

A SIX DEGREE-OF-FREEDOM ADAPTIVE FLIGHT CONTROL ARCHITECTURE FOR TRAJECTORY FOLLOWING

Eric N. Johnson* and Anthony J. Calise†

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

J. Eric Corban‡

Guided Systems Technologies, Inc., McDonough, GA 30253-1453

Abstract

For reusable launch vehicles, it is common to separate the guidance and flight control problem into an outer loop and an inner loop. The inner loop uses aerodynamic and propulsive controls to achieve a commanded attitude. The attitude commands are generated by an outer loop. This outer loop uses inner loop commands and control variables to achieve desired positions/velocities. This paper describes improvements and evaluation results for combined inner-outer loop adaptive flight control architecture for reusable launch vehicles. This architecture includes the online training of a neural network to correct for force and moment model errors. The result is a system that can respond to large model errors (significant failure scenarios) in both inner loop attitude control as well as in trajectory tracking. The resulting system can be used to improve performance and/or to increase guidance, navigation, and control system tolerance to model error, failures, and the environment. Simulation results are presented for failure scenarios.

Introduction

Reliable and affordable access to space along with a global engagement capability are recognized as critical requirements. New U.S. Air Force initiatives are in place to create a truly integrated AeroSpace force, and include technology developments that will enable a hypersonic cruise reconnaissance/strike vehicle. NASA has established aggressive goals for reduction in both the cost of operations and turn-around time of future Reusable Launch Vehicles (RLVs), and seeks to obtain greatly enhanced safety in operations. Autonomous guidance and control (G&C) technologies are recognized as critical to the objective of achieving

reliable, low-cost, aircraft-like operations into space. Specifically, next generation G&C systems must be able to fly a variety of mission scenarios, as well as handle dispersions, failures and abort requirements in a robust fashion¹. This paper is concerned with the development of autonomous G&C systems for RLVs, and in particular with the ability to handle large dispersions and failures through adaptation.

Neural Network (NN)-based direct adaptive control has emerged as an enabling technology for practical reconfigurable flight control systems. In the USAF Reconfigurable Control for Tailless Fighter Aircraft (RESTORE) program, adaptive nonlinear control was combined with on-line real-time parameter identification and on-line constrained optimization to demonstrate the capability of a next generation aircraft control system with redundant control actuation devices to successfully adapt to unknown failures and damage. The reconfigurable flight control system was based on a dynamic inversion control law augmented by an on-line NN. Here, the NN continuously learns (i.e., adapts) to produce a signal that is used to cancel the error between the plant model (used by the dynamic inversion control law) and the true vehicle dynamics². The program culminated in successful flight demonstration of the adaptive controller on the X-36³.

This approach to failure and damage tolerant control has been improved to handle control saturation, unmodeled actuator dynamics, and quantized control, and applied to an autopilot design for the X-33. The X-33 was designed to be a sub-orbital aerospace vehicle intended to demonstrate technologies necessary for future RLVs. Features of the design include a linear aerospike rocket engine, vertical take-off, and horizontal landing. A simulation study has shown that NN augmented non-linear adaptive flight control provides an approach that maintains stable performance under large variations in the vehicle and environment. This has a two-fold benefit, by increasing safety in the presence of unanticipated failures and by reducing the tuning required per mission due to small changes in vehicle/environment/payload configuration⁴.

* Lockheed Martin Assistant Professor of Avionics Integration, Member AIAA.

E-mail: Eric.Johnson@aerospace.gatech.edu

† Professor, Fellow AIAA.

E-mail: Anthony.Calise@aerospace.gatech.edu

‡ GST President, Member AIAA. E-mail: corban@mindspring.com

Adaptive guidance technology is required, however, to realize the full potential of adaptive control in application to RLVs. For some classes of failures, one would expect to be able to continue to track the nominal trajectory (i.e., guidance commands). But for others, a loss in thrust and/or control power will prevent successful tracking of the nominal solution, or even family of solutions. The combination of adaptive guidance and adaptive control is required to successfully manage a wide class of potential failures in autonomous launch systems. Ascent, reentry, and abort trajectories can be complicated in failure scenarios by constraints that result from a reduction in control power (possibly induced by the failure) and the potential for control saturation. The capability for on-board trajectory regeneration is thus essential to establish the ability to overcome such in-flight failures. The further addition of adaptive tracking of the trajectory will allow for successful mission completion in an expanded set of failure scenarios, and provides the time necessary for diagnosis of failures and for the regeneration of the desired trajectory.

There are a number of mature technologies for fault detection and isolation that are appropriate for use in detecting, isolating and identifying various classes of failures. There are, as well, mature research efforts focused on the task of on-line system identification^{5,6}. On-line system identification can be used to detect and model many additional classes of failures, but require time to produce a valid result. The proposed autonomous G&C architecture will draw upon this technology base to produce on-line an approximate model of the failed system. Note however, that for trajectory re-planning, it is not sufficient to capture the local impact of the failure. In many cases it will be necessary to fully isolate the cause of degraded performance, so that its impact can be modeled across the flight envelope.

Recognizing that significant time is required to construct a model of the failed system, it will be necessary to devise a proper guidance strategy for this interim period. In this paper, Pseudo-Control Hedging (PCH)⁴ is used to locally modify the nominal trajectory commands so that the vehicle follows a feasible path with the same overall objectives as the nominal trajectory. The modified (i.e., hedged) trajectory commands are locally achievable despite a reduction in control power and/or control saturation. This strategy does not require the identification of a failed system model for implementation. Note that the requirement for optimality is relaxed in the absence of a model reflecting the failure. Once an approximate model of the failed system is available, the impact of the failure on the mission can be assessed and the mission objectives

can be intelligently reassigned.

One such approach that may prove suitable for the purpose of online trajectory optimization is a hybrid method for trajectory optimization⁷⁻⁹. The hybrid designation in this context refers to the combined use of analytical and numerical methods of solution. The approach combines optimal control with collocation techniques, and has the demonstrated potential of being able to determine profiles with modified targeting in a closed-loop fashion on-board the vehicle. For vehicles with air-breathing propulsion it may be possible to further simplify the on-board trajectory regeneration problem using reduced order modeling methods based on Energy-State methods¹⁰.

We focus in this paper on the development of a strategy for closed-loop trajectory tracking. This strategy is designed to provide for (1) adaptive closed-loop guidance during nominal operation; (2) adaptive closed-loop guidance and guidance command modification (i.e., local trajectory reshaping) to maintain feasible guidance commands in the time period between the occurrence of a failure and the completion of modeling the failure followed (if necessary) by trajectory regeneration for orbit retargeting or abort; and (3) for tracking of the trajectory in all cases when subject to potentially significant variations in the force equations.

The PCH method⁴ is used to implement this approach to direct adaptive guidance and control. An adaptive inner-loop provides fault tolerance as in the X-36 and X-33 applications previously discussed. An adaptive outer-loop has been introduced that provides closed-loop guidance for tracking of the reference trajectory¹¹. The outer-loop adapts to force perturbations, while the inner-loop adapts to moment perturbations. The outer-loop is "hedged" to prevent adaptation to inner-loop dynamics. The hedge also enables adaptation while at control limits, and mitigates any need for frequency separation of inner and outer-loop dynamics.

The paper begins by giving a description of the PCH method and an adaptive inner-loop design. This is followed by a section that describes the development of an adaptive outer-loop for trajectory following. The paper concludes with an application of the developed methods to the X-33 RLV demonstrator simulation.

Adaptive Autopilot Design

The purpose of the Pseudo-Control Hedging (PCH) method is to prevent the adaptive element of a control system from trying to adapt to selected system input characteristics. For example, one would not want an adaptive trajectory following controller to continuously ask for more-and-more plant input when the vehicle is

off course and this plant input is saturated. A plain-language conceptual description of the method is: *The reference model is moved in the opposite direction (hedged) by an estimate of the amount the plant did not move due to system characteristics the control designer does not want the adaptive control element to ‘see’.* To formalize the language in the context of an adaptive control law involving dynamic inversion, ‘movement’ is simply an appropriate system signal. Preventing the adaptive element from ‘seeing’ a system characteristic means to prevent that adaptive element from seeing the system characteristic as model tracking error.

The specific case of the PCH applied to a particular Model Reference Adaptive Control (MRAC) architecture employing approximate dynamic inversion is illustrated in Figure 1. The adaptive element shown in the figure is an online trained NN correcting for errors in the approximate dynamic inversion.

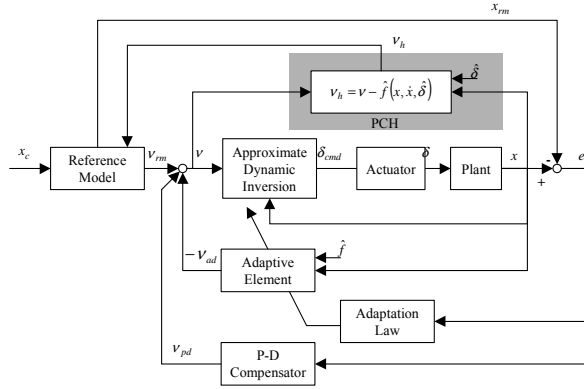


Figure 1 - MRAC including an approximate dynamic inversion; the pseudo-control hedge component utilizes an estimate of actuator position

The design of a suitable NN adaptive element and its associated update law for the controller architecture illustrated in Figure 1 is in the literature, as are proofs of boundedness^{2,4}. The design of the pseudo-control hedge compensator for the controller architecture illustrated in Figure 1 is now described. For simplicity, consider the case of full model inversion, in which the plant dynamics are taken to be of the form

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

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

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

where v is the pseudo-control signal, and represents a desired \dot{x} that is expected to be approximately achieved by δ_{cmd} . That is, this dynamic inversion

element is designed without consideration of the actuator model (i.e., ‘perfect’ actuators). This command (δ_{cmd}) will not equal actual control (δ) due to actuator dynamics.

To get a pseudo-control hedge (v_h), an estimated actuator position ($\hat{\delta}$) is determined based on a model or a measurement. In cases where the actuator position is measured, it is regarded as known ($\hat{\delta} = \delta$ in this case). This estimate is then used to get the difference between commanded pseudo-control and the achieved pseudo-control

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

$$v_h = v - \hat{f}(x, \dot{x}, \hat{\delta}) \quad (4)$$

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

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

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

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

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

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

Note: the pseudo-control hedge signal v_h affects reference model output v_{rm} only through changes in reference model state.

The following subsections discuss the theory associated with this application of PCH. There are two fundamental changes associated with PCH that affect existing boundedness theorems for the NN architecture. Firstly, there is a change to the model tracking error dynamics (e), which is the basis for adaptation laws presented in the earlier work^{2,3}. Secondly, the reference model is not stable by its selection, due to the fact that it is now coupled with the rest of the system.

Model Tracking Error Dynamics

The complete pseudo-control signal for the system introduced earlier is

$$v = v_{rm} + v_{pd} - v_{ad} + v_r \quad (8)$$

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

$$v_{pd} = [K_d \quad K_p] e, \quad (9)$$

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

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

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

$$\dot{e} = Ae + B[v_{ad}(x, \dot{x}, \hat{\delta}) - f(x, \dot{x}, \delta) + \hat{f}(x, \dot{x}, \hat{\delta})] \quad (11)$$

where

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

$$B = \begin{bmatrix} I \\ 0 \end{bmatrix} \quad (13)$$

Model error to be compensated for by v_{ad} is defined as

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

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

$$\dot{e} = Ae + B[v_{ad}(x, \dot{x}, \hat{\delta}) - \Delta(x, \dot{x}, \hat{\delta})] \quad (15)$$

Eqn (15) is of the same form as the model tracking error dynamics seen in previous work^{2,3}. As a result, the boundedness of a NN adaptation law given by earlier results can be used with some modification⁴.

Stability Properties of the Reference Model

Assumptions made up to this point allow the adaptive element to continue to function when the actual control signal has been replaced by *any* arbitrary signal. This completely arbitrary signal does not necessarily stabilize the overall system. Therefore, system characteristics to be removed from the adaptation must be limited to items that are a function of the commanded control, such as saturation, quantized control, linear input dynamics, and latency. This class of system characteristics will be referred to in this section as an actuator model. In general, this actuator model could also be a function of plant state.

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

$$\ddot{x} = \hat{f}(x, \dot{x}, \hat{\delta}) + \Delta(x, \dot{x}, \hat{\delta}) \quad (16)$$

When the actuator is ideal, one obtains

$$\ddot{x} = \hat{f}(x, \dot{x}, \delta_{cmd}) + \Delta(x, \dot{x}, \hat{\delta}) \quad (17)$$

So, when the actuator position and command differ, the adaptation occurs as though the command had corresponded to the actual. Also, the system response is as close to the command as was permitted by the actuator model.

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

Neural Network

In this section, a NN is described for use as the adaptive element (v_{ad}). Single Hidden Layer (SHL) Perceptron NNs are universal approximators in that they can approximate any smooth nonlinear function to within arbitrary accuracy, given a sufficient number of hidden layer neurons and input information. Figure 2 shows the structure of a SHL NN. The following definitions are convenient for further analysis.

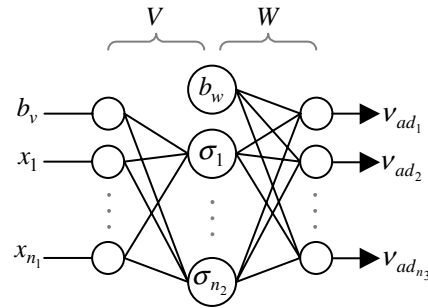


Figure 2 - The Single Hidden Layer (SHL) Perceptron Neural Network

The input-output map of the NN can be expressed as

$$v_{ad_k} = b_w \theta_{w,k} + \sum_{j=1}^{n_2} w_{j,k} \sigma_j \quad (18)$$

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

$$\sigma_j = \sigma \left(b_v \theta_{v,j} + \sum_{i=1}^{n_1} v_{i,j} x_i \right) \quad (19)$$

Here n_1 , n_2 , and n_3 are the number of input nodes, hidden layer nodes, and outputs respectively. The scalar

function σ is a sigmoidal activation function that represents the ‘‘firing’’ characteristics of the neuron, e.g.

$$\sigma(z) = \frac{1}{1 + e^{-az}} \quad (20)$$

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

$$V = \begin{bmatrix} \theta_{v,1} & \cdots & \theta_{v,n_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_1,1} & \cdots & v_{n_1,n_2} \end{bmatrix} \quad (21)$$

$$W = \begin{bmatrix} \theta_{w,1} & \cdots & \theta_{w,n_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{bmatrix} \quad (22)$$

and define a new sigmoid vector

$$\bar{\sigma}(z) = [b_w \quad \sigma(z_1) \quad \sigma(z_2) \quad \cdots \quad \sigma(z_{n_1})]^T \quad (23)$$

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

$$\bar{x} = [b_v \quad x_1 \quad x_2 \quad \cdots \quad x_{n_1}]^T \quad (24)$$

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

$$v_{ad} = W^T \bar{\sigma}(V^T \bar{x}) \quad (25)$$

Consider a SHL perceptron approximation of the nonlinear function Δ , introduced in Eqn (15), over a domain D of \bar{x} . There exists a set of ideal weights $\{W^*, V^*\}$ that bring the output of the NN to within an ε -neighborhood of the error $\Delta(x, \dot{x}, \delta) = \Delta(\bar{x})$. This ε -neighborhood is bounded by $\bar{\varepsilon}$, defined by

$$\bar{\varepsilon} = \sup_{\bar{x}} \|W^T \bar{\sigma}(V^T \bar{x}) - \Delta(\bar{x})\| \quad (26)$$

The universal approximation theorem implies that $\bar{\varepsilon}$ can be made arbitrarily small given enough hidden layer neurons. The matrices W^* and V^* can be defined as the values that minimize $\bar{\varepsilon}$. These values are not necessarily unique. The NN outputs are represented by v_{ad} where W and V are estimates of the ideal weights. Define

$$Z = \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \quad (27)$$

and let $\|\cdot\|$ imply the Frobenius norm. Define the derivative of the sigmoids as

$$\bar{\sigma}_z(z) = \frac{\partial \bar{\sigma}(z)}{\partial z} = \begin{bmatrix} 0 & \cdots & 0 \\ \frac{\partial \sigma(z_1)}{\partial z_1} & & 0 \\ & \ddots & \\ 0 & & \frac{\partial \sigma(z_{n_2})}{\partial z_{n_2}} \end{bmatrix} \quad (28)$$

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

$$r = (e^T P B)^T \quad (29)$$

where $P \in \mathfrak{R}^{2n \times 2n}$ is the positive definite solution to the Lyapunov Equation $A^T P + PA + Q = 0$. Where a reasonable choice for $Q > 0$ is⁴

$$Q = \begin{bmatrix} K_d K_p & 0 \\ 0 & K_d K_p^2 \end{bmatrix} \frac{1}{\frac{1}{4} n_2 + b_w^2} \quad (30)$$

The robustifying signal is chosen to be

$$v_r = -[K_{r0} + K_{r1} (\|Z\| + \bar{Z})] r \quad (31)$$

with $K_{r0}, K_{r1} > 0, \in \mathfrak{R}^{n \times n}$. Previous theoretical results⁴ guarantee uniform ultimate boundedness of tracking errors, weights, and plant states, when \dot{W} and \dot{V} satisfy

$$\dot{W} = -\left\{ \bar{\sigma} - \bar{\sigma}_z V^T \bar{x} \right\} r^T + \lambda \|r\| W \Gamma_w \quad (32)$$

$$\dot{V} = -\Gamma_v \left\{ \bar{x} (r^T W^T \bar{\sigma}_z) + \lambda \|r\| V \right\} \quad (33)$$

with $\Gamma_w, \Gamma_v > 0$ and $\lambda > 0$.

Adaptive Trajectory Following Controller

It is common practice to approach the guidance and control problem by independent design of inner and outer loops. The purpose of the inner-loop is to use the control effectors to achieve a desired attitude and angular velocity with respect to the Earth or to the relative wind (i.e., angle of attack and sideslip angle). The outer-loop generates inner-loop commands to achieve a desired trajectory.

The theory presented thus far has pertained directly to the inner-loop portion of the problem. Introduction of adaptation in a traditional outer-loop that is to be coupled with this adaptive inner-loop is problematic. In particular, adaptation of the outer loop to the dynamics of the inner loop (which will appear to the outer loop as model error) is not desired. Here, PCH is used to remove the effect of inner-loop dynamics and any inner-loop adaptation from the outer loop process. The result is a trajectory following system that can respond to force perturbations while the inner-loop responds to moment perturbations.

A block diagram of the combined inner and outer loops is shown in Figure 3. The outer-loop, which is enclosed by the gray-bordered box on the left-hand side of the figure, provides direct force effector commands (such as engine throttle commands), as well as attitude command adjustments to the inner-loop. The inner-loop, which is enclosed in the gray-bordered box on the right-hand side of the figure, uses moment-generating effectors to achieve the attitude commands generated by the outer loop. A single NN is employed to serve the adaptation needs of both the inner and outer loops. The NN thus has six outputs, which are used to correct for force and moment model errors in each of the axes. In Figure 3, the symbols p and v represent position and velocity respectively. The pseudo-control has been delineated as linear acceleration (a) and angular acceleration (α). The symbol Δq_{OL} represents attitude angle corrections from the outer-loop.

As shown in the figure, the outer-loop reference model is driven by a trajectory prescribed in terms of commanded position and velocity. The filtered trajectory commands, modified by the hedge signal, are combined with the output of proportional plus derivative control of the trajectory following error and the appropriate NN outputs to produce the pseudo-control in each axis of control. As described previously for the inner loop design, the pseudo-control serves as the input to the model inversion process, and the NN output signal serves to cancel the errors that result from the dynamic inversion of an approximate model of the

plant. An equivalent structure is used for the inner loop.

Error dynamics for the integrated system can be written in the form of Eqn (15), so previously developed boundedness results above can be employed. Consider the attitude control problem given by

$$\begin{aligned}\dot{\theta} &= q \\ \dot{q} &= \alpha(p, v, \theta, q, \delta_m, \delta_f)\end{aligned}\quad (34)$$

where θ is vehicle attitude angle (e.g., pitch angle), q is vehicle angular velocity (e.g., pitch angular rate), α is vehicle angular acceleration (note: not angle of attack) which is a function of states and controls, p is vehicle position (e.g., altitude), v is vehicle velocity (e.g., vertical speed), δ_m is the moment control (e.g., idealized elevator), and δ_f is the force control (e.g., flap or thrust). Note that the conventional dependency on angle of attack and sideslip is captured by the dependency on velocity and attitude.

A model for vehicle motion can be used to develop a dynamic inverse of attitude motion, providing an expression for desired moment controls as a function of desired angular acceleration:

$$\begin{aligned}\alpha_{des} &= \hat{\alpha}(p, v, \theta, q, \delta_{m_{des}}, \delta_f) \\ \delta_{m_{des}} &= \hat{\alpha}^{-1}(p, v, \theta, q, \alpha_{des}, \delta_f)\end{aligned}\quad (35)$$

However, actuators have limits, so although $\delta_{m_{des}} = \hat{\alpha}^{-1}$ (moment controls are a function of vehicle

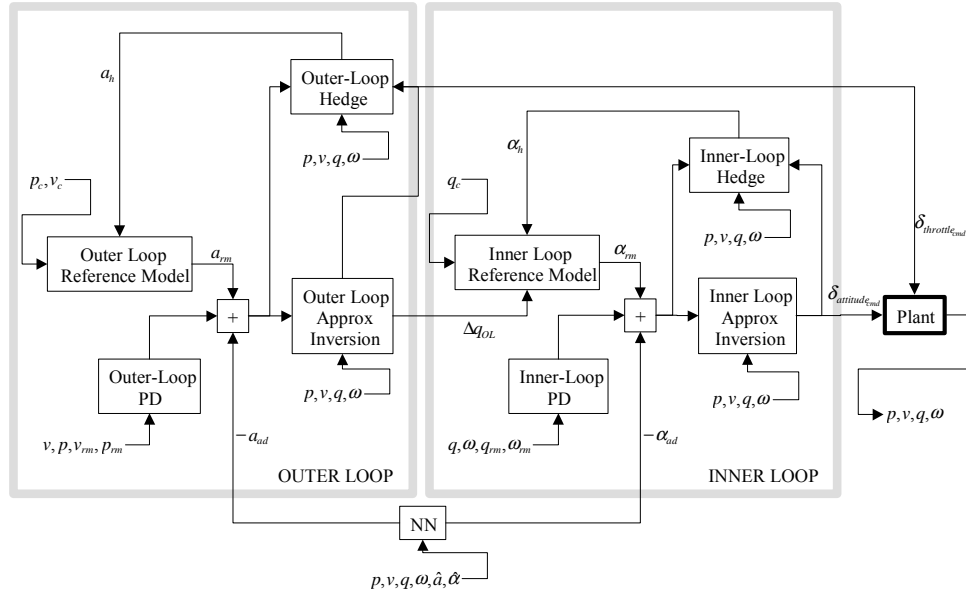


Figure 3 - Inner and outer-loop adaptive flight control architecture that utilizes pseudo-control hedge to de-couple the adaptation process

state and an external command), $\delta_m = \delta_{m_{des}}$ only when sufficient control power exists. Thus α_{des} and $\hat{\alpha}(p, v, \theta, q, \delta_m, \delta_f)$, an estimate of angular acceleration, are not necessarily equal.

One must select the desired angular acceleration. A reference model is introduced (θ_{rm}, q_{rm}) that one would like the plant to match. A reasonable choice for desired angular acceleration given these approximately linearized dynamics is proportional-derivative control

$$\alpha_{des} = K_p(\theta_{rm} - \theta) + K_d(q_{rm} - q) + \alpha_{des_{rm}} - \alpha_{ad} \quad (36)$$

where, K_p is the proportional gain, K_d is the derivative gain, $\alpha_{des_{rm}}$ is the angular acceleration signal associated with the reference model, and α_{ad} is the output of a NN, further discussed later. The reference model desired angular acceleration could be chosen as

$$\alpha_{des_{rm}} = K_p(\theta_c + \theta_{ol} - \theta_{rm}) + K_d(q_c - q_{rm}) \quad (37)$$

where θ_c, q_c is an external reference command (e.g., attitude and angular rate of a reference trajectory) and θ_{ol} is an incremental vehicle attitude angle command due to an outer loop controller, discussed further below.

The reference model states can be updated using the relationships

$$\begin{aligned} \dot{\theta}_{rm} &= q_{rm} \\ \dot{q}_{rm} &= \alpha_{des_{rm}} - [\alpha_{des} - \hat{\alpha}(p, v, \theta, q, \delta_m, \delta_f)] \end{aligned} \quad (38)$$

where the latter term in square brackets is the inner-loop PCH signal. When the PCH and θ_{ol} signals are zero, the reference model will exhibit a linear second order response to external reference attitude commands. This PCH signal will normally be zero when moment actuators are not at limits. This simplifies to

$$\begin{aligned} \dot{\theta}_{rm} &= q_{rm} \\ \dot{q}_{rm} &= -K_p(\theta_{rm} - \theta) - K_d(q_{rm} - q) + \alpha_{ad} + \hat{\alpha}(p, v, \theta, q, \delta_m, \delta_f) \end{aligned} \quad (39)$$

Now consider the outer loop control problem given by

$$\begin{aligned} \dot{p} &= v \\ \dot{v} &= a(p, v, \theta, q, \delta_m, \delta_f) \end{aligned} \quad (40)$$

where a is vehicle *linear* acceleration, a function of states and controls.

A model for vehicle motion can be used to develop a dynamic inverse of linear motion¹¹. This provides an expression for the force controls and incremental

attitude commands as a function of desired linear acceleration:

$$\begin{aligned} a_{des} &= \hat{a}(p, v, \theta_{des}, q, \delta_m, \delta_{f_{des}}) \\ \begin{bmatrix} \delta_f \\ \theta_{ol} \end{bmatrix} &= \hat{a}^{-1}(p, v, \theta, q, \delta_m, \dot{v}_{des}) \end{aligned} \quad (41)$$

However, actuators have limits. Also, one might impose limits on the magnitude of outer loop attitude commands ($\theta_{ol} = \theta_{des} - \theta$). In addition, the plant will not respond instantly to these outer loop attitude

commands (θ_{des}). So although $\begin{bmatrix} \delta_f \\ \theta_{ol} \end{bmatrix} = \hat{a}^{-1}$, a_{des} and $\hat{a}(p, v, \theta, q, \delta_m, \delta_f)$ are not necessarily equal.

A reference model is introduced (p_{rm}, v_{rm}) that one would like the plant to match. A reasonable choice for the approximately linearized dynamics is proportional-derivative control

$$\begin{aligned} a_{des} &= K_{pg}(p_{rm} - p) + K_{dg}(v_{rm} - v) \\ &+ a_{des_{rm}} - a_{ad} \end{aligned} \quad (42)$$

where K_{pg} is the outer loop proportional gain, K_{dg} is the outer loop derivative gain, $a_{des_{rm}}$ is the linear acceleration signal associated with the reference model, and a_{ad} is the output of a neural network, discussed further below.

The reference model desired linear acceleration could be chosen as

$$a_{des_{rm}} = K_{pg}(p_c - p_{rm}) + K_{dg}(v_c - v_{rm}) \quad (43)$$

where p_c, v_c is an external reference command (e.g., position and velocity of a reference trajectory). The reference model states can be updated using the relationships

$$\begin{aligned} \dot{p}_{rm} &= v_{rm} \\ \dot{v}_{rm} &= a_{des_{rm}} - [a_{des} - \hat{a}(p, v, \theta, q, \delta_m, \delta_f)] \end{aligned} \quad (44)$$

where the latter term in square brackets is the outer-loop PCH signal. When this PCH signal is zero, the reference model will exhibit a linear second order response to external reference position commands. This PCH signal will be zero when the inner loop (attitude) responds ‘‘fast’’, when no imposed limitations on attitude commands are enforced, and when force control actuators are not at limits. This simplifies to

$$\begin{aligned} \dot{p}_{rm} &= v_{rm} \\ \dot{v}_{rm} &= -K_{pg}(p_{rm} - p) - K_{dg}(v_{rm} - v) + a_{ad} + \hat{a}(x, v, \theta, q, \delta_m, \delta_f) \end{aligned} \quad (45)$$

Consider a new reference model tracking error vector

defined by

$$e = \begin{bmatrix} p_{rm} - p \\ v_{rm} - v \\ \theta_{rm} - \theta \\ q_{rm} - q \end{bmatrix} \quad (46)$$

which represents the error between the reference model and the plant. A time derivative of this vector is

$$\dot{e} = \begin{bmatrix} v_{rm} - v \\ -K_{pg}(p_{rm} - p) - K_{dg}(v_{rm} - v) + a_{ad} + \hat{a} - a \\ q_{rm} - q \\ -K_p(\theta_{rm} - \theta) - K_d(q_{rm} - q) + \alpha_{ad} + \hat{\alpha} - \alpha \end{bmatrix} \quad (47)$$

which can be re-written as

$$\dot{e} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -K_{pg} & -K_{dg} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -K_p & -K_d \end{bmatrix} e + \begin{bmatrix} 0 \\ a_{ad} + \hat{a} - a \\ 0 \\ \alpha_{ad} + \hat{\alpha} - \alpha \end{bmatrix} \quad (48)$$

This represents linear dynamics (Hurwitz by selection of PD gains), with a forcing function containing neural network output and model error. This system is of the form given in Eqn (15).

A drawback of the architecture described above is that acceleration is not utilized as a feedback signal (although position, velocity, attitude, and angular rates are). One can achieve a considerably more rapid adaptation to force errors if acceleration feedback is utilized to train the NN, since errors are directly measured in this case, rather than eventually showing up as reference model tracking error. With linear-acceleration feedback, a steepest descent approach can be used to train the NN online to correct errors in linear acceleration output. As before, a single NN can still be utilized for both force and moment adaptation. From, Eqn (29) one has

$$r_{1,2,3} = K_p(\theta_{rm} - \theta) + \frac{K_d K_p}{2}(q_{rm} - q). \quad (49)$$

For the linear acceleration components one has the similar expression

$$r_{4,5,6} = K_{pg}(p_{rm} - p) + \frac{K_{dg} K_{pg}}{2}(v_{rm} - v) \quad (50)$$

which does not utilize any linear-acceleration feedback.

A replacement for Eqn (50) which regards linear-

acceleration error as known is

$$r_{4,5,6} = K_{pg}(a_{ad} - a + \hat{a}) \quad (51)$$

where a is the measured linear acceleration of the vehicle. Here, NN weights are updated based on minimizing error (r) by adjusting NN weights, where rate of adjustment is proportional to the sensitivity of the output of the NN to any given weight, given the current NN input. This represents a departure from the theoretical results above. Results presented below utilize Eqn (49) and (51) to determine r .

RLV Numerical Simulation Results

The proposed guidance and control architecture was tested in the Marshall Aerospace Vehicle Representation in C (MAVERIC), which is a guidance and control simulation tool used for RLV research and in another simplified simulation tool. The results presented here correspond to the X-33 and extend from launch to Main Engine Cut-Off (MECO), simulated with MAVERIC. Typical missions include vertical launch and peak Mach numbers of approximately 8, altitudes of 180,000 feet, and peak dynamic pressures of 500 Knots Equivalent Air Speed (KEAS). During ascent, vehicle mass drops by approximately a factor of 3, and vehicle inertia by a factor of 2 due to fuel consumption. The inner and outer-loop flight control architecture illustrated in Figure 3 was used to generate the results that follow. The primary deviation from the description given previously for the idealized example is that the throttle will be open-loop. That is, the nominal throttle command is employed, and linear force adaptation will occur for horizontal and vertical deviations from the nominal trajectory.

The inner-loop approximate inversion consisted of multiplying desired angular acceleration by an estimate of vehicle inertia, and using a fixed-gain control allocation matrix based on the existing baseline X-33 control allocation system¹². The outer-loop approximate inversion is a transformation of acceleration commands to attitude commands, which included only an estimate of the effect of thrust tilt and a fixed linear model for the relationship between aerodynamic-angle changes and aerodynamic force coefficients. This conversion involves estimated thrust, vehicle mass, and dynamic pressure.

NN inputs were angle-of-attack, side-slip angle, bank angle, vehicle angular rate, body-axis velocity, and estimated pseudo-control ($\hat{a}, \hat{\alpha}$). Four middle layer neurons were used; learning rates on W were unity for all axes and learning rates for V were 20 for all inputs. For the inner-loop, K_p and K_d were chosen based on a natural frequency of 1.0, 1.5, and 1.0 rad/sec for the

roll, pitch, and yaw axes respectively and a damping ratio of 0.7. For the outer-loop, they corresponded to 0.5, 0.2, and 0.1 rad/sec for the x, y, and z body axis directions respectively, all with damping ratio of unity.

Aerodynamic surface actuator and main engine thrust vectoring position and rate limits are included in the inner-loop PCH signal. The PCH also has knowledge of axis priority logic within the control allocation system, which appears as input saturation. The implementation included PCH compensation for notch filters that could be designed to prevent excitation of specific aeroelastic modes, although these filters were not used for the results presented here. Closed-loop control is not used for main engine throttling (the nominal schedule is used). The outer-loop hedge design also includes inner-loop dynamics.

Two failure cases are now discussed. The first is a failure simulated by a large perturbation in the normal force coefficient. The second superimposes large moment perturbations due to multiple control surface failures. In all cases the adaptive autopilot (inner loop) described above is employed. Closed-loop trajectory following is also employed in all cases, either with or without the contribution of an adaptive term.

Perturbation in Normal Force Coefficient

In this case, a failure is to be simulated by a fixed perturbation of the normal force coefficient, C_N , by the amount of 0.5. The simulated failure is injected 60 seconds into the flight, and remains for the duration. This is representative of the failure of an aerodynamic fairing or wing. No direct knowledge of the failure is given to the guidance and control system.

Figure 4 presents the nominal trajectory, along with simulation of closed-loop guided flight with the aid of outer-loop adaptation. The plus symbols placed on the trajectory represent 30-second intervals of flight. Sixty seconds into the flight, designated by the circle, the 0.5 perturbation of C_N is introduced. The vehicle responds to the simulated failure by pitching nose down so that the reference trajectory can be maintained.

Figure 5 presents a comparison of the z-axis model error with the outer-loop adaptation signal used to correct for it. The impact of the normal force perturbation introduced at 60 seconds is evident.

Multiple Failures

In this case, the normal force coefficient, C_N , is again perturbed 60 seconds into the flight by 0.5. In addition, the right body flap freezes, and the right outer elevon goes hard-over (trailing edge up).

Figure 6 presents the nominal trajectory in terms of altitude versus downrange. Superimposed is the guided

flight *without* the benefit of outer loop adaptation. Again, the circle indicates the point where the failure is introduced. Dramatic deviation of the flight path occurs, which near the end of the trajectory results in very high dynamic pressures. The corresponding control surface time histories are presented in Figure 7. The excessive control activity in the later half of the flight indicates an impending loss of control. Without outer loop closed-loop guidance, performance is even worse as expected.

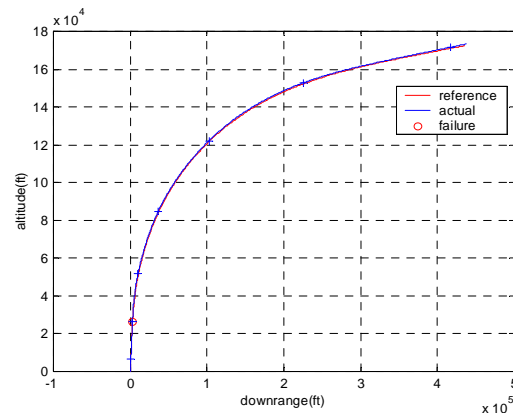


Figure 4 – Altitude versus downrange distance for the nominal trajectory and guided flight with outer-loop adaptation. A 0.5 perturbation of the normal force coefficient was introduced at 60 seconds.

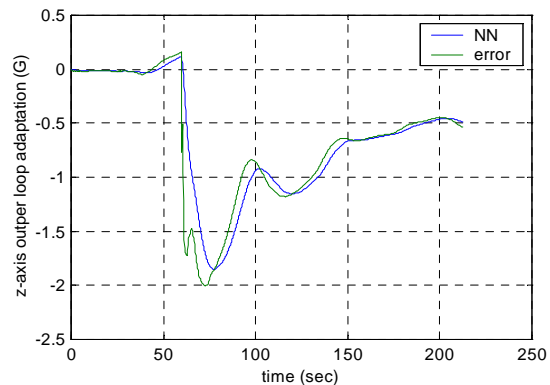


Figure 5 – Z-axis acceleration model error. A 0.5 perturbation of the normal force coefficient was introduced at 60 seconds.

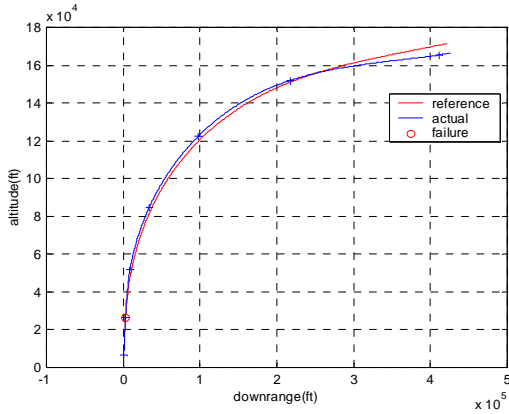


Figure 6 – Altitude versus downrange distance for the nominal trajectory and guided flight without outer-loop adaptation. Multiple failures introduced at 60 seconds.

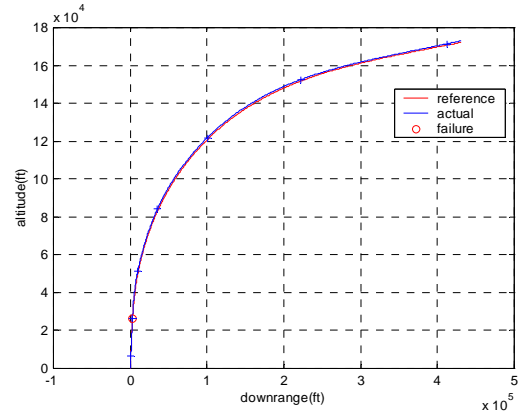


Figure 8 – Altitude versus downrange distance for the nominal trajectory and guided flight with outer-loop adaptation. Multiple failures introduced at 60 seconds.

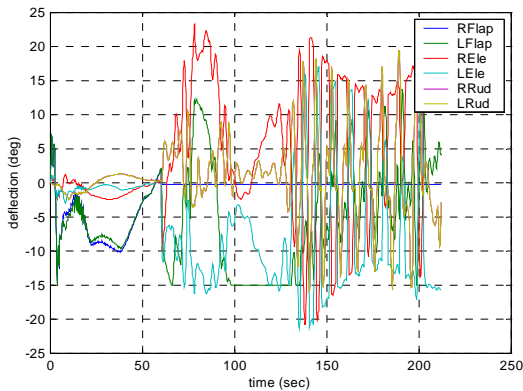


Figure 7 – Control surfaces without outer-loop adaptation. Multiple failures introduced at 60 seconds.

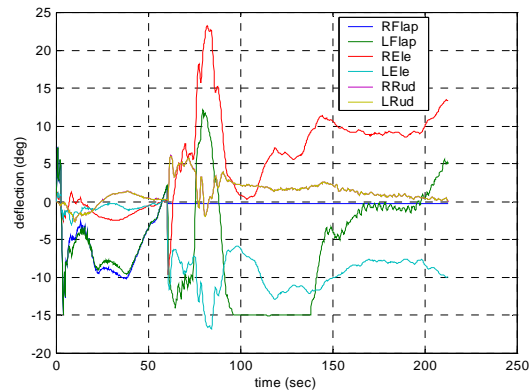


Figure 9 – Control surfaces with outer-loop adaptation. Multiple failures introduced 60 seconds into the flight. Left flap is saturated at -15° for significant portion of flight.

Figure 8 again presents a comparison of X-33 guided flight with the nominal trajectory, but this time aided by adaptation in the outer loop. Despite introduction of the multiple failures at 60 seconds, the reference trajectory is tracked. The corresponding control surface time histories are presented in Figure 9, and are now more reasonable. Note that the right flap is frozen near zero, and that the left flap is saturated at -15° for a significant portion of the flight.

These results indicate the ability of the developed architecture to manage large unknown force and moment perturbations. Also, the trajectory commands were locally altered when required to ensure the prescribed path remained feasible, and used to return the vehicle to the nominal trajectory when sufficient control power was available to do so.

Future Work

Current efforts include the coupling of real-time hybrid trajectory optimization⁷⁻¹⁰ with this adaptive trajectory controller within a single system. Figure 10 is a representative 3-dimensional plot of a nominal trajectory (for an RLV design based on the X-33) produced by the hybrid optimization tool¹³ and flown using the adaptive trajectory controller. Current efforts also include the extension to entry adaptive trajectory following.

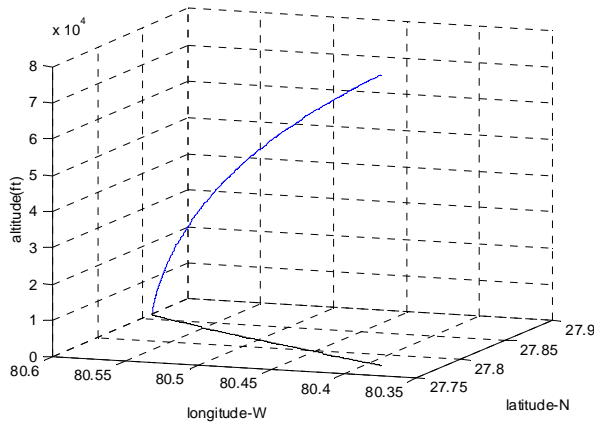


Figure 10 – 3-dimensional plot of a nominal trajectory produced by the hybrid optimization tool and flown using the adaptive trajectory controller.

Conclusions

Previous results relating neural network adaptive control for systems with control authority limitations have been extended to include both inner and outer loop flight control. The result is an inner/outer loop flight control system, which can adapt to both force and moment model errors of systems with realistic control authority limitations. This architecture enables adaptive guidance and control, giving time for system identification and trajectory re-shaping functions that can be performed. The architecture was tested on a simulation of the X-33 reusable launch vehicle technology demonstrator with scenarios that involved both force and moment errors and which caused input saturation. Current efforts include the incorporation of real-time trajectory reshaping tools into a single system.

Acknowledgements

This work was supported in part by the NASA Marshall Space Flight Center, Grant NAG3-1638, and in part by the U.S. Air Force Wright Laboratories, Contract F33615-00-C-3021.

References

- ¹Hanson, J., "Advanced Guidance and Control Project for Reusable Launch Vehicles," AIAA-2000-3957, *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2000, Denver, CO.
- ²Calise, A., 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.
- ³Brinker, J. and Wise, K., "Flight testing of a reconfigurable flight control law on the X-36 tailless

fighter aircraft," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, pp. 903-909.

⁴Johnson, E., Calise, A., Rysdyk, R., and El-Shirbiny, H., "Feedback Linearization with Neural Network Augmentation Applied to X-33 Attitude Control," AIAA-2000-4157, *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2000, Denver, CO.

⁵*Reconfigurable Systems for Tailless Fighter Aircraft – RESTORE*, Final Report, September 1999, Boeing, AFRL-VA-WP-TP-1999-30XX.

⁶*Reconfigurable Systems for Tailless Fighter Aircraft – RESTORE*, Final Report, September 1999, Lockheed Martin, AFRL-VA-WP-TR-1999-3078.

⁷Calise, A.J., Melamed, N., Lee, S., "Design and Evaluation of a 3-D Optimal Ascent Guidance Algorithm," *AIAA Journal of Guidance, Control and Dynamics*, Vol 21, No. 6, Nov.-Dec., 1998, pp 867-875.

⁸Gath, P.F., Calise, A.J., "Optimization of Launch Vehicle Ascent Trajectories with Path Constraints and Coast Arcs," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, March-April 2001.

⁹Calise, A.J., et al, "Further Improvements to a Hybrid Method for Launch Vehicle Ascent Trajectory Optimization," AIAA-2000-4261.

¹⁰Corban, J.E., Calise, A.J., and Flandro, G., "Rapid Near-Optimal Aerospace Plane Trajectory Generation and Guidance," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 6, Nov.-Dec., 1991.

¹¹Johnson, E., Calise, A., and Corban, J., "Reusable Launch Vehicle Adaptive Guidance and Control Using Neural Networks," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2001, Montreal, Canada.

¹²Hanson, 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 and Exhibit*, 1998, Boston, MA.

¹³Calise, A.J., Brandt, N., "Generation of Launch Vehicle Abort Trajectories Using a Hybrid Optimization Method," AIAA-2002-3504, *Guidance, Navigation, and Control Conference*, 2002, Monterey, CA.