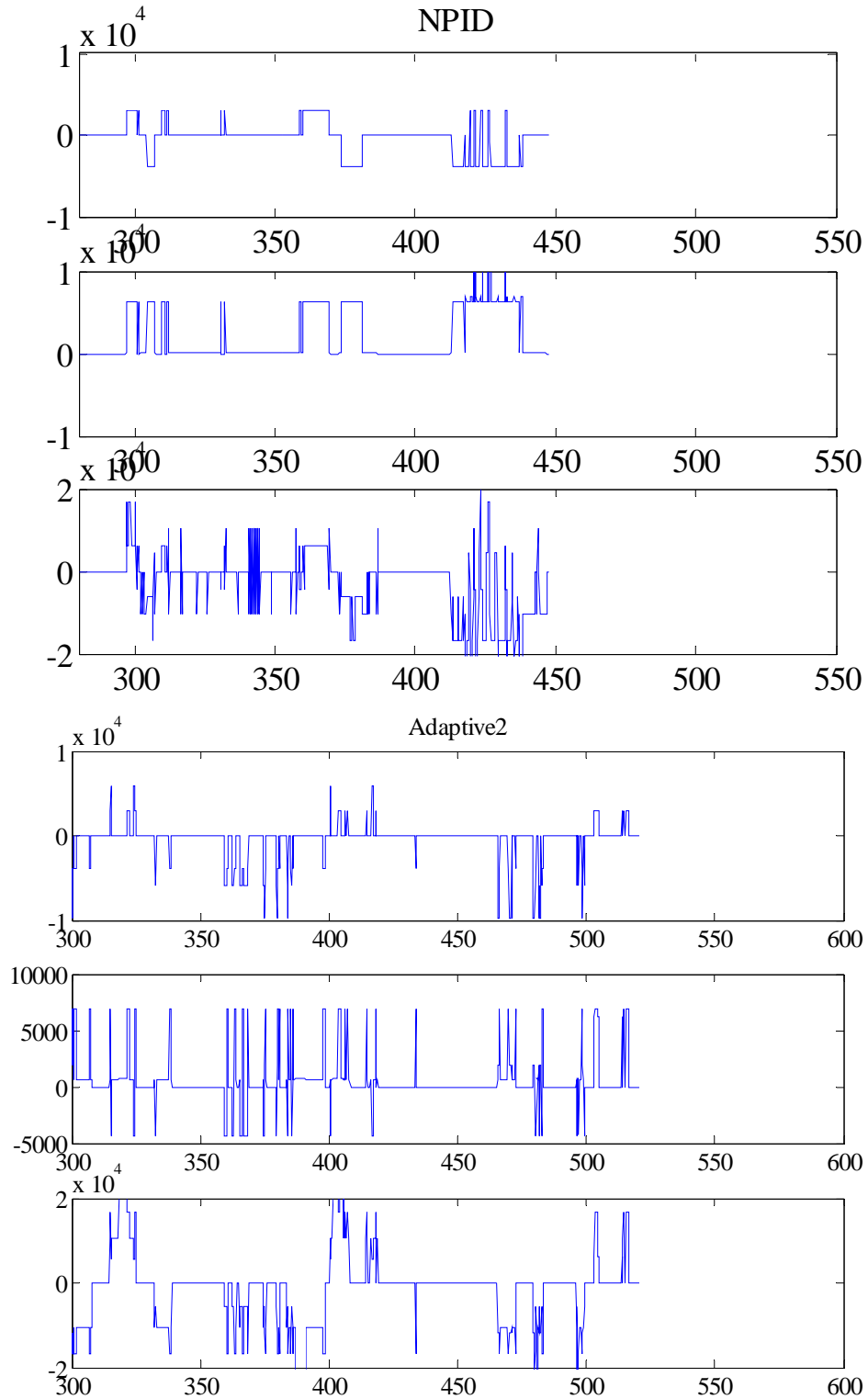


Figure 12. RCS Moments in Entry.

Figure 13 shows a comparison of the NPID to two versions of the adaptive controller, *adaptive*, the version prior to the changes discussed above, and *adaptive2*, the version tested after those changes. Note that on tests where all three controllers succeed, the gain scheduled NPID tends to do slightly better. However, there are several tests where the NPID fails the test criteria and does not score at all. The Figure demonstrates that the changes present in *adaptive2* show an incremental improvement over the prior version.

Table 2 shows a history of test results since November 2002. The test criteria themselves 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 modified test criteria. After the changes discussed previously, the algorithm was re-scored according to the new criteria, and our overall score improved slightly due to having passed one of the previously failed test cases.

Table 2. 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

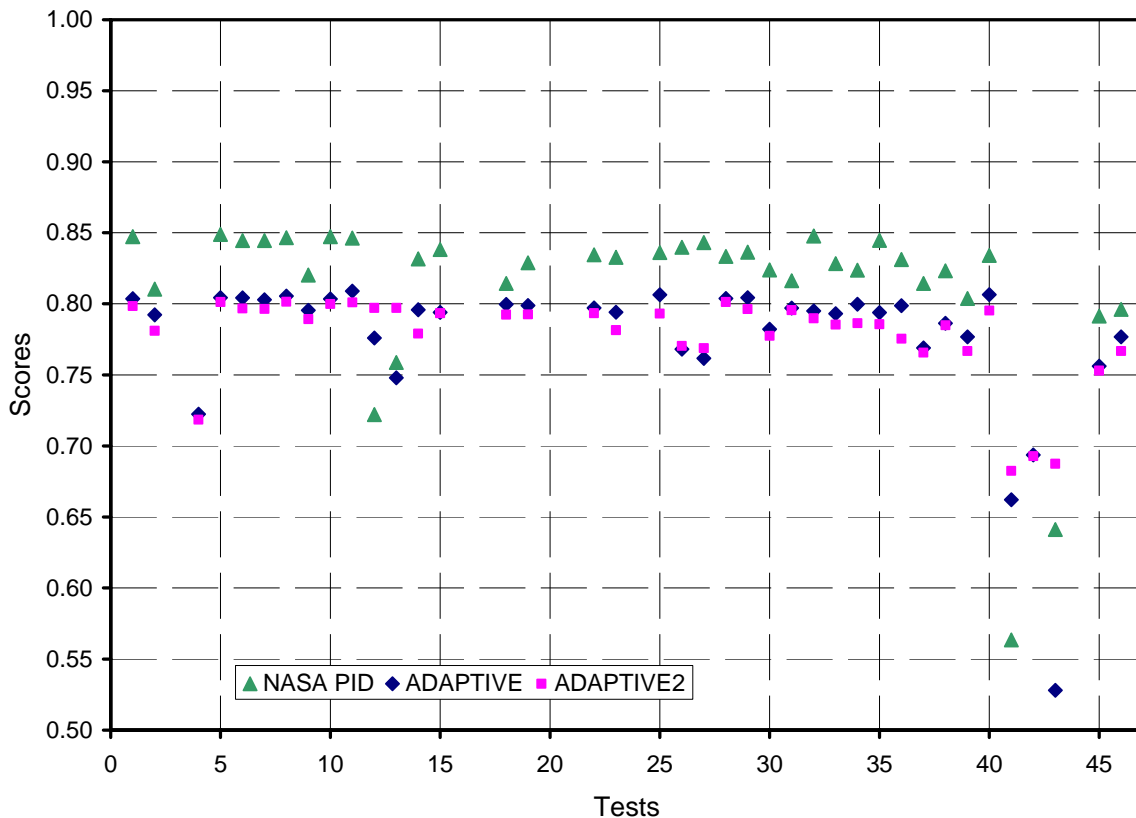


Figure 13 Comparison of test results

IV. Conclusions And Future Study

This paper 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. A example is given to illustrate the manner in which the design can be improved for a particular failure scenario, by modifying aerodynamic control allocation in entry. We believe that further improvements are possible, both by further refinement of the adaptive control algorithm and by implementing changes in the guidance algorithm.

Acknowledgment

Research supported by NASA Marshall Space Flight Center, under grant number NAG 8-1638.

References

- ¹Hanson, 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.
- ²Hanson, J. of NASA MSFC. Personal Communication. April 3, 2000.
- ³Hall, 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.
- ⁴Calise, A., and Rysdyk, R. "Nonlinear Adaptive Flight Control Using Neural Networks" *Control Systems Magazine*, December 1998.
- ⁵Calise 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.
- ⁶Brinker, 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.
- ⁷Calise 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.
- ⁸Sharma, 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.
- ⁹Corban, 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.
- ¹⁰Idan, M., Johnson, M., and Calise, A. "A Hierarchical Approach to Adaptive Control for Improved Flight Safety" *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 6, 2002.
- ¹¹Kanehige, J., and Gundy-Burlet, K. "Integrated neural Flight and Propulsion Control System" *AIAA Guidance, Navigation, and Control Conference*, 2001.
- ¹²Johnson, 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.
- ¹³Johnson, M., Calise, A., Johnson, E., "Evaluation of an Adaptive Method For Launch Vehicle Flight Control" *AIAA Guidance, Navigation, and Control Conference*, 2003.
- ¹⁴Johnson, 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.
- ¹⁵Hanson, J., Hall, C., Mulqueen, J., Advanced Guidance and Control for Hypersonics and Space Access. *JANNAF Interagency Propulsion Committee Meeting*, 2003.
- ¹⁶Shtessel, 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.
- ¹⁷Zhu, 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.
- ¹⁸Hodel, A. S., Callahan, R. Autonomous Reconfigurable Control Allocation (ARCA) for Reusable Launch Vehicles. *AIAA Guidance, Navigation, and Control Conference*, 2002.
- ¹⁹Doman, 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.
- ²⁰Slotine, J., and Li, W. *Applied Nonlinear Control*. Prentice Hall, 1991.
- ²¹Lewis, 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.
- ²²Stevens, B. and Lewis, F., *Aircraft Control and Simulation*, John Wiley & Sons, 1992.

²³Narendra, K. and Annaswamy, A., *Stable Adaptive Systems*, Prentice Hall, 1989.

²⁴Funahashi, K., "On the Approximate Realization of Continuous Mappings by Neural Networks", *Neural Networks*, Vol. 2, 1989.

²⁵Hornik, K., Stinchcombe, M. and White, H., "Multilayer Feedforward Networks are Universal Approximators", *Neural Networks*, Vol. 2, 1989.