JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 28, No. 6, November–December 2005

Nonlinear Adaptive Neural Controller for Unstable Aircraft

S. Suresh,* S. N. Omkar,† and V. Mani‡
Indian Institute of Science, Bangalore 560 012, India
and
N. Sundararajan§
Nanyang Technological University, Singapore 63978, Republic of Singapore

A model reference indirect adaptive neural control scheme that uses both off-line and online learning strategies is proposed for an unstable nonlinear aircraft controller design. The bounded-input–bounded-output stability requirement for the controller design is circumvented using an off-line, finite interval of time training scheme. The aircraft model is first identified using a neural network with linear filter (also known as time-delayed neural network) with the available input–output data for a finite time interval. The finite time interval is selected such that this time interval is less than the critical time interval for the aircraft from its stability point of view (similar to the time to double). A procedure to select this critical time interval is also presented. For a given reference model and the identified model, the controller neural-network weights are adapted off-line for the same time interval. The off-line trained neural controller ensures the stability and provides the necessary tracking performance for the unstable aircraft. If there is a change in the aircraft dynamics or characteristics, the trained neural identifier and controller are also adapted online. The theoretical results are validated using the simulation studies based on a locally nonlinear longitudinal high-performance fighter aircraft similar to the F-16. The neural controller design proposed is also compared with the feedback error learning neural control strategy in terms of the tracking ability and control efforts for various level flight conditions and fault conditions such as modeling uncertainties and partial control surface loss. We also present the robustness of the aircraft under extreme wind and noise conditions.

Introduction

OVER the past three decades, adaptive control theory has evolved as a powerful methodology for designing nonlinear feedback controllers for systems with parameter uncertainties. The fundamental issues of adaptive control for linear systems such as selection of controller architecture, assumption on a priori system knowledge, parameterization of adaptive systems, establishment of error models, adaptive law, and analysis of stability have been extensively addressed. These results have been reported in several textbooks dealing with the design and analysis of adaptive systems.[1–3] However, most practical systems are nonlinear in nature. Adaptive control of such nonlinear systems is therefore an intense area of research. Novel techniques in adaptive control of nonlinear systems are facilitated through advances in geometric nonlinear control theory and, in particular, feedback linearization methods[2] and backstepping methods.[3] A key assumption in these methods is that the system nonlinearities are known a priori.

During the past decade, a large amount of research work has been carried out in neural control theory almost independently from adaptive nonlinear control research.[4,5] Neural networks possess an inherent structure suitable for mapping complex characteristics, learning, and optimization. The feasibility of applying neural-network architectures for identification and control of nonlinear systems was first demonstrated through numerical studies in Ref. 6. In these studies neural networks are mostly used as approximate models for unknown nonlinearities, thus removing the need for a priori knowledge of system nonlinearities. The nonlinear relationship between the input and output data is represented by the neural-network parameters, also known as weights.

The role of neural network in adaptive neural flight techniques is to learn some underlying relationship between the given input-output data and approximate the control law. Different architectures and training schemes have been used for this purpose.[6–13] Because of their powerful ability of approximating nonlinear functions and control laws, flight controllers incorporating neural network have been extensively studied. A detailed survey on the application of neural networks for nonlinear flight control is presented in Ref. 14. The online learning ability of neural networks is demonstrated by using modeling uncertainties and partial control surface loss in Refs. 15–18. A reconfigurable flight-control algorithm that is trained to distribute control authority among remaining surfaces in a timely fashion without explicit knowledge of a given failure condition is reported in Refs. 12, 19, and 20. A complete survey of the relevant literature is given in Ref. 21.

Among the various neural-network-based flight-control schemes, the feedback error learning scheme is quite common. In the feedback-error learning-neural-control (FENC) scheme, the control architecture uses a conventional controller in the inner loop to stabilize the system dynamics, and the neural controller acts as an aid to the conventional controller for compensating the nonlinearity. Under severe modeling uncertainties, fault conditions, and time-varying nonlinear dynamics of the plant, the neural network is adapted online to ensure better tracking ability, provided the basic conventional controller satisfies bounded-input–bounded-output (BIBO) stability requirement. Because the conventional controller is not designed for the new conditions, the control effort required by the FENC scheme is usually high when compared to the adaptive neural controllers. Recently[22] different conventional and neural controllers were designed for various fault and nominal conditions. A switching technique based on performance measure is used to select the appropriate controllers. But identifying the fault conditions and switching to appropriate controller is often difficult. To overcome these difficulties, a model reference indirect adaptive neural control (MRIANC) for unstable plants that incorporates a new scheme with both off-line and online learning is presented.

Received 23 August 2004; revision received 30 November 2004; accepted for publication 25 February 2005.

*Research Scholar, Department of Aerospace Engineering. Member AIAA.

†Senior Scientific Officer, Department of Aerospace Engineering.

‡Professor, Department of Aerospace Engineering.

§Professor, School of Electrical and Electronic Engineering. Associate Fellow AIAA.

This paper presents a discrete-time control design procedure using MRIANC that incorporates an off-line and online learning scheme for an "unstable" aircraft. The controller design makes an assumption on the aircraft dynamics that the states and the response/outputs of the aircraft do not escape to infinity in a finite interval of time, that is, the response/outputs of the aircraft remain bounded up to a certain interval of time, referred to as critical time, and afterward they start growing unbounded. Using this mild assumption on aircraft dynamics, an off-line learning procedure is presented to design neural controller. This off-line trained neural controller will stabilize the aircraft dynamics and thereby overcomes the requirement of a BIBO stable aircraft model for identification and control. The input–output recursive equations and a unique control law for the aircraft dynamics are established using the concept presented in Ref. 23. The recursive input–output model and a unique control law are approximated using the identifier network N_I and controller networks N_c, respectively. First, the identifier network N_I is trained off-line using the input–output data generated (between 0 s to critical time) from the unstable aircraft model, for various initial conditions and bounded inputs. Next, the controller network N_c is trained using the arbitrary bounded reference output generated (between 0 s to critical time) from the reference model and the identifier neural network. The multilayer perceptron with a linear filter (also known as time delayed) network is used as basic building blocks for identifier and controller networks. The identifier and controller network are adjusted using static and dynamic backpropagation learning algorithms, respectively.[6] Under fault conditions, the identifier and controller network will quickly learn (online) these changes and provide an appropriate control input to maintain a satisfactory tracking performance. The off-line trained neural-network weights are used as a starting point for online adaptation. The off-line learning scheme eliminates the BIBO stability requirement for neural controller design, and the online adaptation process overcomes the excessive control effort problem. The advantage of the proposed neural control scheme is demonstrated using simulation studies.

The nonlinear perturbed equations for the longitudinal dynamics of a high-performance aircraft are considered for simulation studies. The perturbed equations of the unstable aircraft model are used for designing a pitch-rate command control. Though the aircraft is trimmed at various flight conditions, the input-output data generated from one particular flight condition are used to train the neural networks off-line. The off-line trained neural network is tested at various level flight conditions and is also adapted online for various fault conditions such as modeling uncertainty and partial control surface loss. The robustness of the neural controller is tested under severe wind conditions and noise in measurements.

The proposed neural controller performance is compared with FENC scheme.[16] In the FENC scheme, the unstable aircraft is stabilized by using state feedback control design.[24,25] The gains are selected such that the controller is able to stabilize the aircraft at different level flight conditions. The neural controller present in the outer loop is used to provide the necessary tracking performance. During fault conditions, the neural controller weights are adjusted such that the aircraft follows the command accurately.

Problem Definition

Aircraft Model

The aircraft model used in this study is similar to the high-performance fighter aircraft model as reported in Ref. 26. For all practical purposes the lateral and longitudinal modes are decoupled, and the aircraft is powered by an after burning turbofan jet engine. The longitudinal dynamics of the aircraft is considered for this study. The local nonlinear perturbed equations of the aircraft in level flight condition ($V_t = 150$ ft/s, $\alpha = 15$ deg, and $h = 2000$ ft) are described by the following equations:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + Bu + g(x), \qquad x(0) = x_0$$

$$\boldsymbol{y}_p = Cx + Du \tag{1}$$

where

$$A = \begin{bmatrix} -0.0376 & -0.22 & -0.0246 & -9.81 & -0.1116 \\ -0.03 & -3.2797 & 0.9188 & 0 & 0.0129 \\ 1.219 & -0.5117 & -2.2150 & 0 & -3.2529 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -20 \end{bmatrix}$$

$$B = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 20]^T$$

$$C = [0 \;\; 0 \;\; 1 \;\; 0 \;\; 0], \qquad D = 0$$

where $x \in \Re^5$ are states of the system containing velocity V_t, angle of attack α, pitch rate q, pitch attitude θ, and actuator dynamics; $u \in \Re$ is the elevator input δ_e, and $\boldsymbol{y}_p \in \Re^1$ is the output q. The function $g(\,)$ is a smooth and bounded nonlinear function:

$$g(x) = \begin{bmatrix} 0.009x_1x_2 + 0.00212x_2^2 - 0.009x_1x_3 - 0.12x_1x_2x_3 - 0.0012x_2^3 \\ -0.1914x_2^2 - 0.0271x_2x_3 + 0.00017x_3^2 \\ -0.2712x_3^2 - 0.0145x_3x_2 + 0.00071x_1x_2^2 - 0.00721x_2^3 \\ 0 \\ 0 \end{bmatrix}$$

The function $g(\,)$ contains the second- and higher-order terms of the state variables x (Ref. 27). The elevator input u belongs to a class of bounded signals U, where

$$u(t) = \{u : \|u(t)\| \leq \delta \quad \forall\, t \geq 0\}, \qquad \delta \in \Re^+$$

it is assumed that the constant $\delta \in \Re^+$ is known.

The continuous-time representation of the aircraft linear model is transformed into discrete-time model as

$$x(k+1) = A_d x(k) + B_d u(k) + \bar{g}(x, k)$$

$$\boldsymbol{y}_p(k) = C_d x(k) + D_d u(k) \tag{2}$$

where the system matrices A_d, B_d, C_d, and D_d are obtained by using zero-order hold with a sampling time of 0.02 s and $\bar{g}(x, k)$ is obtained using numerical discretization technique. The short-period poles of the linear model given in Eq. (2) are $-3.2683 \pm 0.4944i$, and the phugoid poles are $0.5022 \pm 1.8306i$.

The aircraft is also trimmed at different flight conditions ($V_t = 250$ ft/s, $V_t = 400$ ft/s, and $V_T = 500$ ft/s), and the local nonlinear perturbed equations are derived. Data for trimmed level flight at sea level, with the center of gravity shifted further away by 5%, are given in Table 1. The center of gravity is shifted further such that the aircraft is unstable at all flight conditions considered in this paper. The neural flight-control system is designed to stabilize the aircraft and also to follow the pilot pitch-rate command signal at all flight conditions. For this purpose, the MRIANC scheme is employed.

Model Reference Indirect Adaptive Neural Control

The objective of MRIANC scheme is defined quantitatively as follows: given an aircraft model, a reference model R_M and a reference pilot input r, the problem is to determine the elevator input to the aircraft u^* (which will be the output of the neural-network

Table 1 Trim data for the aircraft model

Speed V_T, ft/s	AOA α,[a] deg	Elevator δ_e, deg	Throttle (per unit)	Altitude h, ft
150	15	−13	0.6884	2000
250	13	−3.8759	0.1971	2000
400	3.45	−2.2732	0.13	3000
500	2.12	−2.067	0.1552	5000

[a]AOA = angle of attack.

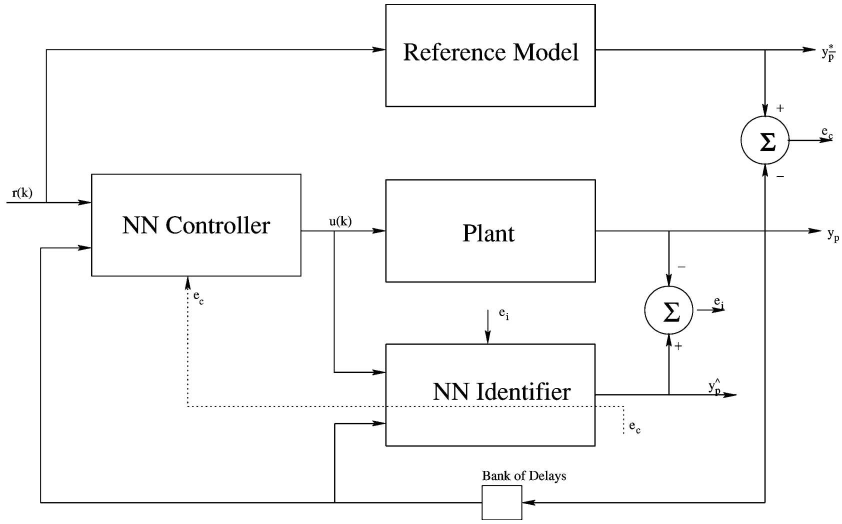

Fig. 1 Model reference indirect adaptive control.

controller) so that the response of the aircraft $\boldsymbol{y}_p$ follows the reference model. The architecture of the MRIANC is shown in Fig. 1. In MRIANC, two neural networks, namely, identifier network N_I and controller network N_c, are used. The identifier neural network is used to approximate the input–output relationship of the aircraft dynamics, and the controller network is used to approximate the unique control law that forces the aircraft output to follow the reference model output accurately.

The method of indirect adaptive control relies on the ability to derive the control law given the identifier model, for a class of systems. In Ref. 6, a model reference indirect adaptive control scheme is used to provide a good tracking performance for an unknown nonlinear plant. But the unknown plant is assumed to be BIBO stable. Because the aircraft considered in this paper is unstable, the method presented in Ref. 6 will not stabilize the aircraft and provide the desired tracking performance. Hence, in this paper we present an off-line and online learning strategy to stabilize the unstable aircraft and also to provide a good tracking performance. The off-line training procedure will be used to overcome the BIBO stability requirement of the plant model. Now, the problem essentially has two parts, the first of which is to derive the identifier model such that the neural model follows the aircraft dynamics accurately in some sense:

$$\|\hat{\boldsymbol{y}}_p(t) - \boldsymbol{y}_p(t)\| < \epsilon\ \forall\, t \in [0, T]$$

where $\hat{\boldsymbol{y}}_p(t)$ is the neural model predicted output and ϵ is a small positive integer.

The second part is to determine the controller network for a given identifier model and to ensure that the aircraft output follows the reference model outputs accurately. The convergence of the controller neural network depends on the accurate modeling of the identifier network. The two different parts of model reference indirect adaptive control problem for unstable aircraft are discussed in detail in the following sections.

Identification

In this section, we first present a strategy to choose an appropriate neural-network model and then discuss the off-line learning scheme to identify the unstable aircraft dynamics.

Formulation

One of the basic requirements in using the neural-network (NN) architectures to identify the nonlinear dynamical systems is the capability of these architectures to accurately model the behavior of a large class of dynamical systems that are encountered in real-world problems. This leads to the question of whether a given NN architecture is able to approximate the input-output response of an unstable

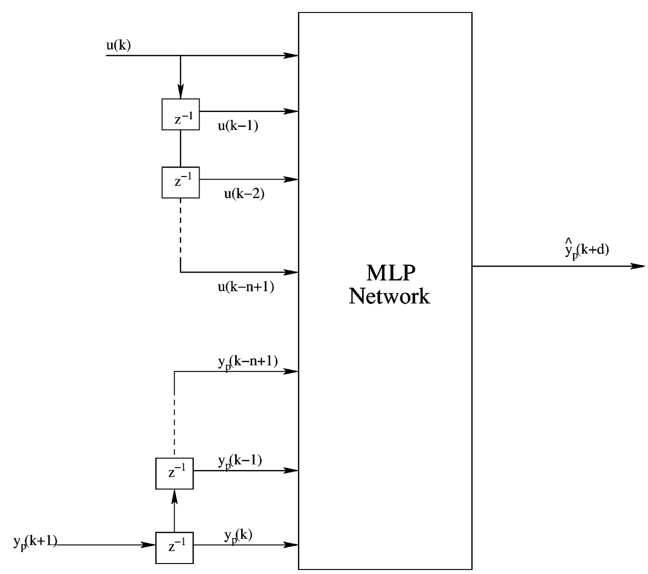

Fig. 2 Neural network architecture.

aircraft in some appropriate sense. The input–output response of the aircraft is represented in terms of network architecture and its weights. Therefore, the representation capabilities of the given network depend on whether there exists a set of weight matrix w_f such that the NN configuration approximates the behavior of a given system.

For an observable system of order n, it is well known[23] that the states $x(k+1)$ can be represented as a function of $y(k)$, $y(k-1)$, $\ldots y(k-n+1), u(k), u(k-1), \ldots, u(k-n+1)$ so that the aircraft dynamics given in Eq. (2) in the neighborhood of the equilibrium point can also be represented as

$$\boldsymbol{y}_p(k+d) = F[\boldsymbol{y}_p(k), \ldots, \boldsymbol{y}_p(k-n+1), u(k), \ldots, u(k-n+1)] \tag{3}$$

where d is the relative degree (or equivalent delay) of the system. It is assumed that the order of the system n and relative degree are specified, whereas the nonlinear function $F[\,]$ is unknown.

Based on the preceding equation, following the approach given in Ref. 6, one can construct a neural-network model as shown in Fig. 2. The inputs to the neural network are the present input and past n inputs and outputs/response of the aircraft. The interconnection of static multilayer perceptron and dynamic elements (past inputs and outputs) is proposed for modeling the input-output response of the system described by Eq. (3). Such networks are called neural networks with linear filter (also known as time-delayed neural networks). The input–output behavior ($u \rightarrow \hat{\boldsymbol{y}}_p$) of a two-layer sigmoidal neural network ($N_I^{l,n_h,m}$) with l inputs, m outputs, and n_h hidden neurons is given by

$$\hat{\boldsymbol{y}}_p(k+d) = N_I[\boldsymbol{y}_p(k), \ldots, \boldsymbol{y}_p(k-n+1), u(k), \ldots, u(k-n+1), w_f]$$

$$\hat{\boldsymbol{y}}_p(k+d) = N_I[V, w_f] \tag{4}$$

where w_f is the weight matrix.

Let V be the input to the neural network at any instant k and let $N_I[\,]$ be the neural network approximation for the function $F[\,]$. If we suppose that the system and the neural-network model are initially at the same state (i.e., $\hat{x}_0 = x_0$), then we have to prove that there exists an optimal weight vector $\boldsymbol{w}_f^*$ such that the input–output behavior ($u \rightarrow \hat{\boldsymbol{y}}_p$) of the neural-network model [Eq. (4)] approximates, in some sense, the input–output behavior ($u \rightarrow \boldsymbol{y}_p$) of the aircraft dynamics [Eq. (2)] . The optimal weight vector $\boldsymbol{w}_f^*$ that approximates the system is given as

$$\boldsymbol{w}_f^* := \arg \min_{w_f \in B(w)} \left\{ \sup_{V, y_p \in \aleph} \|\hat{\boldsymbol{y}}_p(k+d) - \boldsymbol{y}_p(k+d)\| \right\} \tag{5}$$

where $\aleph$ is the set consisting of all network inputs V and target vector $\boldsymbol{y}_p$. $B(w)$ is a (large) compact set of weight vector, $B(w) := \{w_f : \|w_f\| \leq \delta\}$ denotes a ball of radius δ. In adaptive law, the estimated weight vector w_f is also restricted to $B(w)$. The neural-network weights are adapted based on the identification error $e_i(k)$ between the actual aircraft response and the neural-network model:

$$e_i(k) = \hat{\boldsymbol{y}}_p(k+1) - \boldsymbol{y}_p(k+1) \tag{6}$$

Because the inputs to the neural networks are independent of the present output of the aircraft, static backpropagation training algorithm is used to adapt the network weights.

If the given system to be identified is BIBO stable, then for any bounded input $u(k)$ the output $\boldsymbol{y}_p(k)$ will also be bounded. Hence, the input V to the neural network is also bounded, that is, the set containing the input and aircraft response $\aleph$ belong to a compact set. By universal approximation property of neural networks, it is possible to approximate any function to desired accuracy if the inputs and outputs belong to compact sets. Hence, the neural-network weight vector will converge to optimal value.[28–30] Because the aircraft considered in this paper is unstable, the response of the aircraft can grow unbounded for bounded elevator input, that is, input–output data set does not belong to compact set. Hence the neural identifier model might not converge to an optimal value.

For the purpose of identification of unstable aircraft dynamics, we make a mild assumption on the boundedness on aircraft state/output responses.

Assumption 1: Let us assume that for a given class of bounded input u and finite initial conditions x_0, the aircraft response does not escape to infinity in finite interval of time:

$$\sup_{t \in [0, T_c]} \|\boldsymbol{y}_p(t)\| \leq \Delta \tag{7}$$

where Δ is a known real positive number. The time T_c is referred to as the critical time.

The critical time T_c depends on the structure of the system and also on the class of bounded input signals. Using the preceding statement, we assume that the response of the aircraft is less than Δ within the time interval $[0, T_c]$, for all bounded input u. Hence, the set containing all possible bounded input and aircraft response in the time interval $[0, T_c]$ belongs to a compact set. Now, we can state the following definition.

Definition: A network is said to satisfy the universal approximation property, if on any compact subset $C \subset \Re^n$ of the input space it is always possible to find an appropriate number of hidden neurons and optimal weight vectors (not necessarily unique), such that any continuous functions can be approximated to any arbitrary level of accuracy.

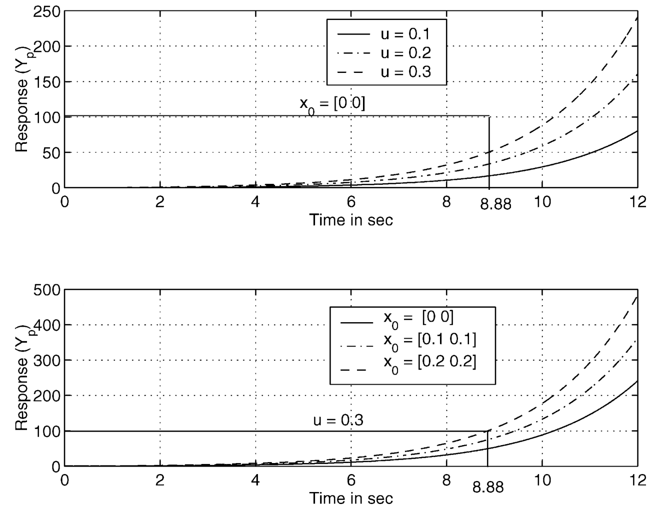

Fig. 3 Selection of T_c: response of the system (8) under various initial conditions and inputs.

Note that the preceding assumption and the definition prove the existence of the optimal network parameters. The details on convergence of the static backpropagation learning algorithm are given in Ref. 31. Thus the proposed off-line learning scheme will be able to identify the unstable aircraft dynamics within the time interval $[0, T_c]$. To illustrate the procedure for selecting the critical time T_c, a simple example is presented next.

Selection of T_c

Let us consider an unstable discrete time system $S(z)$ as defined here:

$$S(z) = \frac{z+2}{z^2 + 1.5z - 1} \tag{8}$$

The preceding system is subject to step signals with various initial conditions. The bounds on inputs and initial conditions are selected as ± 0.3 and ± 0.2 rad, respectively. For various step inputs of 0.1, 0.2, and 0.3 rad and initial conditions, the response of the system described in Eq. (8) is shown in Fig. 3. We can observe that the system responses go to infinity after some time T for a given input and initial conditions. Let Δ be equal to 100, the selection of Δ depending on the sensor capabilities. Now, the critical time T_c is selected such that the response of the system is always less than Δ for various inputs and initial conditions as already defined. From Fig. 3, we can observe that the response of the system is always less than 100 in [0,8.88] time interval, and so the critical time T_c can be selected as 9 s.

Simulation Results

In this section, we present the simulation studies for identification of longitudinal dynamics of an unstable aircraft. The dynamics of the aircraft are defined in Eq. (2). For this study, the bound Δ, critical time T_c, and sampling time t_s are selected as 5 rad, 10 s, and 0.02 s, respectively. The value of bound Δ and critical time T_c are found using the concept explained in the preceding section. The input to the elevator is generated using pseudo random signal for 0–10-s duration with the magnitude bounded between ± 0.15 rad. For various initial conditions, the input signal δ_e and the pitch-rate response q are measured. This data set is used as training set for identifier neural network N_I.

The order of the aircraft dynamics n and its relative degree d are assumed to be four and one, respectively. Hence, along with the current input of the elevator, four past elevator inputs and pitch-rate response of the aircraft model form the input vector to the neural network N_I. The network architecture used in the simulation study is given by $N_f^{9,25,1}$ (9 input nodes, 25 hidden nodes, and 1 output node). The optimal number of hidden nodes required to approximate the input–output representation is selected by using the technique

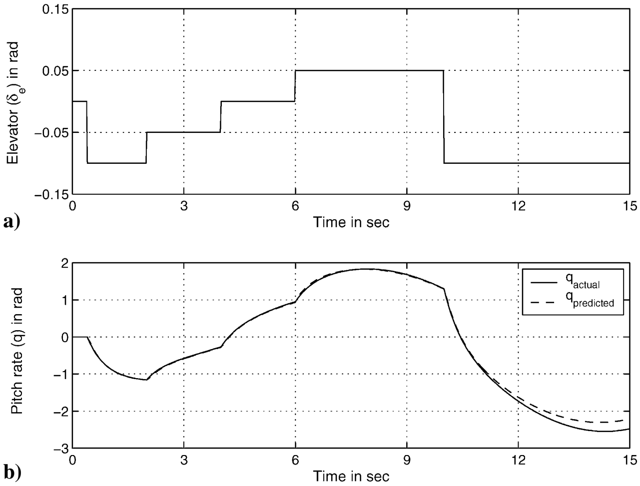

Fig. 4 Simulation result: identification of longitudinal dynamics of unstable aircraft.

presented in Refs. 32 and 33. Bipolar sigmoidal function is used as an activation function in the hidden and output layer. To accelerate the learning process, the input–output data are normalized between ±0.8 (Ref. 34). The network parameters are adapted such that the error between the actual plant output and network predicted output is less than 0.002 mean square error (MSE). The trained network is tested for its generalization ability with pseudorandom signal for 15-s duration. The network response and actual output of the neural network are computed. Figure 4a shows the input to the elevator, and Fig. 4b shows the pitch-rate q response of the neural-network model and actual aircraft model. Because the neural network is trained between the time interval [0, 10], the network response is not good after the critical time 10. This can be observed clearly in Fig. 4b. The network predicted pitch-rate $\hat{q}$ output follows the actual output q of the aircraft closely up to T_c, and after T_c the network response does not follow the actual response accurately. From Fig. 4b, we can say that the identifier network N_I is able to capture the nonlinear mapping $F[\,]$ of unstable aircraft dynamics very well up to time T_c.

Neural Controller Design

In this section we consider a strategy to choose an appropriate controller network to stabilize the unstable aircraft dynamics and also follow the arbitrary reference output signal generated from the reference model. We first formulate the control problem of unstable aircraft and later discuss the learning strategy.

The objective of the control problem is to find the control input $u(k)$ such that the aircraft output $\boldsymbol{y}_p(k)$ tracks any arbitrary output sequence $\boldsymbol{y}_p^*(k)$. The reference model is selected based on the flying quality requirements for fighter aircraft.[35,36] The desired output is an arbitrary sequence $\boldsymbol{y}_p^*(k)$ generated from the reference model R_m between the time interval $[0, T_c]$. Because the reference model is also a class of dynamical system, the output $\boldsymbol{y}_p^*(k+d)$ can be represented by its past inputs $r(k)$ and outputs $\boldsymbol{y}_p^*(k)$:

$$\boldsymbol{y}_p^*(k+d) = \bar{F}_m\big[\boldsymbol{y}_p^*(k), \ldots, \boldsymbol{y}_p^*(k-n+1), r(k), \ldots, r(k-n+1)\big] \tag{9}$$

$\bar{F}_m(\,)$ is a smooth (continuous) function.

If the asymptotic stability of the zero dynamics together with a well-defined relative degree ensures the existence of a control input that can make the aircraft follow any arbitrary signal $\boldsymbol{y}_p^*$ (Ref. 37), then the controller has the form

$$u(k) = G\big[\boldsymbol{y}_p(k), \ldots, \boldsymbol{y}_p(k-n+1), u(k), \ldots, u(k-n+1), \boldsymbol{y}_p^*(k+d)\big] \tag{10}$$

where function map $G[\,]$ exists and is unique. Hence, the aim of controller neural network is to approximate the function map $G[\,]$.

Substituting Eq. (9) in Eq. (10),

$$u(k) = \bar{G}\big\{\boldsymbol{y}_p(k), \ldots, \boldsymbol{y}_p(k-n+1), u(k), \ldots, u(k-n+1), \bar{F}_m\big[\boldsymbol{y}_p^*(k), \ldots, \boldsymbol{y}_p^*(k-n+1), r(k), \ldots, r(k-n+1)\big]\big\} \tag{11}$$

The input u de pends on the past aircraft response and reference signals $\boldsymbol{y}_p, \boldsymbol{y}_p^*, r$, and if the aircraft is able to track any arbitrary sequence, then $\boldsymbol{y}_p(k+d) = \boldsymbol{y}_p^*(k+d)$. Hence, we can replace $\boldsymbol{y}_p^*$ by $\boldsymbol{y}_p$ in the preceding equation:

$$u(t) = K[\boldsymbol{y}_p(k), \ldots, \boldsymbol{y}_p(k-n+1), r(k), \ldots, r(k-n+1)] \tag{12}$$

where $K[\,]$ is a smooth function.

The mapping $K[\,]$ is not known. Hence, it is difficult to calculate the target $u(k)$ (elevator input δ_e) to train the controller network N_c. The neural-network architecture shown in Fig. 2 is used to approximate the unknown function $K[\,]$. The inputs to the network N_c are the past reference inputs and response of the aircraft. The parameters of controller network are updated using a dynamic backpropagation algorithm.[6] The error e_c between the reference output $\boldsymbol{y}_p^*$ and aircraft response $\boldsymbol{y}_p$ is backpropagated through the identifier network N_I to calculate the error at the output neurons of network N_c. Figure 2 shows explicitly the delayed inputs that are fed to the network N_I. Let us suppose that the aircraft response depends on $u(k)$, $u(k-1)$, $u(k-2)$, and $u(k-3)$, and hence these are fed to the network N_I by delays from the single output δ_e of N_c. Initially we do not consider the other inputs such as past outputs of the aircraft to N_I, as they are not important to this discussion. For training N_c, we need to find the error at the output node of N_c, which is directly connected to the input node 1 of the network N_I. So, we have to backpropagate the error e_c through N_I to reach the node 1 in its input layer. Because the aircraft response depends on many previous inputs, the correct procedure to calculate the error at output node of N_c is the dynamic backpropagation algorithm,[6] that is, propagate the error through delay line (z^{-1}). In practice, this would mean that we have to backpropagate the error e_c up to all input nodes of N_I and then add appropriately delayed versions of these errors to get the error at the output node of N_c.

The pseudorandom reference inputs are generated between the time interval $[0, T_c]$, and the reference output $\boldsymbol{y}_p^*$ is computed. These data are used to train the network N_c. A performance index J for training process is defined as

$$J(t) = \frac{1}{N}\sum_{k=1}^{N}(e_c)^2 \tag{13}$$

where $e_c = \boldsymbol{y}_p^*(k) - y_p(k)$. The network weights are adjusted such that the performance index J is minimized.[6]

The estimation of error at output node of N_c will be accurate only when the identifier N_I network approximates the system response well. Because the identification model is valid up to time T_c, the calculation of error at network N_c will also be accurate in the time interval $[0, T_c]$. Hence, the controller network parameters are adapted between the time interval $[0, T_c]$. The trained neural network is used as a starting point for further adaptation, in case of parameter uncertainty.

For the simulation study, pitch-rate command control is used as a reference model. The reference model (pitch-rate command control) is defined based on the desired flying quality requirements.[35,36] To train the controller network N_c, the pseudorandom pulse reference input r and the arbitrary output sequence for pitch rate q are generated for 10-s duration. Similarly, 20 data sets are generated for various random signal and initial conditions. These data sets are used to adapt the controller weight matrices. The past four input and output signals and present input signal are fed as inputs to the controller N_c. The output of the controller N_c is the elevator deflection δ_e and is the input to the aircraft model. The difference between the aircraft and the reference model is backpropagated through the network N_I to adapt the weights of network N_c. The controller network N_c structure used in the simulation studies is $N_c^{9,25,1}$, that is, 9 input nodes, 25 hidden nodes, and 1 output node. The controller

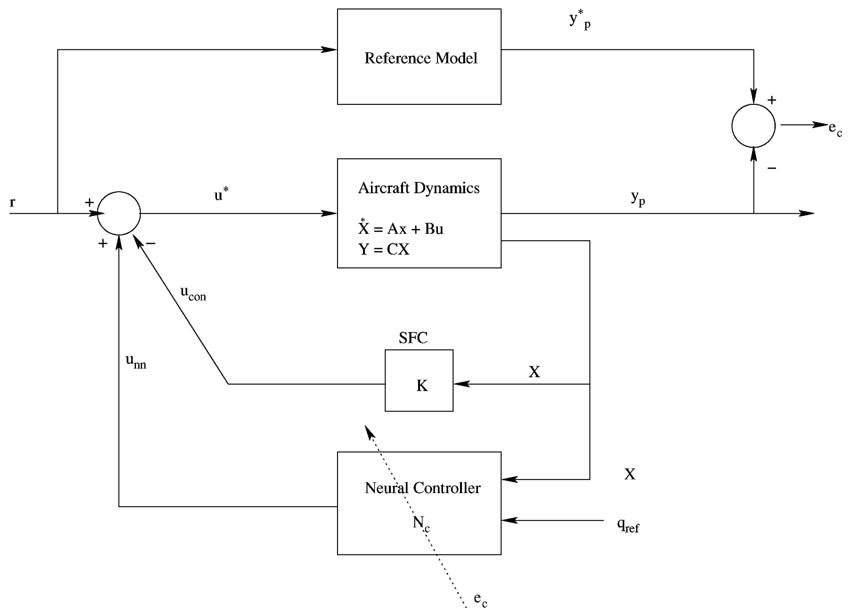

Fig. 5 Block diagram of FENC scheme.

network architecture is also selected based on the procedure given in Refs. 32 and 33. The N_c weights are adapted until the performance index J is less than 0.002.

Feedback-Error Learning Neural-Control Scheme

In this section, we present the FENC for an unstable aircraft. The block diagram of the FENC scheme is shown in Fig. 5. In the FENC scheme, the conventional state feedback controller in the inner loop is used to stabilize the aircraft, and the neural controller in the outer loop approximates the unknown nonlinearity and provides the necessary tracking performance. The neural controller is trained to minimize the deviation between the reference signal y^* and actual output of the aircraft. The control effort applied to the aircraft is the sum of the conventional and neural controller signals:

$$u(k) = u_{\mathrm{nn}}(k) - u_{\mathrm{con}}(k) + r(k) \tag{14}$$

where u_{nn} is neural-network output and u_{con} is the control input from the state feedback controller (SFC).

The conventional SFC is designed based on the linearized model at level flight condition ($V_T = 150$ ft/s). The controller gain matrix is

$$K = [0 \quad -0.08 \quad -0.28 \quad -0.49]$$

The SFC is able to stabilize the aircraft at various level flight conditions. In this study, radial basis function network is used to approximate the unknown nonlinearity. The stability and convergence of the preceding approach is discussed in Ref. 16.

Simulation Results and Discussion

The performance capabilities of the SFC, FENC, and MRIANC schemes are tested with reference pulse input of 0.05 rad at various level flight conditions. The differential equations describing the aircraft motion are solved using the Runga–Kutta higher-order method. The reference pitch-rate response and aircraft response with different controllers for pulse input at level flight condition ($V_T = 150$ ft/s) are shown in Fig. 6. The control effort required by the flight controllers is presented in Fig. 6a (the elevator inputs are within the limit $\delta_e \leq \pm 24$ deg), and the pitch-rate and attitude responses are shown in Figs. 6b and 6c. From this figure, we can observe that the MRIANC scheme stabilizes the aircraft and also follows the pitch-rate command accurately. The control effort required by the proposed MRIANC scheme is much less than that of the FENC and SFC schemes. The control effort is calculated by using area

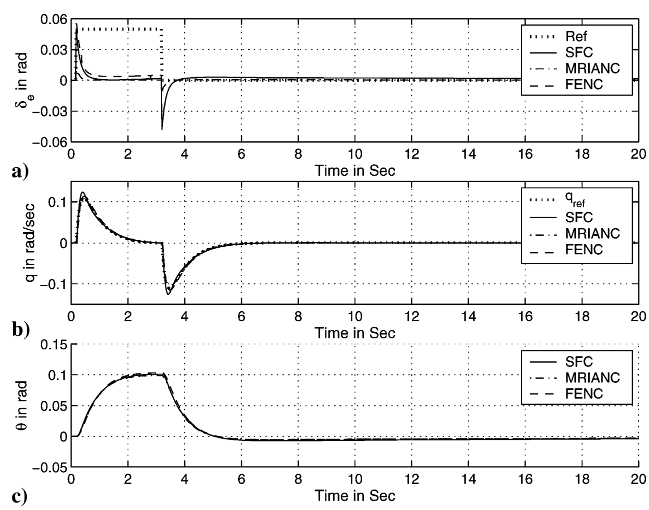

Fig. 6 Controller responses for pulse reference input at nominal flight condition (V_T = 150 ft/s).

under the elevator deflection curve. The qualitative performance measures like maximum error, rms error, and maximum control deflection and control effort are calculated for the flight controllers at different level flight conditions and are given in Table 2. From Table 2, we can observe that the proposed MRIANC scheme performs better than the FENC and SFC schemes. For example, the proposed MRIANC scheme requires a maximum of 0.0154 rad of elevator deflection at 150 ft/s, whereas the FENC scheme requires 0.0592 rad. The maximum control surface deflection in the FENC scheme is approximately three to four times more than the proposed MRIANC scheme. This fact can be clearly observed from the control effort required. Similar performance can be observed for other flight conditions. The simulation study indicates that the proposed off-line learning strategy is able to stabilize the unstable aircraft and also provide good tracking performance. Now, we present the online learning ability of the neural network for various fault conditions.

Response Under Fault Conditions

To study the robustness of the flight controllers, modeling uncertainties and partial control surface loss conditions are considered. The neural network in the MRIANC and FENC schemes are adapted online for fault conditions such that the aircraft follows the reference pitch-rate command accurately. The controller neural network N_c

Table 2 Quantitative results for controllers at different flight conditions

		MRIANC				FENC			
		q error				q error			
Trim V_T, ft/s	Test cases	Max. rad/s	RMS rad/s	Max $\|\delta_e\|$, rad	Cont. effort	Max rad/s	RMS rad/s	Max $\|\delta_e\|$, rad	Cont. effort
150	Nominal	0.0051	0.0014	0.0151	0.0154	0.0101	0.0095	0.0592	0.0632
250		0.0118	0.0087	0.0196	0.0261	0.0181	0.0115	0.0671	0.0734
400		0.0185	0.0154	0.0205	0.0274	0.0251	0.0261	0.0782	0.0819
500		0.0081	0.0112	0.0235	0.0309	0.0201	0.0097	0.0813	0.0859
150	$A = 0.3A$	0.0111	0.0054	0.0251	0.0069	0.0313	0.0189	0.0752	0.0364
250		0.0202	0.0107	0.0366	0.0124	0.0478	0.0275	0.0891	0.0412
400		0.0339	0.0199	0.0465	0.0164	0.0592	0.0361	0.0910	0.0495
500		0.0174	0.0082	0.0495	0.0212	0.0375	0.0136	0.0976	0.0521
150	$B = 0.3B$	0.0106	0.0033	0.0285	0.0319	0.0321	0.0125	0.0992	0.1757
250		0.0154	0.0108	0.0491	0.0425	0.0481	0.0248	0.1071	0.2123
400		0.0320	0.0148	0.0601	0.0612	0.0595	0.0465	0.1482	0.2941
500		0.0289	0.0208	0.0673	0.0734	0.0595	0.0465	0.1482	0.3427
150	$A = 1.5A$	0.0118	0.0076	0.0125	0.0117	0.0138	0.0096	0.0582	0.1221
250		0.0151	0.0110	0.0172	0.0174	0.0191	0.0099	0.0637	0.1514
400		0.0171	0.0171	0.0199	0.0251	0.0214	0.0212	0.0521	0.1721
500		0.0212	0.0076	0.0210	0.0297	0.0252	0.0146	0.0712	0.1914
150	$A = 1.75A$	0.0091	0.0056	0.0113	0.0099	0.0151	0.0109	0.0551	0.1197
250		0.0142	0.0141	0.0165	0.0167	0.0121	0.0056	0.0612	0.1487
400		0.0154	0.0121	0.0187	0.0247	0.0234	0.0197	0.0489	0.1697
500		0.0197	0.0101	0.0203	0.0281	0.0272	0.0212	0.0701	0.2021

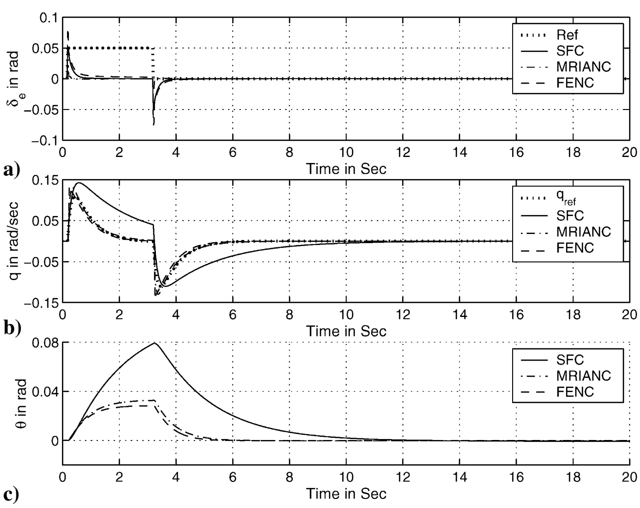

Fig. 7 Response of the controllers for 70% modeling uncertainty ($A = 0.3A$) at $V_T = 150$ ft/s.

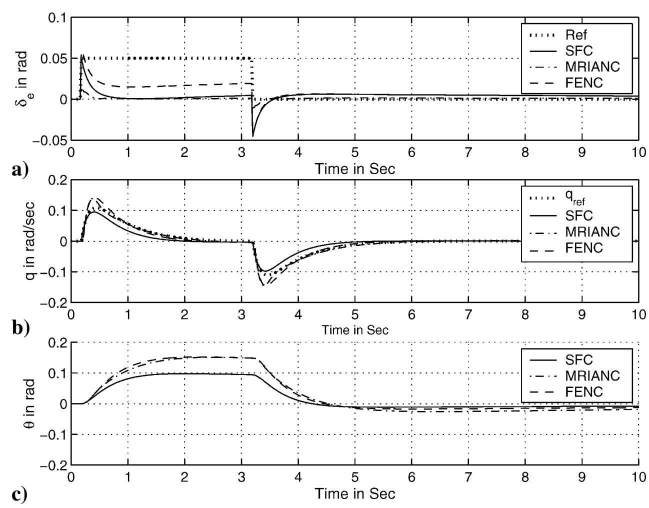

Fig. 8 Response of the controllers for multiplicative uncertainty ($A = 1.75A$) at $V_T = 150$ ft/s.

is updated for a window of time (approximately 30 s) with learning rate of 0.3. During this time interval, pseudorandom pulse input signal is injected into reference input channel. The reference inputs and pitch-rate command signals are used to adapt the control weight matrices. The online adaptation of controller weight matrices is stopped, if the performance of the controller is within the specified limit.

System Matrix Fault

Let us first consider the 70% modeling uncertainty problem, that is, the elements of the system matrix A given in Eq. (1) are modified as $A = 0.3A$. The controller weight matrix is adapted online. The response of the flight controllers after online adaptation at $V_T = 150$ ft/s is shown in Fig. 7. The control effort required is shown in Fig. 7a, and the pitch-rate response is shown in Fig. 7b. From Fig. 7a, we can observe that the control effort required to follow the pitch-rate command is within the maximum limit. From this figure, we can also see that the proposed MRIANC scheme has better tracking performance and lesser control effort than the FENC and SFC schemes. The preceding result demonstrates the online training ability of the neural controller to adapt to any damage level. The performance of the flight controller at various level flight conditions under modeling uncertainty is given in Table 2. From Table 2, we can say that the MRIANC scheme has better tracking performance and requires less control effort.

To study the robustness of the proposed control scheme, the system matrix A is also multiplied with factors 1.5 and 1.75. The response of the flight controller at $V_T = 150$ ft/s with multiplicative uncertainty of 175% ($A = 1.75A$) is shown in Fig. 8. The pitch-rate and attitude response and control surface deflection of SFC, MRIANC, and FENC are shown in Figs. 8a–8c. From these figures, we can say that the proposed MRIANC scheme stabilized the aircraft model and also provided necessary tracking performance. The same controller is tested at different flight conditions, and the performance of the controller is tabulated in Table 2. Table 2 also presents the performance of flight controller with factor of 1.5. The simulation results clearly demonstrate the ability of the proposed MRIANC scheme under various fault conditions.

Control Surface Loss

Similarly, the performance of the closed-loop systems is studied for partial control surface loss. The results for 70% ($B = 0.3B$) control surface loss at $V_T = 400$ ft/s are presented. The response of the neural controller after online adaptation is shown in Fig. 9.

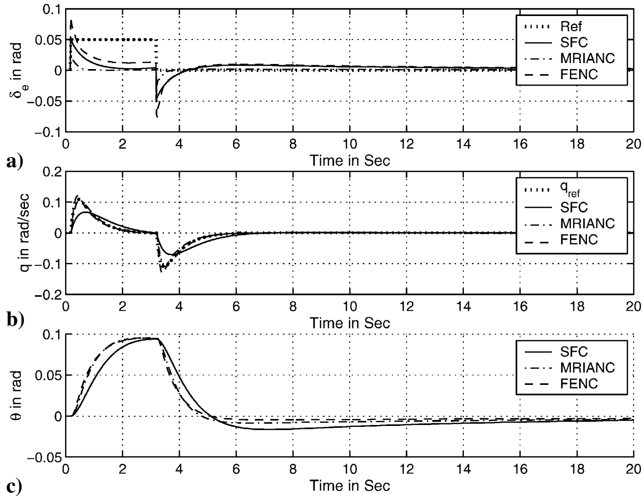

Fig. 9 Response of the controllers for partial control surface loss ($B = 0.3B$) at $V_T = 400$ ft/s.

The elevator input under fault conditions is shown in Fig. 9a, and the aircraft response is shown in Fig. 9b. From Fig. 9a, we can observe that the input to the elevator is within the maximum limit. From Fig. 9b, we can observe that the SFC is not able to stabilize the aircraft alone under 70% control surface loss condition at $V_T = 400$ ft/s, whereas the same SFC is able to stabilize the aircraft at 150- and 250-ft/s level flight conditions. Hence, the feedback gains K are modified such that the plant is stable under control surface fault. The controller neural network in the FENC scheme is adapted online with new gain values for SFC. This condition clearly indicates the requirement of an online adaptive controller. Using the online learning scheme, FENC and MRIANC are able to stabilize the aircraft, but the control effort required by the FENC scheme is more than by the proposed scheme. The quantitative results for 70% control surface loss at various level flight conditions are given in Table 2, from which we can clearly observe that the controller parameters can be reconfigured and the proposed neural controller can accurately track the reference output under parameter uncertainties and partial control surface loss conditions.

Response Under Gust Disturbance and Noise Rejection

The aircrafts are highly susceptible to atmospheric turbulence that commonly occurs during flight. To determine the gust rejection specifications of the closed-loop system, the vertical wind gust noise is taken to have a spectral density given in Dryden form as[35,36]

$$\phi_w(\omega) = \frac{2L_w\sigma^2}{\phi U}\frac{1 + 12(L_w/U)^2\omega^2}{[1 + 4(L_w/U)^2]^2} \tag{15}$$

where ω is the frequency in rad/s, σ is the turbulence standard deviation, L_w is the turbulence scale length, and U is the flight velocity. The turbulence scale length at 4000-ft altitude is $L_w \approx 883$ ft. The turbulence standard deviations are defined in statistical terms and classified as light, severe, and moderate, with $\sigma = 5.1$ ft/s for light wind, $\sigma = 10$ ft/s for moderate wind, and $\sigma = 20$ ft/s for severe wind conditions. The aircraft is flown in severe wind conditions, and therefore a standard deviation of 18 ft/s is chosen. The frequency range of concentration of gust disturbance is found to increase with speed. This represents the worst-case scenario when disturbances can excite both the phugoid and short-period modes. Taking this as the benchmark, the gust rejection specification is to reject all disturbances below 13 rad/s.

In general the sensor used for measuring pitch rate will have an built-in 50-Hz low-pass filter, but the sensor outputs are noisy and biased with noise concentration in the region above 30 rad/s. These noises coupled with the mechanical vibrations of the airframe could lead to erroneous measurements making the control of aircraft difficult. Thus, high-frequency specification is to reject all noise above 30 rad/s.

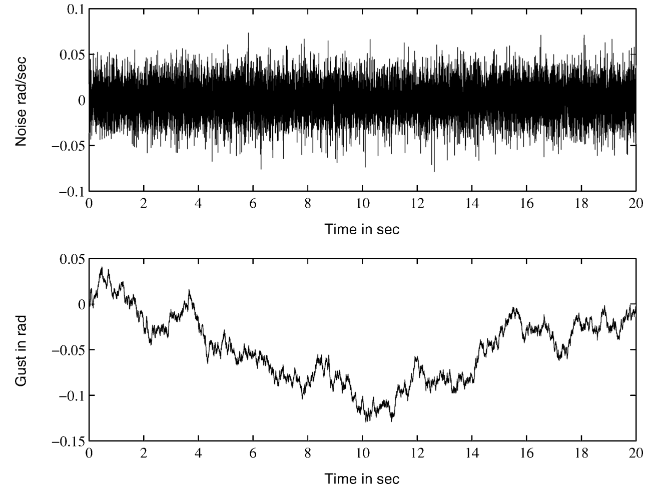

Fig. 10 Sensor noise and gust disturbance.

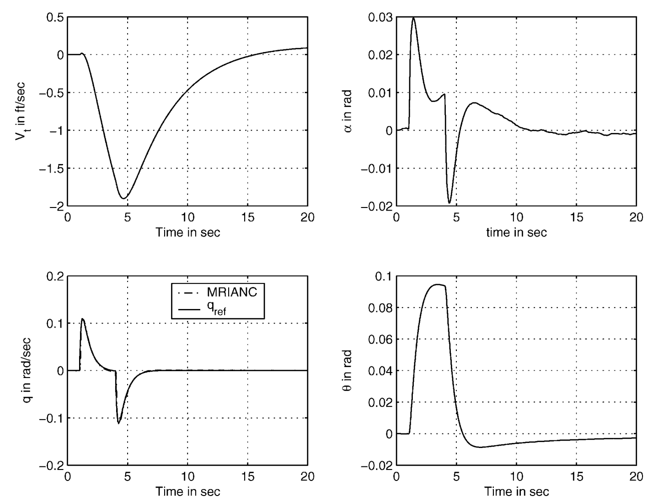

Fig. 11 Response of the MRIANC for severe gust and noise condition at $V_T = 150$ ft/s.

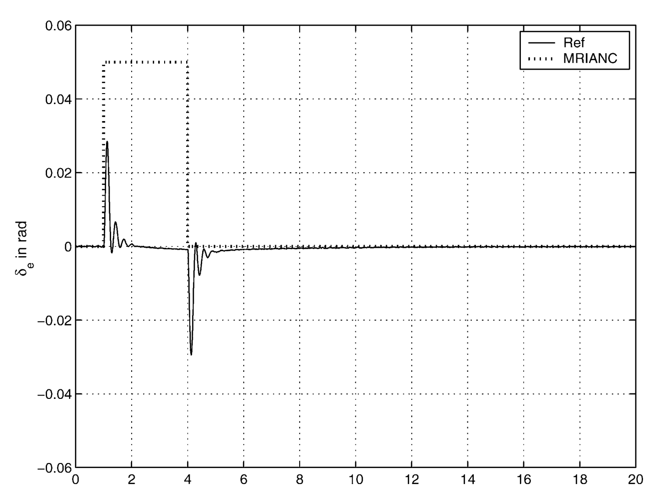

Fig. 12 Control surface deflection under severe gust and noise conditions at $V_T = 150$ ft/s.

Sensor noise is added by passing white noise through a high-pass shaping filter $0.02(s + 30)/s + 100$, which corresponds to a high-frequency gain of 0.02. The atmospheric gust disturbance and sensor noise added to the system are shown in Fig. 10. The simulation studies are carried out at 150 ft/s flight condition with the vertical gust and noise in the measurement. The response of the MRIANC is shown in Fig. 11. The control surface deflection under this condition is shown in Fig. 12. From this figure, we can observe that the neural controller rejects the noise and gust very well and also that the control surface deflection is within the limit.

Conclusions

A discrete-time model reference indirect adaptive neural control scheme that incorporates off-line and online learning strategies is presented for unstable nonlinear aircraft model. Using a mild assumption on the aircraft model, an off-line training scheme is developed. The bounded-input–bounded-output stability requirement for neural controller design is circumvented by using the off-line training scheme. The off-line trained neural identifier and controller are adapted online for model uncertainties and control surface loss. The proposed control scheme is compared with well-known feedback-error learning neural control scheme. The online training/reconfiguration shows that the proposed control scheme requires less control effort to provide better tracking performances. The robustness of the proposed neural flight control is tested under severe wind and noise conditions, and the simulation results indicate that the proposed control scheme rejects the disturbance very well. The simulation studies based on an F-16 aircraft model demonstrate the benefits of using the proposed control scheme.

Acknowledgments

The last author acknowledges gratefully the support extended to him by the Departments of Computer Science and Automation and Aerospace Engineering, Indian Institute of Science, Bangalore, to visit the Institute as I. G. Sarma, ARDB Memorial Professor, during which time this study was undertaken. The authors are very thankful to the reviewers for their comments and suggestions, which have improved the quality of presentation.

References

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

[2]Marino, R., and Tomei, P., *Nonlinear Control Design, Geometric, Adaptive, Robust*, Prentice–Hall, 1995.

[3]Isidori, A., *Nonlinear Control Systems*, Springer-Verlag, New York, 1989.

[4]Lewis, F. L., Jagannathan, S., and Yesildirek, A., *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor and Francis, London, 1998.

[5]Gupta, M. M., and Rao, D. H., *Neuro-Control Systems: Theory and Applications*, IEEE Press, New York, 1994.

[6]Narendra, K. S., and Parthasarathy, K., "Identification and Control of Dynamical System Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, 1990, pp. 4–27.

[7]Anderson, C. W., "Q-Learning with Hidden Unit Restarting," *Advance in Neural Information Processing Systems*, Vol. 5, Morgan Kaufmann, San Mateo, CA, 1993, pp. 81–88.

[8]Puskorius, G. V., and Feldkamp, L. A., "Model Reference Adaptive Control with Recurrent Networks Trained by the Dynamic DEKF Algorithm," *International Joint Conference on Neural Networks*, Vol. 2, June 1992, pp. 106–113.

[9]Hovakimyan, N., Nardi, F., Calise, A. J., and Lee, H., "Adaptive Output Feedback Control of a Class of Nonlinear Systems Using Neural Networks," *International Journal of Control*, Vol. 74, No. 12, 2001, pp. 1161–1169.

[10]Polycarpou, M. M., "Stable Adaptive Neural Control Scheme for Nonlinear Systems," *IEEE Transactions on Automatic Control*, Vol. 41, No. 3, 1996, pp. 447–451.

[11]Gili, P. A., and Battipede, M., "Adaptive Neurocontroller for a Nonlinear Combat Aircraft Model," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 910–917.

[12]Shin, D.-H., and Kim, Y, "Reconfigurable Flight Control System Design Using Adaptive Neural Newtworks," *IEEE Transactions on Control System Technology*, Vol. 12, No. 1, 2004, pp. 87–100.

[13]Sundararajan, N., Sarachandran, P., and Wang, Z., "Robust Neuro-H-∞ Controller Design for Aircraft Auto-Landing," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 40, No. 1, 2004, pp. 158–168.

[14]Calise, A. J., "Neural Networks in Nonlinear Aircraft Flight Control," *IEEE Aerospace and Electronics System Magazine*, Vol. 11, No. 7, 1996, pp. 5–10.

[15]Sadhukhan, D., and Feteih, S., "F8 Neuro-Controller Based on Dynamic Inversion," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 1, 1996, pp. 150–156.

[16]Li, Y., Sundararajan, N., and Saratchandran, P., "Neuro-Controller Design for Nonlinear Flighter Aircraft Maneuver Using Fully Tuned RBF Networks," *Automatica*, Vol. 37, No. 8, 2001, pp. 1293–1301.

[17]Kim, B., and Calise, A. J., "Nonlinear Flight Control Using Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 26–33.

[18]Li, Y., Sundararajan, N., and Saratchandran, P., "Stable Neuro-Flight Controller Using Fully Turned Radial Basis Function Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, 2001, pp. 665–674.

[19]Ha, C. M., Weis, Y. P., and Bessolo, J. A., "Reconfigurable Aircraft Flight Control System Via Neural Network," AIAA Paper 92-1075, Feb. 1992.

[20]Kim, K.-S., Lee, K.-J., and Kim, Y., "Reconfigurable Flight Control System Design Using Direct Adaptive Method," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, 2003, pp. 543–550.

[21]Ng, G. W., "Neural Networks Control Strategies," *Application of Neural Networks to Adaptive Control of Nonlinear Systems*, Research Studies Press, Hertfordshire, England, U.K., 1997, pp. 43–77.

[22]Thambi, G. K., Principe, J. C., Motter, A. M., Cho, J., and Lan, J., "Multiple Model Based Flight Control Design," *45th Midewest Symposium on Circuits and Systems*, Vol. 3, Inst. of Electrical and Electronics Engineers, 2002, pp. III-133–III-136.

[23]Leontartis, I. J., and Billings, S., "Input Parametric Models for Non-Linear Systems, Part I: Deterministic Non-Linear Systems," *International Journal of Control*, Vol. 41, No. 2, 1985, pp. 303–328.

[24]Shim, K. H., and Sawan, M. E., "Near Optimal State Feedback Design for Singularly Perturbed System by Unified Approach Using Delta Operators with an Aircraft Implementation," *IEEE Conference on Decision and Control*, Vol. 3, Dec. 1998, pp. 3177–3182.

[25]Zhao, Q., and Jiang, J., "Reliable State Feedback Control System Design Against Actuator Failures," *Automatica*, Vol. 34, No. 10, 1998, pp. 1267–1272.

[26]Nguyen, L. T., Ogburn, M. E., Gilbert, W. P., Kibler, K. S., Brown, P. W., and Deal, P. L., "Simulator Study of Stall/Post Stall Characteristics of a Fighter Airplane with Relaxed Longitudinal Static Stability," NASA TP 1538, Dec. 1979.

[27]Wang, J. L., and Sundararajan, N., "Extended Nonlinear Flight Conroller Design for Aircraft," *Automatica*, Vol. 32, No. 8, 1996, pp. 1187–1193.

[28]Barron, A. R., "Universal Approximation Bounds for Superpositions of a Sigmoidal Function," *IEEE Transactions on Information Theory*, Vol. 39, No. 3, 1993, pp. 930–945.

[29]Hornik, K., Stinchcombe, M., and White, H., "Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks," *Neural Networks*, Vol. 3, No. 5, 1990, pp. 551–560.

[30]Hornik, K., Stinchcombe, M., and White, H., "Multilayer Feedforward Networks Are Universal Approximators," *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359–366.

[31]Cybenko, G., "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems*, Vol. 2, No. 4, 1989, pp. 303–314.

[32]Suresh, S., Omkar, S. N., Mani, V., and GuruPrakash, T. N., "Lift Coefficient Prediction at High Angle of Attack Using Recurrent Neural Network," *Aerospace Science and Technology*, Vol. 7, No. 8, 2003, pp. 595–602.

[33]Chan, D. S., and Jain, R. C., "A Robust Back Propagation Learning Algorithm for Function Approximation," *IEEE Transactions on Neural Networks*, Vol. 5, No. 3, 1994, pp. 467–479.

[34]LeCun, Y., "Efficient Learning and Second-Order Methods," *A Tutorial at 7th International Conference on Neural Information Processing*, Vol. 6, 1993.

[35]Franklin, J. A., "Flight Investigation of the Influence of Turbulence on Longitudinal Flying Qualities," *Journal of Aircraft*, Vol. 9, No. 4, 1972, pp. 273–281.

[36]Rynaski, E. G., "Flying Qualities in the Time Domain," AIAA Paper 85-1849, Aug. 1985.

[37]Narendra, K. S., "Neural Networks for Control: Theory and Practice," *Proceedings of the IEEE*, Vol. 84, No. 10, 1996, pp. 1385–1406.