Journal of Applied Fluid Mechanics, Vol. 6, No. 1, pp. 115-120, 2013.
Available online at www.jafmonline.net, ISSN 1735-3572, EISSN 1735-3645.
DOI: 10.36884/jafm.6.01.19503

An Aeroelastic Metamodel Based on Experimental Data for Flutter Prediction of Swept Rectangular Wings

M. Mohammadi-Amin[1†] and B. Ghadiri[2]

[1]*Aerospace Research Institute, Tehran 14665-834, Iran*
[2]*Tarbiat Modares University, Tehran 14115-385, Iran*

†*Corresponding Author Email: mmohammadi@ari.ac.ir*

(Received January 1, 2011; accepted July 15, 2011)

ABSTRACT

An aeroelastic metamodel was designed and implemented for prediction of flutter speed and frequency of swept rectangular wings based on experimental data and artificial neural networks (ANN). The ANN is a supervised multilayer perceptron that was trained based on an experimental data set involves flutter characteristics of various cantilever rectangular wing models. Some data were not learned to ANN and were maintained as test cases. The activation functions were tangent hyperbolic and linear function in the hidden and output layers respectively. For learning process, the normalized form of the inputs and outputs were given to the ANN. The ANN learned the relation between the inputs and outputs and was trained for predicting output parameters. It is observed that ANN results are in good agreement with experimental data as well as results of an aeroelasticity code developed using an analytical aerodynamic model. So this ANN can be used for quick prediction of flutter characteristics of swept rectangular wings and also for the study of the effects of various parameters on flutter characteristics of swept rectangular cantilevered wings.

Keywords: Artificial neural network, Experimental aeroelasticity, Flutter, Swept wings.

NOMENCLATURE

b_k	bias of a neuron	y_k	activation potential of a neuron
E	mean square error	δ_k^{m+1}	local gradient at neuron k layer m+1
f^l	activation function for layer l	φ_k	activation function of neuron k
w_{kj}	synaptic weight of the neuron	v_k	net input signal

1. INTRODUCTION

Flutter is a dynamic aeroelastic instability that involves the interaction of the elastic, inertia and unsteady aerodynamic forces. In general, solving such aeroelastic problems computationally requires interaction between a structural analysis code and an aerodynamic analysis code where the information about geometry deformation and changing aerodynamic forces and moments is sent back and forth between the two modules. Just an analysis of wing deformation in static flight requires several runs with both structural and aerodynamic codes before reaching convergence to a solution, which is time consuming even on fast processors. On the other hand, artificial neural networks are computational entities that simulate the functioning of the brain and are, in principle, capable of modeling any nonlinear input-output relationship to any degree of accuracy. Artificial neural networks have proven to be able to learn and generalize correlations that would be difficult or almost impossible to explain analytically and where the problems are highly nonlinear in nature. Hence, ANN's have been used in a variety of engineering applications, including aeroelastic problems. In aeroelasticity domain, heretofore, ANN's have been used for modeling nonlinear unsteady aerodynamic effects Marques and Anderson (2001), predicting aeroelastic behavior of aircraft Pesonen and Agarwal (2002), modeling nonlinear aeroelasticity of morphing wings Natarajan (2002), predicting nonlinear oscillations in the aeroelastic response Voitcu and Wong (2003), analyzing the flutter behavior of a simple wing that under went multiple weight variation changes Pitt and Haudrich (2004), and for simulation of unstable aeroelastic responses Wang (2004). For example, in the work of Pesonen and Agarwal, a neural net was designed to predict the shape of a flexible wing in static flight conditions using results from a structural analysis and an aerodynamic analysis

performed with traditional computational tools. Then, another network was also designed and trained to predict airfoil maximum lift at low Reynolds numbers where wind tunnel data was used for the training. Finally, neural net was designed and trained to predict the aeroelastic behavior of a wing without the need to iterate between the structural and aerodynamic solvers.

In the present work, an efficient artificial neural network base on aeroelastic experimental data was developed and implemented for prediction of flutter characteristics of swept rectangular cantilever wings. In the following sections, details of ANN methodology, illustrative diagrams and explanatory discussion about results are presented.

2. ARTIFICIAL NEURAL NETWORK METHODOLOGY

In terms of new methodologies for multi-dimensional estimation, neural networks are a promising technology because of their ability to be trained and used for investigation of systems that involve nonlinear dynamics. Because of this proven capacity, neural networks have been applied in system identification. In this research, a supervised multilayer 14-12-6 perceptron ANN with 12 neurons in the hidden layer is designed and employed for predicting flutter speed and frequency of swept wings. An artificial neural network (ANN) is a massively parallel distributed processor made up of interconnected processing units. The fundamental information processing unit is called as neuron or node, which is the mathematical abstraction of the neuron in the biological science. Figure 1 presents the block diagram of the mathematical model of a neuron. As shown, the signal x_i at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}. According to its activation function, the neuron fires when the weighted sum of the input signals exceeds the externally applied threshold input or bias, denoted by b_k. In mathematical terms, the neuron model can be described as follows,

$$\nu_k = \sum_{j=1}^{N} w_{kj} . x_j + b_k \tag{1}$$

and

$$y_k = \varphi_k(\nu_k) \tag{2}$$

where x_j is the input signal, w_{kj} is the synaptic weight of the neuron, N is the total number of input signals of the neuron, b_k is the bias, υ_k is the net input (activation potential) of the neuron, φ_k is the activation function of neuron k, and y_k is the output signal of the neuron.

The neurons are arranged in layers and are each connected to the neurons in the preceding layer for input and the following layer for output. Data are passed through weighted connections. The network acquires the knowledge from the presented data by adjusting the values of its synaptic weights during learning process. With suitable weights, the ANN can model any function. One popular and successfully applied ANN model is the multi-layer perceptron (MLP). The MLP has a multilayer feed-forward configuration, and it is trained in a supervised (target-oriented) manner with the highly popular error back-propagation learning algorithm. Typically the MLP has input, hidden, and output layers. Figure 2 shows the configuration of the MLP with one hidden layer which is used in our model. Note that a synaptic weight is associated with each connection. Error back-propagation learning consists of two passes through the network on a layer-by-layer basis: a forward pass and a backward pass.

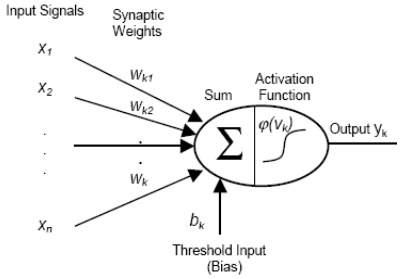

Fig. 1. Neuron Model

In the forward pass, the passive neurons of the input layer (input neurons) merely broadcast input data values (input patterns) over weighted connections to the neurons in the hidden layer (hidden neurons). Then each hidden neuron in the first hidden layer broadcasts the weighted sum of its inputs through its activation function to next hidden layer or the output layer. The neurons in the output layer (output neurons) pass the weighted sum of their inputs through their activation functions to generate the actual results of the network. Hidden neurons have no direct connection to input or output. The hidden layer is introduced to permit the MLP to model nonlinear functions of greater complexity. During the forward pass the synaptic weights of the network are all fixed. In mathematical terms, the net input to neuron k in layer m+1(L+1 is the output layer) is,

$$\nu_k^{m+1} = \sum_{j=1}^{N} w_{kj}^{m+1} . x_j^{m} + b_k^{m+1} \tag{3}$$

The output of neuron k will be,

$$y_k^{m+1} = \varphi_k^{m+1}(\nu_k^{m+1}) \tag{4}$$

In the backward pass, the actual output signals of the network are subtracted from the desired outputs (targets) to produce output error signals, and then those error signals are propagated backward (against the direction of synaptic connections) through the network. During the backward pass the synaptic weights are all adjusted in accordance with a predetermined mathematical criterion to reduce the output error of the network. The error signal at the output neuron j at iteration i (i.e., presentation of the i^{th} training sample) is defined by,

$$e_j^{L+1} = T_j(i) - y_j^{L+1}(i) \tag{5}$$

where $T_j(i)$ is the desired output for output neuron j at iteration i. The usual criteria to guide the learning process is to minimize the cost function E,

$$E = \frac{1}{2}\sum_{i\in S}\sum_{j\in C}[e_j^{L+1}(i)]^2 \qquad (6)$$

where the set C includes all the neurons in the output layer of the network, and the set S includes all the training samples (patterns) contained in the training dataset. Usually the Delta learning rule is used as the error correction rule to update synaptic weights, as shown in figure 3 for synaptic weight w_{kj} associated with the connection between neuron j in layer m and neuron k in layer m+1, the amount of weight adjustment is determined by,

$$\Delta w_{kj} = -\eta\frac{\partial E}{\partial w_{kj}} = \eta\sum_{i\in S}\delta_k^{m+1}(i)\,y_j^m(i) \qquad (7)$$

where η is the learning rate, and δ_k^{m+1} is the local gradient at neuron k in layer m+1,

$$\delta_k^{m+1} = -\frac{\partial E}{\partial \nu_k^{m+1}} \qquad (8)$$

For output neurons ($m = L$),

$$\delta_k^{L+1} = \varphi_k'(\nu_k^{L+1})e_k^{L+1} \qquad (9)$$

For hidden neuron ($m < L$),

$$\delta_k^{m+1} = \varphi_k'(\nu_k^{m+1})\sum_n w_{nk}^{m+2}\delta_n^{m+2} \qquad (10)$$

Then the synaptic weights will be updated,

$$w_{kj}^{new} = w_{kj}^{old} + \Delta w_{kj} \qquad (11)$$

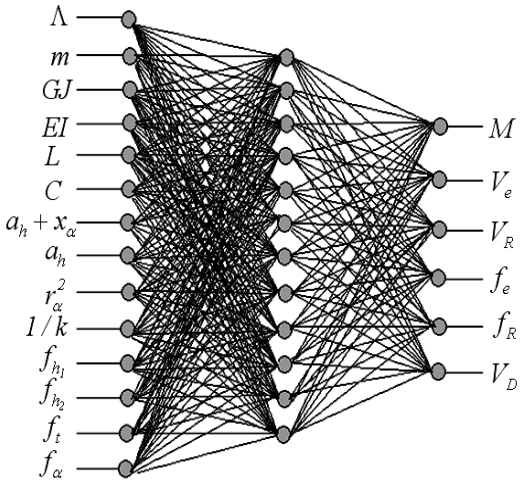

Fig. 2. Multi-layer perceptron with one hidden layer

As it is obvious in the formulation, the activation functions should be continuous. The shapes of activation functions, in hidden and output layers, are the ANN architecture design parameters for the designer. In our approach, the activation functions are tangent hyperbolic function and linear function in the hidden and output layers respectively. For learning process, the normalized form of the inputs and outputs are given to the ANN. The ANN learns the relation between the inputs and outputs and is trained for predicting output parameters.

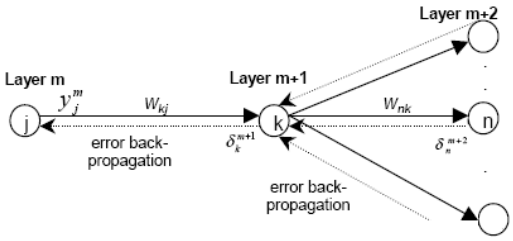

Fig. 3. Delta Learning Rule

In this work, base on the physics of the problem and according to the experimental data set that was used Barmby *et al*. (1950), we have chosen and examined some different combinations of inputs, outputs and number of neurons in hidden layer to obtain the most efficient configuration of the ANN. Then, the results were investigated base on the solution accuracy and required computational effort. The final perceptron (figure 2) which has the best accuracy and computational effort is in 14-12-6 configuration with 12 neurons in the hidden layer and learning rate equals to 0.004. As shown in the figure 2 and listed in table 1, inputs include geometrical parameters (length, chord, sweep angle, C.G. location and elastic axis position), structural parameters (mass per length unit, bending and torsional stiffness, radius of gyration and natural frequencies) and wing/air mass ratio and outputs include Mach number of flutter, flutter speed and frequency and divergence speed.

It must be mentioned that Barmby and his coworkers have investigated flutter phenomenon of swept wing models experimentally and analytically. The experimental work dealt with models as cantilevered at their roots with different geometry and mass properties and in order to facilitate analysis, they used models that were uniform and untapered (rectangular). Also, in their study, angle of sweep ranged from 0 to 60 degrees and Mach numbers extended to approximately 0.85. For demonstrating purposes, we train the ANN with both experimental and analytical output data reported by Barmby *et al*. (1950). The experimental data are denoted with subscript 'e' and analytical data are denoted with subscript 'R'. Also, Divergence speed (VD) is a theoretically calculated value.

3. RESULT AND DISCUSSION

To have optimum accuracy and computational cost, it is necessary to examine different learning rates in neural network training. Figure 4 shows the convergence history of the problem for different values of learning rates. The best results base on solution accuracy and computational effort are obtained for learning rate equals to 0.004. As it is clear, for greater learning rate values, we have less accuracy but more convergence rate. Figure 5 shows the upper bound of the stable region for designed ANN. As is shown in the figure, by increasing the learning rate from 0.04 to 0.07, one can see more

serious oscillations in convergence history and faster error increase. By increasing the learning rate to 0.08, the learning diagram diverges.

The trained neural network was tested for inputs that it had not seen before. At first, the predictions of flutter characteristics base on analytical data are compared with the calculated counterparts reported by Barmby *et al*. (1950).

Table 1 Inputs and outputs of the designed ANN

Type	Symbol	Description
Input	Λ	Angle of sweep
	m	Mass of wing per unit length
	GJ	Torsional stiffness
	EI	Bending stiffness
	l	Length of wing
	c	Wing chord
	$a_h + x_\alpha$	Nondimensional c.g. position
	a_h	Nondim. elastic axis position
	r^2_α	Square of nondim. radius of gyration of wing about elastic axis
	μ	Wing/air mass ratio ($m/\square\square b^2$)
	f_{h1}	First bending natural frequency
	f_{h2}	Second bending natural frequency
	f_t	First torsion natural frequency
	$f_\square$	Uncoupled first torsion frequency
Output	M	Mach number at flutter
	V_e	Experimental flutter speed
	V_R	Theoretical flutter speed
	f_e	Experimental flutter frequency
	f_R	Theoretical flutter frequency
	V_D	Theoretical divergence speed

As seen in the tables 2 and 3, agreement between calculated and predicted values is excellent and the ANN has successfully learned the mathematical logic governing the calculation of trained data. It means that for new cases, the ANN can quickly predict results of analytical method without need to solve the governing equations. Also, it indicates the ANN code works correctly.

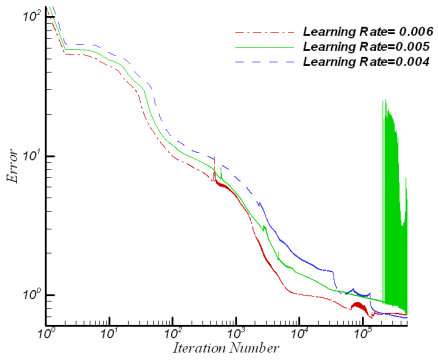

Fig. 4. Learning history for different learning Rates

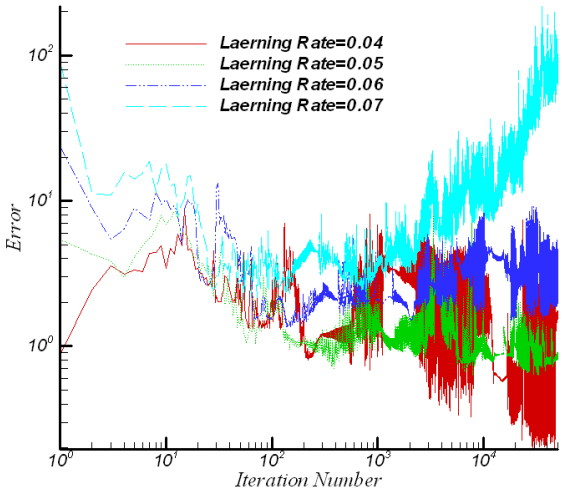

Fig. 5. Learning convergence history towards unstable region

Table 2 Analytical flutter characteristics and ANN predictions

model	V_R (m/s)	V_{R_ANN}	f_R (cps)	f_{R_ANN}
24	101.0	101.7	44	43.08
30B	95.66	98.26	44	43.53
30C	97.90	98.31	46	46.07
40A	117.5	120.8	41	40.52
64	40.67	37.04	32	29.53
73	95.21	98.12	46	44.61
73'	162.261	164.84	39	39.53
85-1	96.55	94.47	43	40.42
91-1	103.2	106.8	15	17.40

The analytical method used by Barmby *et al*. (1950) in aeroelastic calculations is simple and does not have enough accuracy in some cases. Therefore, for these cases, experimental data must be provided. Tables 4, 5 and 6 give the values of experimental flutter characteristics and the predictions by ANN. For some cases, there are also the results of an aeroelastic code verified for flutter characteristics prediction of wing (described in appendix section). As seen, the results are in good agreement and a few appeared differences are due to the complex nonlinear nature of the experimental data.

Table 3 Theoretical divergence velocities and ANN predictions

model	V_D (m/s)	V_{D_ANN} (m/s)
24	103.7	99.29
30B	118.9	118.0
30C	118.4	119.6
40A	158.6	160.0
64	36.34	36.03
73	117.5	117.1
73'	154.2	150.7
85-1	87.61	88.13
91-1	152.4	149.0

Table 4 Experimental flutter velocities and ANN predictions

model	$V_{exp.}$ (m/s)	V_{ANN} (m/s)	V_{code} (m/s)
24	125.6	121.0	……..
30B	120.2	118.1	117.8
30C	126.9	134.6	……..
40A	105.0	96.23	100.6
64	37.50	40.08	35.85
73	86.27	86.56	88.95
73'	125.1	123.3	……..
85-1	143.9	146.1	144.3
91-1	56.77	57.11	……..

In table 5, flutter frequency of models 24 and 30C are not presented because there are neither experimental nor aeroelastic code results for them. Nevertheless, the ANN could predict the values of flutter frequency of these models, 34.85 and 32.60 respectively. Because of the lack of experimental data for the mentioned models and also inability of developed aeroelastic code to calculate flutter characteristics of them, the capability of the ANN in accurate experiment-based prediction of flutter characteristics of such models becomes more considerable and interesting.

Table 5 Experimental flutter frequencies and ANN predictions

model	$f_{exp.}$ (cps)	f_{ANN} (cps)	f_{code} (cps)
30B	……..	50.36	50.39
40A	……..	56.83	55.24
64	……..	13.31	14.48
73	29	26.78	28.41
73'	22	25.05	……..
85-1	35	35.37	36.22
91-1	12.5	13.65	……..

Table 6 Experimental flutter Mach numbers and ANN predictions

model	M_{exp}	M_{ANN}
24	0.76	0.79
30B	0.45	0.45
30C	0.81	0.79
40A	0.30	0.29
64	0.24	0.26
73	0.57	0.56
73'	0.82	0.86
85-1	0.41	0.42
91-1	0.37	0.38

Another capability of the designed ANN is that we can use it for parametric analysis and study of the effects of various parameters on flutter characteristics. Figure 6 shows the effect of sweep angle variation on flutter speed value for model 73. As seen, the predicted trends of the code and the ANN are similar and the difference between the values of two methods decreases towards the middle. The major advantage of the ANN approach is that it does not require multiple runs to obtain such trend lines. In this way, considerable saving in computational time and cost will be earned.

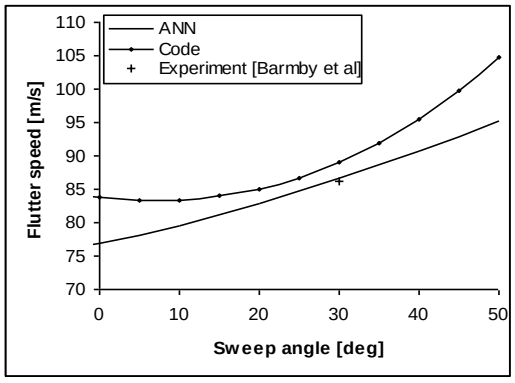

Fig. 6. Effect of sweep angle on flutter speed of model 73

Figure 7 represents the effect of wing/air mass ratio on flutter speed for model 73 that its data was not included in the training data. As shown, agreement between the code results and the ANN predictions is excellent. We can obtain similar trend curves regards the effects of the other parameters on the flutter characteristics using the ANN.

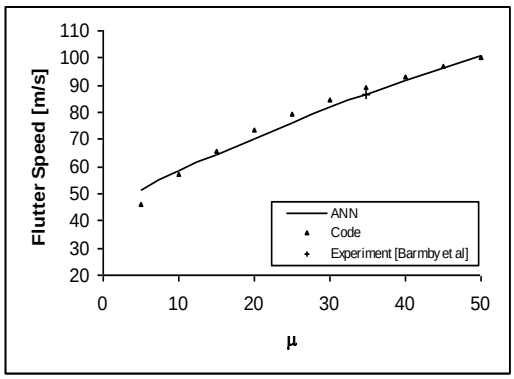

Fig. 7. Effect of wing/air mass ratio on flutter speed of model 73

It is noteworthy that the ANN result accuracy depends on the ANN type, number of inputs and outputs, number of hidden layers, number of neurons in the hidden layers, and number of data which are used for ANN training. As said before, in this work, the optimal case with minimum error and computational effort was obtained using different ANN configurations.

4. CONCLUSION

In this paper, the utilization of artificial neural networks for wing aeroelastic studies was explored. In this respect, an efficient artificial neural network was developed and used for flutter prediction of swept wings successfully. The ANN is a supervised multilayer perceptron that was trained based on an experimental data set involves flutter characteristics of various rectangular wing models. It was seen that the ANN results are in good agreement with the experimental data. So, the resultant ANN can be used for quick prediction of flutter characteristics of swept rectangular wings with proper accuracy. Also, the designed ANN may be implemented for investigation of the effects of various parameters on the flutter characteristics of swept rectangular cantilever wings.

REFERENCES

Barmby, J.G., H.J. Cunningham and I.E. Garrick (1950). Study of effects of sweep on the flutter of cantilever wings, *NACA Technical Note 2121*, Langley aeronautical laboratory.

Marques, F.D. and J. Anderson (2001). Identification and prediction of unsteady transonic aerodynamic loads by multi-layer functional, *Journal of Fluids and Structures*, 15, 83-106.

Natarajan, A. (2002). *Aeroelasticity of morphing wings using neural networks*, PhD Thesis of aerospace engineering, Virginia Polytechnic Institute and State University.

Pesonen, U.J., and R.K. Agarwal (2002). Artificial neural network prediction of aircraft aeroelastic behavior, AIAA 2002-0947.

Pitt, D.M. and D.P. Haudrich (2004). Artificial neural network for multiple aeroelastic analysis, *Proceedings of 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, California, USA.

Voitcu, O. and Y.S. Wong (2003). An improved neural network model for nonlinear aeroelastic analysis, *44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Virginia, USA.

Wang, Z. (2004). *Time-domain simulations of aerodynamic forces on 3D configurations, unstable aeroelastic responses, and control by neural network systems*, PhD Thesis, Virginia Polytechnic Institute and State University.