JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 28, No. 1, January–February 2005

Neural-Network Augmentation of Existing Linear Controllers

Manu Sharma[*]
Barron Associates, Inc., Charlottesville, Virginia 22901
and
Anthony J. Calise[†]
Georgia Institute of Technology, Atlanta, Georgia 30332

A method to augment existing linear controllers with a multilayer neural network is presented. The neural network is adapted online to ensure desired closed-loop response in the face of parametric plant uncertainty; no off-line training is required. The benefit of this scheme is that the neural-network output is simply added to the nominal control signal, thereby preserving the existing control architecture. Furthermore, the nominal control signal is only modified if the desired closed-loop response is not met. This method applies to a large class of modern and classical linear controllers. Stability guarantees are provided via Lyapunov-like analysis, and the efficacy of this scheme is illustrated through two numerical examples.

Introduction

NEURAL-NETWORK (NN)-BASED adaptive control methods in various forms have emerged as enabling technologies for practical control when either plant parameters or operating environment, or both, are uncertain. A popular use of NNs for control is in conjunction with dynamic inversion. In this setting, an adaptive NN is used to cancel errors that arise from inexact inversion of the plant dynamics. This approach has been investigated extensively in the area of flight control, particularly for missile and aircraft applications.[1−3]

One drawback to this approach is that unless the existing control architecture is already based on inversion, it must be replaced with one that is. This can be undesirable if the plant already has a controller that performs well within its operational envelope, or if the control variable is nonminimum phase. In such a case it is more expedient to augment the existing controller with an NN than to replace it entirely. In this setting, the NN provides augmentation to ensure that commands are still tracked when the existing controller is deficient.

Kim and Lewis present an approach that augments an optimal controller with an NN for application to robotic manipulators.[4] Their approach takes advantage of the structure of the dynamical equations for robots to treat the problem. A method applicable to more general nonlinear plants is presented by McFarland and Stansberry.[5] However, this work is limited to second-order systems with full relative degree. Rovithakis presents an approach applicable to nonlinear, affine systems, and allows for nonlinear controllers.[6] However, this work only allows uncertainties in the internal systems dynamics, that is, the control influence function $(g(x))$ is required to be known exactly. Additionally, only the regulation problem is addressed.

Although none of the work in the preceding references explicitly accounts for dynamic compensators, an approach that does so is presented in Ref. 7. This work applies to plants that are locally linear and allows for linear dynamic compensators. However, its architecture does not allow for compensation in the feedback path and implicitly assumes that all compensator states go to zero when command tracking is achieved.

The work presented herein extends that of Refs. 4, 5, and 7 by expanding upon the class of existing controllers to which an NN-based adaptive element can be added. The existing control architecture considered includes several classical and modern forms, including multi-input/multi-output (MIMO) dynamic compensators, to provide greater generality. Full-state feedback is required, which is also the case in the references already cited. However, output tracking is treated with no restrictions on relative degree. As in the cited references, uncertainties in the plant model are assumed to satisfy the matching condition. This restriction arises from the fact that the developed approach does not seek to modify the control architecture.

This paper begins by presenting the control architecture to which an NN can be augmented. Subsequently, the details of augmenting the control signal with an NN output and the construction of the tracking error (to be used for NN update) is presented. The NN architecture along with weight update rules is given next. Two numerical examples to illustrate the effectiveness of the design are also included.

Design Methodology

System Definition

This section establishes the architecture under which an NN will be augmented to an existing closed-loop system. To begin, consider the nonlinear dynamical system $\mathcal{G}_p$, given by

$$\dot{x}(t) = f(x(t), u(t)) \tag{1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. Additionally let $\mathcal{G}_{p0}$, given by

$$\dot{x}_0(t) = Ax_0(t) + Bu(t) \tag{2}$$

with $x_0 \in \mathbb{R}^n$, represent the linear approximation of this system used to build the existing linear controller. Here, A and B are constant matrices of appropriate dimensions. It is assumed that the nonlinear system $\mathcal{G}_p$ in Eq. (1) can be written in terms of $\mathcal{G}_{p0}$ as

$$\dot{x}(t) = Ax(t) + B[u(t) + \Delta(x(t), u(t))] \tag{3}$$

where $\Delta(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ is a nonlinear function that represents the modeling error between $\mathcal{G}_p$ and $\mathcal{G}_{p0}$. The presence of modeling error might be caused by plant nonlinearities and errors that arise when the system operates away from its linearization point. The form of Eq. (3) implicitly assumes that the modeling error is matched, that is, $\Delta(\cdot,\cdot)$ lies in the column space of B. Furthermore, the full-state feedback nature of the current architecture does not

Presented as Paper 2001-4163 at the Guidance, Navigation, and Control Conference, Montreal, Canada, 7 August 2001; received 28 July 2003; revision received 8 April 2004; accepted for publication 9 April 2004.

[*] Research Scientist, 1410 Sachem Place, Suite 202; sharma@bainet.com. Member AIAA.

[†] Professor, School of Aerospace Engineering; anthony.calise@ae.gatech.edu. Fellow AIAA.

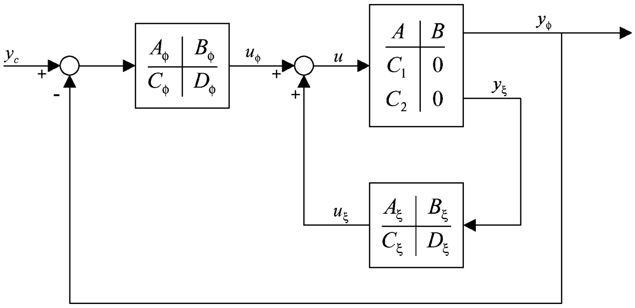

Fig. 1 Existing control architecture.

allow uncertainty in the plant output. However, recent extensions to output feedback that are not limited by the matching assumption can be found in Refs. 8 and 9.

To address a large set of control applications, suppose that the existing controller can be cast into a form consisting of two loops: an inner loop with a dynamic compensator in the feedback path and an outer loop with a dynamic compensator in the feedforward path, as illustrated in Fig. 1.

Let the plant output be composed of two signals

$$y(t) = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x(t) \tag{4}$$

$$= \begin{bmatrix} y_\phi \\ y_\xi \end{bmatrix} \tag{5}$$

where $y_\phi \in \mathbb{R}^k$ and $y_\xi \in \mathbb{R}^l$ represent the measurements provided to the compensators $\mathcal{G}_\phi$ and $\mathcal{G}_\xi$, respectively. Thus, $y \in \mathbb{R}^{k+l}$ represents the total plant output. In addition, let y_ϕ be designated the controlled output, thus the command y_c is required to be of the same dimension as y_ϕ. Further assume that the closed-loop system is stable and achieves the desired response when $\Delta(\cdot,\cdot) = 0$.

Let the inner-loop compensator $\mathcal{G}_\xi$ be given by

$$\dot{\xi}(t) = A_\xi \xi(t) + B_\xi y_\xi(t) \tag{6}$$

$$u_\xi(t) = C_\xi \xi(t) + D_\xi y_\xi(t) \tag{7}$$

where $\xi(t) \in \mathbb{R}^p$; $u_\xi(t) \in \mathbb{R}^m$; and A_ξ, B_ξ, C_ξ, D_ξ are constant matrices of appropriate dimension. Additionally, let the outer-loop compensator $\mathcal{G}_\phi$ be given by

$$\dot{\phi}(t) = A_\phi \phi(t) + B_\phi[y_c(t) - y_\phi(t)] \tag{8}$$

$$u_\phi(t) = C_\phi \phi(t) + D_\phi[y_c(t) - y_\phi(t)] \tag{9}$$

where $\phi(t) \in \mathbb{R}^q$; $u_\phi(t) \in \mathbb{R}^m$; and A_ϕ, B_ϕ, C_ϕ, D_ϕ are also constant matrices of appropriate dimension. Then the nominal control u_0 can be constructed as a sum of the contributions from $\mathcal{G}_\xi$ and $\mathcal{G}_\phi$, and the nominal controller defined as the combination of these two compensators.

$$u_0(t) = u_\xi(t) + u_\phi(t) \tag{10}$$

Now a closed-loop description of the plant and the compensators $\mathcal{G}_\xi$ and $\mathcal{G}_\phi$ can be constructed as

$$\begin{bmatrix} \dot{\xi}(t) \\ \dot{\phi}(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} A_\xi & 0 & B_\xi C_2 \\ 0 & A_\phi & -B_\phi C_1 \\ BC_\xi & BC_\phi & A + B(D_\xi C_2 - D_\phi C_1) \end{bmatrix} \begin{bmatrix} \xi(t) \\ \phi(t) \\ x(t) \end{bmatrix}$$
$$+ \begin{bmatrix} 0 \\ B_\phi \\ BD_\phi \end{bmatrix} y_c(t) + \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix} \Delta(x(t), u(t)) \tag{11}$$

$$y_\phi(t) = [0 \quad 0 \quad C_1] \begin{bmatrix} \xi(t) \\ \phi(t) \\ x(t) \end{bmatrix} \tag{12}$$

which can be written more compactly as

$$\dot{z}(t) = \hat{A} z(t) + \hat{B} y_c(t) + B_\Delta \Delta(x(t), u(t)) \tag{13}$$

$$y_\phi(t) = C z(t) \tag{14}$$

where $z(t) \in \mathbb{R}^{n+p+q}$. It is assumed that $\hat{A}$ is Hurwitz, and that in the absence of modeling error, that is, $\Delta(\cdot,\cdot) = 0$, the nominal controller provides the desired closed-loop dynamics. Furthermore, the realization in Eqs. (13) and (14) is assumed to be minimal; however, if it is not, one that is can be easily constructed.

Construction of Error Dynamics and Control Augmentation

To form an error signal representing the deviation of the true plant response from that desired, a model representing the ideal closed-loop dynamics $\mathcal{G}_m$ is used. Let $\mathcal{G}_m$ be given by

$$\dot{z}_m(t) = \hat{A} z_m(t) + \hat{B} y_c(t) \tag{15}$$

$$y_m(t) = C z_m(t) \tag{16}$$

where $z_m(t) \in \mathbb{R}^{n+p+q}$, $y_m \in \mathbb{R}^k$, and $\hat{A}$, $\hat{B}$, C must be the same matrices as those in Eq. (13). If both the response model and the controller are fed the same reference signal y_c, an error between the closed-loop plant and the response model can be formed. Let this error be defined as

$$e(t) \overset{\triangle}{=} z_m - z \tag{17}$$

From this, the error dynamics can be expressed as

$$\dot{e}(t) = \hat{A} e(t) - B_\Delta \Delta(x(t), u(t)) \tag{18}$$

Thus when the model error $\Delta(\cdot,\cdot) \equiv 0$, any tracking error, possibly because of different initial conditions in the closed-loop plant and response model, decays asymptotically to zero. However if the model error is nonzero, it acts as an undesirable forcing term on the error dynamics.

Consider, then, the following augmentation of the nominal control,

$$u(t) = u_0(t) - u_c(t) \tag{19}$$

where $u_0(t)$ is the output of the nominal controller in Eq. (10) and $u_c = u_{\mathrm{ad}} - \nu_r$ represents a corrective signal that will serve to reduce the effects of the model-error $\Delta(\cdot,\cdot)$. It will also ensure that the desired response is achieved under off-nominal conditions.[5] Applying the augmented-control signal in Eq. (19) to the plant yields the following closed-loop dynamics:

$$\dot{z}(t) = \hat{A} z(t) + \hat{B} y_c(t) + B_\Delta[\Delta(x(t), u(t)) - u_c(t)] \tag{20}$$

which allows the error dynamics in Eq. (18) to be rewritten as

$$\dot{e}(t) = \hat{A} e(t) + B_\Delta[u_c(t) - \Delta(x(t), u(t))] \tag{21}$$

Thus, if the corrective signal u_c can be constructed so that it cancels the effect of the uncertainty $\Delta(\cdot,\cdot)$, stability of the error dynamics is guaranteed, whereby tracking of the response model is also achieved. Augmentation of the nominal controller in this way is illustrated in Fig. 2.

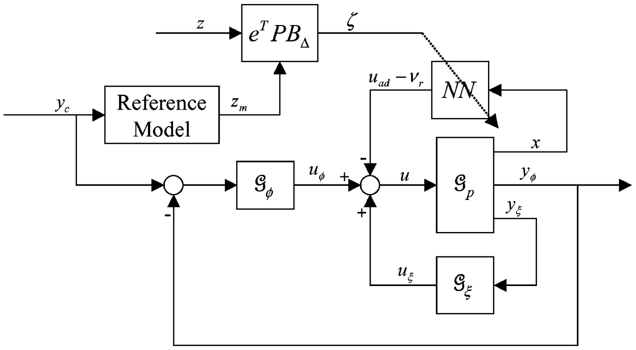

Fig. 2 Neural-network augmented control loop.

Restrictions on the Uncertainty

The requirement for the corrective signal to cancel the uncertainty in Eq. (21) brings up additional restrictions because, by Eq. (19), $\Delta(\cdot,\cdot)$ is a function of u_c itself, that is, $\Delta(x,u)\equiv\Delta(x,u_0-u_c)$. Thus the existence of a solution to the fixed-point problem

$$u_c = \Delta(x, u_0 - u_c) \tag{22}$$

is required. A sufficient condition for the existence of a solution to this problem is that $\Delta(\cdot,\cdot)$ be a contraction mapping,[10] that is,

$$\left\| \frac{\partial\Delta}{\partial u_c} \right\| < 1 \tag{23}$$

(Although any norm on $\mathbb{R}^n$ can be used, the Euclidean norm is used here for convenience.) An expression for $\Delta(\cdot,\cdot)$ can be obtained by differencing Eqs. (1) and (3) and multiplying by $(B^TB)^{-1}B^T$ from the left to get

$$\Delta = (B^TB)^{-1}B^T[f(x,u) - Ax - Bu] \tag{24}$$

Then, the contraction mapping requirement is satisfied when

$$\left\| \frac{\partial\Delta}{\partial u_c} \right\| = \left\| (B^TB)^{-1}B^T[f_u - B] \right\| < 1 \tag{25}$$

where $f_u = \partial f/\partial u$.

To gain further insight, consider the single-input case, and let $b\in\mathbb{R}^n$ represent the control-input vector. This simplifies Eq. (25) to

$$\left\| b^T[f_u - b] \right\| \Big/ \|b\|^2 < 1 \tag{26}$$

and leads to the conditions

$$b^Tf_u\Big/\|b\|^2 > 0 \qquad \text{and} \qquad b^Tf_u\Big/\|b\|^2 < 2 \tag{27}$$

which are essentially conditions on the projection of f_u onto b. However because the control designer cannot choose f_u, they can instead be interpreted as restrictions on b. In a geometric sense, the first of these conditions requires that $\theta\in(-\pi/2,\pi/2)$, where θ is the angle between b and f_u, that is, there should be no unexpected control reversal.

The second condition places a restriction on the estimated control power, and can be simplified to

$$\|b\| > \tfrac{1}{2}\|f_u\|\cos\,\theta \tag{28}$$

In the most restrictive case, when the two vectors are parallel, this condition requires that the linearized plant not underestimate the true control power by more than a factor of two. These conditions are independent of the NN architecture and the type of adaptation law employed.

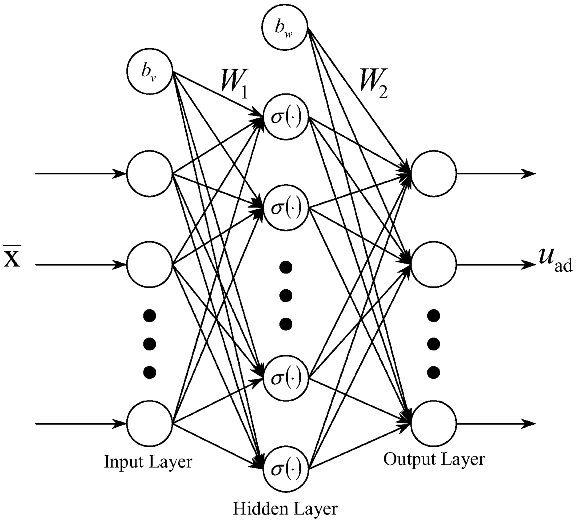

Fig. 3 Multilayer neural-network structure.

Neural Network

The development of the control augmentation in the preceding sections is sufficiently general to allow the use of any adaptive element capable of approximating the model error. However, the remaining development proceeds with a specific NN, the three-layer perceptron shown in Fig. 3, to construct the corrective signal u_{ad} online. The input-output map of the network is given in matrix form by

$$u_{\text{ad}} = W_2^T\boldsymbol{\sigma}\big(W_1^T\bar{x}\big) \tag{29}$$

where $\bar{x}$ represents the NN input, $\boldsymbol{\sigma}(\cdot)$ is a vector of scalar, sigmoidal activation functions $\sigma_0(\cdot)$, which represent the firing of neurons, and W_1 and W_2 are the network interconnection weights. The NN input and the interconnection matrices include elements that account for bias signals that are a part of the network. The structure of these matrices is not defined here, but can be found in Refs. 1 and 11. The scalar activation function $\sigma_0(\cdot)$ is given by

$$\sigma_0(w) = 1/(1 + e^{-\bar{a}w}) \tag{30}$$

where the constant parameter $\bar{a}$ is known as the activation potential.

The Universal Approximation Theorem ensures that, given a sufficient number of hidden-layer neurons, there exist a set of constant weights W_1 and W_2 such that the network approximates the model error to arbitrary accuracy.[12] These weights need not be known and are not required for implementation. They are introduced only to derive a stable adaptive algorithm. During operation, the network weights are computed online and designated $\hat{W}_1$ and $\hat{W}_2$. $\hat{W}_{1,2}$ are not required to converge to their ideal values to ensure command tracking, and in practice they generally do not converge.

To ensure that the NN weights remain bounded, the NN output is augmented by a robustifying term. The construction of this term is given here:

$$\nu_r = K_z(\|\hat{\mathcal{Z}}\|_F + \bar{\mathcal{Z}})(\zeta/\|\zeta\|)\|e\| + K_v\zeta \tag{31}$$

$$\hat{\mathcal{Z}} = \begin{bmatrix} \hat{W}_1 & 0 \\ 0 & \hat{W}_2 \end{bmatrix} \tag{32}$$

$$\zeta = e^TPB_\Delta \tag{33}$$

In Eq. (31), K_z, $K_v > 0$ are constant gains, $\|\hat{\mathcal{Z}}\|_F$ is the Frobenius norm of the current $\hat{\mathcal{Z}}$ matrix, and $\bar{\mathcal{Z}}$ is an upper bound on $\mathcal{Z}$ (the matrix of ideal NN weights). The term ζ can be viewed as a filtered error, and it is what drives the update of the network. The error term e is defined in Eq. (17), and P is the positive-definite solution to the Lyapunov equation $\hat{A}^TP + P\hat{A} + Q = 0$, where $\hat{A}$, B_Δ are from

Eq. 13 and Q is a positive-definite matrix. Commonly, $Q = I$ is used, but Q can be selected by the designer to weight the tracking of some of the states above others. The interconnection weights between different layers represent the states of the network, and the expressions defining their update are given by

$$\dot{\hat{W}}_1 = -\Gamma_1\left[\bar{x}\zeta\hat{W}_2^T\hat{\boldsymbol{\sigma}}' + \lambda_1|\zeta|\hat{W}_1\right]$$

$$\dot{\hat{W}}_2 = -\Gamma_2\left[\zeta\left(\hat{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}'\widehat{W}_1^T\bar{x}\right)\zeta + \lambda_2|\zeta|\hat{W}_2\right] \tag{34}$$

where $\hat{\boldsymbol{\sigma}}$ is a vector of the activation functions in Eq. (30) evaluated at $\widehat{W}_1^T\bar{x}$ and $\hat{\boldsymbol{\sigma}}' = d\hat{\boldsymbol{\sigma}}(w)/dw$ is the Jacobian of $\hat{\boldsymbol{\sigma}}$. The remaining terms are constants: $\Gamma_{1,2} = \Gamma_{1,2}^T > 0$ are the learning rates of the first and second network layers, and $\lambda_{1,2} > 0$ are e-modification gains. The adaptation law in Eq. (34) employs e-modification, which is a commonly used technique to prevent parameter drift in adaptive systems.[13] However, a number of alternatives to e-modification are available, such as σ-modification,[13] incorporation of dead zones,[14] and parameter projection.[15] This now leads to the main result of this paper.

Theorem: If the following assumptions hold—

1) The command signal y_c is bounded and continuously differentiable;

2) The matrix $\hat{A}$ in Eq. (13) is Hurwitz;

3) The condition in Eq. (25) is met;

4) The NN input is chosen as $\bar{x} = [z^T \quad y_c \quad \dot{y}_c \quad \nu_{\mathrm{ad}} \quad \|\hat{\mathcal{Z}}\| \quad 1]$;

5) The bound on the ideal weights is known, $\|\mathcal{Z}^*\|_F \leq \bar{\mathcal{Z}}$;

6) The NN weights are updated as in Eq. (34)—

then all signals in the system comprised of Eqs. (21) and (34) remain bounded.

Proof: See Refs. 3 and 16 for details. □

Remark: Assumption 4 requires a fixed-point iteration because the NN output u_{ad} is also used as an input. In practice, this algebraic loop can be broken by delaying the feedback of u_{ad} to the NN by one step. Numerous applications have confirmed that this yields essentially the same solution as that obtained by iterating the NN map until a fixed-point solution is obtained. This also implies that the NN itself possess a fixed-point solution for u_{ad}, which is guaranteed by use of bounded NN-basis functions. A different approach that avoids the fixed-point assumption using the mean value theorem can be found in Refs. 17 and 18. However, it introduces a difficult-to-verify assumption regarding a bound on the derivative of f_u.

Simple Example

Plant and Controller

As a demonstration of the methodology presented herein, consider the plant and controller depicted in the block diagram in Fig. 4, with the nominal plant and controller given by

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad C_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{35}$$

$$K = \begin{bmatrix} 4 & 8 \end{bmatrix} \tag{36}$$

$$Kp = 7.3, \qquad K_i = 2.2 \tag{37}$$

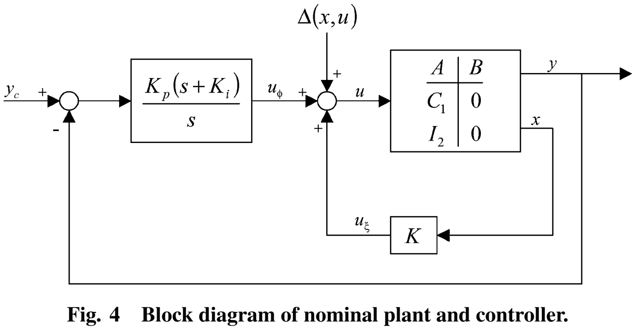

Fig. 4 Block diagram of nominal plant and controller.

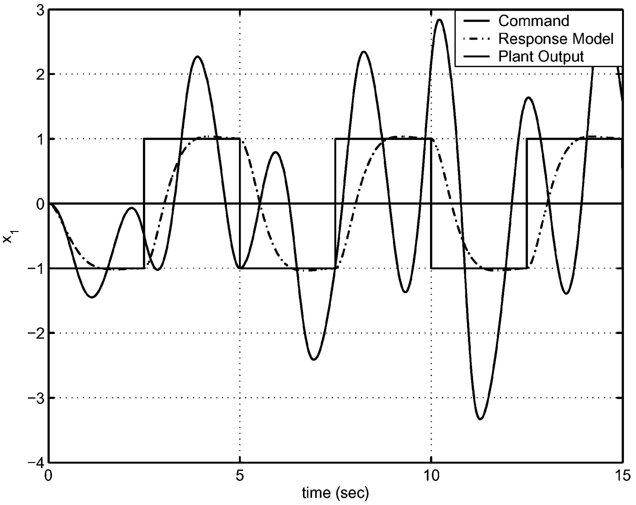

Fig. 5 Command and response with nominal control.

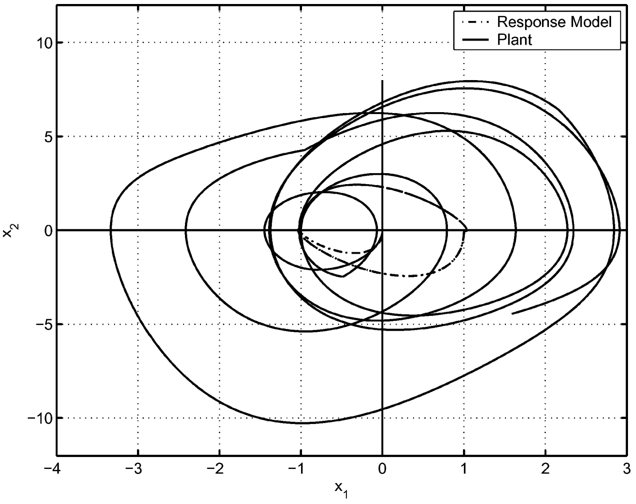

Fig. 6 States with nominal control.

The dynamics of the true plant is given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 + 2x_2 - x_1^2 x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix} u \tag{38}$$

so that

$$\Delta(x, u) = 4x_2 - x_1^2 x_2 - \tfrac{1}{2}u \tag{39}$$

The open-loop plant is unstable and exhibits a limit cycle. The nominal controller is unable to stabilize the true plant, though the states remain bounded. Figure 5 shows the command and the response of both the true plant and the response model. A phase portrait of the plant and corresponding reference model states is given in Fig. 6.

Reference Model and Neural Network

The reference model for the desired response is identical to the nominal closed-loop dynamics and is given by

$$\dot{z}_m(t) = \hat{A}z_m(t) + \hat{B}y_c(t)$$

$$\hat{A} = \begin{bmatrix} 0 & -2.2 & 0 \\ 0 & 0 & 1 \\ 7.3 & -16.3 & -6 \end{bmatrix}, \qquad \hat{B} = \begin{bmatrix} 2.2 \\ 0 \\ 7.3 \end{bmatrix}, \qquad z_m = \begin{bmatrix} \phi \\ x_1 \\ x_2 \end{bmatrix} \tag{40}$$

Table 1 Simple example neural-network parameters

Parameter	Value
Γ_1	100
Γ_2	1000
λ	0.01
K_z	10
K_v	1
$\bar{Z}$	100
$\bar{a}$	1

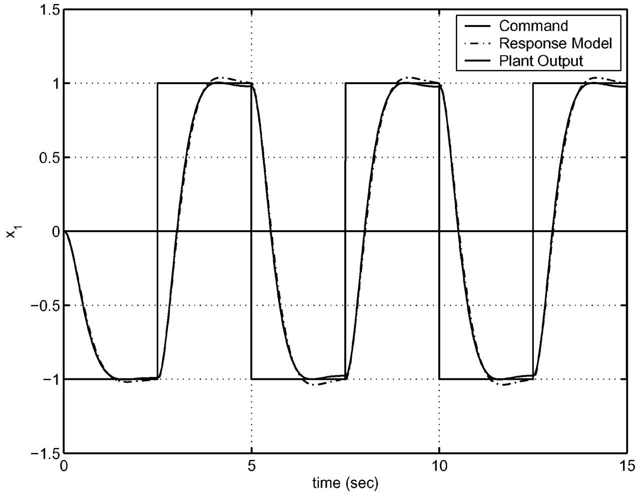

Fig. 7 Command and response with augmented control.

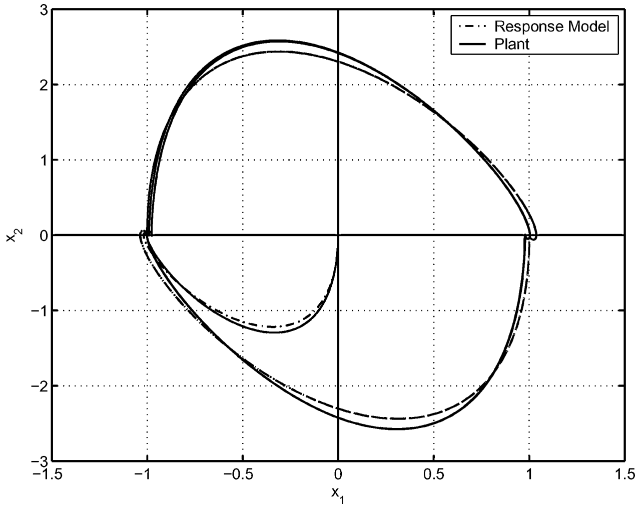

Fig. 8 States with augmented control.

where x_1, x_2, $\phi \in \mathbb{R}$ are the plant and integrator states respectively. The output of the NN is added to the control signal as shown in Fig. 2. The inputs to the network are the plant and controller states, and the learning rates and associated parameters are given in Table 1.

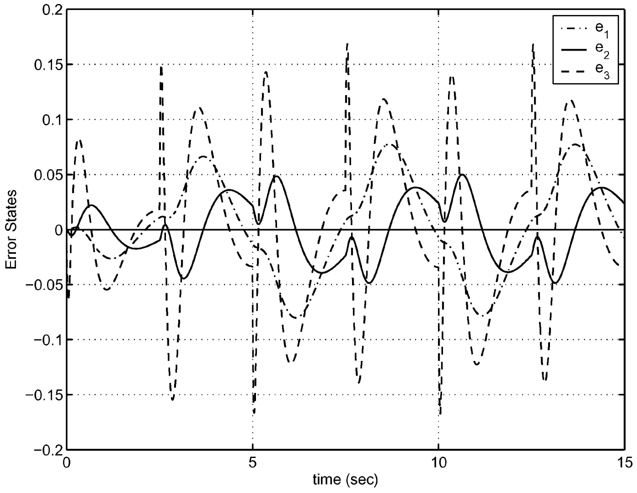

Fig. 9 Tracking errors.

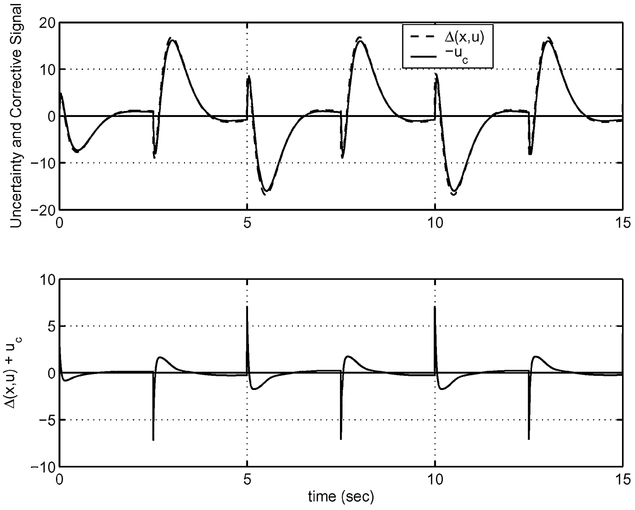

Fig. 10 Plant uncertainty.

Numerical Results

Results from a numerical simulation of the true plant with augmented control are given in Figs. 7–12. These results demonstrate that the NN aids the existing controller in stabilizing the plant and providing the desired response. Figure 7 shows the command and the response of both the plant and the response model, and Fig. 8 shows a phase portrait of the plant and corresponding reference model states. The tracking errors are given in Fig. 9, where e_1 is the integrator-state error, and e_2 and e_3 are the position and velocity tracking errors respectively. This figure shows that the position error is kept to 2.5% of the peak value of y_m. The magnitude of these errors can be further reduced (or relaxed) by adjusting the NN learning rates appropriately. A time history of the uncertainty $\Delta(\cdot,\cdot)$, is shown in Fig. 10. Figure 11 shows the nominal and adaptive control signals along with the augmented control signal. This figure shows that the NN output is smooth and illustrates that the magnitude of the augmented control is on the same order as the nominal control. Finally, Fig. 12 gives select NN weight histories to show that they remain bounded.

Flight-Control Example

For further illustration of the application of the control method developed herein, a flight-control example is also included. Let the short-period dynamics of an aircraft be given by

$$\begin{bmatrix}\dot{\alpha}\\ \dot{q}\end{bmatrix} = \begin{bmatrix} Z_\alpha/U_0 & 1\\ M_\alpha & M_q\end{bmatrix}\begin{bmatrix}\alpha\\ q\end{bmatrix} + \begin{bmatrix} Z_\delta/U_0\\ M_\delta\end{bmatrix}\delta \tag{41}$$

where α, q, and δ denote angle of attack, pitch rate, and elevator deflection, respectively. The dynamics in Eq. (41) can also be represented in transfer function form as

$$G_{\alpha,\delta} = \frac{(Z_\delta/U_0)(s + M_\delta U_0/Z_\delta - M_q)}{s^2 - (Z_\alpha/U_0 + M_q)s + Z_\alpha M_q/U_0 - M_\alpha} = \frac{k_\alpha(s + a_\alpha)}{D(s)} \tag{42}$$

$$G_{q,\delta} = \frac{M_\delta[s + (M_\alpha Z_\delta - M_\delta Z_\alpha)/U_0 M_\delta]}{s^2 - (Z_\alpha/U_0 + M_q)s + Z_\alpha M_q/U_0 - M_\alpha} = \frac{k_q(s + a_q)}{D(s)} \tag{43}$$

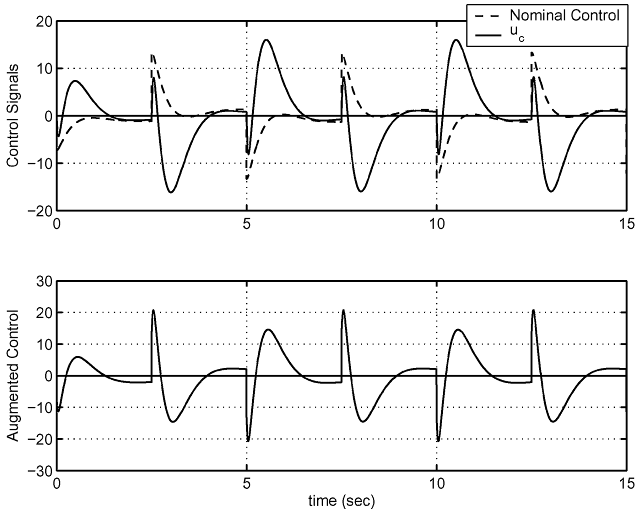

Fig. 11 Control signals.

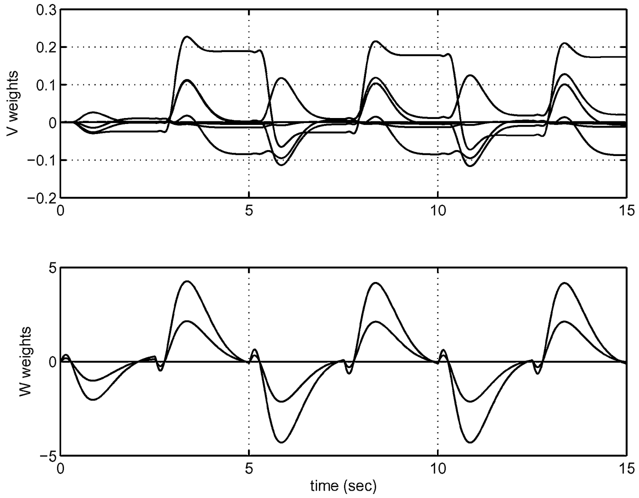

Fig. 12 Select neural-network weights.

Because of the presence of U_0 and M_δ in the numerator of Eq. (42), the zero of the α transfer function is often much faster than its poles. This holds true for a large class of aircraft whose control effectors are primarily moment-producing devices, for example, tail-controlled aircraft. Neglecting this zero simplifies Eq. (42) to $G_{\alpha,\delta} = k_\alpha a_\alpha / D(s)$ and allows a transfer function from q to α to be constructed as

$$G_{\alpha,q} = \frac{k_\alpha a_\alpha}{k_q(s + a_q)} \tag{44}$$

Baseline Autopilot

The baseline autopilot used in this example is an α command system composed of two loops: an inverting inner loop and a classical outer loop. The inner loop has an explicit model following form and inverts the pitch-rate transfer function $G_{q,\delta}$ in Eq. (43). The details of the inverting design are not included here for brevity, but are given in Refs. 1–3. When the aerodynamic stability and control derivatives of the aircraft are perfectly known, the inverting controller yields closed-loop q dynamics of the form $q(s)/q_c(s) = 1/(\tau_q s + 1)$, where τ_q is a design parameter to specify the desired closed-loop bandwidth.

The outer loop is designed through an analysis of the block diagram in Fig. 13. Assuming perfect plant knowledge, that is, $\Delta(\cdot,\cdot) = 0$, the loop gain can be written as

$$G(s) = \frac{K_p k_\alpha a_\alpha (s + K_i)}{k_q s(\tau_q s + 1)(s + a_q)} \tag{45}$$

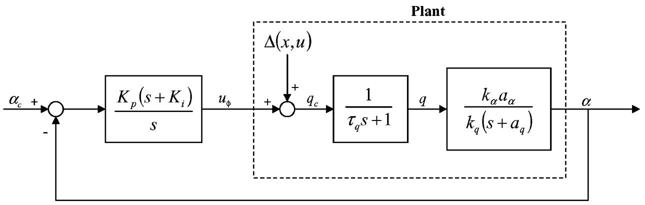

Fig. 13 Example 2: control architecture.

Then selecting $K_i = a_q$ creates a pole-zero cancellation and simplifies the expression for the loop gain to

$$G(s) = \frac{K_p k_\alpha a_\alpha}{k_q s(\tau_q s + 1)} \tag{46}$$

from which the closed-loop transfer function can be constructed as

$$H(s) = \frac{K_p k_\alpha a_\alpha / k_q \tau_q}{s^2 + (1/\tau_q)s + (K_p k_\alpha a_\alpha / k_q \tau_q)} \tag{47}$$

Because this transfer function is in the standard second-order form, the values for K_p, K_i, and τ_q needed to provide the desired closed-loop response are

$$K_i = a_q, \qquad K_p = k_q \omega_n / 2\zeta k_\alpha a_\alpha, \qquad \tau_q = 1/2\zeta\omega_n \tag{48}$$

This result hinges on accurate knowledge of the plant dynamics. If the dynamics are uncertain, as is often the case, the closed-loop dynamics will not have the expected second-order response—unless an NN is used to accommodate the plant uncertainties.

Neural-Network Augmentation

In this example, NN augmentation is performed by treating the inner-loop controller and the transfer function $G_{\alpha,q}$ from Eq. (44) as the plant and the proportional–integral (PI) gain as the controller as shown in Fig. 13. This representation allows the terms required for the augmentation described in Eq. (20) to be specified as

$$\mathcal{G}_{p0} \sim \left[\begin{array}{cc|c} -a_q & \dfrac{k_\alpha a_\alpha}{k_q} & 0 \\ 0 & \dfrac{-1}{\tau_q} & \dfrac{1}{\tau_q} \\ \hline 1 & 0 & 0 \end{array}\right] \tag{49}$$

$$\mathcal{G}_\phi \sim \left[\begin{array}{c|c} 0 & 1 \\ \hline K_p K_i & K_p \end{array}\right] \tag{50}$$

with $\mathcal{G}_\xi = 0$. Finally, because the autopilot is required to track α commands, but the plant contains some unmatched uncertainties, the NN training error is weighted towards α error and integral of α error to improve tracking performance. This weighting is done by appropriately selecting the matrix Q. The values used for the design parameters are summarized in Table 2.

Simulation Results

The evaluation of both the baseline and augmented autopilots was performed using a nonlinear, three-degree-of-freedom (pitch-plane) simulation of a flying-wing aircraft. The simulation uses table-look-up aerodynamics that include compressibility effects. Second-order actuator models with bandwidth, position, and rate limits representative of those used in unmanned combat air vehicle (UCAV) class aircraft were also included.

Table 2 Flight-control example neural-network parameters

Parameter	Value
Γ_1	100
Γ_2	100
λ	0.001
K_z	1
K_v	2
$\bar{Z}$	10
$\bar{a}$	0.001
Q	diag{10, 1, 0.1}

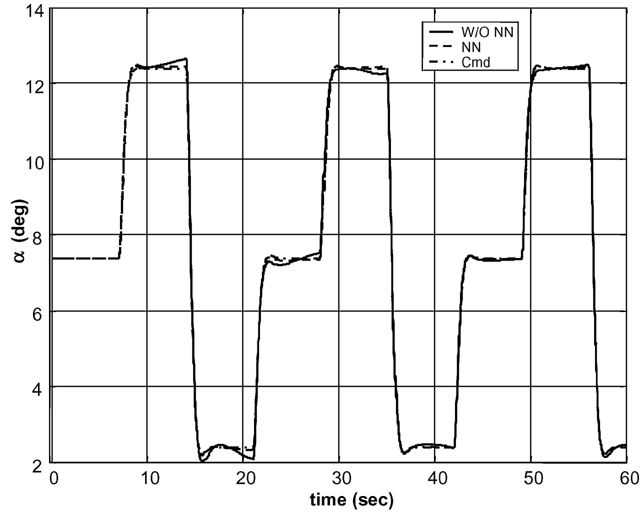

Fig. 14 Example 2: command tracking (*V* = 300 ft/s).

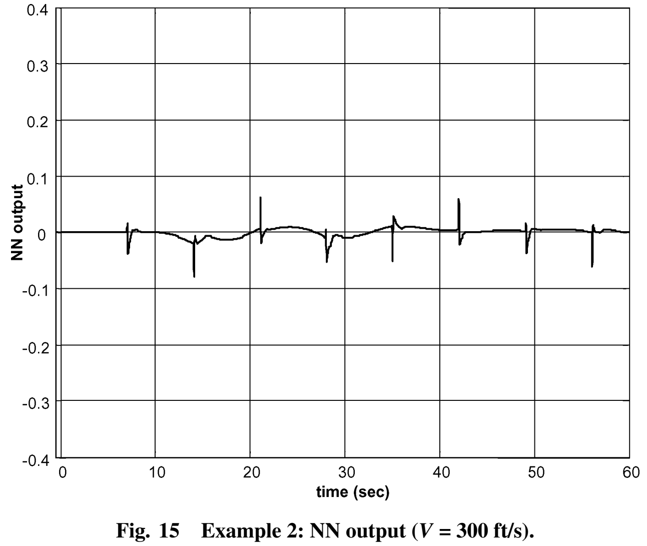

Fig. 15 Example 2: NN output (*V* = 300 ft/s).

For the purpose of illustration, it is assumed that the values for the aerodynamic parameters in Eq. (41) are only known for a single flight condition ($V = 300$ ft/s, alt $= 5000$ ft). Then the gains K_p and K_i are designed for this flight condition only, and no gain scheduling is performed. Figure 14 gives α tracking from both the baseline and augmented autopilots at the nominal flight condition. The significant features of these results are that the baseline autopilot provides reasonable closed-loop response at the design condition, and the response from the augmented controller is nearly identical to that from the baseline autopilot. Hence the NN is not very active at the design condition. The latter point is also illustrated in Fig. 15, which gives the NN output ν_c. No dead zones are used to achieve these results.

Next, the aircraft is flown at a speed of 600 ft/s (same altitude). Figure 16 gives the response of the baseline autopilot, alone,

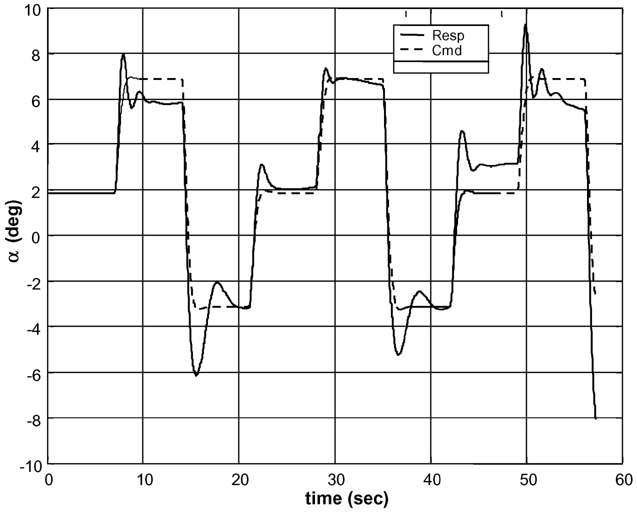

Fig. 16 Example 2: command tracking without NN (*V* = 600 ft/s).

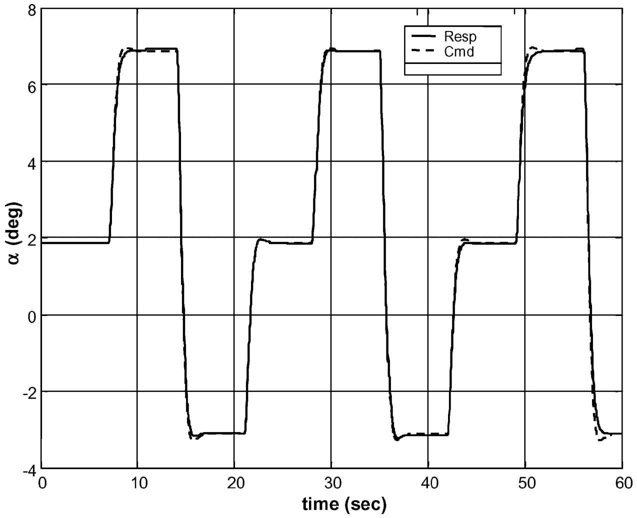

Fig. 17 Example 2: command tracking with NN (*V* = 600 ft/s).

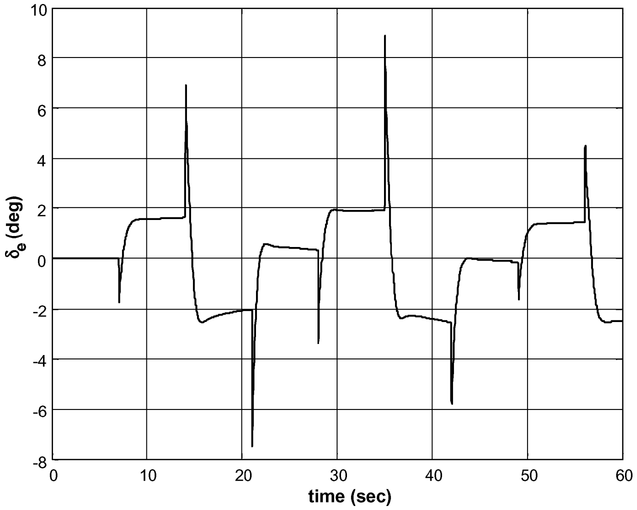

Fig. 18 Example 2: elevator deflection (*V* = 600 ft/s).

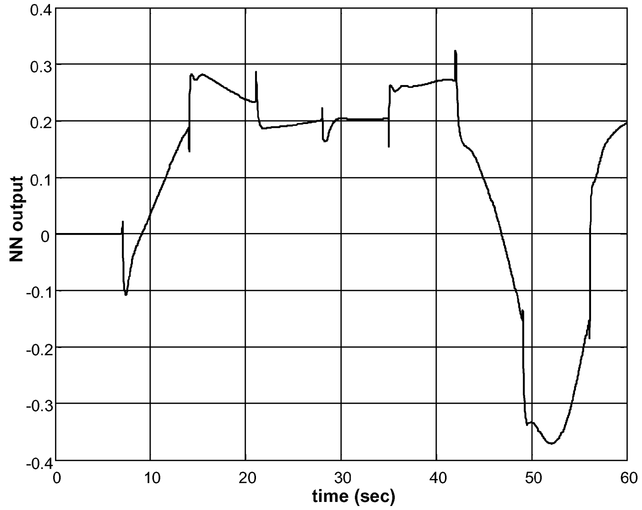

Fig. 19 Example 2: NN output (V = 600 ft/s).

at this condition and shows that it does not provide satisfactory command tracking. In fact, the vehicle departs around $t = 57$ s. Note that magnitude of the command doublets is the same as the previous test condition, but the aircraft is trimmed at a lower α as a result of increased speed. Figure 17 shows the α tracking response from the augmented autopilot and shows that it is able to maintain stability and provide good closed-loop response. Figures 18 and 19 give the control signal and NN output for this case and show that these signals are reasonable in magnitude and well behaved.

Conclusions

An approach to augment existing linear controllers with a neural-network-based adaptive element is presented. The proposed scheme provides the benefit of adaptation to cancel errors in the plant model and ensure command tracking with minimal changes to the existing control architecture. The plant must be affine in the control, and plant uncertainties are required to satisfy a matching condition. Furthermore, full-state information is required. The viability of the method is illustrated through two examples.

Acknowledgments

Research supported by U.S. Air Force Wright Laboratory Air Force Base, Eglin, and by the Air Force Office of Scientific Research.

References

[1] McFarland, M. B., and Calise, A. J., "Multilyer Neural Networks and Adaptive Nonlinear Control of Agile Anti-Air Missiles," AIAA Paper 97-3540, Aug. 1997.

[2] Rysdyk, R. T., and Calise, A. J., "Adaptive Model Inversion Flight Control for Tilt-Rotor Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 3, 1999, pp. 402–407.

[3] Calise, A. J., Lee, S., and Sharma, M., "Development of a Reconfigurable Flight Control Law for the X-36 Tailless Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 896–902.

[4] Kim, Y. H., and Lewis, F., *High-Level Feedback Control with Neural Networks*, World Scientific Publishing, London, 1998, pp. 150–170.

[5] McFarland, M. B., and Stansbery, D., "Adaptive Nonlinear Autopilot for Anti-Air Missiles," *Proceedings of the AIAA Missile Sciences Conference*, AIAA, Reston, VA, 1998; also available from Defense Technical Information Center, AD-A356503.

[6] Rovithakis, G. A., "Robustifying Nonlinear Systems Using High-Order Neural Network Controllers," *IEEE Transactions on Automatic Control*, Vol. 44, No. 1, 1999, pp. 102–108.

[7] Campa, G., Sharma, M., Calise, A. J., and Innocenti, M., "Neural Network Augmentation of Linear Controllers with Application to Underwater Vehicles," *Proceedings of the 2000 American Control Conference*, Vol. 1, No. 6, American Automatic Control Council, Dayton, OH, 2000, pp. 75–79.

[8] Calise, A. J., Yang, B.-J., and Craig, J. I., "An Augmenting Adaptive Approach to Control of Flexible Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, 2004, pp. 387–396.

[9] Yang, B.-J., Hovakimyan, N., and Calise, A. J., "Output Feedback Control of an Uncertain System Using an Adaptive Observer," *Proceedings of the Conference on Decision and Control*, Vol. 2, IEEE Press, New York, 2003, pp. 1705–1710.

[10] Khalil, H. K., *Nonlinear Systems*, 2nd ed., Prentice–Hall, Upper Saddle River, NJ, 1996, pp. 64–67.

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

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

[13] Narendra, K. S., and Annaswaymy, A. M., "A New Adaptive Law for Roubust Adaptation Without Persistent Excitation," *IEEE Transactions on Automatic Control*, Vol. 32, No. 2, 1987, pp. 134–145.

[14] Naik, S. M., Kumar, P. R., and Ydstie, B. E., "Robust Continuous-Time Adaptive Control by Parameter Projection," *IEEE Transactions on Automatic Control*, Vol. 37, No. 2, 1992, pp. 182–197.

[15] Peterson, B. B., and Narendra, K. S., "Bounded Error Adaptive Control," *IEEE Transactions on Automatic Control*, Vol. 27, No. 6, 1982, pp. 1161–1168.

[16] Sharma, M., "A Neuro-Adaptive Autopilot Design for Guided Munitions," Ph.D. Dissertation, Georgia Inst. of Technology, Atlanta, May 2001.

[17] Ge, S. S., Hang, C., Lee, T., and Zhang, T., *Stable Adaptive Neural Network Control*, Kluwer Academic, Boston, 2002, pp. 139–146.

[18] Kim, N., Calise, A., and Hovakimyan, N., "Several Extension in Methods for Adaptive Output Feedback Control," *Proceedings of the American Control Conference*, 2004 (in press).