JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 26, No. 1, January–February 2003

Neural Network Approach for Nonlinear Aeroelastic Analysis

Ovidiu Voitcu* and Yau Shu Wong†
University of Alberta, Edmonton, Alberta T6G 2G1, Canada

A new approach is proposed, based on the use of artificial neural networks, for predicting nonlinear aeroelastic oscillations. Our objective is to reconstruct the asymptotic state of the nonlinear behavior of an aeroelastic model when only a limited segment of the transient data is known. An original neural network architecture is proposed and is used to predict the nonlinear motions of an aeroelastic system modeling a self-excited two-degree-of-freedom airfoil oscillating in pitch and plunge. When a segment of the transient state of the given signal is used for training, the neural network is capable of correctly predicting the corresponding limit-cycle oscillations, damped oscillations, or unstable divergent oscillations. The network training set consists of numerically generated data or data obtained from a wind-tunnel experiment. A neural network used in conjunction with a wavelet decomposition is presented, which proves to be capable of extracting the values of the damping coefficients and frequencies from the predicted signal. Neural networks, thus, prove to be useful tools in nonlinear aeroelastic analysis.

Nomenclature

$E(\boldsymbol{w})$	=	mean-squared one-step prediction error (performance index)
$\tilde{e}_w(t)$	=	current one-step prediction error, $s(t) - \tilde{y}_w(t)$
$\hat{e}_w(t)$	=	current multistep prediction error, $s(t) - \hat{y}_w(t)$
f^1, f^2	=	transfer function of the first and second layer, respectively
n_1	=	number of neurons in the first layer of a neural network
$s(t)$	=	current observation of the given time series
$s(t_0+1)$, $s(t_1)$	=	first and last training value, respectively
$s(t_1+1)$, $s(t_2)$	=	first and last value to be predicted, respectively
$\boldsymbol{s}_p(t)$	=	vector $[s(t-1), \ldots, s(t-p)]^T$
t	=	current time step
$v_1^1(t)$	=	net input of the first layer
$\boldsymbol{w}$	=	vector of all weights of a neural network
w_j	=	generic weight of a neural network
$w_{k,h}^1$, $w_{1,k}^2$	=	neural network weights of the first and second layer, respectively
$\tilde{y}_w(t)$	=	current one-step prediction, $\Phi_w[s(t-1), \ldots, s(t-p)]$
$\hat{y}_w(t)$	=	current multistep prediction, $\Phi_w[\hat{y}_w(t-1), \ldots, \hat{y}_w(t-p)]$
$\hat{\boldsymbol{y}}(\boldsymbol{w})$	=	vector $[\hat{y}_w(t_1+1), \ldots, \hat{y}_w(t_2)]^T$
$y_1^1(t)$, $y_1^2(t)$	=	current output of the first and second layer, respectively
z^{-1}	=	one-step delay operator
$\|\cdot\|$	=	Euclidean norm

Introduction

IN recent years, considerable research has been focused on nonlinear aeroelastic analysis,[1] a research field with great impact on safe aircraft design. An important problem when designing a fighter aircraft is the prediction of nonlinear oscillations in the aeroelastic response. Of special interest are the oscillations with increasing amplitude, which eventually lead to structural failure, and the limit-cycle oscillations (LCO), which may cause structural fatigue and may impair the pilot's ability to perform the combat mission. A particular area in need of development is the prediction of LCO. Many methods have been proposed for predicting the oscillation frequency, but they fail to predict the onset or the amplitude of the LCO.[2,3]

A common approach in the study of nonlinear aeroelasticity is to construct a mathematical model and to solve the resulting differential system by numerical or analytical methods.[1] However, this approach could be computationally expensive and can only be used when all system parameters are known. Moreover, numerical results may not be reliable in some applications due to the instability of the discretization algorithm or to the failure to capture essential features of the physical process modeled.[4] Moreover, when performing ground vibration or flight tests, only information about the nonlinear response of an aircraft to a given excitation input is available. Hence, it is necessary to develop a prediction technique based only on the known response data.

The objective of our present work is to predict a long-term behavior of a nonlinear response of an aeroelastic model when only a limited segment of the response data is available. The ability to predict the asymptotic response based on the limited information of a given initial dynamics, such as data obtained from experiments, flight test, or numerical simulation, could be important in practical applications. For instance, an engineer who wishes to design a stable system can apply certain control laws if he knows that the system response will be leading to an unstable oscillation. Once a long-term prediction is performed, it would be useful to extract important features of the nonlinear dynamics, such as the damping ratios and modal frequencies.

In this paper, a new approach based on the use of artificial neural networks (ANNs) for predicting nonlinear aeroelastic oscillations is proposed. ANNs are systems of interconnected simple elements (artificial neurons) that perform parallel processing of a set of input data, resulting in a set of output values. The arrangement of neurons and the connection strengths (weights) provide the neural networks with the ability to approximate highly nonlinear relationships between a set of input parameters and a set of output parameters to any desired accuracy.[5,6] Therefore, ANNs are capable of dealing with problems that are difficult to solve by classical mathematical or computational methods.

ANNs have recently been used in applications such as damage detection,[7] aerodynamic design,[8] estimation of air-data parameters,[9] detection of air-frame ice based on the dynamic response of the aircraft to known elevator inputs,[10] estimation of the strain on the vertical tail structure as a function of the lateral and normal acceleration measured on several points on the empennage during various flight

Received 22 April 2002; presented as Paper 2002-1286 at the AIAA/ASME/ASCE/AHS/ASC 43rd Structures, Structural Dynamics, and Materials Conference, Denver, CO, 22–25 April 2002; revision received 12 September 2002; accepted for publication 15 September 2002.

*Ph.D. Student, Department of Mathematical and Statistical Sciences, 632 Central Academic Building. Student Member AIAA.

†Professor, Department of Mathematical and Statistical Sciences, 632 Central Academic Building.

maneuvers,[11] etc. Because of their universal approximation property, neural networks have been widely used for system identification and adaptive control. This makes ANNs particularly attractive in flight control problems.[12] Applications in helicopter stabilization,[13] adaptive control of antiair missiles,[14] and active control of aeroelastic response,[15] including active flutter suppression,[16] have been reported.

The idea of using neural networks to predict the asymptotic behavior of a nonlinear dynamics is quite new. Denegri and Johnson[3] proposed a three-layer ANN to provide the frequency and amplitude of the LCOs of an aircraft wing when given the Mach number of the desired flight condition, a coding of the store carriage configuration and the values of some parameters resulted from linear flutter analysis. In this paper we propose a different approach, which extends earlier work.[17] An original ANN architecture is developed, and the transient data of the aeroelastic response are used to compute the values of the connection weights by applying a nonlinear optimization algorithm. The developed neural network is then used to construct a long-term nonlinear trajectory by a multistep prediction process. Namely, the values predicted by the ANN are used as network inputs to generate subsequent predictions. The proposed ANN architecture has important features that control the stability of the nonlinear optimization process and the error propagation in the multistep prediction process. The frequencies and dampings of the predicted signal can subsequently be extracted by using a different neural network. The proposed method was tested on both simulated data resulted from an aeroelastic model and experimental data, but the ANN does not require any specific information about the aeroelastic system being considered. Moreover, to our knowledge, neural networks have not been reported so far to perform long-term multistep prediction of nonlinear dynamics. However, some references can be found in applying ANN for one-step prediction and for short-term multistep prediction.[18,19]

Nonlinear Dynamic Prediction

Assuming we know a set of consecutive terms of a time series $s(1),\dots,s(t_1)$, we are interested in predicting the future values $s(t_1+1),\dots,s(t_2)$, for some $t_2>t_1$. The usual approach is to search for a nonlinear recurrence relation that is satisfied in an approximative manner by a segment of the known data points: $s(t)=\Phi[\boldsymbol{s}_p(t)]+\tilde{e}(t)$. The nonlinear mapping Φ and the order p of the recurrence have to be determined such that the preceding relation holds for $t=t_0+1,\dots,t_1$ (for some t_0). The predicted values for the unknown terms can then be generated by one of the formulas $\tilde{y}(t)=\Phi[s(t-1),\dots,s(t-p)]$ (one-step prediction), or $\hat{y}(t)=\Phi[\hat{y}(t-1),\dots,\hat{y}(t-p)]$ [multistep prediction, or recursive prediction, where $\hat{y}(t)\overset{\text{def}}{=}s(t)$ for $t=t_0+1,\dots,t_1$]. Clearly, the first formula can only be used if the correct values $s(t-1),\dots,s(t-p)$ are known before the moment t. In this paper, we are interested in the latter type of prediction, assuming that only a limited transient data set is known and that the subsequent values never become available through measurement or computations. Finding the mapping Φ out of an enormous number of possible choices is an extremely difficult task. We propose to search for Φ among the mappings that define the output of an ANN as a function of its inputs.

ANNs

An ANN is a set of interconnected simple processing elements (artificial neurons), each receiving several numbers (real or complex) as inputs and generating a single number as output. The network is usually composed of successive layers of neurons, such that the outputs of all neurons in each layer are provided as inputs to each neuron in the next layer. The inputs to all neurons in the first layer form the network input, whereas the outputs of all neurons in the last layer (the output layer) form the network output. The interneuron connection strengths (weights) are numbers that can be determined by learning from a set of examples of correct network outputs to given inputs.

We will search for the mapping Φ among the nonlinear mappings Φ_w that express the dependence of the output of an ANN on the p network inputs. The form of Φ_w depends on the chosen network architecture. In the next section, we will illustrate how an ANN architecture can be designed to perform our specific task. The ANN output also depends on the values of the network weights $\boldsymbol{w}$.

After choosing a specific neural network architecture and a certain value for p, acceptable values for the network weights have to be determined such that the network can successfully predict the unknown values $s(t_1+1),\dots,s(t_2)$. A common approach is to choose the vector $\boldsymbol{w}$ that minimizes the mean-squared one-step prediction error (the performance index)

$$E(\boldsymbol{w})=\sum_{t=t_0+1}^{t_1}\frac{[\tilde{e}_w(t)]^2}{(t_1-t_0)}$$

The process of minimization of $E(\boldsymbol{w})$ is called a network training and can be achieved by applying a nonlinear optimization method such as the conjugate gradient algorithm, variations of Newton's method, etc.

In our experiments, the ANN weights are first initialized with small random values. Then, at each iteration, a sweep of the training data set is performed and $\boldsymbol{g}^{\text{new}}=\nabla E(\boldsymbol{w}^{\text{old}})$ is computed using the current weight vector $\boldsymbol{w}^{\text{old}}$. The current search direction is set to be $\boldsymbol{p}^{\text{new}}=-\boldsymbol{g}^{\text{new}}+[\|\boldsymbol{g}^{\text{new}}\|/\|\boldsymbol{g}^{\text{old}}\|]^2\boldsymbol{p}^{\text{old}}$ according to the conjugate gradient method. The weights are then updated according to the formula $\boldsymbol{w}^{\text{new}}=\boldsymbol{w}^{\text{old}}+\alpha_w\boldsymbol{p}^{\text{new}}$, where α_w is the learning rate. The search direction is set to $\boldsymbol{p}^{\text{new}}=-\boldsymbol{g}^{\text{new}}$ at the beginning of training and after every 20 iterations, to accelerate the convergence of the conjugate gradient method.

At each training step, α_w was multiplied by $\zeta_w=1.1$ if $E(\boldsymbol{w}^{\text{new}})<E(\boldsymbol{w}^{\text{old}})$ and divided by ζ_w otherwise. For α_w not to decay to zero, a constant lower bound $\hat{\alpha}_w$ for α_w was set to 0.001. At the beginning of training α_w was set to $\hat{\alpha}_w$. To avoid a heavy oscillatory behavior of the learning rate, an adaptive upper bound α_w^* for α_w was initially set to 10.0. Then, whenever α_w decreased, α_w^* was divided by $\zeta_w^*=1.001$, otherwise it remained unchanged. As a result, both α_w and α_w^* eventually stabilized at the same value α_w^0.

Proposed ANN Architecture and Training Algorithm

Based on the result reported by Cybenko,[5] the mapping Φ just mentioned can be approximated to any desired accuracy by the output $y_1^2(t)=\Phi_{\boldsymbol{w}}[\boldsymbol{s}_p(t)]$ of a two-layer feedforward ANN as shown in Fig. 1:

$$y_1^2(t)=\mathrm{f}^2\left\{w_{1,0}^2+\sum_{k=1}^{n_1}w_{1,k}^2\mathrm{f}^1\left[v_1^1(t)\right]\right\}\tag{1}$$

where $\mathrm{f}^1(v)=\tanh(v)$ and $\mathrm{f}^2(v)=v$, provided that the number n_1 of neurons in the first layer (hidden layer) is sufficiently large. Note that the biases $w_{k,0}^1$ and $w_{1,0}^2$ are particular weights corresponding to constant inputs equal to one. Moreover,

$$v_1^1(t)=w_{k,0}^1+\sum_{h=1}^{p}w_{k,h}^1 s(t-h)\tag{2}$$

denotes the net input of the first layer. Trying to implement this in a straightforward manner leads to a poor prediction performance because the linear transfer function in the output layer usually leads to a strong error propagation in a multistep prediction process. The approach proposed is to build the function Φ_w starting with a simple form and then to cautiously evolve toward more complex forms.

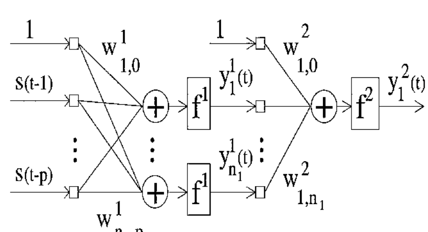

Fig. 1 Two-layer feedforward network.

First define Φ_w as the output of a two-layer feedforward network as illustrated in Fig. 1:

$$\Phi_w[\boldsymbol{s}_p(t)] = y_1^2(t) = w_{1,1}^2 \tanh\left[v_1^1(t)\right] \tag{3}$$

with $n_1 = 1$, $w_{1,0}^2 = 0$, $f^1(v) = \tanh(v)$, and $f^2(v) = v$. The nonlinearity of the mapping Φ_w is given by $\tanh(\cdot)$, and $w_{1,1}^2$ performs a scaling of the network output.

Note that, when the values $s(t_0+1), \ldots, s(t_1)$ are large, $w_{1,1}^2$ will likely become large during the network training process. Consequently, large values for the derivatives

$$\frac{\partial \tilde{y}_w(t)}{\partial w_{1,h}^1} = w_{1,1}^2\left\{1 - \tanh^2\left[v_1^1(t)\right]\right\}s(t-h) \tag{4}$$

result. Subsequently, the gradient of the performance index will also be large, and this may cause destabilization of the training process.

Moreover, it can be shown that for $t \geq t_1 + 1$ the multistep prediction errors can be bounded as follows:

$$|\hat{e}_w(t)| \leq (1 + w_\infty)^{t - t_1 - 1}[\tilde{e}_\infty(t) + w_\infty \hat{e}_1(p)] \tag{5}$$

where $w_\infty \stackrel{\text{def}}{=} |w_{1,1}^2| \max\{|w_{1,h}^1|;\ 1 \leq h \leq p\}$, $\tilde{e}_\infty(t) \stackrel{\text{def}}{=} \max\{|\tilde{e}_w(\tau)|;\ t_1 + 1 \leq \tau \leq t\}$, and

$$\hat{e}_1(p) \stackrel{\text{def}}{=} \sum_{h=1}^{p} |\hat{e}_w(t_1 - h + 1)|$$

It has been noticed experimentally that if $E(\boldsymbol{w})$ is small then the one-step prediction errors $\tilde{e}_w(t)$ for $t \geq t_1 + 1$ and the multistep prediction errors $\hat{e}_w(t_1 - h + 1)$ corresponding to the training points are also small. Hence, to control the magnitude of the multistep prediction errors, it is desirable to have a small value for w_∞. Therefore we propose to modify Φ_w as follows:

$$\Phi_w[\boldsymbol{s}_p(t)] = (c)^2 \tanh\left\{\left[\delta_0 \big/ (c)^2\right] v_1^1(t)\right\} \tag{6}$$

where $\delta_0 = 0.1$ is a constant that sets an upper bound for the derivatives:

$$\frac{\partial \tilde{y}_w(t)}{\partial w_{1,h}^1} = \delta_0\left\{1 - \tanh^2\left[v_1^1(t)\right]\right\}s(t-h) \tag{7}$$

In Eq. (6), c appears as squared to keep the scaling parameter positive throughout network training. The error propagation is now diminished because bound (5) now holds with a smaller $w_\infty = \delta_0 \max\{|w_{1,h}^1|;\ 1 \leq h \leq p\}$. In our experiments, c was initially set to the square root of the amplitude of the training data, which had been scaled to mean value close to zero. It can be verified that c varies slowly during the training process.

In practice, we may have to perform a long-term prediction of two trajectories that have similar transient states but totally different asymptotic states. To provide a better fit of the time series, an additional term is introduced in Eq. (6), obtaining the formula of the output $y_1^2(t) = \Phi_w^t[\boldsymbol{s}_p(t)]$ of a two-layer recurrent network as shown in Fig. 2:

$$y_1^2(t) = \mathrm{f}_c^2\left\{w_{1,0}^2 + w_{1,1}^2 \mathrm{f}_b^{-2}\left[y_1^2(t-1)\right] y_1^1(t)\right\} \tag{8}$$

where $y_1^1(t) = \mathrm{f}^1[v_1^1(t)]$ and the transfer functions are $\mathrm{f}_c^2(v) = (c)^2 \tanh\{\delta_0 v/(c)^2\}$, $\mathrm{f}_b^{-2}(v) = \exp\{-(b)^2(v)^2\}$, and $\mathrm{f}^1(v) = v$. The parameters of the ANN are now contained in the vector

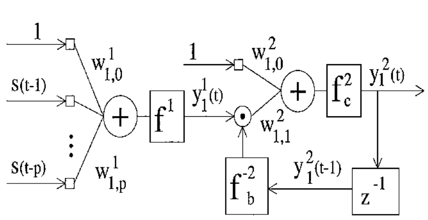

Fig. 2 Proposed neural network architecture.

$\boldsymbol{w} = [b, c, w_{1,0}^1, \ldots, w_{1,p}^1, w_{1,0}^2, w_{1,1}^2]^T$. The network now has a feedback connection, in the sense that the output $y_1^2(t-1) = z^{-1}\{y_1^2(t)\}$ at the previous moment $t-1$ becomes one of the network inputs at the current moment t. In this case, the mapping Φ actually depends on time.

The introduction of the term $\exp\{-(b)^2[y_1^1(t-1)]^2\}$ was suggested by the work of Ozaki,[20] who showed that the amplitude-dependent exponential autoregressive (EXPAR) time series model

$$y(t) = \sum_{h=1}^{p}\left(\phi_h + \pi_h \exp\{-\gamma[y(t-1)]^2\}\right)y(t-h) + \hat{e}(t) \tag{9}$$

is capable of capturing complex nonlinear phenomena such as the amplitude-dependent frequency, the jump phenomena, and the limit cycles. The model was actually extended[21] by replacing the coefficients π_h by Hermite-type polynomials in $y(t-1)$. Here, γ is a positive scaling factor; therefore, we replaced it by a squared term $(b)^2$ in our neural network. The initial value for b was set to 0.1, but any other small nonzero value could be chosen. The EXPAR model has been used for aeroelastic response prediction[22] in the context of statistical time series analysis. In the present paper, we are enriching an ANN architecture with features of the EXPAR time series model, and we will estimate the parameters of the ANN predictor using neural network training methods. The term $\mathrm{f}_b^{-2}[y_1^2(t-1)]$ also reduces the multistep prediction error propagation by reducing the sensitivity of the network output with respect to the inputs. The parameter b eventually vanishes when the ANN is trained to predict a divergent oscillation because the damping effect introduced by the recurrent term tends to impair the prediction accuracy in this case.

The conjugate gradient algorithm was applied to minimize the performance index as discussed in the preceding section. The gradient of the performance index

$$\nabla E(\boldsymbol{w}) = -2 \sum_{t=t_0+1}^{t_1} \tilde{e}_w(t) \nabla \tilde{y}_w(t)$$

was computed using the real time recurrent learning algorithm[23]:

$$\frac{\partial y_1^2(t)}{\partial w_j} = \frac{\delta y_1^2(t)}{\delta w_j} + \frac{\delta y_1^2(t)}{\delta y_1^2(t-1)} \cdot \frac{\partial y_1^2(t-1)}{\partial w_j} \tag{10}$$

where $\delta y_1^2(t)/\delta w_j$ and $\delta y_1^2(t)/\delta y_1^2(t-1)$ are the derivatives of $y_1^2(t)$ regarded as a function of b, c, $w_{1,0}^1, \ldots, w_{1,p}^1$, $w_{1,0}^2$, $w_{1,1}^2$, and $y_1^2(t-1)$ as independent variables. We defined $y_1^2(t_0) \stackrel{\text{def}}{=} s(t_0)$ at the beginning of each sweep of the training data set and $y_1^2(t_1) \stackrel{\text{def}}{=} s(t_1)$ in the testing phase, before predicting $s(t_1+1), \ldots, s(t_2)$. Here, $\nabla y_1^2(t_0)$ was initialized by zero, and at the beginning of each new sweep of the training data points, it was set to the corresponding $\nabla y_1^2(t_1)$ computed from the previous sweep.

Validation

In general, the neural network training is terminated when the value of the performance index reaches a prescribed threshold value θ_E. Once the training is completed, the main issue is to investigate whether $\hat{y}_w(t_1+1), \ldots, \hat{y}_w(t_2)$ provide a good prediction for the unknown values $s(t_1+1), \ldots, s(t_2)$. In time series analysis, it is common to divide the known data into two groups. The first segment is used as training set for the neural network, whereas the second segment is used for validation, in which the distribution of the prediction errors over this set is analyzed. However, there is no rigorous statistical approach to a multistep prediction process, and so it is not clear whether the distribution of the errors over the validation set is relevant. In addition, the success of a short-term prediction may not guarantee a good long-term prediction performance of the neural network. The asymptotic state of the signal may be correctly reconstructed even when the validation set is not correctly predicted. Moreover, there is no clear way to choose correctly the threshold value θ_E. If θ_E is too large, the training is insufficient. However, if it is too small, the overfitting phenomenon occurs, namely, the neural network starts to learn irrelevant particular features of the training data set.

In our present studies, the following approach was used as both a stopping criterion and a validation method. At each training step, after the weight update, the vector $\hat{\mathbf{y}}(\mathbf{w}^{\mathrm{new}})$ was generated, and the distance $\rho^{\mathrm{new}} \stackrel{\mathrm{def}}{=} \|\hat{\mathbf{y}}(\mathbf{w}^{\mathrm{new}}) - \hat{\mathbf{y}}(\mathbf{w}^{\mathrm{old}})\| / \sqrt{(t_2 - t_1)}$ was computed, providing a measure of the variation in the ANN response caused by the current weight update. Because the values of ρ^{new} may oscillate during training, a new distance $\bar{\rho}^{\mathrm{new}}$ was defined by taking the average value of ρ^{new} over the last 100 iterations. When $\bar{\rho}^{\mathrm{new}}$ becomes small and stays small over many training iterations, it means that the sequence generated by the proposed neural network by a multistep prediction remains practically unchanged despite the weight updates. At this point, the network training was stopped and $\hat{\mathbf{y}}(\mathbf{w}^{\mathrm{new}})$ was accepted as the solution of the prediction problem.

Case Studies

The proposed neural network shown in Fig. 2 and the training algorithm have been implemented using C++ programming. In this section, illustrative examples applied to nonlinear aeroelastic systems are presented.

Nonlinear Prediction

To test our proposed nonlinear predictor for practical applications, we first consider the data sets generated by a numerical simulation for an aeroelastic model with structural nonlinearities. The governing equations for a self-excited two-degree-of-freedom airfoil oscillating in pitch and plunge can be expressed as[1]

$$\xi'' + x_\alpha \alpha'' + 2\zeta_\xi(\tilde{\omega}/U^*)\xi' + (\tilde{\omega}/U^*)^2 G(\xi)$$
$$= -[1/(\pi\mu)]C_L(\tau) \tag{11}$$

$$\left(x_\alpha \big/ r_\alpha^2\right)\xi'' + \alpha'' + 2(\zeta_\alpha/U^*)\alpha' + [1/(U^*)^2]M(\alpha)$$
$$= \left[2\big/\left(\pi\mu r_\alpha^2\right)\right]C_M(\tau) \tag{12}$$

where ξ is the plunging displacement, α is the pitch angle about the elastic axis, $G(\xi)$ and $M(\alpha)$ are the nonlinear plunge and pitch stiffness terms, respectively, and $C_L(\tau)$ and $C_M(\tau)$ are the lift and moment coefficients due to aerodynamic forces. The structural nonlinearities are represented by $G(\xi)$ and $M(\alpha)$. In the case of a cubic spring, $M(\alpha)$ is a third-degree polynomial, whereas for a freeplay model, $M(\alpha)$ is a continuous piecewise linear function with two switching points. Similar expressions for $G(\xi)$ can be defined. Note that τ is the nondimensional time, defined as in Ref. 1.

In this paper, the system parameters are chosen such that the resulting aeroelastic response, numerically generated by using a fourth-order Runge–Kutta time integration scheme[4] with respect to τ, corresponds to a LCO (test cases 1–4), damped oscillation (case 5), or divergent oscillation (case 6). The signals considered for testing the proposed neural network predictor represent the time history of the pitch motion for the preceding aeroelastic system in these cases. The time step $\Delta\tau$ used in the Runge–Kutta scheme had the following values: 0.863 (case 1), 0.3 (case 5), and 3.0 (case 6). The signal for case 3 is composed by choosing each second discrete point generated by applying the Runge–Kutta method with $\Delta\tau = 0.5$. The known transient data set was scaled into $(-1, +1)$ in all cases. In test cases 1, 2, and 5, cubic stiffness terms were imposed in both the pitch and plunge degrees of freedom, whereas in test cases 3, 4, and 6, the structural nonlinearity is represented by a freeplay model in the pitch degree of freedom and a linear spring in the plunge degree of freedom. In test cases 2 and 4, the simulated pitch motion corresponding to test cases 1 and 3, respectively, was contaminated with a normal additive noise with zero mean and signal-to-noise ratio equal to 5. In Figs. 3–5 for test cases 1–6, the x axis represents the discrete time and the y axis represents the amplitude of the scaled pitch motion. The two vertical lines mark the time moments t_0 and t_1, indicating the training data used by the ANN. The continuous line represents the simulated pitch motion, and the dashed line represents the neural network prediction. For each case, the values of p (the number of ANN inputs), t_0, t_1, the number of network training iterations I_{tr} performed, and the training time T_{tr} (in seconds), achieved on a Sun workstation, Ultra-10 model, are reported in Table 1.

Table 1 Some ANN parameters and training statistics

	Test case							
Parameter	1	2	3	4	5	6	7	8
p	50	50	50	50	50	200	20	20
t_0	50	50	50	50	50	200	195	195
t_1	273	325	282	323	100	250	300	300
I_{tr}	1317	2000	2220	1000	1100	1000	7000	4000
T_{tr}	69	108	116	54	48	75	264	153

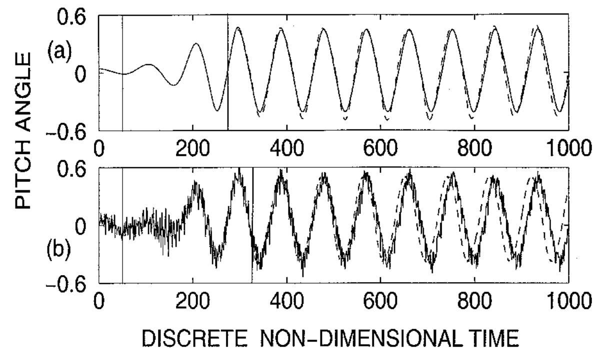

Fig. 3 a) Test case 1: - - -, ANN prediction and ——, pitch motion and b) test case 2: - - -, ANN prediction and ——, noisy pitch motion.

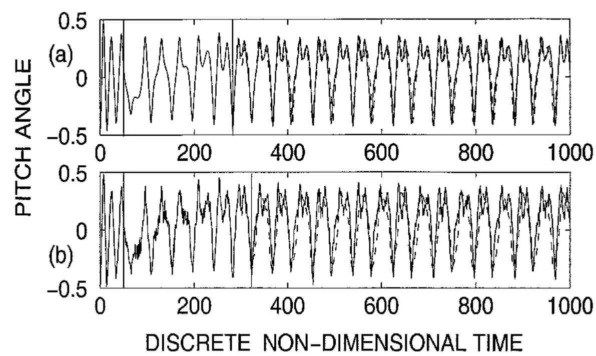

Fig. 4 a) Test case 3: - - -, ANN prediction and ——, pitch motion and b) test case 4: - - -, ANN prediction and ——, noisy pitch motion.

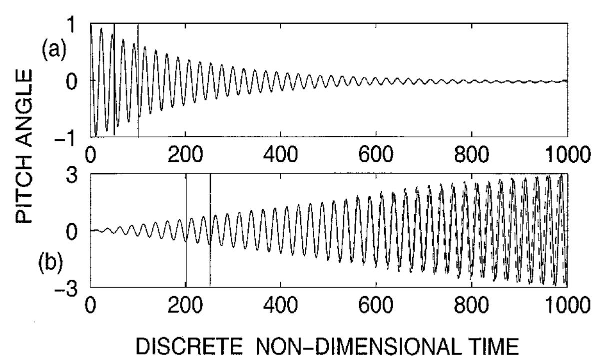

Fig. 5 a) Test case 5: - - -, ANN prediction and ——, pitch motion and b) test case 6: - - -, ANN prediction and ——, pitch motion.

As shown in Figs. 3–5, the neural network predictor reconstructs the LCOs, damped oscillations, or divergent oscillations successfully in each case. It is noticed that the prediction accuracy slightly deteriorates over the long term due to the error propagation. When the signals are contaminated with noise, more training points are needed, but the neural network prediction is not dramatically altered. The phase error becomes noticeable for a long-term prediction (as in test case 2), or some features of the corresponding limit cycle may be lost (as in test case 4), but the amplitude and frequency of the LCOs are almost the same as in the corresponding noise-free cases. One of the attractive features of the proposed technique is

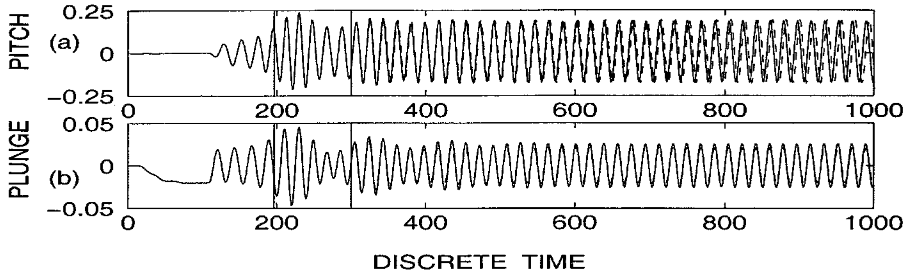

Fig. 6 a) Test case 7: - - -, ANN prediction and ——, pitch motion and b) test case 8: - - -, ANN prediction and ——, plunge motion.

that neural networks are capable of dealing directly with data corrupted by noise. The prediction performance may be improved if a preprocessing phase is introduced to first filter the noisy data set before it is used for neural network training.

Unlike the simulated data for test cases 1–6, the signals in test cases 7 and 8 represent experimental data recorded in a wind-tunnel experiment reported by Ko et al.[24,25] The data are available online at http://aerounix.tamu.edu/aeroel/, and the details of the experiment are described in Refs. 24 and 25. In Fig. 6, results for test cases 7 and 8 are presented, where the continuous line represents the time history of the pitch and plunge motion of an airfoil, respectively. The displayed signals were obtained by consecutively choosing every 10th point of the signals in the file DN04J.DAT (which were sampled every 0.0019 s). The ANN prediction, represented by the dashed line, is in very good agreement with the original signal in both cases. The training statistics and other parameters are reported, similarly to test cases 1–6, in Table 1.

Nonlinear time series models and unscented Kalman filters have also recently been proposed for aeroelastic response predictions. The prediction results using these nonlinear statistical methods applied to the present case studies[22] are comparable to the ANN predictions reported in this section.

Feature Extraction

In addition to performing nonlinear dynamic predictions, neural networks can be trained to extract important features from the nonlinear aeroelastic response. Wong et al.[26] proposed a combined wavelet–neural networks model for extracting damping coefficients and modal frequency values of simulated flutter signals. Suppose we are given a transient segment $\bar{s}(t)$, such as the 50 data points delimited by the vertical line in Fig. 7, sampled with a step of 1/128 from a signal $s(t) = s_1(t) + s_2(t)$, where $s_j(t) = A_j \exp(-a_j t)\sin(f_j t + \phi_j)$, $j = 1, 2$, with $A_1 = A_2 = 1$, $\phi_1 = \phi_2 = 0$, $a_1 = 0.7$, $a_2 = 0.5$, $f_1 = 12\pi$, and $f_2 = 8\pi$. Clearly, it is difficult to extract the corresponding damping coefficients and frequency values directly from such a short transient data set. In Fig. 8, we present a wavelet–neural network model that can efficiently estimate the values of a_1, a_2, f_1, and f_2 when the transient signal $\bar{s}(t)$ is provided. First, a neural network as discussed in the preceding section is trained using $\bar{s}(t)$ and subsequently predicts a long-term nonlinear behavior $s(t)$ (displayed in Fig. 7). Then, $s(t)$ is fed into a wavelet decomposition module, where the two-mode signal is decomposed into two single-mode signals, $s_1(t)$ and $s_2(t)$. The details are described in Ref. 26. A simple way to estimate the damping a and frequency f for a single-mode signal is to train a two-layer feedforward neural network with transfer functions $f^1(v) = 1/(1 + e^{-v})$ and $f^2(v) = v$ to recognize the damping coefficients of damped sine waves $e^{-at}\sin(ft)$ for (a, f) in some fixed bounded set $[a_i, a_f] \times [f_i, f_f]$. Then, given an arbitrary damped sine wave, its frequency f can be easily determined by using the fast Fourier transform. If it is found that $f \in [f_i, f_f]$, then f and a sampling of the sine wave are provided as the neural network input. The corresponding network output will provide an estimate for a.

In our experiments, a set of 100 training pairs and 100 testing pairs (a, f), uniformly distributed in $[0.1, 4.1] \times [3.0, 9.0]$, were simultaneously generated. The damped sine wave $e^{-at}\sin(ft)$, $t \geq 0$, corresponding to each pair was sampled with step $\Delta t = \pi/128$, generating 512 discrete points in each case. The two-layer ANN was trained to output an estimated value for the damping coefficient a

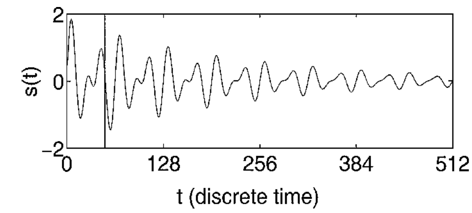

Fig. 7 Simulated two-mode signal.

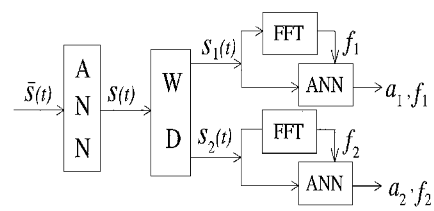

Fig. 8 Wavelet–neural networks model for feature extraction.

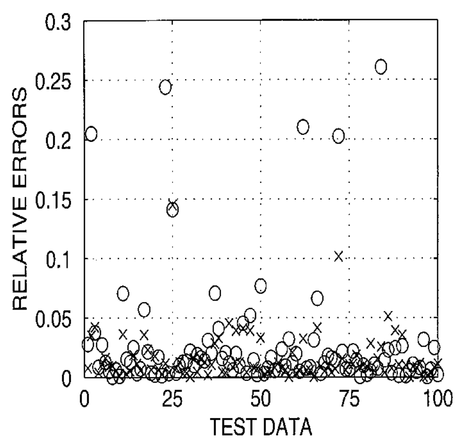

Fig. 9 Relative errors for ANN damping extraction: ○, using 513 inputs and ×, using 9 inputs.

when receiving as inputs the value of f and the 512 sample points of the given single-mode signal. Let y denote the estimate for a, that is, the ANN output value. Thus, the ANN had 513 inputs and one output. In the network implementation, three neurons were used in the first layer. The accuracy of the estimated value for a was reported by the relative output error $|(a - y)/a|$. The network training consisted in applying the conjugate gradient algorithm to minimize the mean-squared relative output error over the 100 training patterns (the performance index E_R) as a function of the 1546 network weights. The training was stopped when $\sqrt{E_R} < 0.015$. In the testing phase, the set of 100 testing pairs (a, f) was used to generate the 513 ANN inputs as described, and the relative output error $|(a - y)/a|$ was computed in each case.

The stopping criterion was reached after 1422 iterations (4 h on a Sun workstation, Ultra-10 model). Even though the network gave responses within 5% accuracy for all training data, 12% of the network responses in the testing phase were associated with large errors in the range of over 5–30% (see Fig. 9). To achieve a better testing performance, the training must be continued until $\sqrt{E_R} \ll 0.015$.

We now propose a more efficient procedure for estimating a, in which the network inputs consist of the value of f and the first eight successive local maximum and minimum values of the corresponding signal. The neural network now has only 9 inputs, 3 neurons in the first layer, and 1 output, and this reduces the number of weights to 34. Providing the value of f to the ANN is essential because the local maximum and minimum values alone $(-1)^k \eta \exp\{[k\pi - \arctan(\eta)]/\eta\}/\sqrt{(1+\eta^2)}$ (for k integer) only determine the quotient $\eta = f/a$.

In this case, the network training stopped after 1440 iterations, and the output errors for the training set were all within 5%. The training process took only 7 min in this case, and more accurate estimates for the damping values were obtained. As shown in Fig. 9, 98% of the estimates were within 5% accuracy and only 2% had error in the range of 10–15%. The result represents a significant improvement in both training time and output accuracy. To obtain all testing errors within 5%, additional training is required.

Clearly, the present wavelet–neural network approach can be extended to a more complex signal consisting of n decaying sine waves.

Discussion

Even though in the present paper we are investigating the use of ANNs for forecasting aeroelastic signals, this approach can be applied to perform a long-term prediction of a trajectory of a general nonlinear dynamic system. Because of the universal approximation property of ANNs, system identification (and, hence, one-step prediction) can be easily performed in that case by a two-layer feedforward ANN with sufficient hidden neurons, provided that sufficient training data points are available. However, given a short transient segment of a trajectory, to provide an accurate long-term multistep prediction we need to do more than a successful system identification; namely, we need to determine a model of the system that is not very sensitive to variations in its inputs. For instance, Volterra kernels have also been successfully used to model nonlinear input–output relationships (see Ref. 27). However, they cannot be used to perform a multistep prediction of the system behavior due to the polynomial input–output relationship, which causes the prediction error to increase rapidly when the actual inputs are replaced by predicted values. The approach we proposed in this paper has taken into consideration the difficulty due to error propagation, and a modified ANN architecture is proposed, with features that make the output less sensitive to the network inputs.

It has been shown that the proposed neural network is capable of predicting LCOs, as well as damped and divergent oscillations. In addition, it has been observed that, in the case of signals that exhibit chaotic behavior, the training of the developed ANN cannot be successfully completed because the distance $\bar{\rho}^{\text{new}}$ defined earlier does not stabilize to a small value. Hence, the proposed method is also capable of predicting the occurrence of chaos. Therefore, it should be able to detect bifurcation phenomena in a given nonlinear system if sufficient transient data sets from system trajectories are given.

It would also be interesting to investigate the possible use of the developed ANN predictor for active flutter suppression. Neural networks have lately been used for that purpose.[15,16] Designing ANN controllers has not been the object of our investigation because we assumed that the actual system outputs beyond a certain time moment t_1 are never known, but one could consider using the constructed neural network predictor for adaptive prediction of the aeroelastic response as follows. If T is the time necessary to retrain the ANN using the last m consecutive system outputs sampled at a rate Δt and to generate a long-term predicted system behavior, then, roughly speaking, at every time moment nT the ANN could be retrained, and an updated long-term prediction could be constructed. Proceeding as described, the occurrence of flutter or LCOs could be predicted, and some damping mechanisms could be initiated to prevent structural failure.

Our damping extraction method is an alternative approach to the one proposed by Johnson et al.,[28] who applied the discrete wavelet transform on both the given signal and a dictionary of so-called singlet functions and performed correlation filtering to determine which singlet function best approximates the frequency and damping characteristics of the given signal at a specific point in time. In our case, a dictionary of ANNs trained to output the damping of a single-mode signal for different frequency intervals could be built and used together with wavelet decomposition to provide real-time estimates of damping coefficients.

Conclusions

Accurate and reliable predictions of nonlinear oscillations in dynamic systems are important in many science and engineering applications. The goal of this paper is to develop ANNs for nonlinear aeroelastic response predictions. An original neural network architecture and a corresponding training algorithm are proposed, and it has been demonstrated that the network is capable of providing accurate predictions for nonlinear aeroelastic responses. A new approach for extracting the damping coefficient from a single-mode signal is also proposed, which greatly reduces the training time of the neural network and provides an improved accuracy. The proposed neural networks appear to be robust in the presence of noise. Various sampling rates were used to generate the network training data, and it has been demonstrated that the ANN is reliable in all cases.

The accuracy of an ANN prediction might be further improved if we could find a way to incorporate in the neural network design specific information about the class of problems that we are trying to solve. Even though more work is required to further improve the performance of the developed methods, we can conclude that ANNs could be useful tools in nonlinear aeroelastic analysis.

Acknowledgments

This work is supported by the Natural Sciences and Engineering Research Council of Canada. The authors thank the reviewers for their valuable suggestions. We are grateful to Thomas W. Strganac for allowing us to use the test data generated by the Aeroelasticity Research Group at the Texas A & M University. We also thank Liping Liu for providing us with the computer code for numerically solving the aeroelastic system (12).

References

[1] Lee, B. H. K., Price, S. J., and Wong, Y. S., "Nonlinear Aeroelastic Analysis of Airfoils: Bifurcation and Chaos," *Progress in Aerospace Sciences*, Vol. 35, No. 3, 1999, pp. 205–334.

[2] Denegri, C. M., Jr., "Limit-Cycle Oscillation Flight-Test Results of a Fighter with External Stores," *Journal of Aircraft*, Vol. 36, No. 5, 2000, pp. 761–769.

[3] Denegri, C. M., Jr., and Johnson, M. R., "Limit-Cycle Oscillation Prediction Using Artificial Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 887–895.

[4] Liu, L., "Mathematical Analysis in Nonlinear Aeroelasticity," Ph.D. Dissertation, Dept. of Mathematical and Statistical Sciences, Univ. of Alberta, Edmonton, AB, Canada, Dec. 2001.

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

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

[7] Yang, S. M., and Lee, G. S., "Structural Damage Identification Using Pole/Zero Dynamics in Neural Networks," *AIAA Journal*, Vol. 39, No. 9, 2001, pp. 1805–1808.

[8] Rai, M. M., and Madavan, N. K., "Aerodynamic Design Using Neural Networks," *AIAA Journal*, Vol. 38, No. 1, 2000, pp. 173–182.

[9] Rohloff, T. J., and Catton, I., "Fault Tolerance and Extrapolation Stability of a Neural Network Air-Data Estimator," *Journal of Aircraft*, Vol. 36, No. 3, 1999, pp. 571–576.

[10] Johnson, M. D., and Rokhsaz, K., "Using Artificial Neural Networks and Self-Organizing Maps for Detection of Airframe Icing," *Journal of Aircraft*, Vol. 38, No. 2, 2001, pp. 224–230.

[11] Kim, D., and Marciniak, M., "Prediction of Vertical Tail Maneuver Loads Using Backpropagation Neural Networks," *Journal of Aircraft*, Vol. 37, No. 3, 2000, pp. 526–530.

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

[13] Enns, R., and Si, J., "Apache Helicopter Stabilization Using Neural Dynamic Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 19–25.

[14] McFarland, M. B., and Calise, A. J., "Neural Networks and Adaptive Nonlinear Control of Agile Antiair Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 3, 2000, pp. 547–553.

[15] Scott, R. C., and Pado, L. E., "Active Control of Wind-Tunnel Model Aeroelastic Response Using Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 6, 2000, pp. 1100–1108.

[16] Bernelli-Zazzera, F., Mantegazza, P., Mazzoni, G., and Rendina, M., "Active Flutter Suppression Using Recurrent Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 6, 2000, pp. 1030–1036.

[17] Wong, Y. S., Popescu, C., and Voitcu, O., "Nonlinear Dynamic Prediction and Feature Extraction," *Proceedings of the 48th Annual Conference of the Canadian Aeronautics and Space Institute*, Canadian Aeronautics and Space Inst., Ottawa, 2001, pp. 97–106.

[18] Wan, E. A., "Time Series Prediction by Using a Connectionist Network with Internal Delay Lines," *Time Series Prediction: Forecasting the Future and Understanding the Past*, edited by A. S. Weigend and N. A. Gershfeld, Addison Wesley Longman, Reading, MA, 1993, pp. 195–217.

[19] Weigend, A. S., Huberman, B. A., and Rumelhart, D. E., "Predicting the Future: A Connectionist Approach," *International Journal of Neural Systems*, Vol. 1, No. 3 , 1990, pp. 193–209.

[20] Ozaki, T., "Non-Linear Time Series Models for Non-Linear Random Vibrations," *Journal of Applied Probability*, Vol. 17, No. 1, 1980, pp. 84–93.

[21] Haggan, V., and Ozaki, T., "Amplitude-Dependent Exponential AR Model Fitting for Non-Linear Random Vibrations," *Time Series*, edited by O. D. Anderson, North-Holland, Amsterdam, 1980, pp. 57–71.

[22] Popescu, C. A., and Wong, Y. S., "A Nonlinear Statistical Approach for Aeroelastic Response Prediction," AIAA Paper 2002-1281, April 1990.

[23] Baltersee, J., and Chambers, J. A., "Nonlinear Adaptive Prediction of Speech with a Pipelined Recurrent Neural Network," *IEEE Transactions on Signal Processing*, Vol. 46, No. 8, 1998, pp. 2207–2216.

[24] Ko, J., Strganac, T. W., and Kurdila, A. J., "Stability and Control of a Structurally Nonlinear Aeroelastic System," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 5, 1998, pp. 718–725.

[25] Ko, J., Strganac, T. W., and Kurdila, A. J., "Adaptive Feedback Linearization for the Control of a Typical Wing Section with Structural Nonlinearity," *Nonlinear Dynamics*, Vol. 18, No. 3, 1999, pp. 289–301.

[26] Wong, Y. S., Lee, B. H. K., and Wong, T. K. S., "Parameter Extraction by Parallel Neural Networks," *Intelligent Data Analysis*, Vol. 5, No. 1, 2001, pp. 59–71.

[27] Kurdila, A. J., Prazenica, R. J., Rediniotis, O., and Strganac, T., "Multiresolution Methods for Reduced-Order Models for Dynamical Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 193–200.

[28] Johnson, J. D., Lu, J., Dhawan, A. P., and Lind, R., "Real-Time Identification of Flutter Boundaries Using the Discrete Wavelet Transform," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, 2002, pp. 334–339.