

1N-08

26555

P20

Neural Network Application to Aircraft Control System Design

Terry Troudet
Sverdrup Technology, Inc.
Lewis Research Center Group
Brook Park, Ohio

and

Sanjay Garg and Walter C. Merrill
Lewis Research Center
Cleveland, Ohio

Prepared for the
Guidance, Navigation and Control Conference
sponsored by the American Institute of Aeronautics and Astronautics
New Orleans, Louisiana, August 12-14, 1991

(NASA-TM-105151) NEURAL NETWORK APPLICATION
TO AIRCRAFT CONTROL SYSTEM DESIGN (NASA)
20 p CSCL 01C

N91-27167

Unclass
0026555

G3/08

NASA



NEURAL NETWORK APPLICATION TO AIRCRAFT CONTROL SYSTEM DESIGN

Terry Troudet
Sverdrup Technology, Inc.
Lewis Research Center Group
Brook Park, Ohio 44142

Sanjay Garg and Walter C. Merrill
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Abstract. The feasibility of using artificial neural networks as control systems for modern, complex aerospace vehicles is investigated via an example aircraft control design study. The problem considered is that of designing a controller for an integrated airframe/propulsion longitudinal dynamics model of a modern fighter aircraft to provide independent control of pitch rate and airspeed responses to pilot command inputs. An explicit model-following controller using H_∞ control design techniques is first designed to gain insight into the control problem as well as to provide a baseline for evaluation of the neurocontroller. Using the model of the desired dynamics as a command generator, a multilayer feedforward neural network is trained to control the vehicle model within the physical limitations of the actuator dynamics. This is achieved by minimizing an objective function which is a weighted sum of tracking errors and control input commands and rates. To gain insight in the neurocontrol, linearized representations of the neurocontroller are analyzed along a commanded trajectory. Linear robustness analysis tools are then applied to the linearized neurocontroller models and to the baseline H_∞ based controller. Future areas of research are identified to enhance the practical applicability of neural networks to flight control design.

1 Introduction. In the past few years, there has been an increasing interest in the control community to exploit the promise of artificial neural networks to solve difficult control problems. However, most of the neural network applications to control design that have appeared in the literature [1,2], either dealt with robotic systems, or with control problems that are mainly of academic interest such as the inverted pendulum problem. Only more recently have neural networks been applied to the control design of more complex problems, e.g. manufacturing process [3]. The objective of this paper is to investigate the applicability of neural networks as controllers for aerospace vehicles with special emphasis on piloted

flight. Towards this objective, results are presented from a preliminary study of neurocontrol design for an integrated airframe/propulsion model of a modern fighter aircraft for the piloted longitudinal landing task. To gain insight in the characteristics of the neurocontroller, linear analysis tools are applied to linearized representations of the neurocontroller and to a baseline H_∞ based controller. Closed loop system performance and robustness of the neurocontroller are evaluated and discussed in relation to the H_∞ based controller.

The paper is organized as follows. The vehicle model and the desired closed-loop dynamics are first discussed, and an explicit model-following H_∞ based control design is presented. The architecture used to train the neurocontroller is then presented and the results of the neurocontroller are evaluated. A performance and robustness analysis is then presented for the neurocontroller and the H_∞ based controller.

2 Vehicle Model. The vehicle model consists of an integrated airframe and propulsion system state-space representation for a modern fighter aircraft powered by a two-spool turbofan engine and equipped with a two-dimensional thrust-vectoring and reversing nozzle.

The flight condition used in this application is representative of the STOL (Short Take-off and Landing) approach-to-landing task, with an airspeed of $V_0 = 120$ Knots, a flight path angle of $\gamma_0 = -3$ deg, and a pitch attitude of $\theta_0 = 7$ deg. The linearized dynamics of the vehicle model are of the form

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}_a, \quad \bar{z} = C\bar{x}; \quad (1)$$

where the state vector is

$$\bar{x} = [u, w, Q, \theta, h, N2, N25, P6, T41B]^T, \quad (2)$$

with

u = aircraft body axis forward velocity (ft/sec)
 w = aircraft body axis vertical velocity (ft/sec)

Q = aircraft pitch rate (rads/sec)
 θ = pitch angle (rads)
 h = altitude (ft)
 $N2$ = engine fan speed (rpm)
 $N25$ = core compressor speed (rpm)
 $P6$ = engine mixing plane pressure (psia)
 $T41B$ = engine high pressure turbine blade temperature ($^{\circ}R$),
 and the control input vector is

$$\bar{u}_a = [WF, \delta TV]^T; \quad (3)$$

with

WF = engine main burner fuel flow rate (lbm/hr)

δTV = nozzle thrust vectoring angle (deg).

The vehicle outputs to be controlled are

$$\bar{z} = [V, Q]^T, \quad (4)$$

where V is the aircraft velocity in ft/sec, and Q is the pitch rate in deg/s. The system matrices A , B , and C are available in Ref.[4]. The open-loop vehicle eigenvalues are:

$$\lambda_1 = 0.07, \lambda_{2,3} = -0.09 \pm j0.23, \lambda_4 = 1.06,$$

$$\lambda_5 = -1.47 \quad \text{--- Airframe modes}$$

and

$$\lambda_6 = -1.40, \lambda_7 = -3.57, \lambda_8 = -6.96,$$

$$\lambda_9 = -89.28 \quad \text{--- Propulsion modes.}$$

Note that the airframe is statically unstable with a highly unstable pitch mode. Open loop analysis also indicated a strong coupling in the response of the controlled outputs \bar{z} to control inputs \bar{u}_a .

The control design objective is to design a control system that provides decoupled command tracking of velocity and pitch rate from pilot control inputs with aircraft responses compatible with Level I handling qualities requirements [5]. The desired response dynamics are selected to be of the form

$$\dot{\bar{x}}_m = A_m \bar{x}_m + B_m \bar{z}_{SEL}, \quad \bar{z}_c = C_m \bar{x}_m; \quad (5)$$

with $\bar{z}_{SEL} = [V_{SEL}, Q_{SEL}]^T$ where V_{SEL} is the pilot velocity command in ft/s and Q_{SEL} is the pilot longitudinal stick deflection in inches, and $\bar{z}_c = [V_c, Q_c]^T$, where the subscript "c" refers to the ideal response in V and Q with units of ft/s and deg/s respectively. The system matrices A_m , B_m and C_m are the state-space representation of the ideal response transfer functions listed in Table 1.

Table 1: Desired Response Transfer Functions.

Notation: $\{k(1/\tau)/[\xi; \omega_n] \equiv k(s + 1/\tau)/(s^2 + 2\xi\omega_n s + \omega_n^2)\}$
$\frac{V_c}{V_{SEL}} = \frac{0.04(3.13)}{[0.89; 0.36]} : \frac{V_c}{Q_{SEL}} = 0.$
$\frac{Q_c}{V_{SEL}} = 0; \frac{Q_c}{Q_{SEL}} = \frac{35.12(0.5)}{[0.89; 2.24]}$

Actuator models were also used in control design and evaluation. The fuel flow actuator was modelled as

$$G_{WF}(s) = \frac{10}{s + 10} \cdot \frac{50}{s + 50}; \quad (6)$$

with a maximum fuel flow rate $|WF|_{max} = 10,000 \text{ lbm/hr}$, and a rate limit $|WF|_{max} = 20,000 \text{ lbm/hr/s}$. Note that the fuel flow here corresponds to the perturbation from the trim value for the linear model. In this study, the value $|WF|_{max}$ is therefore chosen such that the total fuel flow limit will not be exceeded when a perturbation of a magnitude of WF_{max} is commanded. The thrust vectoring actuator is modelled as

$$G_{\delta TV}(s) = \frac{15}{s + 15}; \quad (7)$$

with a maximum thrust vector angle $|\delta TV|_{max} = 10 \text{ deg}$, and a rate limit $|\delta TV|_{max} = 20 \text{ deg/s}$.

As a result, *nonlinearities* appear in the control design and evaluation in the form of actuators position and rate limits.

3 H_{∞} Control Design. Recent advances in H_{∞} control theory [6] and computational algorithms to solve for H_{∞} optimal control laws [7] have enabled the application of this theory to practical complex multivariable control design problems. Many example applications of H_{∞} based control designs for aerospace vehicles have appeared in recent literature [8-10]. Prior to applying a neural network approach to control design for the example vehicle under study, an H_{∞} based control law was obtained as a baseline for the performance and robustness analysis of the neurocontroller.

Within the framework of H_{∞} optimization, the control design problem for this example study was formulated as the model-following problem shown in Fig.1. The three transfer functions that are of interest for such a problem are the sensitivity function $S(s)$, the complementary sensitivity function $T(s)$, and the control transmission function $C(s)$. These represent the transfer functions from the reference commands to tracking errors, controlled variables, and commanded control inputs respectively, i.e. $\bar{e}(s) = S(s)\bar{z}_c(s)$, $\bar{z}(s) = T(s)\bar{z}_c(s)$ and $\bar{u}_c(s) =$

$C(s)\bar{z}_c(s)$. In order to be able to influence both the low-frequency and high-frequency properties of the closed-loop system, it is desirable to find a controller $K(s)$ which minimizes a weighted norm of a combination of these three transfer functions, i.e.:

$$\min \|H(j\omega)\|_{\infty} \text{ with } H(j\omega) = \begin{bmatrix} W_S(j\omega) \cdot S(j\omega) \\ W_T(j\omega) \cdot T(j\omega) \\ W_C(j\omega) \cdot C(j\omega) \end{bmatrix} \quad (8)$$

The weighting functions $W_S(j\omega)$, $W_T(j\omega)$ and $W_C(j\omega)$ are the "knobs" used by the control designer to "tune" the controller $K(s)$ such that the design objectives are met. For instance, choosing W_S to be large at low frequencies ensures good command tracking performance, and choosing W_T to be large at high frequencies ensures robustness to high frequency unmodelled dynamics. W_C is chosen to ensure that control actuation bandwidths, as well as rate and deflection limits, are not exceeded in the control design.

For the aircraft example, the integrated design model, $P(s)$, in Fig.1 consisted of the vehicle model (1) and the actuator models (6) and (7). The ideal response model, $R(s)$, in Fig.1 consisted of the desired model dynamics (5) with a high pass filter ($\frac{s}{s+0.1}$) on the pilot pitch rate command. This high pass filter is added to reflect the fact that pitch rate cannot be commanded in steady-state. The outputs \bar{z} and the errors \bar{e} were scaled by their approximate maximum values to be commanded by the pilot with $V_c^0 = 20$ ft/sec and $Q_c^0 = 3$ deg/sec. The sensitivity weights W_S and the complementary sensitivity weights W_T were chosen as listed in Table 2.

Table 2: Weights for H_{∞} Control Design.

Controlled Variable	W_S	W_T
V	$\frac{33.50s+1000}{335.01s+1}$	$\frac{0.22s}{0.0022s+1}$
Q	$\frac{6.70s+1000}{67.02s+1}$	$\frac{0.044s}{0.00044s+1}$

This choice of W_S and W_T was based on the performance and robustness arguments discussed earlier. The weights W_C consisted of the control commands and rates weighted by the inverse of actuator position and rate limits for WF and δTV listed earlier. Note that the combination of tracking errors \bar{e} and aircraft outputs \bar{z} is used as a controller input instead of \bar{e} and ideal response, \bar{z}_c , to avoid control saturation due to large pilot inputs and undue amplification of inadvertent pilot command noise.

The H_{∞} control design plant as discussed above is of 21st order consisting of the 9th order aircraft model, 2nd order WF actuator model, 1st order δTV

actuator model, 5th order ideal response model, and 1st order W_S and W_T for the two controlled variables. The resulting 21st order H_{∞} optimal controller obtained using the solution algorithm of Ref.[6] was reduced to 13th order by residualizing the high order modes. The maximum eigenvalue of the reduced order controller is $|\lambda|_{max} = 6.83$ rad/sec, which implies that the controller can be implemented digitally with reasonable sampling rates. With this reduced-order controller, the performance results in terms of closed-loop response, control effort and control rate requirements, are shown in Figs.2 and 3 for two cases of pilot command inputs: (1) $V_{SEL} = -20$ ft/s for $t > 0$, $Q_{SEL} = 0.5$ in for $0 < t \leq 3$ sec and $Q_{SEL} = 0$ in for $t > 3$ sec; (2) $V_{SEL} = 20$ ft/s for $t > 0$ and Q_{SEL} same as for command input case 1. From Fig.2, we note that for the pilot command input in case 1 the velocity response obtained with the controller is quite close to the ideal response, and the control input commands and rates are reasonable. For the pilot command input in case 2, the pitch rate response is quite similar to that for case 1; however, the velocity response is degraded from the ideal response. Case 2 is demanding in that the pilot is commanding the aircraft to pitch up as well as accelerate to a higher velocity. As seen in Fig.3, the maximum fuel flow rate is commanded by the controller for an extended period of time in order to track the ideal response. Note that the closed-loop system remains stable in the presence of the actuator limits, and the aircraft velocity response tracks the ideal response in the steady-state.

4 Neurocontrol Design. Although the strength of neural networks lies in their ability to handle nonlinearities in the controlled dynamics, the control design for a linear aircraft model is being considered in this paper to gain insight into the neural network characteristics by using linear analysis tools. As discussed earlier, nonlinearities of concern for practical control design, such as actuator position and rate limits, are included in the design criteria.

The architecture for training the neurocontroller is shown in detail in Fig.4. For each pilot selected trajectory $\bar{z}_{SEL}(t)$, a commanded trajectory $\bar{z}_c(t)$ is generated from (5). Prior to training, the commanded variables $\bar{z}_c(t)$ are discretized and scaled to $\bar{z}_c^*(t_k)$ using the same scaling as for the H_{∞} design. Likewise, the dynamics of the actuators and of the vehicle model are discretized and scaled after normalizing the control input vector by its maximum value ($|WF|_{max}$, $|\delta TV|_{max}$). As for the H_{∞} design, the tracking error at time t_k is the error between the scaled vehicle output vector and its desired scaled value at the same time t_k , i.e. $\bar{e}_s(t_k) = \bar{z}_c^*(t_k) - \bar{z}^*(t_k)$. However, because of the time-discretization of the ac-

tuator dynamics and vehicle model dynamics within the training loop, a commanded control input vector generated at time t_k by the neurocontroller will only affect the aircraft output at time t_{k+2} . Consequently, the tracking error at time t_{k+2} defines the magnitudes of the weights increments at time t_k . Said in another way, due to the time-discretization of the dynamics, the internal representation of the neurocontroller has to be updated at time t_k on the basis of information which will be only available at a later time t_{k+2} . To be consistent with the time-discretized design, knowledge of the anticipated commanded vehicle output at time t_{k+2} , $\bar{z}_c^*(t_{k+2})$, is explicitly provided to the neural network at time t_k during training by means of the commanded error $\bar{e}_z(t_k) = \bar{z}_c^*(t_{k+2}) - \bar{z}^*(t_k)$. This procedure ensures that the proper action will be commanded by the neurocontroller at time t_k to achieve the desired tracking at time t_{k+2} during training. When operating the trained neural network in closed-loop however, the tracking error $\bar{e}_z(t_k)$ will be used as input to the neurocontroller instead of the commanded error $\bar{e}_z(t_k)$ which is not available in the real simulation because it requires knowledge of future pilot command inputs. This means that the trained neural network will be tracking the exact commanded trajectory with a two-step time delay during simulation evaluation. Since the neurocontroller operates in the continuous time domain, this two-step time delay should not adversely affect performance in closed-loop evaluation. That such is the case was confirmed by the closed-loop evaluation results to be presented later.

As shown in Fig.4, the two commanded control inputs are calculated by a two hidden-layer feedforward neural network with eight input units (or four pairs of fan-out units associated to the Q and V variables), and two neurons in the output layer. These pairs consist of the scaled output vector $\bar{z}^*(t_k)$; the commanded error $\bar{e}_z(t_k)$ between the scaled vehicle output vector at time t_k and its desired scaled value at time t_{k+2} ; the discrete time-derivative of the tracking error, $\dot{\bar{e}}_z(t_k)$; and the time-average of the tracking error, $1/t_k \int_0^{t_k} \bar{e}_z(t) dt$. As in the H_∞ design, the motivation behind using the combination of $\bar{z}^*(t_k)$ and $\bar{e}_z(t_k)$ as inputs to the neurocontroller, instead of $\bar{z}^*(t_k)$ and $\bar{z}_c^*(t_{k+2})$, is to allow the neural network to reconstruct the command without direct feedforward of the command. The role of the error rates $\dot{\bar{e}}_z(t_k)$ is to provide the neural network with lead information, and the time-averaged error feedback $1/t_k \int_0^{t_k} \bar{e}_z(t) dt$ is to minimize the steady-state tracking error for step command inputs. (The motivation behind scaling the integral error $\int_0^{t_k} \bar{e}_z(t) dt$ into its time-average was to improve backpropagation learning by bounding the

corresponding input to the neural network. Other alternatives would be to low-pass filter the integral error itself, or to remove the scaling factor $1/t_k$ from the time-averaged error as learning takes place. Because of their potential to improve steady-state tracking, these latter approaches should be considered in future neurocontrol designs.) In Fig.4, the symbol Δ represents a latch that is clocked every δt seconds to update the inputs to the neurocontroller, the actuators and the vehicle model. A network configuration of 15 neurons in the first hidden layer, and 10 neurons in the second hidden layer, is chosen for the neurocontroller. Each neuron of the neurocontroller has the activation function:

$$y = \tanh(x); \quad (9)$$

which limits its output y to the interval $[-1, +1]$ for any input signal x . For a given set of weights of the neural network, the two output neurons yield the normalized commanded control input vector

$$\bar{u}_c^*(t_k) = \left[\frac{WF_c}{|WF|_{max}}, \frac{\delta TV_c}{|\delta TV|_{max}} \right]^T. \quad (10)$$

which is applied to the scaled actuators. After a small time-interval $\delta t = t_{k+1} - t_k$, the actuators yield the normalized actuator control output vector $\bar{u}_a^*(t_{k+1})$ as defined by (6) and (7). The normalized actuator control output vector $\bar{u}_a^*(t_{k+1})$ is subsequently applied as input to the scaled vehicle model over the time-interval $[t_{k+1}, t_{k+2}]$, and changes the state vector of the vehicle model from $\bar{x}(t_{k+1})$ to $\bar{x}(t_{k+2})$. In order to maximize the tracking performance while minimizing the costs associated with high control effort and high control rate requirements, the neural network is trained to minimize an objective function that includes tracking errors, control effort and control rate requirements

$$J(t_k) = \frac{1}{2} (\bar{e}_z^T(t_{k+2}) \cdot \bar{\rho} \cdot \bar{e}_z(t_{k+2}) + \bar{u}_a^{*T}(t_{k+1}) \cdot \bar{\lambda} \cdot \bar{u}_a^*(t_{k+1}) + \dot{\bar{u}}_a^{*T}(t_{k+1}) \cdot \bar{\mu} \cdot \dot{\bar{u}}_a^*(t_{k+1})) \quad (11)$$

where $\bar{e}_z(t_{k+2})$ is the error between the scaled commanded vector $\bar{z}_c^*(t_{k+2})$ and the scaled vehicle output $\bar{z}^*(t_{k+2})$. The matrices $\bar{\rho}$, $\bar{\lambda}$ and $\bar{\mu}$ are 2×2 diagonal matrices whose coefficients can be adapted so as to modify the characteristics of the neurocontroller in order to achieve a practical performance/control-effort trade-off. Expression (11) is of the same form as the objective function used in Ref.[11] to design a neurocontroller for the same airframe/propulsion system, but without simulating the actuator dynamics within the training loop. In Ref.[11], it was found

that training the neural network to minimize only the tracking error led to high control effort and high control rate requirements. When the actuator dynamics were included in the closed-loop evaluation, this resulted in a highly oscillatory pitch rate response and a limit cycle behavior in velocity/fuel-flow response. However, a satisfactory trade-off between tracking performance and control effort could be achieved with finite values of $\bar{\lambda}$ and $\bar{\mu}$. Since the bandwidth limiting effect of the actuators is now explicitly taken into account within the training loop, much improvement in performance/control-effort trade-off is expected from the minimization of (11).

The backpropagation algorithm [12] was used to find the set of weights of the neurocontroller which minimize the objective function (11) over the set of pilot input commands. In order to backpropagate (11), a single layer feedforward neural network (perceptron) was used in place of the vehicle model in the training architecture of Fig.4. This neural network emulator had 11 input units (corresponding to the two normalized actuator control outputs and to the nine state variables of the vehicle model), and 9 linear output neurons (corresponding to the nine state variables of the vehicle model). Likewise, two feedforward neural networks were used to emulate the discretized dynamics of the actuators. The second-order dynamics of the fuel flow actuator were simulated by a three-layer network of linear and *linear-thresholding* neurons. As shown in Fig.5, constraining fuel flow effort and fuel flow rate requirements is achieved by thresholding the linear neurons of the two last layers. The first-order dynamics of the thrust vectoring actuator were simulated by the two-layer neural network shown in Fig.6. Constraining the effort and rate requirements of the thrust vectoring actuator is achieved by means of linear-thresholding neurons.

The layers of an $(N + 1)$ -layer neural network can be labeled by an index p from 0 to N , $p = 0$ denoting the input layer. Layer p has $\nu(p)$ elements consisting of $[\nu(p) - 1]$ neurons and one unit that is permanently "on" and used to define the thresholds of the neurons of the $(p + 1)^{th}$ layer. With symmetric activation functions of the type (9), the threshold of a neuron is defined as the value of its input signal above which its output is positive, and below which its output is negative. During training, the thresholds are updated with backpropagation in a manner similar to the updating of the weights [12].

The weight connecting the i^{th} neuron of the p^{th} layer to the j^{th} neuron of the $(p + 1)^{th}$ layer is denoted as $w_{j,(p+1):i,p}$. The threshold of the j^{th} neuron of the $(p + 1)^{th}$ layer thus corresponds to $w_{j,(p+1):\nu(p),p}$. For a single feedforward pass of the

neural network, a weight increment is given by

$$\delta w_{j,(p+1):i,p} = \alpha \alpha_{i,p} \Delta_{j,(p+1)} \quad (12)$$

where α is the steepest descent coefficient, $\alpha_{i,p}$ is the output of the i^{th} neuron of the p^{th} layer, and $\Delta_{j,(p+1)}$ is the effective error at the output of the j^{th} neuron of the $(p + 1)^{th}$ layer. The effective errors $\Delta_{k,(p+2)}$ in the $(p + 2)^{th}$ layer are backpropagated to the $(p + 1)^{th}$ hidden layer to give the effective errors in the $(p + 1)^{th}$ layer, as

$$\Delta_{j,(p+1)} = f'(x_{j,(p+1)}) \times S_{j,(p+1)}$$

with

$$S_{j,(p+1)} = [\sum_{k=1}^{\nu(p+2)} \Delta_{k,(p+2)} w_{k,(p+2):j,(p+1)}] \quad (13)$$

and where $f'(x_{j,(p+1)})$ is the value of the derivative of the neural activation function for an input $x_{j,(p+1)}$ of the j^{th} neuron of the $(p + 1)^{th}$ layer. In the output layer, the effective errors $\Delta_{j,N}$ are the gradients of the objective function (11)

$$\Delta_{j,N} = f'(x_{j,N}) \frac{\partial J}{\partial o_{j,N}} \quad (14)$$

Whenever the neural activation is not differentiable over the range of all possible neuron input values (as is the case for the linear-thresholding neurons used for emulating the actuators), f' should be constructed to preserve the characteristics of a monotonous continuous function. For example, the linear-thresholding activation function which is defined as

$$\begin{aligned} f_{lth}(x) &= x \quad \text{if } |x| \leq 1, \\ f_{lth}(x) &= 1 \quad \text{if } x \geq 1, \\ f_{lth}(x) &= -1 \quad \text{if } x \leq -1. \end{aligned} \quad (15)$$

is clearly not differentiable over $[-\infty, +\infty]$. Since f_{lth} is piecewise differentiable, it would seem a-priori natural to define f_{lth}' as $f_{lth}'(x) = 1$ if $|x| \leq 1$, and $f_{lth}'(x) = 0$ if $|x| > 1$. With this definition of f_{lth}' however, any time a neuron input x_0 would take a value outside of $[-1, +1]$ during training, the neuron output would remain trapped to 1, if $x_0 > 1$, or -1, if $x_0 < -1$. For such neuron input values, the weights of the incoming connections would remain frozen, and this would bias the learning. In order to permit the neurons full access to the output state space during training, f_{lth}' is thus defined as

$$\begin{aligned} f_{lth}'(x_{i,p}) &= 1 \quad \text{if } |x_{i,p}| \leq 1 \quad \text{or if } x_{i,p} \cdot S_{i,p} < 0, \\ f_{lth}'(x_{i,p}) &= 0 \quad \text{otherwise.} \end{aligned} \quad (16)$$

which will ensure that the weights be properly incremented during training. $S_{j,p}$ which appears in (16) is defined in (13). The serial arrangement of the neurocontroller, the neuro-emulator of the actuators, and the neuro-emulator of the vehicle model, constitutes a larger neural network through which the objective function (11), $J(t_k)$, can be backpropagated through time [2] using Eqs.(13)-(16). The connections between neurocontrollers and neuro-emulators which were used as backpropagating channels are indicated in Fig.7 over a period of three time-steps δt , and the weights increments are calculated using (12).

The commanded trajectories used to train the neural network were generated as follows. The pilot selected pitch rate was a doublet centered at a time t_c between 2.5s and 5s, with the characteristics: $Q_{SEL}(t) = Q_0$ for $t \leq t_c$; $Q_{SEL}(t) = -Q_0$ for $2t_c \geq t > t_c$; $Q_{SEL}(t) = 0$ for $t > 2t_c$. Note that Q_{SEL} corresponds to pilot longitudinal stick deflection with units in inches. The pilot selected airframe velocity was a step function characterized by $V_{SEL}(t) = 0$ for $t \leq 0$ and $V_{SEL}(t) = V_0$ for $t > 0$. The maximum intensities $|Q_0|$ and $|V_0|$ of the randomly selected input commands were bounded by $Q_{max} = 0.5$ in and $V_{max} = 20$ ft/s. This maximum value of Q_{SEL} corresponds to a maximum pitch rate command of about 3 deg/sec. Random sets of input trajectories were generated from uniform distributions of Q_0 , t_c and V_0 over $[-Q_{max}, Q_{max}]$, $[2.5s, 5s]$ and $[-V_{max}, V_{max}]$ respectively. The commanded variables $Q_c(t)$ and $V_c(t)$ were filtered from $Q_{SEL}(t)$ and $V_{SEL}(t)$ over a period of 12s with a time-step $\delta t = 0.02s$. These types of commanded trajectories represent typical pilot command inputs.

Training was performed in two phases. In the gross-tuning phase of the training, a set of 4000 commanded trajectories was randomly generated, and the synaptic weights were updated at every time $t_k = k\delta t$ after backpropagating $J(t_k)$ through the neural network. This was done once for each trajectory of the training data set with a steepest-descent coefficient $\alpha = 0.001$. In the fine-tuning phase of the training, the synaptic weights were updated following a moving-window scheme: at every time t_k , the weights were incremented after backpropagating through the neural network the time-integral of the objective function calculated over n_w sampled points or during a period of $n_w \cdot \delta t$ seconds, i.e. $\sum_{i=1}^{n_w} J(t_{k+i})$. As the width of the moving window was progressively increased to cover an entire commanded trajectory, i.e. $n_w = 12sec/0.02sec = 600$, the steepest descent coefficient α was progressively reduced from the initial value of 0.001 to 0.0001. In total, the neurocontroller was trained with approximately 10,000 commanded

trajectories.

5 Neurocontrol Performance. The evaluation architecture of the neurocontroller in closed-loop is shown in Figure 8. The neurocontroller was tested on *step* pitch rate input commands, different from the *doublets* used in training. The input commands chosen to illustrate the neurocontrol performance were defined by the step pitch rate command $Q_{SEL}(t) = 0.5$ in for $t \leq 3sec$, $Q_{SEL}(t) = 0$ for $t > 3sec$; applied simultaneously with one of the following classes of step velocity commands: $V_{SEL}(t > 0) = -20ft/sec$ (case 1); $V_{SEL}(t > 0) = 20ft/sec$ (case 2).

When training the neural network without giving any consideration to the cost associated with large control efforts and large control rates, i.e. $\bar{\lambda} = \bar{\mu} = \bar{0}$ in Eq.(11), the neurocontroller learns very satisfactorily to track the commanded outputs. However, the fuel flow is quite irregular, and both control input commands generated by the neurocontroller ride the actuator rate limits. A study of the trade-off between tracking performance and control effort requirement was conducted by training the neural network with $\bar{\lambda}$ and $\bar{\mu}$ of the form $\bar{\lambda} = diag[\lambda_{WF}, \lambda_{\delta TV}]$ and $\bar{\mu} = diag[\mu_{WF}, \mu_{\delta TV}]$, with the same training characteristics and the same matrix elements of \bar{p} used earlier. As in Ref.[11], the tracking error is found to actually decrease for small increases in values of $\bar{\lambda}$ and $\bar{\mu}$.

The results from this trade-off study are shown in Figs.9-10 for cases 1 and 2 with the choice of parameters $\bar{p} = diag[\rho_V, \rho_Q] = diag[2000, 20]$, $\lambda = 0.01$, $\mu = 0.1$. The pitch rate response follows the commanded trajectory very smoothly, in spite of the thrust vectoring requirement δTV reaching the actuator rate limit at the initiation and end of the command. However, within the proposed training scheme, any attempt to lower the rate of thrust vectoring by increasing $\mu_{\delta TV}$ resulted in a loss of tracking performance. In case 1, neurocontrol is very satisfactory both in pitch rate and velocity response. In case 2, neurocontrol tracking is still very satisfactory in pitch rate response, but is slightly less satisfactory in velocity response owing to the physically demanding effort of increasing simultaneously aircraft speed and pitch angle.

In order to estimate the effect of providing the neurocontroller with lead information during training, the above process was repeated without feeding the discrete time-derivatives of the tracking error, i.e. $\dot{e}_z(t_k)$, to the neural network during training. Without constraining control efforts and rates ($\bar{\lambda} = \bar{\mu} = \bar{0}$), the tracking performance deteriorated significantly with the appearance of some ringing in

the pitch rate response and a limit cycle behavior in the velocity/fuel-flow response. The fuel flow requirement and fuel flow rate were both much more oscillatory than when lead information was provided to the neurocontroller during training. The fuel flow rate oscillated between the maximum and minimum rate limit during and beyond the 12 sec training period. A more oscillatory behavior was also noted for the control effort and rate of the thrust vectoring. However, the situation improved significantly when constraints on control efforts and rates were applied during training. In this case, a satisfactory trade-off between performance and control-effort was reached for values of $\bar{\lambda}$ and $\bar{\mu}$ in the vicinity of $\lambda_{WF} = \lambda_{\delta TV} = 0.02$, $\mu_{WF} = 0.2$ and $\mu_{\delta TV} = 1.0$. The results showed a similar velocity/fuel-flow response with and without lead information, but showed a noticeable degradation in the pitch-rate/thrust-vectoring response in comparison to the situation where lead information was provided to the neurocontroller. This degradation in tracking performance resulted from the large value of the pitch rate constraint $\mu_{\delta TV}$ (one order of magnitude larger than before), which was needed to decrease the tracking overshoots. In summary, lead information enabled the neurocontroller to overcome ringing and limit cycle behavior while increasing tracking performance. Thus, within the present scheme of neural computation, any dynamic characteristics required to achieve desirable performance had to be incorporated into the neural network with an appropriate choice of inputs. An extension of the present neural architecture to generate such dynamic characteristics could be a *feedforward* neural network with *intermediate feedback inputs*, i.e. a recurrent neural architecture as a dynamic neurocontroller.

6 Analysis of the Controllers. From a comparison of the closed-loop response for the two command cases with the H_{∞} based reduced order controller (Figs.2 and 3) and the neurocontroller (Figs.9 and 10), it is evident that the neurocontroller provides improved command tracking although at the expense of increased control rate activity, both for δTV and WF . Also the pitch vectoring control requirements are higher and the fuel flow activity exhibits oscillatory behavior for the neurocontroller.

Note that the results presented so far have been with the nominal vehicle model used for control design. Since this model is only a simplified version of the vehicle dynamics, an important criterion for design of controllers for flight vehicles is that of robustness. Robustness is defined here as maintaining performance and stability in the presence of uncertainties associated with the modelling process. Modelling uncertainties are due to neglected high order

dynamics, parameter changes due to change in flight conditions and the margin of error associated with estimating model parameters based on analytical tools and experimental data. A classic specification for robustness, also used in the military specifications for design of flight control systems [5], is that of stability margins, specifically gain and phase margin [14]. The tools to determine these margins are fairly well developed for linear systems - classical Bode analysis for single-input single-output systems [14] and modern singular value and structured singular value analysis for multi-input multi-output systems [15, 16]. For nonlinear systems, one way to determine robustness is to conduct Monte Carlo type simulations using all possible combinations of modelling uncertainties that can be expected. Another approach is to linearize the closed-loop system at various points along a given trajectory and then apply the linear analysis tools. The latter approach is less time consuming and provides more insight into the characteristics of the nonlinear system. Furthermore, this latter approach allows to perform a similar analysis for the linear H_{∞} based reduced order controller and the nonlinear neurocontroller, for small perturbations along a given trajectory.

Since the vehicle model used in this analysis is linear, only linear small perturbation models of the neurocontroller at different points along a given trajectory are needed to perform the type of robustness analysis discussed earlier. Considering the closed loop system response with the neurocontroller for the case 2 command inputs, corresponding to the results presented in Fig.10, the linear neurocontroller models were generated at times $t = 0.5, 2, 4, 6, 8$ and 10 secs. The first three points in time correspond to transient control activity whereas the last three represent steady-state type command tracking with monotonically decreasing tracking error. Note that the neurocontroller as shown in Fig.8 consists of 4 sets of scaled (normalized) inputs: the time-averaged errors $1/t \int_0^t \bar{e}(t) dt$, the error rates $\dot{\bar{e}}_z(t)$, the errors $\bar{e}_z(t)$ and the controlled outputs $\bar{z}^*(t)$. The scaling, the time-averaged error and derivative action were embedded within the neurocontroller during the linearization process to find a control structure consistent with the structure of the H_{∞} based controller which has only the errors (\bar{e}) and the controlled outputs (\bar{z}) as the inputs. The frequency response Bode plots of the linearized neurocontroller models were obtained to gain insight into the characteristics of the control action. Bode gain plots for the thrust vectoring angle (δTV) response to all the inputs to the controller linearized at $t = 0.5$ sec are shown in Fig.11. The Bode gain plots for the H_{∞} based controller are shown in

Fig.12. An example variation in the neurocontroller characteristics with the change in magnitude of the inputs to the controller along the trajectory is shown in Fig.13 in terms of the Bode gain plots for pitch rate error (e_Q) to thrust vectoring angle (δTV) response. Fig.13 shows that the neurocontroller gains decrease with time. This type of behavior was exhibited by all the other input/output Bode plots of the linearized neurocontroller models. So in effect, the neurocontroller can be thought of as a set of linear controllers with the controller parameters being a strong function of the magnitude and direction (relative magnitude) of the inputs to the controller. Note that since the H_∞ based controller is linear, its dynamics are independent of the magnitudes of the controller inputs.

From Fig.11 we note that the neurocontroller exhibits PID (Proportional + Integral + Derivative) control type behavior from the error inputs (e_V and e_Q) to the thrust vectoring angle (δTV) output. This was also the case for the e_V and e_Q to WF response, and was true all along the trajectory as shown partially (for e_Q input) by the plots in Fig.13. This dynamic behavior of the neurocontroller for the error inputs is directly due to allowing feedback of the integral and derivative errors. Since no such dynamics were added to feedback of V and Q to the neurocontroller, the neurocontroller exhibits only proportional type behavior from these inputs.

Comparing Figs.11 and 12, we first note that the magnitude of the e_V and e_Q to δTV response is much lower for the H_∞ based controller compared to the particular linearized neurocontroller models. This was also true for the error inputs to WF response. This result is a further confirmation that the control effort and control rate requirements to track a given set of commands will be higher for the neurocontroller. Although the dynamic behavior of the H_∞ based controller is more complex than the neurocontroller, some integral and derivative action is evident in the e_Q to δTV response. The integral action was built into the H_∞ based controller through the choice of the sensitivity weighting, however, unlike for the neurocontrol design the error rate information was not explicitly provided in the H_∞ controller. The H_∞ control synthesis procedure is such that it naturally builds in the amount of lead (error rate) information into the controller that is necessary to meet the control design objectives specified through the weighted quantities. As evident from Figs.11 and 12, the H_∞ based controller provides lead at a lower frequency in the e_Q to δTV response as compared to the linearized neurocontroller.

Another difference between the H_∞ based con-

troller and the neurocontroller is the compensation from the measurements of the controlled plant outputs (V and Q) to the control inputs (WF and δTV). As mentioned earlier, this compensation is a "constant" (varying with input magnitude) gain from the controller inputs to outputs for the linearized neurocontroller. However, as seen from Fig.12, the H_∞ based controller has dynamics associated with this part of the control compensation and also has higher compensation gains than the linearized neurocontroller (Fig.11). The controller structure used for the H_∞ and the neurocontrol design is consistent with the classical approach of flight control design wherein an inner loop compensation ($\bar{z} \rightarrow \bar{u}$) is designed first to provide stability augmentation and place the augmented plant dynamics within the handling qualities specifications; and then the outer loop compensation ($\bar{e} \rightarrow \bar{u}$) is designed to provide decoupled command tracking to reduce pilot workload. The significance of the difference between the H_∞ based controller and neurocontroller "inner loop" compensation was studied further by considering failures in the outer compensation loops, i.e failure in the error sensors. Eigenvalue analysis showed that the closed-loop system with H_∞ based controller will remain stable for failures in any or both of the error sensor loops whereas the closed-loop system with the neurocontroller linearized at $t = 0.05$ sec was unstable for failure in either or both of the error loops. The response of the closed-loop system for case 2 commands and failure in the e_Q loop is shown in Fig.14 for both the H_∞ based controller and the nonlinear neurocontroller. The H_∞ based controller still tracks the velocity command and provides stable response in pitch rate whereas the neurocontroller gives a highly unstable response. So the H_∞ based controller is using the plant measurements (\bar{z}) in a manner consistent with the classical idea of providing inner loop plant augmentation. How to formulate the neurocontrol design problem such that the resulting controller exploits the plant measurement information to provide inner loop stability augmentation is an area of future research.

Stability margin analysis was performed for the linearized neurocontroller models and the H_∞ based controller to quantify robustness of the control designs. Among the linearized neurocontroller models, stability margins were worst for the one linearized around $t = 0.05$ sec, so only those results are discussed here. Structured singular value analysis [17] showed that the H_∞ based controller has guaranteed multivariable gain margins of -3.7 to 6.6 dB (gain factor of 0.65 to 2.1) and phase margins of ± 30 deg for simultaneous loop gain or phase changes at the plant

output (V and Q) and margins of -3.8 to 7.2 dB and ± 32.5 deg at the plant input (WF and δTV). For the linearized neurocontroller, these multivariable margins were only -0.6 to 0.6 dB and ± 3.4 deg for loop gain variations at the plant output, and -0.9 to 1.1 dB and ± 6.6 deg at the plant input. The low stability margins with the neurocontroller are indicative of poor robustness in that the closed loop system might be unstable for small uncertainties in the plant dynamics. Since the multivariable margins can sometimes be conservative, the stability robustness of the closed-loop system was further evaluated using the more classical approach of "breaking" one loop at a time, i.e. one loop open and other loops closed. This one-loop-at-a-time analysis confirmed the poor stability margins of the neurocontroller. The closed-loop response of the system with the H_∞ based controller and the nonlinear neurocontroller for an added delay of $\tau_d = 0.05$ sec in the two control channels (WF and δTV) is shown in Fig.15. This value of τ_d corresponds to a phase loss of 8 deg at a frequency of 3 rads/sec, which is the frequency that corresponds to the guaranteed multivariable phase margin of 6.6 deg for the linearized neurocontroller, and it is quite representative of the kinds of time delays to be expected in practical implementation of complex flight control designs. From Fig.15 we note that the H_∞ based control shows very little degradation in tracking performance in the presence of time delay, whereas the neurocontroller exhibits limit cycle behavior in the pitch controlled variable. A factor that may contribute to this lack of robustness is the fact that the neuro-command rides the thrust vectoring rate limit during initial and final transients. In contrast, the neuro-command is well below the fuel flow rate limit, which results in robust velocity tracking in the presence of time-delay. Improving phase robustness characteristics of neurocontrollers and investigating their gain robustness characteristics are areas that warrant further study.

In the neurocontrol design, the weights of the neural network (the *internal representation* of the neurocontroller) were chosen to minimize the objective function (11) over an exhaustive set of pilot input commands to the nominal vehicle model by using the backpropagation algorithm. No information on modelling uncertainties and no constraint on "off-nominal" actuator dynamics were provided to the neural network during training. Without any constraint other than control effort and rate limits, the trained neural network learned to control the nominal vehicle model as efficiently as possible (and within the resolution of backpropagation). Consequently, the robustness of the neurocontroller as trained in section 4

is mostly subject to the generalization ability of the backpropagation algorithm (in the present context, *generalizing* means *providing stable control for "off-nominal" vehicle model dynamics that were not used during training*). Because backpropagation is known in general to have a limited ability to generalize [18], the robustness of the neurocontroller as trained in section 4 could have been expected to be quite limited.

Within the neural architecture of Fig.4, one possible approach to enhance the robustness of the neurocontroller may be to include all modelling uncertainties in the training data set. Another possibility might be to modify the objective function (11) used to train the neurocontroller to reflect some of the characteristics of the functional (8) which is minimized in the H_∞ based control design.

7 Conclusions. The applicability of neural networks for flight control design was analyzed through the process of designing a model-following neurocontroller for the example of an integrated airframe/propulsion model of a modern fighter aircraft for the piloted longitudinal landing task. For this two control inputs - two control outputs example, the control design problem was set up as the task of following the trajectories generated from a model of the desired vehicle response dynamics to pilot command inputs. The neurocontroller was trained by simulating the non-linear dynamics of the actuators including position and rate limits. The choice of the objective function and its minimization over entire commanded trajectories were found to be critical to the neurocontrol design. A satisfactory trade-off between tracking performance and control effort could be achieved by an appropriate selection of the weights of the objective function.

The neurocontroller shows better performance than a baseline H_∞ based controller designed for the same command tracking problem. However, the neurocontroller commands larger control rates than the H_∞ based controller, specially for thrust vectoring where the neuro-command rides the thrust vectoring rate limit during initial and final transients. The possibility of improving the practicality of the proposed neurocontrol design methodology, to prevent neuro-commands from riding actuator rate limits without significant degradation of tracking performance, is currently being investigated in light of the results from the minimization of the H_∞ based control design.

To gain further insight into the neurocontroller characteristics, linearized small perturbation representations of the neurocontroller were obtained at different time points along a trajectory correspond-

ing to a demanding set of tracking commands. A linear analysis of these linearized neurocontroller models and the H_∞ based controller showed some differences in the controller characteristics. The major difference between the two controllers is that the H_∞ based controller is a "fixed" dynamic controller whose dynamics are "automatically" determined through the synthesis procedure such that the specified criterion is met in the best possible manner, whereas the neurocontroller is an input-output mapping which is highly dependent on the magnitude and direction of the inputs and any desired dynamic characteristics have to be built into the neurocontroller by appropriate selection of inputs. For instance, both the H_∞ based controller and the neurocontroller have lead characteristics (rate feedback) from the tracking error measurements to the control commands; however, the lead characteristic was a result of the synthesis procedure for the H_∞ based controller which used only errors as inputs, whereas for the neurocontroller this lead characteristic could be obtained only by providing error rate as explicit inputs (measurements). Developing neurocontrol design methodologies that can synthesize the dynamics needed by the neurocontroller to achieve the desired performance is an area of future research. A possible approach may lie in the use of recurrent neural architectures.

Linear stability robustness analysis tools were applied to the linearized neurocontroller models and to the baseline H_∞ based controller. These analysis tools showed that the neurocontroller will have very poor stability margins as compared to the H_∞ based controller. The poor phase margins for the neurocontroller were confirmed in simulation wherein time delays of 0.05 sec in both control channels resulted in a limit cycle pitch response with the neurocontroller, while there was little performance degradation with the H_∞ based controller. Since the issue of robustness is critical to practical implementation of flight control systems, a future area of research is to develop methodologies for the synthesis of robust neurocontrollers, and tools to analyze their robustness.

References.

- [1] Special Section on Neural Networks for Control Systems, *IEEE Cont. Sys. Mag.*, Vol.9, No.3, pp. 25-59, April 1989.
- [2] Special Issue on Neural Networks in Control Systems, *IEEE Cont. Sys. Mag.*, Vol.10, No.3, pp 3-87, April 1990.
- [3] White, D., and Sofge, D., "Neural Network Based Process Optimization and Control", *29th IEEE Conference on Decision and Control*, Honolulu, HI, Dec. 1990.
- [4] Garg, S., Mattern, D.L., and Bullard, R.E., "Integrated Flight/Propulsion Control System Design Based on a Centralized Approach", *Journal of Guidance, Control and Dynamics*, Vol.14, No.1, Jan.-Feb. 1991.
- [5] "Military Specification - Flying Qualities of Piloted Airplanes", MIL-F-8785C, USAF, WPAFB, OH, Nov. 1980.
- [6] Doyle, J.C., Glover, K., Khargonekar, P.P., and Francis,

- B.A., "State-Space Solutions to Standard H_2 and H_∞ Control Problems", *IEEE Trans. on Automatic Control*, Vol.34, No.8, Aug. 1989, pp.831-847.
- [7] "MATRIXx Robust Control Module", Integrated Systems Inc., Santa Clara, CA, Dec. 1989.
- [8] Kammer, I., Khargonekar, P.P., and Robel, G., "Design of Localizer Capture and Track Modes for a Lateral Autopilot using H_∞ Synthesis", *IEEE Cont. Sys. Mag.*, Vol.10, No.4, pp. 13-21, June 1990.
- [9] Reichart, R., "Application of H_∞ Control to Missile Autopilot Design", *AIAA Paper 89-3550*, Guidance, Navigation and Control Conf., Boston, MA, Aug. 1989.
- [10] Garg, S., Mattern, D.L., Bright, M.M., and Ousts, P. J., " H_∞ Based Integrated Flight/Propulsion Control Design for a STOVL Aircraft in Transition Flight", *AIAA Paper 90-3335*, Guidance, Navigation and Control Conf., Portland, OR, Aug. 1990.
- [11] Troudet, T., Garg, S., Mattern, D.L., and Merrill, W. C., "Towards Practical Control Design Using Neural Computation", *Int. Joint Conf. on Neural Networks*, Seattle, WA, July 1991.
- [12] Rumelhart, D. E., McClelland, J. L., et al.: *Parallel Distributed Processing - Exploration in the Microstructure of Cognition, Volume 1: Foundations.*, MIT Press, Cambridge, MA, 1986.
- [13] Jordan, M. I.: *Generic Constraints On Underspecified Trajectories*, *Int. Joint Conf. on Neural Networks*, Vol.1, p.217, Washington D.C., June 1989.
- [14] Ogata, K.: *Modern Control Engineering*, Prentice Hall Inc., 1970.
- [15] Lehtomaki, N. A.: *Practical Robustness Measures in Multi-variable Control System Analysis*, Ph.D. Thesis (LIDS-TH-1093), M.I.T., Cambridge, MA, May 1981.
- [16] Doyle, J. C.: *Structured Uncertainty in Control System Design*, *Proceedings of the 24th Conf. on Decision and Control*, Ft. Lauderdale, FL, Dec. 1985, pp. 260-265.
- [17] Apkarian, P. R.: *Structured Stability Robustness Improvement by Eigenspace Assignment Techniques: A Hybrid Methodology*, *Journal of Guidance, Control and Dynamics*.
- [18] Troudet, T., and Merrill, W. C.: *Neuromorphic Learning of Continuous Valued Mappings from Noise Corrupted Data*, *IEEE Trans. on Neural Networks*, Vol.2, No.2, March 1991.

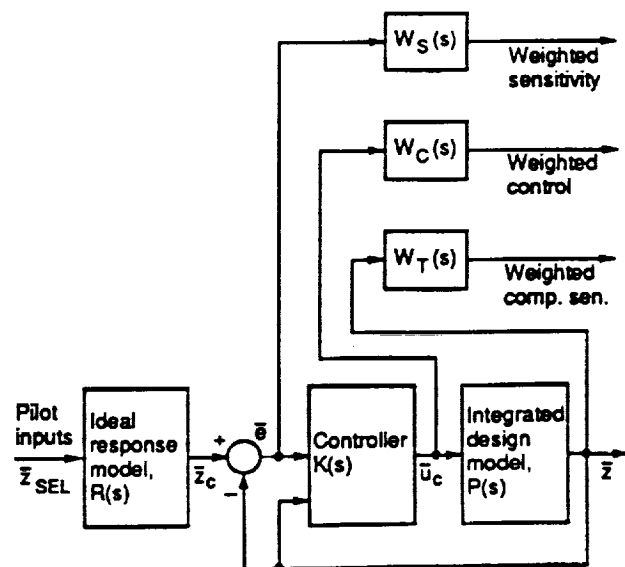


Figure 1.—Block diagram for H_∞ control design.

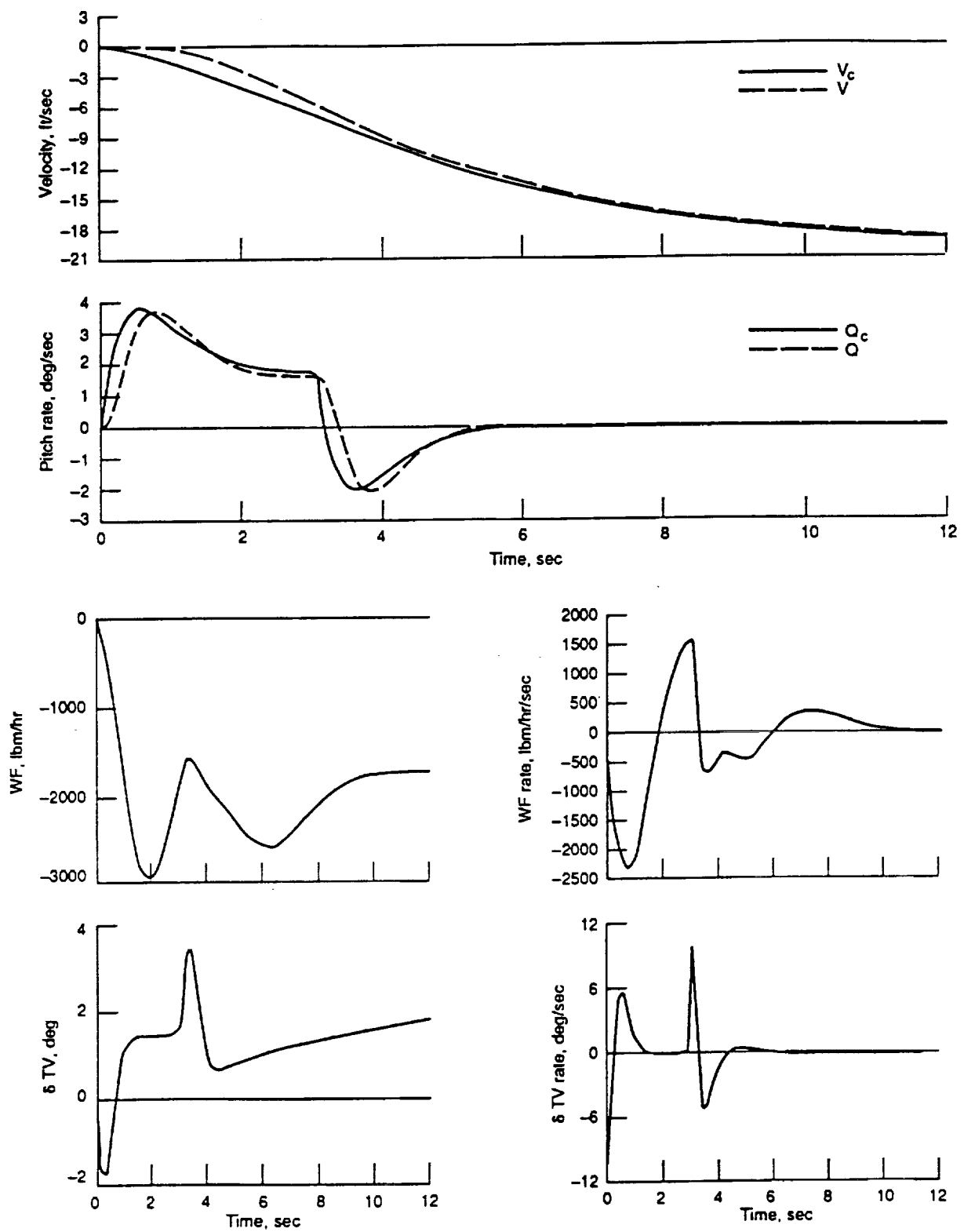


Figure 2.—Closed-loop response and control and control rate requirements with H_{∞} reduced-order controller; case 1.

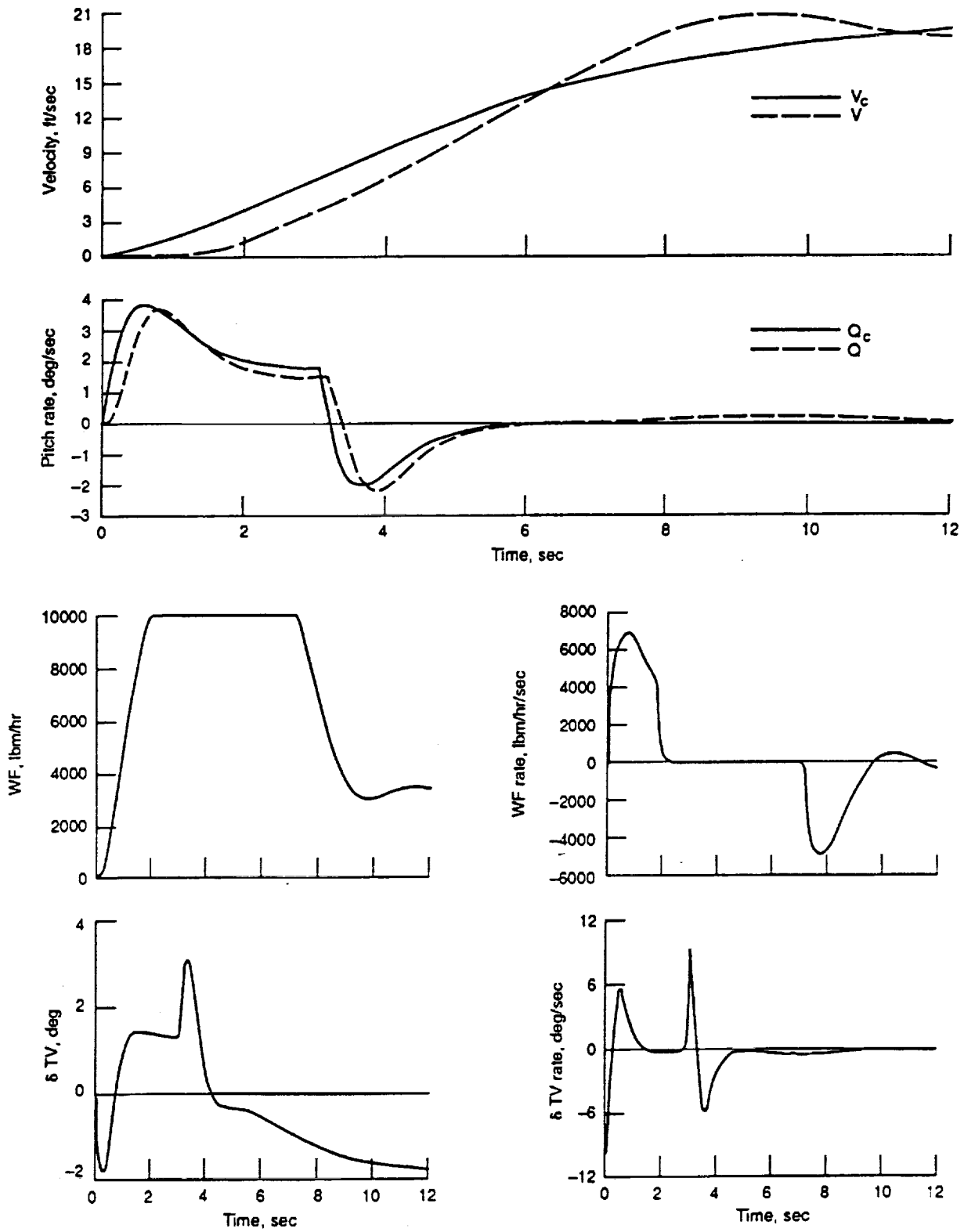


Figure 3.—Closed-loop response and control and control rate requirements with H_∞ reduced-order controller; case 2.

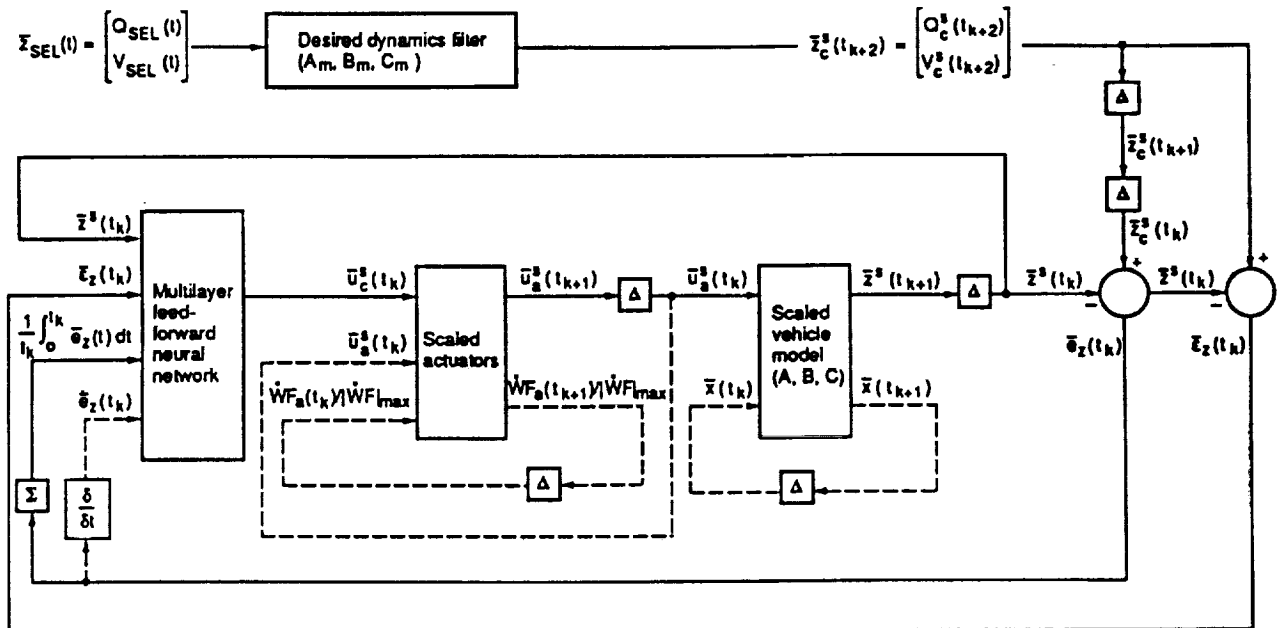


Figure 4.—Training architecture.

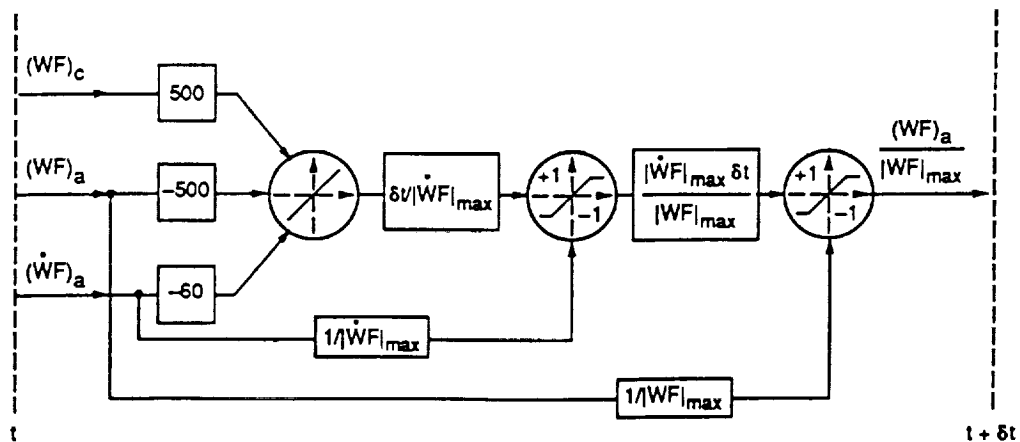


Figure 5.—Neural emulation of fuel flow actuator dynamics.

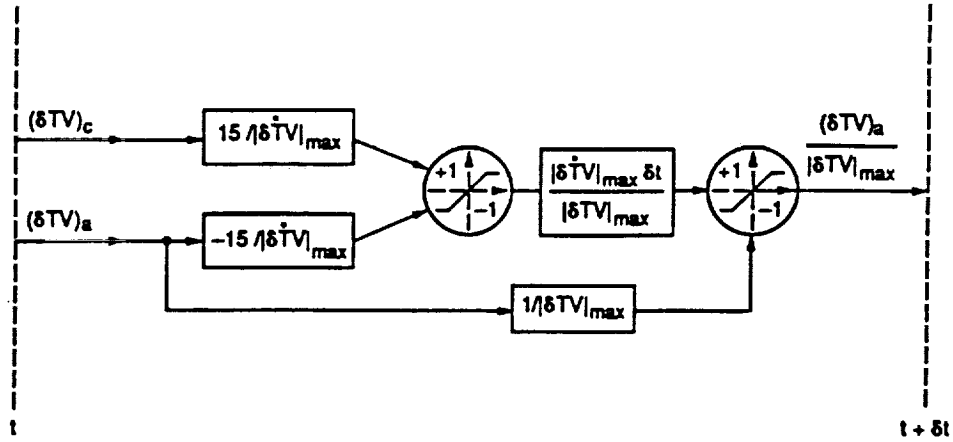


Figure 6.—Neural emulation of thrust vectoring actuator dynamics.

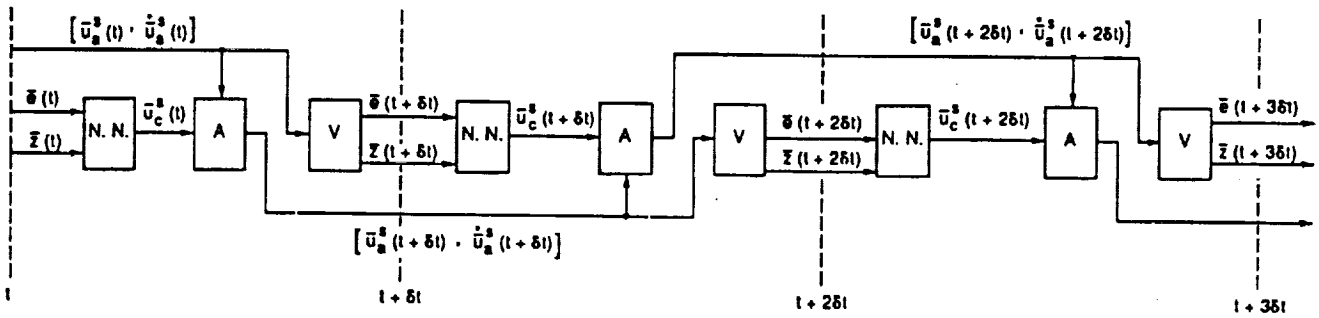


Figure 7.—Data paths of backpropagation of the error during training.

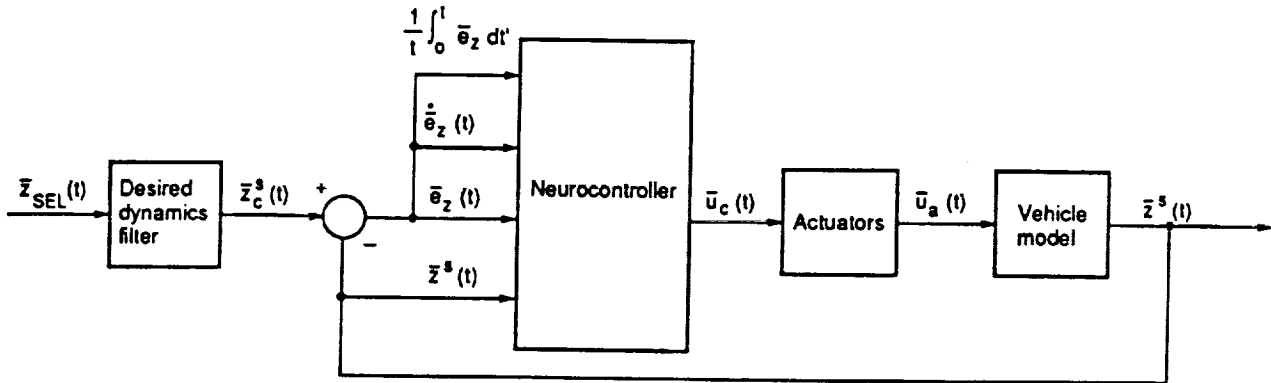


Figure 8.—Evaluation architecture of closed-loop neurocontroller.

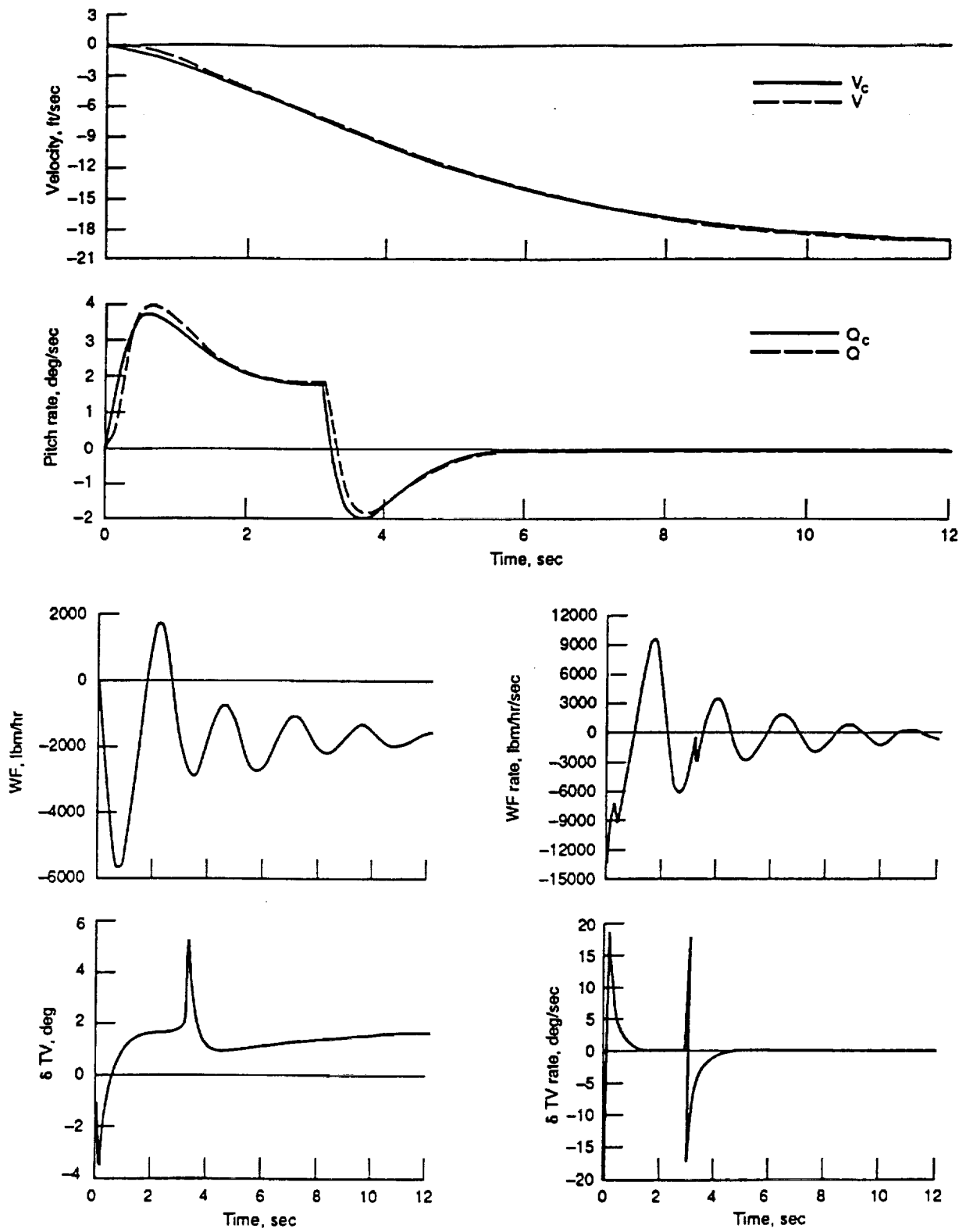


Figure 9.—Closed-loop response and control and control rate requirements with neurocontroller; case 1.

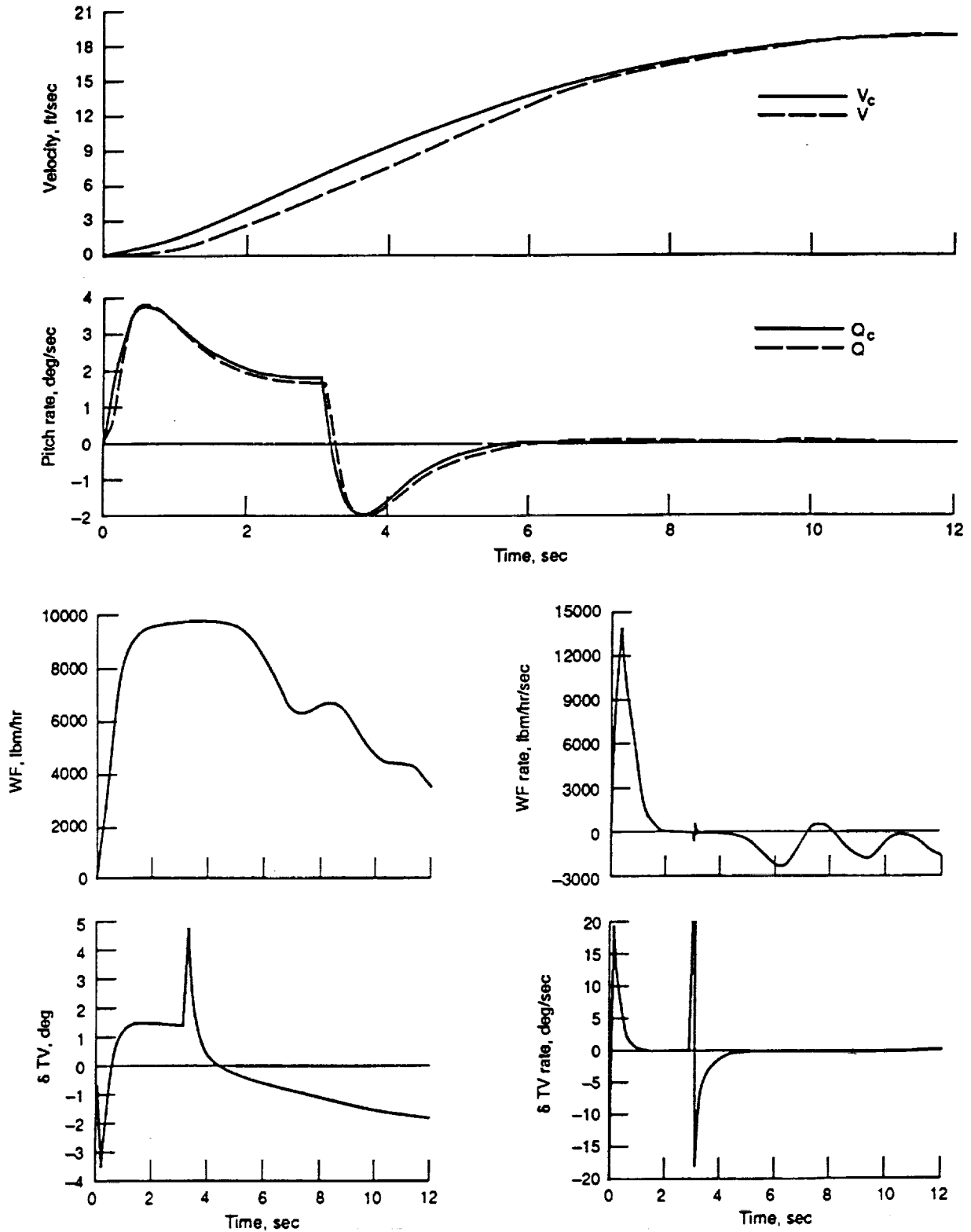


Figure 10.—Closed-loop response and control and control rate requirements with neurocontroller; case 2.

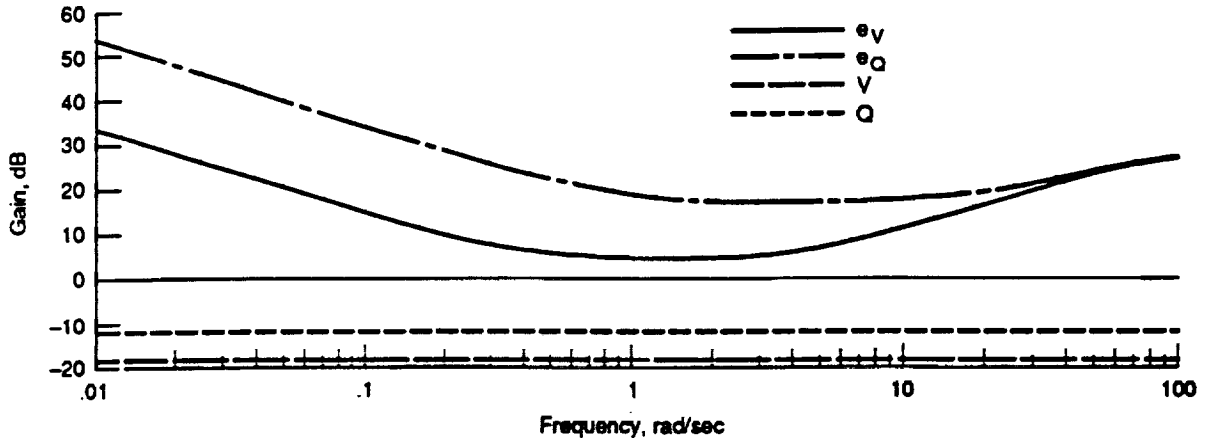


Figure 11.—Bode gain plots for neurocontroller linearized at $t = 0.5$ sec. All inputs to δTV .

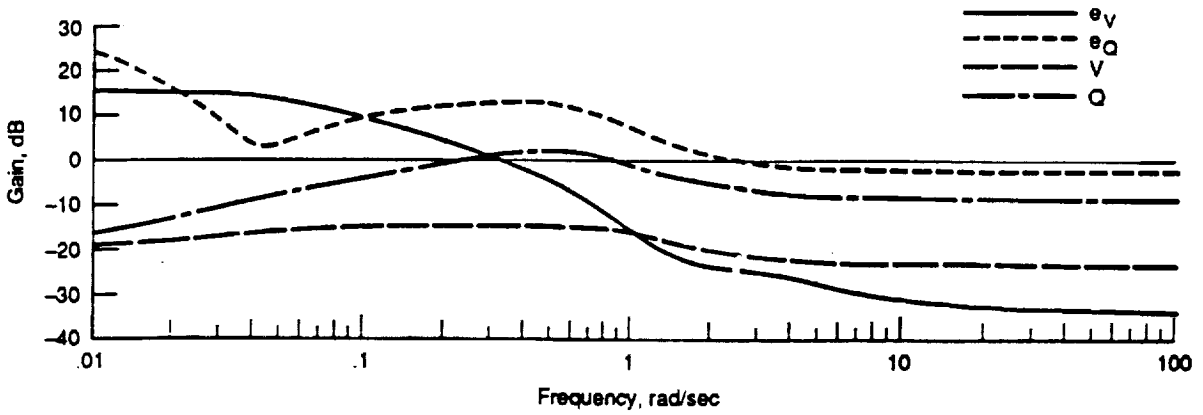


Figure 12.—Bode gain plots for H_∞ based controller. All inputs to δTV .

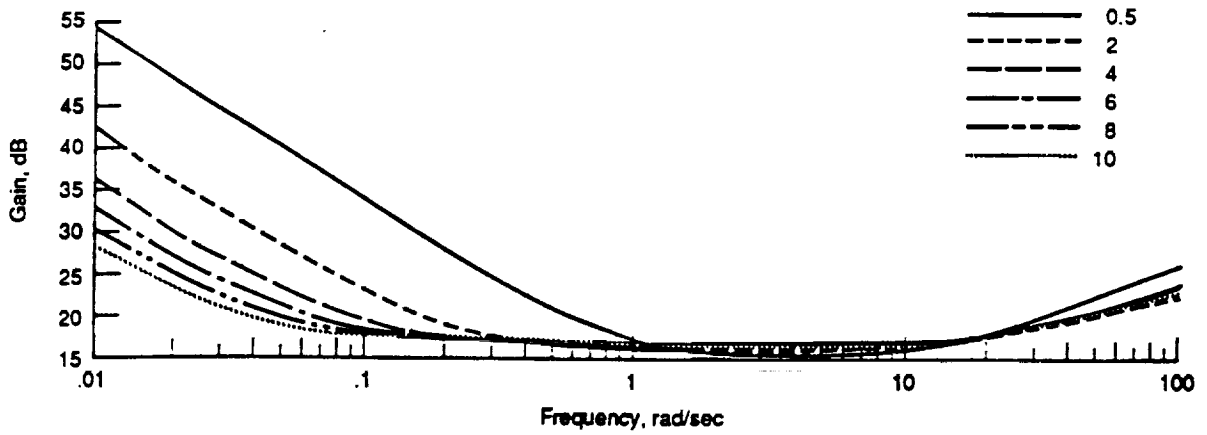


Figure 13.—Bode gain plots for neurocontroller at $t = 0.5, 2, 4, 6, 8$ and 10 sec. e_Q to δTV .

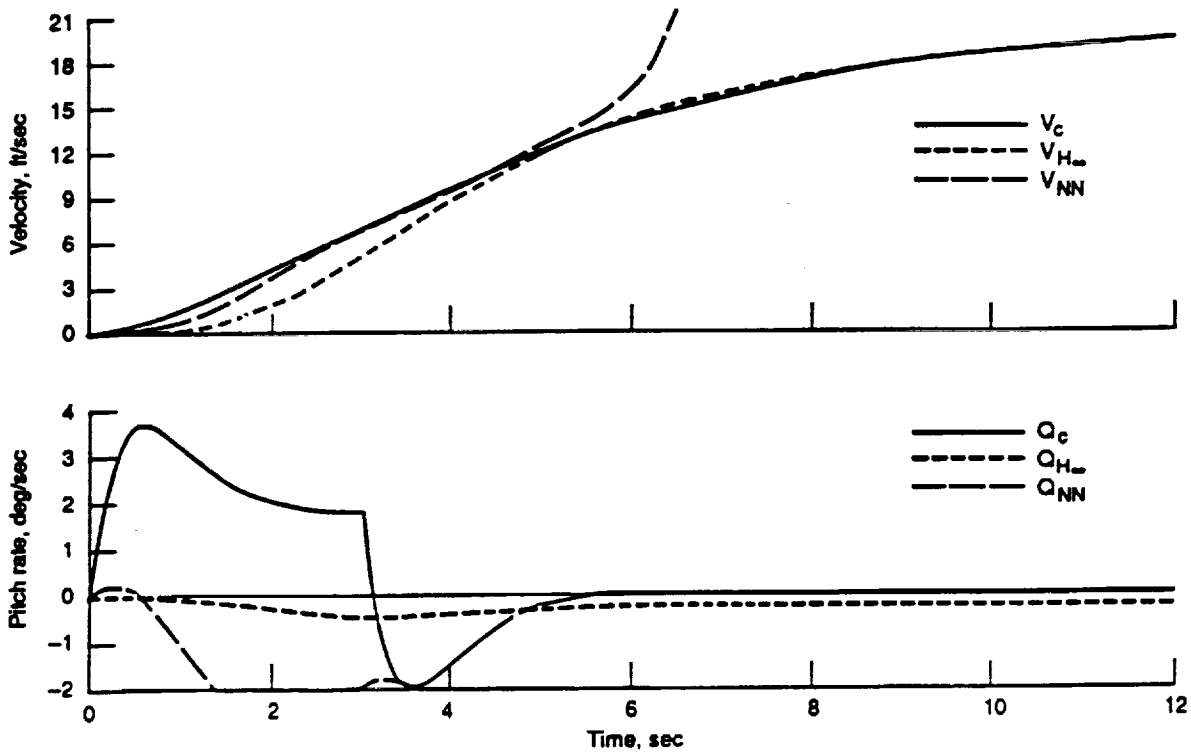


Figure 14.—Closed-loop response to Case 2 commands with e_Q sensor failure.

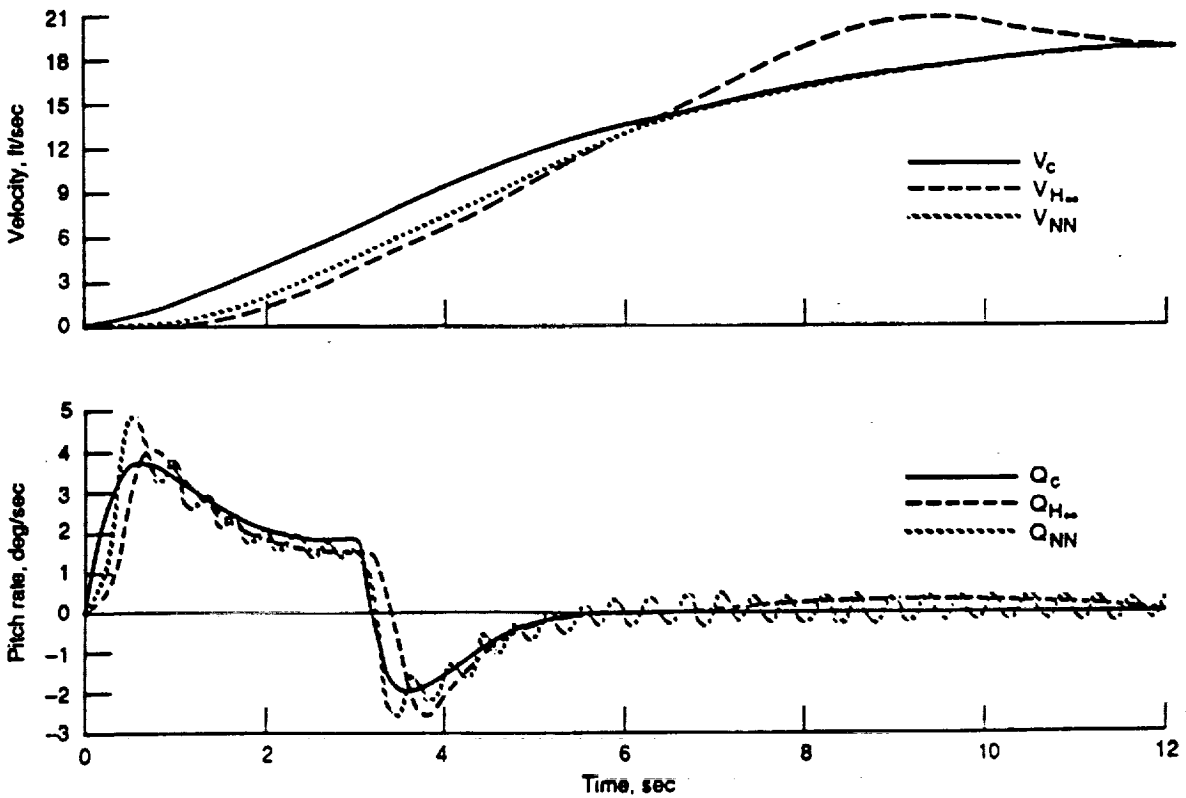


Figure 15.—Closed-loop response to Case 2 commands with time delay of 0.05 sec in both controls.

1. Report No. NASA TM - 105151		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Neural Network Application to Aircraft Control System Design				5. Report Date	
				6. Performing Organization Code	
7. Author(s) Terry Troudet, Sanjay Garg, and Walter C. Merrill				8. Performing Organization Report No. E-6435	
				10. Work Unit No. 505-62-50	
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191				11. Contract or Grant No.	
				13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001				14. Sponsoring Agency Code	
15. Supplementary Notes Prepared for the Guidance, Navigation and Control Conference sponsored by the American Institute of Aeronautics and Astronautics, New Orleans, Louisiana, August 12-14, 1991. Terry Troudet, Sverdrup Technology, Inc., Lewis Research Center Group, 2001 Aerospace Parkway, Brook Park, Ohio 44142; Sanjay Garg and Walter C. Merrill, NASA. Lewis Research Center. Responsible person, Terry Troudet, (216) 433-8524.					
16. Abstract The feasibility of using artificial neural networks as control systems for modern, complex aerospace vehicles is investigated via an example aircraft control design study. The problem considered is that of designing a controller for an integrated airframe/propulsion longitudinal dynamics model of a modern fighter aircraft to provide independent control of pitch rate and airspeed responses to pilot command inputs. An explicit model-following controller using H_∞ control design techniques is first designed to gain insight into the control problem as well as to provide a baseline for evaluation of the neurocontroller. Using the model of the desired dynamics as a command generator, a multilayer feedforward neural network is trained to control the vehicle model within the physical limitations of the actuator dynamics. This is achieved by minimizing an objective function which is a weighted sum of tracking errors and control input commands and rates. To gain insight in the neurocontrol, linearized representations of the non-linear neurocontroller are analyzed along a commanded trajectory. Linear robustness analysis tools are then applied to the linearized neurocontroller models and to the baseline H_∞ based controller. Future areas of research are identified to enhance the practical applicability of neural networks to flight control design.					
17. Key Words (Suggested by Author(s)) H-infinity control; Neural networks; Neurocontrol; Performance/ control trade-off			18. Distribution Statement Unclassified - Unlimited Subject Category 08		
19. Security Classif. (of the report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages 20	22. Price* A03

