JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 18, No. 4, July–August 1995

Neural Networks Approach to AIAA Aircraft Control Design Challenge

C. M. Ha*
Lockheed Fort Worth Company, Fort Worth, Texas 76101

This paper focuses on designing a discrete-time lateral-directional control law for a high-performance aircraft using neural networks. The control law structure is composed of feedback and filter components formulated in the form of a three-layer feedforward neural network whose parameters are adjusted by a gradient descent algorithm to provide stabilization about the aircraft center of mass and asymptotic tracking of pilot command input. The number of parameters was chosen in an ad hoc manner. Only rate gyro and lateral accelerometer outputs are available for feedback, whereas rudder pedal and lateral stick commands are input signals to the filter. Linear simulation results at an operating point within the aircraft's envelope in the presence of atmospheric turbulence and actuator and sensor noises shed light on the ability of neural networks to serve as a practical tool for flight control law designers.

I. Introduction

NUMEROUS classical and modern control law design methodologies have been proposed for designing continuous- or discrete-time longitudinal and lateral-directional control laws for aircraft.[1,2] These time- and frequency-domain design techniques usually apply to a linear, time-invariant state-space dynamic model of the aircraft from either a single-input, single-output (SISO) or a multi-input, multi-output (MIMO) point of view. However, the aircraft's equations of motion are highly nonlinear in six degrees of freedom (6-DOF) throughout its operating envelope. Thus, those who design control laws for aircraft face some difficulties in extending these techniques to real-flight vehicles.

Traditionally, gain scheduling has been employed to compensate for nonlinearity. In this approach, the resulting nonlinear control law for the full-flight envelope is interpolated between various linear controllers scheduled as a function of speed, altitude, and wing loading. However, the desire to enlarge the flight envelope of modern aircraft and to provide enhanced maneuver capabilities may make gain scheduling infeasible due to rapid changes in dynamics. A more sophisticated flight control system that could work well over a wide range of operating conditions is clearly needed.

Adaptive control is intended for plant models that must operate in the presence of uncertainty. It utilizes the available a priori knowledge and adapts to the unknown part of the plant such as time-varying parameters. An adaptive flight control system is capable of adjusting on-line the controller's parameters in response to changing flight characteristics.[3] However, instability in the adaptive control system in the presence of unmodeled dynamics, bounded input, and output disturbances has been observed in numerous studies.[4,5] These problems have motivated the search for a type of flight control system that is capable of learning from flight experience, i.e., the more it is flown, the better it becomes. The need to control increasingly complex dynamic systems such as aircraft in a better way has led to a more general concept of "intelligent control," one that includes higher level decision making, planning, and learning. One approach that has the potential for intelligent control is neural networks that can act as compensation elements in control systems. Neural networks represent a promising technology because of their attractive properties, such as 1) performing massive parallel processing, in contrast to conventional digital computers, in which computations are executed sequentially; 2) providing fault tolerance, since damage to a few interconnection links within the network need not impair the overall performance; and most important, 3) being capable of learning complex dynamics with uncertainty.[6–12]

In recent years, increased interest has developed in process control, robotics, communications, and aerospace applications[13–17] as the control community became aware of the feasibility of rapid and inexpensive neural network implementation through very large scale integrated technology. Reference 15 shows how a three-layer neural network can be used as a real-time gain adjuster in a longitudinal stability augmentation system. A neural network for the preliminary development of an autopilot model to precisely control a high-performance aircraft during flight maneuvers was conceptually demonstrated in Refs. 16 and 17.

In this paper, we investigate the use of neural networks as learning controllers. We utilize a gradient descent algorithm to train the neural network to learn to function as a closed-loop flight control system and to force the dynamics of the aircraft to match that of a specified reference model. The paper is organized as follows. Section II discusses the neural network design methodology. This section also deals with a gradient method for the adjustment of parameters of neural networks. An example application of the methodology to lateral-directional control law design for a modern aircraft is given in Sec. III. The aircraft model is first discussed, the design specifications are then stated, and the equations of the neural network control law are summarized. Section IV presents preliminary closed-loop performance evaluation results for the designed neural network control law at a specified flight condition. Finally, in Sec. V, some concluding remarks are given.

II. Three-Layer Feedforward Neural Networks

A. Theoretical Foundations

This section provides a theoretical background of the neural network design methodology to be used in the control law design. A neural network is defined as a system with time-domain inputs and outputs interconnected via many layers of nonlinear processing elements. Each processing element, called a neuron, consists simply of forming the total weighted sum of all input signals of the preceding layer, adding a "threshold" value to it (not considered in this paper), and then applying a nonlinear sigmoidal activation function to this weighted sum to generate output signals. These output signals are then distributed to processing elements or neurons in the next layer (Fig. 1).

Depending on the interconnection of the neurons to one another, neural networks can be classified as either recursive or feedforward. In a recursive network, each neuron receives as input a weighted output from all other neurons, including itself as a feedback signal. This type of network has found widespread application in biological systems. On the other hand, the feedforward network has no feedback between layers. The output of one neuron multiplied by a weight becomes the input of an adjacent neuron of the next layer.

Received Jan. 2, 1992; revision received Nov. 23, 1993; accepted for publication Oct. 20, 1993.

*Engineering Specialist, Control Law Design and Analysis Group. Member AIAA.

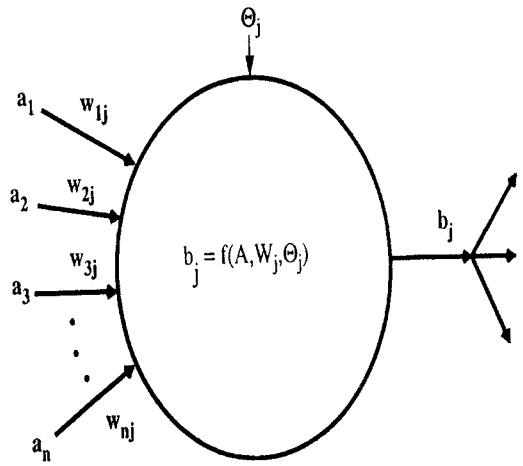


Fig. 1 Neuron.

In this paper, we only consider the feedforward neural network. A typical feedforward network is shown in Fig. 2, where neurons in a given layer receive inputs only from those in the preceding layer. A three-layer neural network with an input layer, a hidden layer, and an output layer can approximately realize any nonlinear continuous mapping to a degree of accuracy over some compact domain.[18] However, this is only an existence proof; it does not show how to find the correct interconnection weights indicate how many neurons should be in each layer. Also, Ref. 18 mentioned that using more than one hidden layer may have the advantage of significantly reducing the number of hidden neurons needed for the same task. As can be seen from Fig. 2, the input to the left layer is the input to the network. The outputs of the right layer are observable; this layer is accordingly called the output layer. On the other hand, the outputs of the intermediate layer are not observable, and this layer is hence called the "hidden layer" of the network.

B. Three-Layer Feedforward Neural Network Equations

Output layer:

$$\begin{aligned} y_k &= f(y_k), \qquad k = 1, 2, \ldots, n_y \\ &= \sum_{i=1}^{n_2} \gamma_{ki} v_i \end{aligned} \tag{1}$$

Hidden layer:

$$\begin{aligned} v_i &= f(v_i) \\ &= \sum_{j=1}^{n_1} \alpha_{ij} z_j \end{aligned} \tag{2}$$

Input layer:

$$\begin{aligned} z_j &= f(z_j) \\ &= \sum_{l=1}^{n_u} \beta_{jl} u_l \end{aligned} \tag{3}$$

where n_y is the number of outputs and n_u is the number of inputs, n_1 is the number of neurons in the input layer and n_2 is the number of neurons in the hidden layer, and γ_{ki}, α_{ij}, and β_{jl} are interconnection weights.

In this paper, the sigmoidal function used is called the hyperbolic tangent, $f(x) = (1 - e^{-2x})/(1 + e^{-2x})$, and its derivative $f'(x) = 1 - f(x)^2$ is shown in Fig. 3. This chosen sigmoidal function has a special feature: $f(x)$ cannot reach either -1 or $+1$ without infinitely large x. Equations (1–3) describe a signal as it feeds forward through the network. That is, when the network is given some input u_l, it responds with some output y_k based on the values of its interconnection weights γ_{ki}, α_{ij}, and β_{jl}. This is called an operational mode, which is one of two ways a neural network can be used. The other way is the so-called learning mode, in which the interconnection weights γ_{ki}, α_{ij}, and β_{jl} are trained so that the neural network outputs y_k have some desired property.

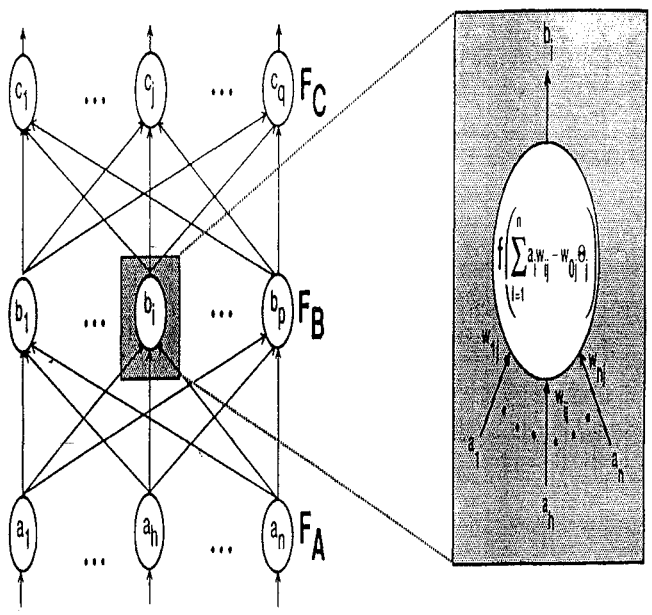


Fig. 2 Three-layer feedforward neural network.

C. Neural Net Training

Here, the interconnection weights γ_{ki}, α_{ij}, and β_{jl} are adjusted by some learning algorithm to optimize an appropriate performance function. Suppose we want the outputs $y_k(t)$ of the network to be $yd_k(t)$ where t runs from 1 to L. The performance function we wish to optimize is the following mean-square error over the interval $[1, L]$:

$$J = 0.5 \sum_{t=1}^{L} \sum_{k=1}^{n_y} [y_k(t) - yd_k(t)]^2 \tag{4}$$

The interconnection weights γ_{ki}, α_{ij}, and β_{jl} are then updated when training according to the equations

$$\begin{aligned} \gamma_{ki(\text{new})} &= \gamma_{ki(\text{old})} - \eta_\gamma \frac{\partial J}{\partial \gamma_{ki}} \\ \alpha_{ij(\text{new})} &= \alpha_{ij(\text{old})} - \eta_\alpha \frac{\partial J}{\partial \alpha_{ij}} \\ \beta_{jl(\text{new})} &= \beta_{jl(\text{old})} - \eta_\beta \frac{\partial J}{\partial \beta_{jl}} \end{aligned} \tag{5}$$

where η_γ, η_α, η_β are step size gains taking values in range (0,1)

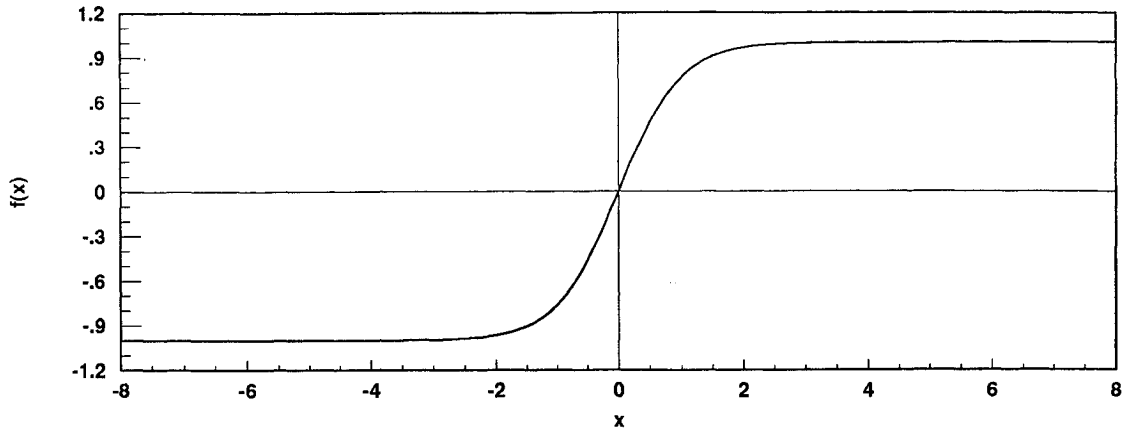
$$\begin{aligned} \frac{\partial J}{\partial \gamma_{ki}} &= \frac{\partial J}{\partial y_k(t)} \frac{\partial y_k(t)}{\partial \gamma_{ki}} \\ \frac{\partial J}{\partial \alpha_{ij}} &= \frac{\partial J}{\partial y_k(t)} \frac{\partial y_k(t)}{\partial \alpha_{ij}} \\ \frac{\partial J}{\partial \beta_{jl}} &= \frac{\partial J}{\partial y_k(t)} \frac{\partial y_k(t)}{\partial \beta_{jl}} \end{aligned} \tag{6}$$

and

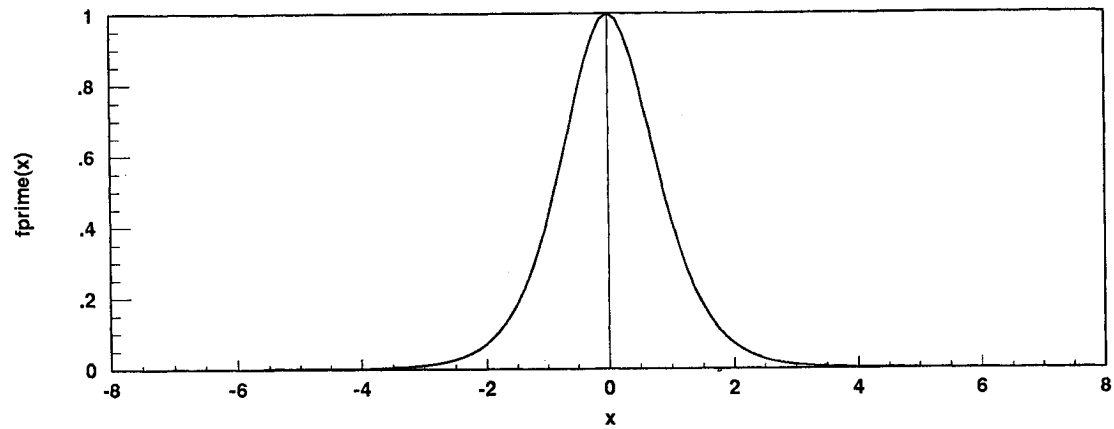
$$\frac{\partial J}{\partial y_k(t)} = \sum_{t=1}^{L} [y_k(t) - yd_k(t)] \tag{7}$$

with

$$\begin{aligned} \frac{\partial y_k(t)}{\partial \gamma_{ki}} &= f'(y_k(t)) \frac{\partial \left[\sum \gamma_{ki} v_i\right]}{\partial \gamma_{ki}} \\ &= \left[1 - f(y_k(t))^2\right] \frac{\partial \left[\sum \gamma_{ki} v_i\right]}{\partial \gamma_{ki}} \\ \frac{\partial y_k(t)}{\partial \gamma_{ki}} &= \left[1 - y_k(t)^2\right] v_i \end{aligned} \tag{8}$$



a) Function



b) Derivatives

Fig. 3 Sigmoid.

$$\frac{\partial y_k(t)}{\partial \alpha_{ij}} = \frac{\partial y_k(t)}{\partial v_i}\frac{\partial v_i}{\partial \alpha_{ij}}$$

$$\frac{\partial y_k(t)}{\partial v_i} = \left[1 - y_k(t)^2\right]\gamma_{ki}$$

$$\frac{\partial v_i(t)}{\partial \alpha_{ij}} = f'(v_i(t))\frac{\partial\left[\sum \alpha_{ij} z_j\right]}{\partial \alpha_{ij}} = \left[1 - f(v_i(t))^2\right]\frac{\partial\left[\sum \alpha_{ij} z_j\right]}{\partial \alpha_{ij}} \qquad (9)$$

$$\frac{\partial v_i(t)}{\partial \alpha_{ij}} = \left[1 - v_i(t)^2\right] z_j$$

$$\frac{\partial y_k(t)}{\partial \beta_{jl}} = \frac{\partial y_k(t)}{\partial v_i}\frac{\partial v_i}{\partial z_j}\frac{\partial z_j}{\partial \beta_{jl}}$$

$$\frac{\partial v_i(t)}{\partial z_j} = \left[1 - v_i(t)^2\right]\alpha_{ij}$$

$$\frac{\partial z_j(t)}{\partial \beta_{jl}} = f'(z_j(t))\frac{\partial\left[\sum \beta_{jl} u_l\right]}{\partial \beta_{jl}} = \left[1 - f(z_j(t))^2\right]\frac{\partial\left[\sum \beta_{jl} u_l\right]}{\partial \beta_{jl}} \qquad (10)$$

$$\frac{\partial z_j(t)}{\partial \beta_{jl}} = \left[1 - z_j(t)^2\right] u_l$$

As can be seen in Eq. (5), the interconnection weights γ_{ki}, α_{ij}, and β_{jl} are updated in the direction of decreasing J, the steepest descent algorithm. Also, it is seen that all signals used to compute the partial derivatives $\partial J/\partial \gamma_{ki}$, $\partial J/\partial \alpha_{ij}$, and $\partial J/\partial \beta_{jl}$ of Eq. (6) seem to be identical to those in the network, except flowing in the opposite direction, justifying the use of the term "back-propagation." In all the simulations carried out in this paper, the initial weights $\gamma_{ki(\text{old})}$, $\alpha_{ij(\text{old})}$, and $\beta_{jl(\text{old})}$ are selected randomly between +0.5 and −0.5. If $\gamma_{ki(\text{old})}$, $\alpha_{ij(\text{old})}$, and $\beta_{jl(\text{old})}$ are initially set to 0.0, then the network will never learn. This is because the steepest descent algorithm is propagated back through the neurons in proportion to $\gamma_{ki(\text{old})}$, $\alpha_{ij(\text{old})}$, and $\beta_{jl(\text{old})}$ [Eqs. (8–10)]. Thus, if the network starts out at a local minimum, it will remain there. It is also worth mentioning that the steepest descent gradient search technique is known to have problems with local minima and saddle points and to have a slow rate of convergence.

III. Application

This section discusses in detail the application of the neural network design methodology to designing a lateral-directional control law for a modern high-performance two-engine aircraft. Detailed aerodynamic and net thrust models of this aircraft along with its nonlinear equations of motion were provided by NASA Ames Dryden Research Facility. The neural network control law capabilities are demonstrated using a linear simulation of the aircraft lateral-directional equations of motion with noisy actuators and sensors and in the presence of atmospheric turbulence. Numerical results are then presented after linear dynamic models of a lateral-directional system are described.

A. Linearized Aircraft Equations of Motion

From linearization of the full, nonlinear, rigid aircraft equations of motion about trim flight conditions, the linear equations for lateral-

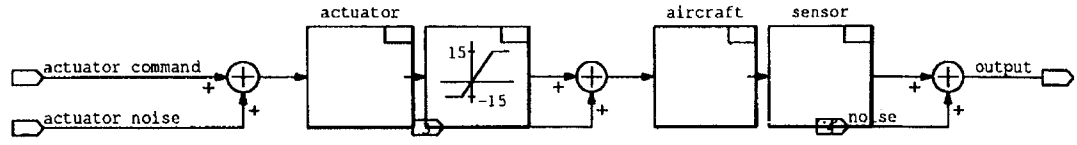


Fig. 4 Open-loop lateral-directional.

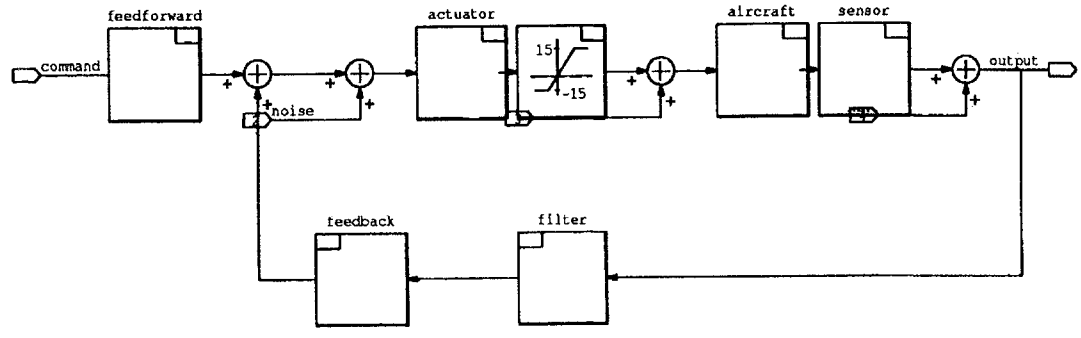


Fig. 5 Closed-loop lateral-directional.

directional motion perturbed from a stability axis are[3]

$$\dot{p}_s = L'_p p_s + L'_r r_s + L'_\beta \beta + L'_{\delta a}\delta_a + L'_{\delta r}\delta_r + L'_\beta w \quad (11)$$

$$\dot{r}_s = N'_p p_s + N'_r r_s + N'_\beta \beta + N'_{\delta a}\delta_a + N'_{\delta r}\delta_r + N'_\beta w$$

$$\dot{\beta} = Y_p p_s + (Y_r - 1)r_s + Y_\beta \beta + (g/V_0)\phi + Y_{\delta a}\delta_a + Y_{\delta r}\delta_r + Y_\beta w$$

$$\dot{\phi} = p_s + \alpha_0 r_s$$

$$a_y = (V_0/g)(\beta + r_s - p_s\alpha_0) - \phi + (d/g)r_s$$

where g is acceleration due to gravity (32.2 ft/s^2), V_0 is trim velocity (ft/s), α_0 is the trim angle of attack (rad), d is accelerometer displacement from center of gravity (c.g.), and

$$L'_k = [L_k + (N_k I_{xz})/I_x]/[1 - I_{xz}^2/(I_x I_z)]$$
$$N'_k = [N_k + (L_k I_{xz})/I_z]/[1 - I_{xz}^2/(I_x I_z)] \quad (12)$$

where $k = p, r, \beta, \delta_a, \delta_r$ and $L_{(\cdot)}$, $N_{(\cdot)}$, and $Y_{(\cdot)}$ are dimensional lateral-directional stability derivatives referred to fixed-body axes.

For a straight-and-level flight condition at 9800 ft with Mach number 0.5, the rigid, stability-axis, lateral-directional equations of motion [Eq. (11)] can be described by the following state-space mathematical model:

$$\dot{x} = Ax + Bu + Ew, \qquad x(0) = x_0$$
$$y = Cx + Du + Lw + v \quad (13)$$

The preceding equations represent a continuous-time process model. The use of this model in a digital neural network flight control system requires a discrete-time model representation. Here, we assume a nominal sampling rate $\Delta t = \frac{1}{50}$ s for the onboard digital flight computer. Equation (13) is then transformed into the following equivalent discrete form:

$$x(t+1) = Fx(t) + Gu(t) + \Gamma w(t), \qquad x(0) = x_0$$
$$y(t) = Hx(t) + Ju(t) + Lw(t) + v(t) \quad (14)$$

where

$$F = e^{A\Delta t}, \qquad G = \int e^{A\Delta\tau} B\,d\tau, \qquad \Gamma = \int e^{A\Delta t} E\,d\tau$$
$$H = C, \qquad J = D$$

and x is stability-axis roll rate p_s (rad/s), stability-axis yaw rate r_s (rad/s), sideslip β(rad), or roll angle ϕ (rad); u is aileron deflection δ_a (rad), elevator deflection δ_e (rad), or rudder deflection δ_r (rad); and y is stability-axis roll rate p_s (rad/s), stability-axis yaw rate r_s (rad/s), sideslip β (rad), roll angle ϕ (rad), or lateral acceleration a_y (ft/s^2).The numerical values of A, B, E, C, D, and L of Eq. (13) and F, G, Γ, H, and J of Eq. (14) are listed in Appendix A. Note that in addition to aileron elevation, elevator elevation is used differentially for roll augmentation. Also, the output matrix H of Eq. (14) has all sensors available for feedback signals. However, due to bias errors, scale factor uncertainty, and noise, air data sensors do not perfectly measure angle of attack α and sideslip angle β during dynamic maneuvers and at elevated g conditions. Also, attitude sensor ϕ is usually not reliable at high pitch angle θ. Thus, air data β and inertial attitude gyro ϕ are not employed for feedback in this study. This restriction limits the neural network flight control system design to the use of accelerometer a_y and body-axis angular rate data p_b and r_b, where

$$p_s(t) = +p_b(t)\cos\alpha(t) + r_b(t)\sin\alpha(t)$$
$$r_s(t) = -p_b(t)\sin\alpha(t) + r_b(t)\cos\alpha(t) \quad (15)$$

B. Lateral-Directional Neural Network Flight Control Design

The primary goal of the lateral-directional neural network flight control system design is to provide the aircraft with a maneuvering capability that exhibits a minimal cross coupling throughout its operational envelope through the use of differential aileron δ_a, differential elevator δ_e, and rudder δ_r (Fig. 4). In terms of MIL-F-8785C level-1 flying qualities,[19] minimal cross coupling manifests as follows:

1) A pure lateral stick input δ_{stick} will result in a well-coordinated steady-state roll about the stability axis ($\beta \approx 0$). The desired response from pilot lateral stick δ_{stick} to stability-axis roll rate $p_{s(\text{model})}$ should be approximately first order, i.e.,

$$p_{s(\text{model})}/\delta_{\text{stick}} = 3/(s+3) \quad (16)$$

where s represents the Laplace operator

$$p_{s(\text{model})}/\delta_{\text{stick}} = 0.0291(z+1)/(z-0.9417)$$

for a sampling time $\Delta t = \frac{1}{50}$ s.

2) A pure rudder pedal input δ_{pedal} will result in a steady-state wing-level sideslip maneuver with minimal roll response ($p_s \approx \phi \approx 0$). The desired response from pilot rudder pedal δ_{pedal} to sideslip angle β_{model} shoud be approximately second order, i.e.,

$$\beta_{\text{model}}/\delta_{\text{pedal}} = 9/(s^2 + 2\times 0.8\times 3s + 9)$$

$$\beta_{\text{model}}/\delta_{\text{pedal}} = (0.0009z^2 + 0.0017z + 0.0009)/(z^2 - 1.9050z + 0.9085) \quad (17)$$

for a sampling time $\Delta t = \frac{1}{50}$ s.

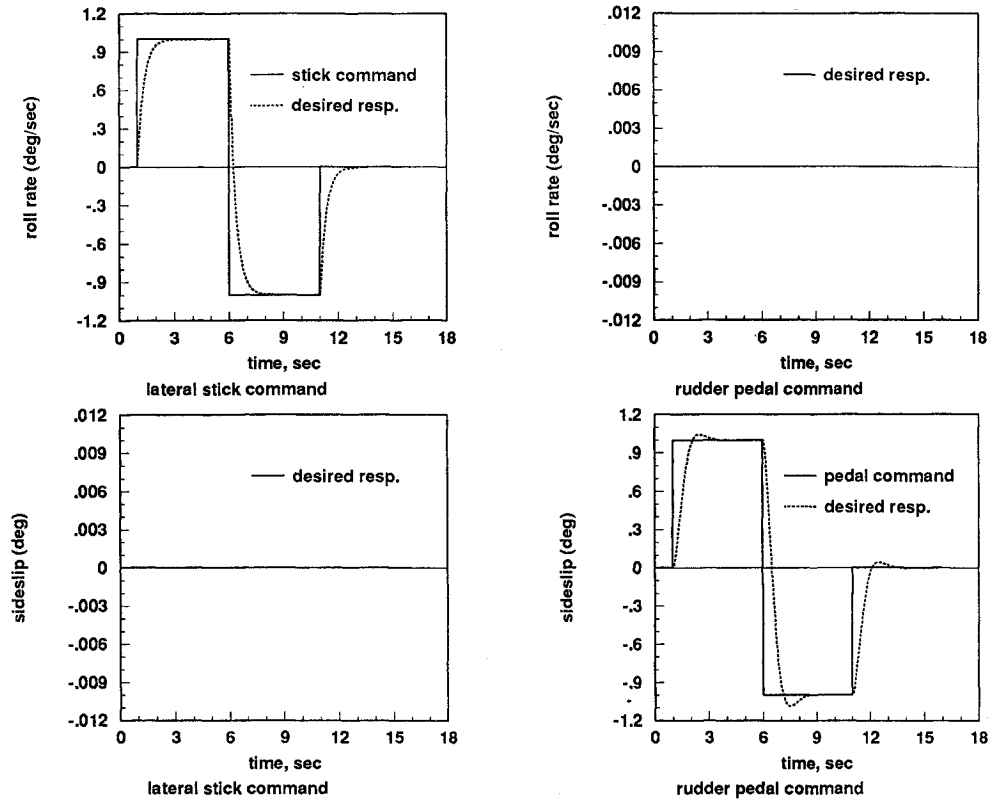


Fig. 6 Desired lateral and directional responses.

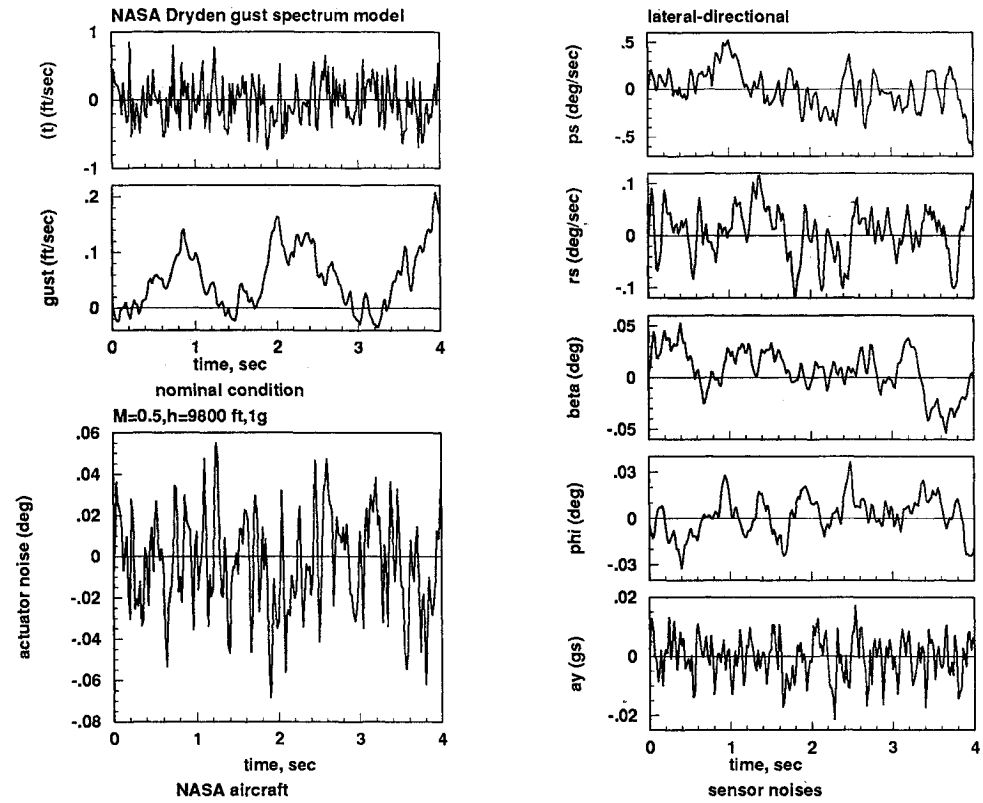


Fig. 7 Atmospheric turbulence, actuator and sensor noises.

Three radians per second is the chosen desired command tracking bandwidth. Although meeting these objectives, the neural network flight control system is expected to adhere to certain constraints, such as minimizing lateral acceleration a_y and avoiding excessive control deflections and rates. In an effort to achieve this goal, the neural network flight control configuration shown in Fig. 5 is proposed. The configuration consists of a feedback controller and a feedforward filter. The controller ensures a stable closed-loop response and good feedback properties such as settling time, damping, and gain and phase margins. Then, freedom to specify the response to a pilot's commands, such as zero steady-state tracking error, and good decoupling between response to roll rate and to sideslip step inputs are provided through the feedforward filter.

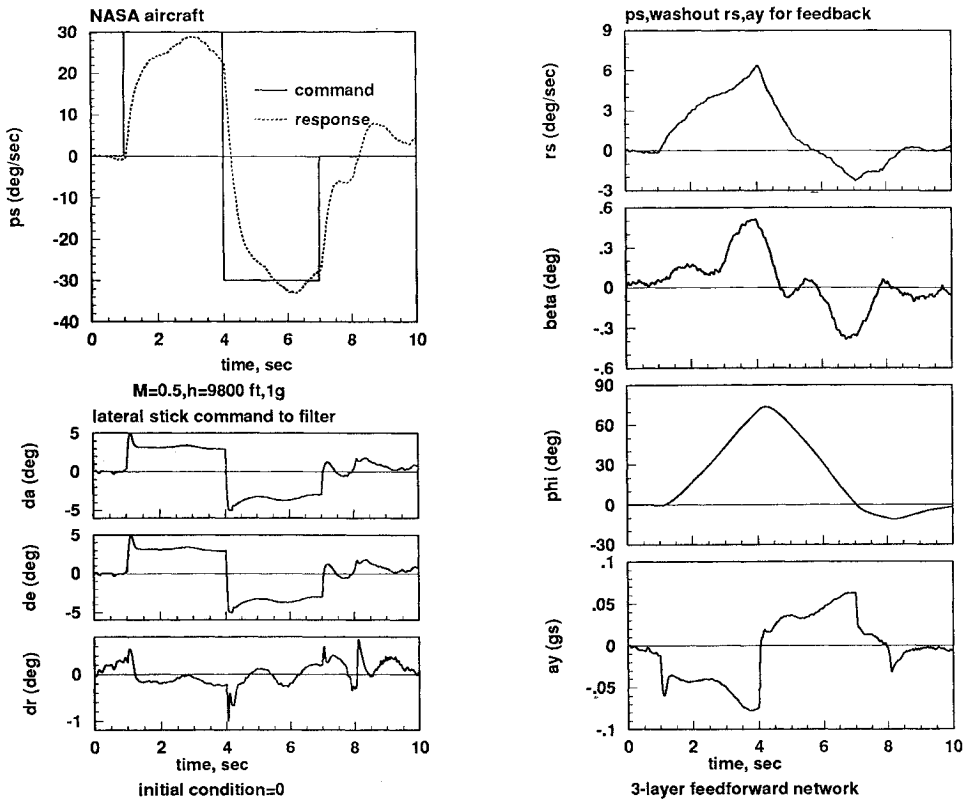


Fig. 8 Lateral stick response.

1. Three-Layer Neural Network Equations for Feedback Controller

Output layer:

$$y_k = \sum_{i=1}^{n_2} \gamma c_{ki} v_i, \qquad k = 1, 2, \ldots, n_y \tag{18}$$

Hidden layer:

$$v_i = f(v_i) = \sum_{j=1}^{n_1} \alpha c_{ij} z_j \tag{19}$$

Input layer:

$$z_j = f(z_j) = \sum_{l=1}^{n_u} \beta c_{jl} u_l \tag{20}$$

where $n_y = 2, n_u = 3, n_1 = 8$, and $n_2 = 8$; the total number of interconnection weights γc_{ki}, αc_{ij}, and βc_{jl} equals $(2+8+3) \times 8 = 104$; and

$$[y_1 \quad y_2] = [\delta_{ac}(t) + \delta_{ec}(t)\delta_{rc}(t)]$$

$$[u_1 \quad u_2 \quad u_3] = \{p_s(t)[0.9901(z-1)/(z-0.9802)]r_s(t)a_y(t)\}$$

where $\delta_{ac}(t)$ is aileron deflection command (rad), $\delta_{ec}(t)$ is elevator deflection command (rad), and $\delta_{rc}(t)$ is rudder deflection command (rad).

2. Three-Layer Neural Network Equations for Feedforward Filter

Output layer:

$$y_k = \sum_{i=1}^{n_2} \gamma f_{ki} v_i, \qquad k = 1, 2, \ldots, n_y \tag{21}$$

Hidden layer:

$$v_i = f(v_i) = \sum_{j=1}^{n_1} \alpha f_{ij} z_j \tag{22}$$

Input layer:

$$z_j = f(z_j) = \sum_{l=1}^{n_u} \beta f_{jl} u_l \tag{23}$$

where $n_y = 2, n_u = 1, n_1 = 8$, and $n_2 = 8$; the total number of interconnection weights γf_{ki}, αf_{ij}, and βf_{jl} equals $(1+8+2) \times 8 = 88$; and

$$[y_1 \quad y_2] = [\delta_{ac}(t) + \delta_{ec}(t)\delta_{rc}(t)], \qquad [u_l] = [\delta_{stick}(t) \text{ or } \delta_{pedal}(t)]$$

where $\delta_{ac}(t)$ is aileron deflection command (rad), $\delta_{ec}(t)$ is elevator deflection command (rad), and $\delta_{rc}(t)$ is rudder deflection command (rad).

Note that the sigmoidal function $f(x)$ is not activated in the output layer of both controller and filter neural network structures [Eqs. (18) and (21)]. This is because we do not want to limit $\delta_{ac}(t)$, $\delta_{ec}(t)$, and $\delta_{rc}(t)$ to the range between $+1$ and -1. However, realistic surface movement [rate and position limits of all surface actuators (see Appendix B)] can be taken into account as follows:

$$\begin{aligned} \delta_{ac}(t) &= 10\left[f\left(\sum \gamma c_{1i}\right) + f\left(\sum \gamma f_{1i}\right)\right] \\ \delta_{ec}(t) &= 7.5\left[f\left(\sum \gamma c_{1i}\right) + f\left(\sum \gamma f_{1i}\right)\right] \\ \delta_{rc}(t) &= 15\left[f\left(\sum \gamma c_{2i}\right) + f\left(\sum \gamma f_{2i}\right)\right] \end{aligned} \tag{24}$$

where $i = 1, 2, \ldots, n_2$ and $f(\cdot)$ is the sigmoidal function.

The stability-axis yaw rate $r_s(t)$ is filtered with a washout $s/(s+1)$ to allow a well-coordinated turn without applying continuous pedal

force, and $0.9901(z-1)/(z-0.9802)$ is the discrete-time transfer function of $s/(s+1)$ for a sampling time $\Delta t = \frac{1}{50}$ s. The number of neurons in each layer was chosen by trial and error. In this study, eight neurons were found to be satisfactory.

C. Neural Network Flight Control System Training

Our objective is to train $104+88=192$ interconnection weights γc_{ki}, αc_{ij}, βc_{jl}, γf_{ki}, αf_{ij}, and βf_{jl} so that the neural network flight control system responds properly to a pilot's stick and pedal input commands. For a lateral stick command, we wish to minimize the following mean-square error over an interval of length L:

$$J_{\text{lateral}} = 0.5 \sum_{t=1}^{L} \left\{ [p_s(t) - p_{s(\text{model})}(t)]^2 + \beta(t)^2 \right\} \tag{25}$$

and for a rudder pedal command

$$J_{\text{directional}} = 0.5 \sum_{t=1}^{L} \left\{ p_s(t)^2 + [\beta(t) - \beta_{\text{model}}(t)]^2 \right\} \tag{26}$$

where typical step responses $p_{s(\text{model})}(t)$ and $\beta_{\text{model}}(t)$ are shown as dotted lines in Fig. 6 and L is the number of data. Since J_{lateral} and $J_{\text{directional}}$ are different, the training has to be carried out separately. This results in two neural network controllers, one for lateral stick command and the other for rudder pedal command.

IV. Preliminary Results

Once training has been completed and the interconnection weights γc_{ki}, αc_{ij}, βc_{jl}, γf_{ki}, αf_{ij}, and βf_{jl} are fixed (given in Appendix E for lateral stick command only), we evaluate the effectiveness of the neural network design technique. Starting from trim, the closed-loop response of the aircraft/discrete-time neural network flight control system was evaluated in a linear time-domain simulation[20] including atmospheric turbulence, actuator and sensor noises (Fig. 7), surface dynamics rate, and position limits for the following maneuver: a ±30 deg/s bank-to-bank from a straight-and-level flight at 9800 ft and 0.5 Mach number where angle of sideslip β is held to within ±1.0 deg and a_y is maintained to within $\pm 0.1g$ (Fig. 8). The neural network controller seems to provide reasonably good tracking of ±30 deg/s roll rate command. It is also noted that reasonable aileron δ_a, elevator δ_e, and rudder δ_r deflections are observed in the maneuver.

V. Conclusions

The successful functioning of a lateral-directional neural network control law designed for a highly maneuverable aircraft has been demonstrated. The three-layer neural network flight control system was trained to optimize an appropriately constructed performance function via a gradient descent algorithm. Time histories of the closed-loop response to pilot inputs shed light on the ability of the neural network design technique to serve as a potential tool for engineers involved in flight control law design and analysis. In what follows we list some of the strengths and weaknesses of the neural network methodology with regard to its application to flight control design:

1) The design of a neural network control law does not result in a high-order controller that may be difficult to implement. Even integral control action by appending integrators to the linear aircraft model to provide desirable feedback properties is not needed.

2) Classical measures of stability gain and phase margins require nonlinear simulation evaluation because of the presence of nonlinear sigmoidal functions in the neural network controller. However, this could not be done for the present study because of time constraints. References 21 and 22 demonstrate that a neural network controller can be trained to guarantee stability in the presence of high-frequency unmodeled dynamics and parameter variation and to maintain nominal performance in situations where failures occur.

3) At present, due to limited computational power and speed of the airborne flight computers, the neural network training has to be carried out off-line.

Appendix A

The following matrices of the continuous- and discrete-time, linear, state-space models [Eqs. (13) and (14)] representing the aircraft lateral-directional equations of motion were obtained from linearization of the full 6-DOF nonlinear equations about a trim flight condition of Mach = 0.5 (V_0 = 539.0989 ft/s), altitude = 9800 ft, and $\alpha_0 = 0.0798$ (rad).

Continuous-Time Model

$$A_{\text{lateral-directional}} = \begin{bmatrix} -2.2162 & 1.3968 & -27.0705 & 0.0000 \\ -0.0745 & -0.5745 & 4.6833 & 0.0000 \\ 0.0797 & -0.9968 & -0.1925 & 0.0594 \\ 1.0000 & 0.0800 & 0.0000 & 0.0000 \end{bmatrix}$$

$$B_{\text{lateral-directional}} = \begin{bmatrix} 9.7142 & 9.7806 & -1.4283 \\ 0.1288 & 1.2054 & -2.7868 \\ -0.0022 & -0.0164 & -0.0363 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix}$$

$$E_{\text{lateral-directional}} = [-27.0705 \quad 4.6833 \quad -0.1925 \quad 0.0000]^T$$

$$C_{\text{lateral-directional}} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \\ 0.0000 & 0.0000 & -3.2260 & 0.0000 \end{bmatrix}$$

$$D_{\text{lateral-directional}} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ -0.0369 & -0.2740 & -0.6079 \end{bmatrix}$$

$$L_{\text{lateral-directional}} = [0. \quad 0. \quad 0. \quad 0. \quad 0.]^T$$

$$x_{0(\text{lateral-directional})} = [0. \quad 0. \quad 0. \quad 0.]^T$$

Discrete-Time Model ($\Delta t = \frac{1}{50}$s)

$$F_{\text{lateral-directional}} = \begin{bmatrix} 0.9562 & 0.0324 & -0.5270 & -0.0003 \\ -0.0014 & 0.9876 & 0.0933 & 0.0001 \\ 0.0016 & -0.0198 & 0.9948 & 0.0012 \\ 0.0196 & 0.0019 & -0.0052 & 1.0000 \end{bmatrix}$$

$$G_{\text{lateral-directional}} = \begin{bmatrix} 0.1901 & 0.1918 & -0.0286 \\ 0.0024 & 0.0238 & -0.0554 \\ 0.0001 & -0.0004 & -0.0002 \\ 0.0019 & 0.0019 & -0.0003 \end{bmatrix}$$

$$\Gamma_{\text{lateral-directional}} = [-0.5270 \quad 0.0933 \quad -0.0052 \quad -0.0052]^T$$

$$H_{\text{lateral-directional}} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \\ 0.0000 & 0.0000 & -3.2260 & 0.0000 \end{bmatrix}$$

$$J_{\text{lateral-directional}} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ -0.0369 & -0.2740 & -0.6079 \end{bmatrix}$$

$$L_{\text{lateral-directional}} = [0. \quad 0. \quad 0. \quad 0. \quad 0.]^T$$

$$x_{0(\text{lateral-directional})} = [0. \quad 0. \quad 0. \quad 0.]^T$$

Appendix B

All surfaces have identical actuators. These actuators have a first-order response modeled by

$$\delta_i(s) = [20/(s+20)]\delta_{ic}(s)$$

$$\delta_i(z) = [0.1667(z+1)/(z-0.6667)]\delta_{ic}(z)$$

$$\text{for} \qquad \Delta t = \tfrac{1}{50}\ \text{s}$$

where i refers to the elevator, aileron, or rudder. The position and rate limits for the surfaces are

Position Limit

$$(\delta_{\text{elevator}})_{\max} = +15 \text{ deg and } -20 \text{ deg}$$

$$(\delta_{\text{aileron}})_{\max} = \pm 20 \text{ deg}$$

$$(\delta_{\text{rudder}})_{\max} = \pm 30 \text{ deg}$$

Rate Limit

$$(\delta_{\text{elevator}})_{\max} = \pm 24 \text{ deg/s}$$

$$(\delta_{\text{aileron}})_{\max} = \pm 24 \text{ deg/s}$$

$$(\delta_{\text{rudder}})_{\max} = \pm 24 \text{ deg/s}$$

Actuator noise is modeled by shaping white noise of 0.1 deg intensity with a first-order low-pass filter $[1/(0.01s+1)]$.[3]

Appendix C

All sensors are modeled as unity transfer functions, and sensor noise is modeled by shaping white noise of the proper spectral density with a first-order low-pass filter $[1/(\tau s+1)]$.[3]

Sensor	τ, s	Intensity
Roll rate gyro	0.08	2.0 deg/s
Yaw rate gyro	0.08	0.5 deg/s
Sideslip angle	0.16	0.3 deg
Bank angle	0.16	0.2 deg
Lateral acceleration	0.008	0.04 g

Appendix D

For control system design purposes, the following Dryden vertical gust spectrum model is used[3]:

$$\dot{w} = -2(V_0/L)w + 2\sigma/(\pi L/V_0)^{1/2}\xi, \qquad w(0) = 0$$

where

$$V_0 = 539.0989 \text{ ft/s}, \qquad L = 2500 \text{ ft}, \qquad \sigma = 6 \text{ ft/s}$$

and ξ is the zero mean, Gaussian, white noise. Thus

$$\dot{w} = -0.4313w + 3.1439\xi, \qquad w(0) = 0$$

$$E\{\xi(\tau_1)\xi(\tau_2)\} = 1.0\delta(\tau_1 - \tau_2) \text{ ft/s}$$

Appendix E

Numerical values of the interconnection weights γc_{ki}, αc_{ij}, βc_{jl}, γf_{ki}, αf_{ij}, and βf_{jl} in Eqs. (18–20) and Eqs. (21–23) for lateral stick response are presented:

$$[\beta c]_{8\times 3} = \begin{bmatrix} -0.2933 & 0.9906 & -0.7888 \\ 0.1614 & -0.9599 & 2.3340 \\ -0.1812 & 0.8408 & 1.6988 \\ 0.0082 & -0.4973 & -2.6614 \\ 0.1174 & -1.0173 & -1.0517 \\ -0.2334 & -0.1173 & 2.4466 \\ -0.6967 & 0.3335 & 0.9258 \\ -0.0552 & 0.8868 & 1.7224 \end{bmatrix}$$

$$[\alpha c]_{8\times 8} = \begin{bmatrix} 0.5324 & -0.0474 & -0.1535 & -0.2095 & -0.2868 & 0.6613 & 0.1965 & 0.3825 \\ 0.4673 & 1.1428 & 1.0642 & 0.5663 & 0.7239 & -0.2605 & 0.5403 & 0.6572 \\ 0.3130 & -0.0236 & 0.6896 & -0.5674 & 0.0416 & 1.4436 & 0.3051 & -0.2052 \\ -0.1164 & -0.7943 & -0.6493 & 0.8276 & 1.7994 & -0.4553 & -0.1678 & -0.1493 \\ 0.8026 & 0.3242 & 0.6964 & -1.7588 & -0.4803 & -0.4368 & 0.5873 & -0.0601 \\ 0.1596 & 0.8557 & 0.2835 & 0.0995 & -0.3905 & -0.1834 & 0.5459 & 0.0922 \\ 0.6153 & 0.4822 & -0.5218 & 0.5879 & -2.3691 & -0.9021 & -0.2088 & -0.3442 \\ 0.1816 & 1.1633 & 0.9709 & -0.5985 & -1.7248 & -1.5946 & 0.8610 & 0.9943 \end{bmatrix}$$

$$[\gamma c]_{2\times 8} = \begin{bmatrix} -0.2599 & -0.4440 & 2.0700 & 0.2596 & -1.1010 & -0.6035 & 2.6221 & 0.9177 \\ -0.1206 & 1.3657 & 0.0636 & 1.6724 & 0.0584 & -0.2963 & 0.6017 & 1.1451 \end{bmatrix}$$

$$[\beta f]_{1\times 8} = [0.7523 \quad -0.0917 \quad -0.0208 \quad -4.1711 \quad -0.4991 \quad -0.4722 \quad 9.5981 \quad -2.0367]$$

$$[\alpha f]_{8\times 8} = \begin{bmatrix} -0.1514 & 0.4012 & 0.3557 & 0.7053 & 0.3817 & -0.0998 & 1.2246 & -0.3249 \\ 0.3713 & 0.4317 & -0.4227 & 0.4153 & -0.1776 & -0.1000 & 1.7820 & -0.3130 \\ -0.8856 & -0.7859 & 3.4498 & 0.8910 & 0.3702 & 0.5682 & 4.8058 & 0.8036 \\ -0.1302 & -0.1760 & -1.1508 & -0.4841 & 0.4381 & 0.2049 & 0.1756 & 0.0405 \\ 0.1397 & 0.2178 & -0.0677 & 0.3129 & 0.0023 & -0.1233 & 1.6738 & -0.2378 \\ -0.0730 & 0.0747 & 0.4648 & -0.4139 & 0.0590 & -0.2254 & -1.8315 & -0.2407 \\ -0.2420 & -0.3102 & -0.7514 & 0.4423 & -0.0631 & 0.0229 & 0.5713 & -0.0213 \\ 0.4842 & 0.3211 & -0.5470 & -0.0809 & -0.8371 & -0.4019 & -0.0760 & -0.7117 \end{bmatrix}$$

$$[\gamma f]_{2\times 8} = \begin{bmatrix} -0.4585 & 1.2635 & -3.5426 & -0.1791 & 0.7215 & -1.6585 & 1.4204 & 1.5699 \\ -0.7323 & 0.1874 & 0.1183 & 1.2107 & 0.3423 & -0.2250 & 0.2405 & -0.9629 \end{bmatrix}$$

Acknowledgments

The author is grateful to J. C. Virnig, D. W. Whatley, D. S. Bodden, and Y. P. Wei of the Control Law Design and Analysis Group for helpful advice and fruitful discussions. The author also thanks J. A. Bessolo for generating the linearized lateral-directional aircraft model used in this study.

References

[1] McRuer, D., and Graham, D., "Eighty Years of Flight Control: Triumphs and Pitfalls of the Systems Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 4, No. 4, 1981, pp. 353–362.

[2]Bryson, A. E., Jr., "New Concepts in Control Theory, 1959–1984," *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 4, 1985, pp. 417–425.

[3]Elliot, J. R., "NASA's Advanced Control Law Program for The F-8 Digital Fly-By-Wire Aircraft," *IEEE Transactions on Automatic Control*, Vol. AC-22, No. 5, 1977, pp. 753–757.

[4]Narendra, K. S., and Annaswamy, A. M., *Stable Adaptive Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[5]Sastry, S. S., and Bodson, M., *Adaptive Control: Stability, Convergence, and Robustness*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[6]Bavarian, B., "Introduction to Neural Networks for Intelligent Control," *IEEE Control Systems Magazine*, April 1988, pp. 3–7.

[7]Franklin, J. A., "Historical Perspective and State of the Art in Connectionist Learning Control," *Proceedings of the Twenty-Eighth Conference on Decision and Control* (Tampa, FL), Dec. 1989, pp. 1730–1736.

[8]Antsaklis, P. J., "Neural Networks for Control Systems," *IEEE Control Systems Magazine*, Vol. 9, No. 3, 1989, pp. 25–59.

[9]Antsaklis, P. J., "Neural Networks in Control Systems," *IEEE Control Systems Magazine*, April 1990, pp. 3–5.

[10]Werbos, P. J., "Neural Networks for Control: An Overview," *Proceedings of the 1990 American Control Conference* (San Diego, CA), 1990, pp. 983–984.

[11]Gupta, M. M., "Fuzzy Neural Network Approach to Control Systems," *Proceedings of the 1990 American Control Conference* (San Diego, CA), 1990, pp. 3019–3022.

[12]Sofge, D. A., and White, D. A., "Neural Network Based Process Optimization and Control," *Proceedings of the Twenty-Ninth Conference on Decision and Control* (Honolulu, HI), 1990, pp. 3270–3276.

[13]Ha, C. M., Wei, Y. P., and Bessolo, J. A., "Identification of Aircraft Dynamical Systems Via Neural Networks," Lockheed Fort Worth Co., Engineering Research Rept., June 1991.

[14]Caglayan, A. K., and Allen, S. M., "A Neural Net Approach to Space Vehicle Guidance," *Proceedings of the 1990 American Control Conference* (San Diego, CA), 1990, pp. 1839–1842.

[15]Burgin, G. H., and Schnetzler, S. S., "Artificial Neural Networks in Flight Control and Flight Management Systems," *Proceedings of the 1990 NAECON* (Dayton, OH), 1990, pp. 567–573.

[16]Josin, G. M., "Preliminary Development of a Neural Network Autopilot Model for a High Performance Aircraft," Neural Systems, Inc., Vancouver, BC, Canada, June 1989.

[17]Ha, C. M., Wei, Y. P., and Bessolo, J. A., "Aircraft Flight Control System Design Via Neural Networks," Lockheed Fort Worth Co., Engineering Research Rept., June 1991.

[18]Chen, F. C., "Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control," *IEEE Control Systems Magazine,* April 1990, pp. 44–48.

[19]Military Standard — Flying Qualities of Piloted Vehicles, MIL-STD-1797 (USAF), ASD/ENES, Wright–Patterson AFB, OH, March 1987.

[20]MATRIX$_x$, Version 7.24 (Oct. 1991), Integrated Systems, Inc., Santa Clara, CA.

[21]Ha, C. M., Wei, Y. P., and Bessolo, J. A., "Performance of Neural Network-Based Controller in the Presence of Bounded Uncertainty," Lockheed Fort Worth Co., Engineering Research Rept., June 1991.

[22]Ha, C. M., Wei, Y. P., and Bessolo, J. A., "Reconfigurable Aircraft Flight Control System Via Neural Networks," Lockheed Fort Worth Co., Engineering Research Rept., June 1991.