Control on 3-D Fixable Wing Flutter Using an Adaptive Neural Controller

Dr. Mauwafak Ali Tawfik
Machines & Equipments Engineering Department, University of Technology / Baghdad
Dr.Mohammed Idris Abu-Tabikh
Machines & Equipments Engineering Department, University of Technology / Baghdad
Hayder Sabah Abd Al-Amir
Mechanical Engineering Department, Institute of Technology / Baghdad
Email: has_04@yahoo.com

Received on: 16/1/2012 & Accepted on: 3/5/2012

ABSTRACT

An adaptive neural controller to control on flutter in 3-D flexible wing is proposed. The aeroelastic model was based on the coupling between structure-of the equivalent plate (wing) and the aerodynamic model that is based on a hybrid unsteady panel methodTime domain simulations were used to examine the dynamic aeroelastic instabilities of the system (e.g. the onset of flutter and limit cycle oscillation). The structure of the controller consists of two models namely modified Elman neural network (MENN) and feedforward multi-layer Perceptron (MLP). The MENN model is trained with off-line and on-line stages to guarantee that the outputs of the model accurately represent the plunge motion of the wing and this neural model acts as the identifier. The feedforward neural controller is trained off-line and adaptive weights are implemented on-line to find the generalized control action (function of addition lift force), which controls the plunge motion of the wing. The general back propagation algorithm is used to learn the feedforward neural controller and the neural identifier. The simulation results show the effectiveness of the proposed control algorithm; this is demonstrated by the minimized tracking error to zero approximation with very acceptable settling time.

Keywords: Aeroelasticity, Flutter, Adaptive Control, Neural Networks.

السيطرة على رفرفة جناح ثلاثي الأبعاد باستخدام مسيطر عصبي متكيف

الخلاصة

تم في هذا البحث اقتراح مسيطر عصبي متكيف للسيطرة على الرفرفة لنموذج جناح مرن ثلاثي الأبعاد. بني نموذج المرونة الهوائية على مبدأ التزاوج بين نموذج الصفيحة المكافئة (الجناح) و النموذج الدينامكي الهوائي الذي تم استخدام طريقة الأشرطة غير المستقرة الهجينة. تم تحديد المنطقة التي يكون فيها النظام غير مستقر من خلال فحص استجابته مع الزمن حيث تم إيجاد السرعة التي تبدأ عندها ظاهرة الرفرفة والتذبذب الدوري المحدد. تتكون هيكلية المسيطر من نموذجين هما

https://doi.org/10.30684/etj.30.16.9
2412-0758/University of Technology-Iraq, Baghdad, Iraq

الشبكة العصبية المحسنة لألمن(MENN) و بيرسبترون متعدد الطبقات (MLP) .لقد تم تأهيل نموذج (MENN) في مرحلتين هما مرحلة الخط المغلق ومرحلة الخط المفتوح لضمان تطابق مخرج النموذج العصبي مع مخرج منظومة الجناح وهو الحركة العمودية لتكوين النموذج العصبي المعرف. تم تأهيل المسيطر العصبي الأمامي من خلال الخط المغلق ثم تم تحديث الأوزان لهذا المسيطر من خلال الخط المفتوح لإيجاد فعل المسيطر العام(الذي هو دالة قوة رفع إضافية) المطلوب للسيطرة على الحركة العمودية.تم استخدام خوارزمية الانتشار الخلفي لتأهيل النموذجين. كانت نتائج المحاكاة لهذا المسيطر العصبي فعالة من خلال تقليل الرفرفة إلى صفر وبزمن استقرار مناسب .

INTRODUCTION

The performance of aircraft is often limited by adverse aeroelastic interactions such as flutter. Flutter is defined as: a dynamic instability of a flight vehicle associated with the interaction of aerodynamic, elastic, and inertial forces. If flutter can be controlled at cruise speeds, lighter wings can be designed and consequently more efficient airplanes. It is therefore, in the aircraft designer's best interest to design innovative ways in which flutter can be controlled without making the resulting structure too heavy.

Nowadays the researchers pay pronounced attention to the control of flutter in 3-D wing model, while earlier published works were often catered for the control of flutter using rigid wing model.

Many researches in this field proposed different flutter controllers, Palaniappan, et al. [1] developed a feedback algorithm for the control of flutter. The actuators are jets in the walls through which there is a small mass flow, either by way of blowing or suction. Afkhami and Alighanbair [2] presented nonlinear controller to control flutter. Integral-input-to-state stability concept is utilized for the construction of a feedback controller. Haiwei and Jinglong [3] proposed the robust flutter analysis of a nonlinear 2-D wing section with structural and aerodynamic uncertain using μ-method. The parametric uncertainty was adopted to describe the uncertainties in structure and aerodynamics.

Recently, the intelligent algorithm like neural networks and fuzzy were introduced in aeroelastic filed as controller or flutter prediction device. Melin and Castillo[4] combined adaptive model-based control using neural networks with the method for modeling using fuzzy logic, and fractal theory to obtain a new hybrid neuro-fuzzy-fractal method for control of nonlinear dynamic aircraft. The adaptive controller can be used to control chaotic and unstable behavior in aircraft systems. Chen, et al. [5] presented an approach using artificial neural networks (ANN) algorithm for predicting the flutter derivatives of rectangular section models without wind tunnel tests. Marques et al. [6] presented an active aeroelastic control strategy for vibration suppression of a flexible smart non-linear wing based on fuzzy logic. The finite element method has been used to model the wing structural-dynamics. The vortex-lattice method has been used for the unsteady aerodynamic model. The fully coupled fluid-structural interaction model [7] is used in the present work.

In this work the designated model is adopted to predict the flutter condition in the wing. A hybrid panel-discrete vortex unsteady method combined with the numerical lifting line method is used to describe the aerodynamic model. While the equivalent plate technique [8] which relays on a solved plate equation by an assumed mode method is used to represent the structure wing model.

The contribution of the present work is the utilization of a relatively simple approximation neural network to identify the posture of the fully coupled fluid-structural interaction wing system and to design an adaptive neural controller.

FLUTTER ANALYSIS

Flutter is a self-feeding and potentially destructive vibration where aerodynamic forces on an object couple with a structure's natural mode of vibration to produce rapid periodic motion. In flutter condition the wing undergoes plunge or pitch or both motions during the flight [9].

Time domain simulation are used to examine the dynamic aeroelastic instabilities of the system (e.g. the onset of flutter and limit cycle oscillation (LCO)) as done in Ref [10]. The simulation is performed by solving fluid-structural interactions problem for different velocities and initial conditions. It is well known that the initial conditions may affect the stability of a system; however this effect is not found in present case study. It was found that LCO appear at U=153m/sec and never appears at speed less than it what ever the initial conditions. Therefore the flutter speed is 153m/sec and the proposed controller must give a good performance at speed higher than that value (unstable region).

Figures (1, 2, 3 and 4) show the generalized displacement responses at speed 120,135, 153 and 160 m/sec respectively. It is clear at velocity 120 m/sec and below the system is stable and does not need controller at this range of velocity. But at velocity of 135 m/sec the stability becomes less and decay time becomes more. Unstable responses appear at velocities above 153 m/sec. This behaviour is clear at velocity 160 m/sec, as shown in figure (4).

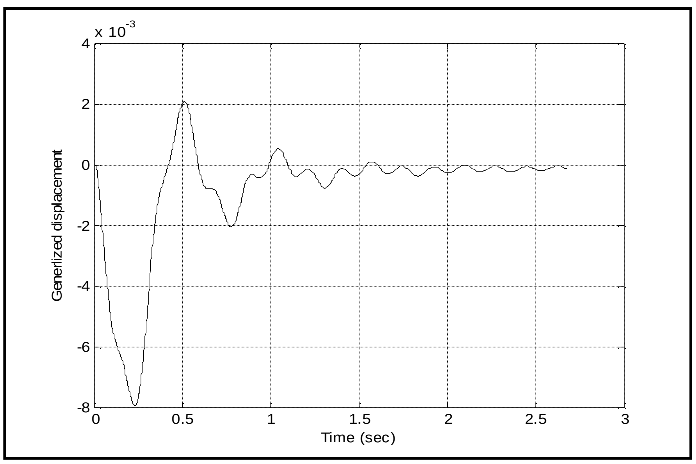

Figure (1): Time history of generalized displacement of the wing at 120 m/s speed .

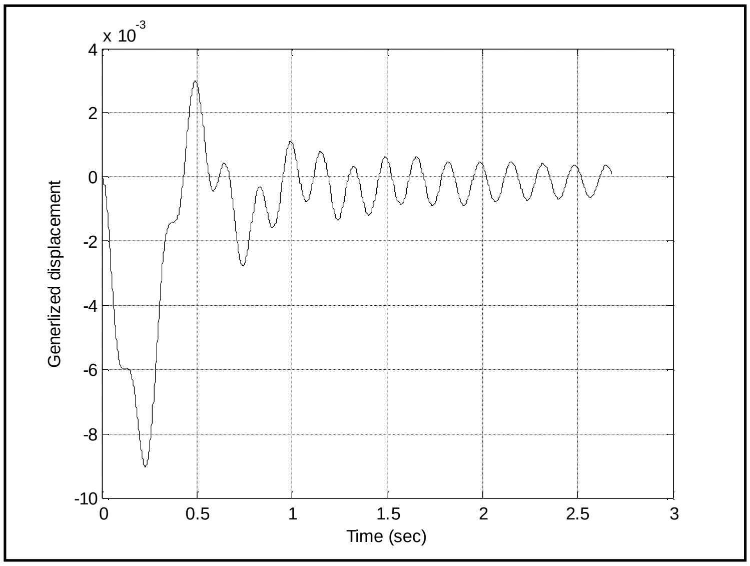

Figure (2): Time history of generalized of the wing at 135 m/s speed .

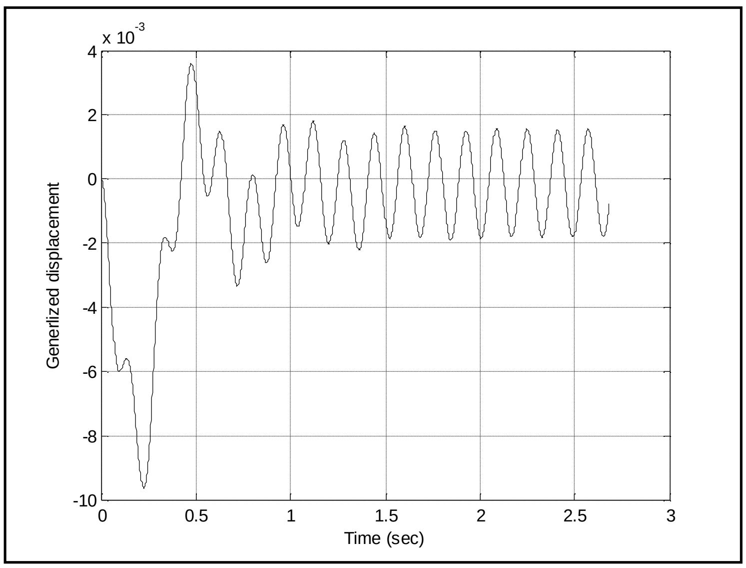

Figure (3): Time history of generalized displacement of the wing at 153 m/s speed ,flutter condition .

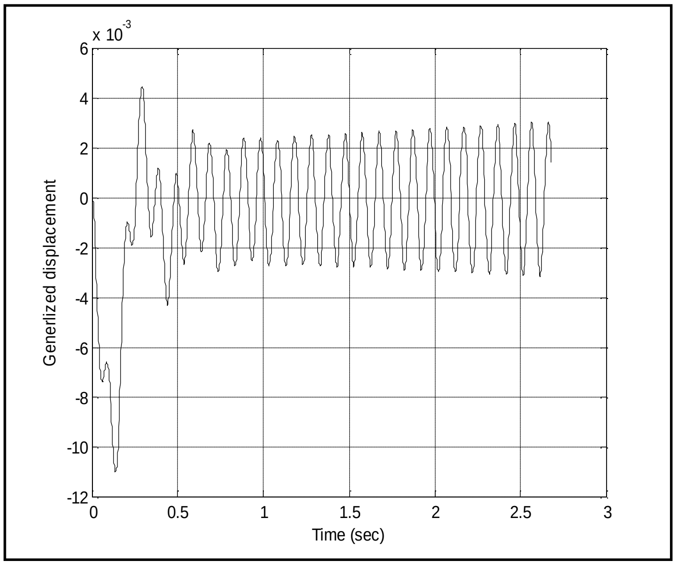

Figure (4): Time history of generalized displacement of the wing at 160m/s speed.

ADAPTIVE NEURAL CONTROL METHODOLOGY

The approach to control wing motion depends on the available information about the system and the control objectives. The wing system is considered as modified Elman recurrent neural networks model. The first step in the procedure of the control structure is the identification of dynamics of wing system from the input-output data. Then a feedforward neural controller is designed using feedforward multi-layer Perceptron neural network to find controller action that control on the plunge wing motion.

The proposed structure of the adaptive nonlinear neural controller can be given in the form of block diagram as shown in figure (5). It consists of:

1- Neural Network Identifier of Wing.

2- Feedforward Neural Controller.

In the following sections, each part of the proposed controller will be explained in details.

WING SYSTEM NEURAL NETWORK IDENTIFIER

The modified Elman recurrent neural network model is applied to construct the wing system neural network identifier as shown in figure (6) [11]. The nodes of input, context, hidden and output layers are highlighted. The network uses two configuration models, series-parallel and parallel identification structures, which are trained using dynamic back-propagation algorithm. The structure shown in figure (6) is based on the following equations [11]:

$$g(k) = F\{VH\overline{G}(k), VC\overline{g^{o}}(k), bias\overline{Vb}\} \qquad \dots (1)$$

$$O(k+1) = L\{Wg(k), bias\overline{Wb}\} \qquad \dots (2)$$

Where VH,VC and W are weight matrices, $\overline{Vb}$ and $\overline{Wb}$ are weight vectors and F is a non-linear vector function. The multi-layered modified Elman neural network, shown in figure (6), is composed of many interconnected processing units called neurons or nodes.

The network weights are denoted as follows:

VH : Weight matrix of the hidden layers.

VC : Weight matrix of the context layers.

$\overline{Vb}$: Weight vector of the hidden layers.

W : Weight matrix of the output layer.

$\overline{Wb}$: Weight vector of the output layer.

L : Denotes linear node.

H : Denotes nonlinear node with sigmoidal function

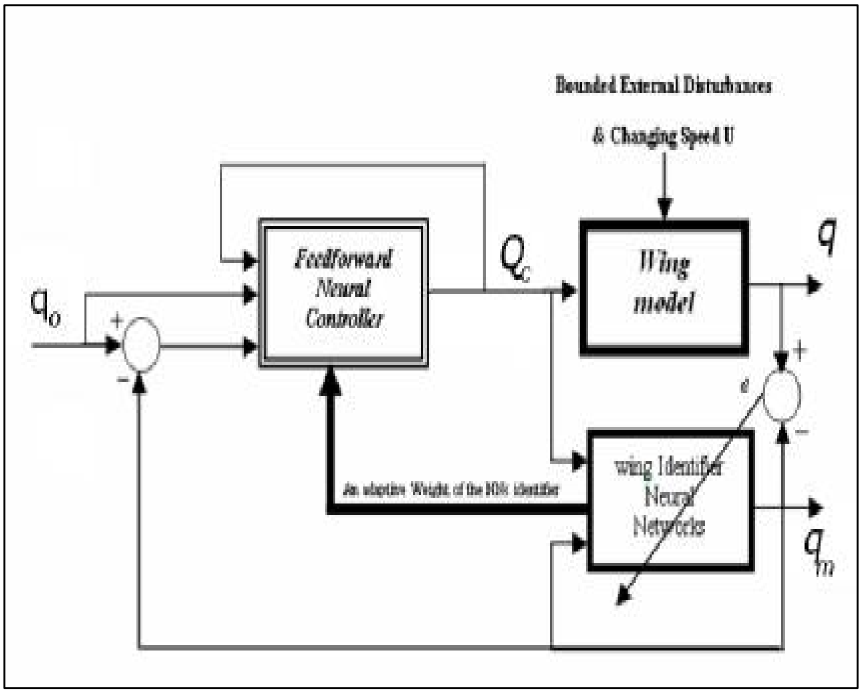

Figure (5): The proposed structure of the adaptive nonlinear Neural Controller for the wing.

In order to improve the ability of network memory, self-connections, with fixed value l , are introduced into the context units of the network to give these units a certain amount of inertia [10]. The introduction of self-connections in the context units increases the possibility of modelling high-order systems by Elman network.

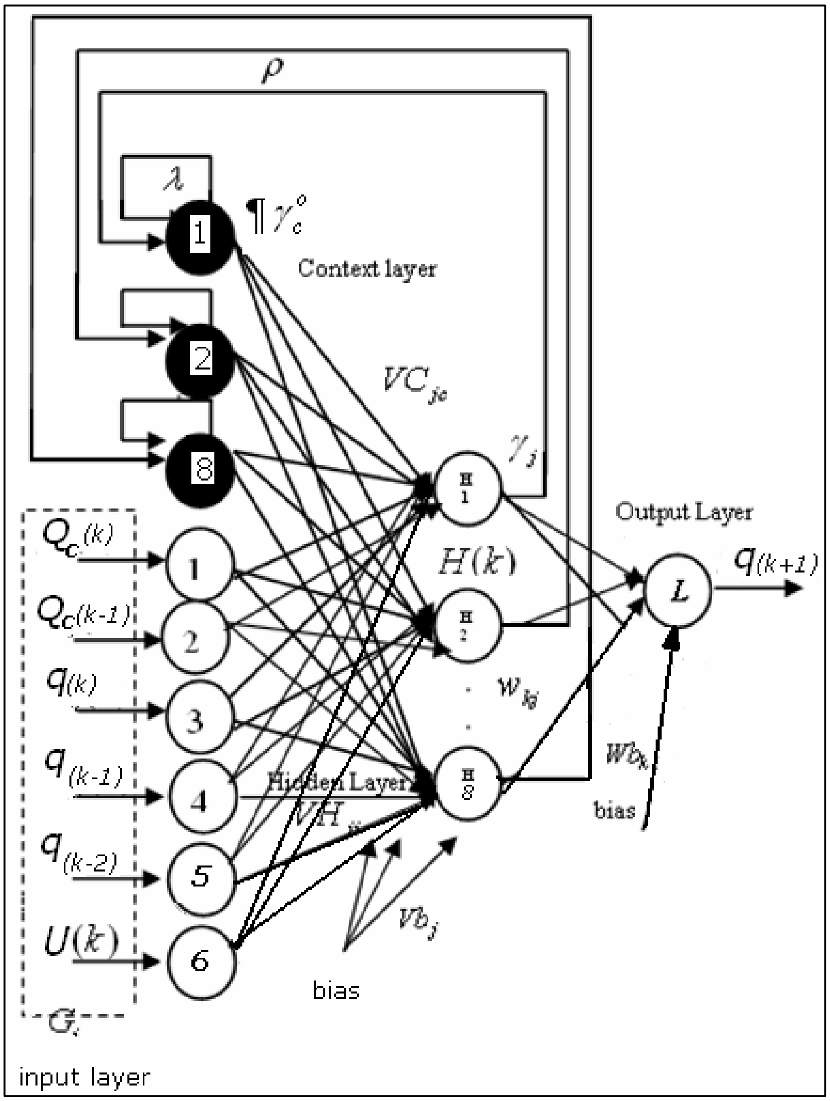

Figure (6): The Modified Elman Recurrent Neural Networks Acts as the plunging motion of the wing.

The output of the context unit in the modified Elman network is given by [11]:

$$g_c^o(k) = l g_c^o(k-1) + r g_c(k-1) \qquad \dots (3)$$

where $g_c^o(k)$ and $g_c(k)$ are the outputs of the context and hidden units respectively. l is the feedback gain of the self-connections and r is the connection weight from the hidden units (j^{th}) to the context units (c^{th}) at the context layer. The value of l and r are selected randomly between (0 and 1).

To explain these calculations, consider the general j^{th} neuron in the hidden layer. The inputs to this neuron consist of an i– dimensional vector, where i is the number of the input nodes. Each of the inputs has *VH* and *VC* weights associated with it.
$\overline{Vb}$ is the weight vector for the bias input that is set equal to -1 to prevent the neurons quiescent. The first calculation within the neuron consists of calculating the weighted sum net_j of the inputs as [11 and 12]:

$$net_j = \sum_{i=1}^{nh} VH_{ji} \times G_i + \sum_{c=1}^{C} VC_{jc} \times g_c^o + bias \times Vb_j \qquad \dots (4)$$

Where j.is the number of the hidden nodes, c is the number of the context nodes and $\overline{G}$ is the input vector. The output of the identifier is the modelling plunge motion in generalized form and is defined as: q_m
The learning algorithm will be used to adjust the weights of dynamical recurrent neural network. Dynamic back propagation algorithm is used to train the Elman network. The sum of the square of the differences between the desired output q and neural network identifier output q_m is given by equation (5).

$$E = \frac{1}{2}\sum_{i=1}^{np}(q - q_m)^2 \qquad \dots (5)$$

where *np* is the number of patterns.
The connection matrix between hidden layer and output layer is W_{kj}.

$$\Delta W_{kj}(k+1) = -h\frac{\partial E}{\partial W_{kj}} = -h\frac{\partial E}{\partial q_m(k+1)}\frac{\partial q_m(k+1)}{\partial o_k}\frac{\partial o_k}{\partial net_k}\frac{\partial net_k}{\partial W_{kj}} \qquad \dots (6)$$

where h is learning rate.

$$\Delta W_{kj}(k+1) = h \times g_j \times e_k \qquad \dots (7)$$

$$W_{kj}(k+1) = W_{kj}(k) + \Delta W_{kj}(k+1) \qquad \dots (8)$$

The connection matrix between input layer and hidden layer is VH_{ji}

$$\Delta VH_{ji}(k+1) = -h\frac{\partial E}{\partial VH_{ji}} = -h\frac{\partial E}{\partial q_m(k+1)}\frac{\partial q_m(k+1)}{\partial o_k}\frac{\partial o_k}{\partial net_k}\frac{\partial net_k}{\partial g_j}\frac{\partial g_j}{\partial net_j}\frac{\partial net_j}{\partial VH_{ji}} \quad \dots (9)$$

$$\Delta VH_{ji}(k+1) = h\times f(net_j)'\times G_i\sum_{k=1}^{K} e_k W_{kj} \quad \dots (10)$$

$$VH_{ji}(k+1) = VH_{ji}(k) + \Delta VH_{ji}(k+1) \quad \dots (11)$$

The connection matrix between context layer and hidden layer is VC_{ji}.

$$\Delta VC_{jc}(k+1) = -h\frac{\partial E}{\partial VC_{jc}} = -h\frac{\partial E}{\partial q_m(k+1)}\frac{\partial q_m(k+1)}{\partial o_k}\frac{\partial o_k}{\partial net_k}\frac{\partial net_k}{\partial g_j}\frac{\partial g_j}{\partial net_c}\frac{\partial net_c}{\partial VC_{jc}} \quad \dots (12)$$

$$\Delta VC_{jc}(k+1) = h\times f(net_j)'\times g_c^o\sum_{k=1}^{K} e_k W_{kj} \quad \dots (13)$$

$$VC_{jc}(k+1) = VC_{jc}(k) + \Delta VC_{jc}(k+1) \quad \dots (14)$$

FEEDFORWARD NEURAL CONTROLLER

The Feedforward Neural Controller (FFNC) is essential to stabilize the tracking error of the wing system when the response of the wing is drifted from the desired condition during transient state and kept the steady-state tracking error at zero. The controller generates controller action that minimizes the cumulative error between the desired condition and the output response of the wing. The FFNC is supposed to learn the adaptive inverse model of the wing with off-line and on-line stages to calculate wing's reference input control action and will keep the wing stable without flutter state in the presence of any disturbances or dynamics parameters changing.

To achieve FFNC, a multi-layer Perceptron model is used as shown in figure (7) [13]. The network notations are as follows:

$Vffc$: Weight matrix of the hidden layers.

$\overline{V}bffc$: Weight vector of the hidden layers.

$Wffc$: Weight matrix of the output layer.

$\overline{Wb}\,ffc$: Weight vector of the output layer.

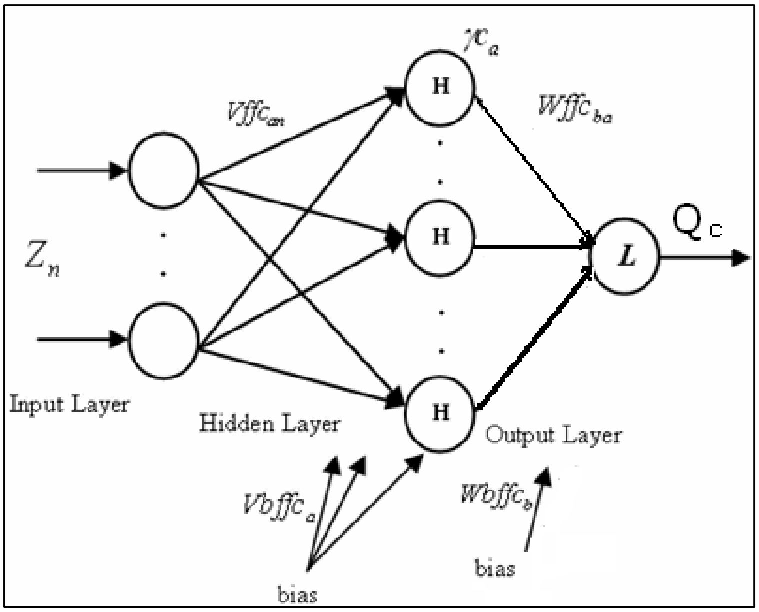

Figure (7): The Multi-Layer Perceptron Neural Networks of the Feedforward Neural Controller[13].

To explain these calculations, consider the general a^{th} neuron in the hidden layer shown in figure (7). The inputs to this neuron consist of an n–dimensional vector, where n is the number of the input nodes. Each input has an associated weight of $Vffc$. The first calculation within the neuron is to calculate the weighted sum of the inputs, $netc_a$ as [13, 14 and 15]:

$$netc_a = \sum_{a=1}^{nhc} Vffc_{an} \times Z_n + bias \times Vbffc_a \qquad \dots (15)$$

Where *nhc* is the number of the hidden nodes and

$$Z_n = [e(m); e(m-1); q_m(m); q_m(m-1); Q_c(m-1); Q_c(m-2)].$$

Next, the output of the neuron h_a is calculated as the continuous sigmoid function of the $netc_a$ as:

$$gc_a = H(netc_a) \qquad \dots (16)$$

$$H(netc_a) = \frac{2}{1+e^{-netc_a}} - 1 \qquad \dots (17)$$

Once the outputs of the hidden layer have been calculated, they are passed to the output layer.

In the output layer, the linear neuron is used to calculate the weighted sum *netco* of its inputs, which are the output of the hidden layer as:

$$netco_b = \sum_{a=1}^{nhc} Wffc_{ba} \times gc_a + bias \times Wbffc_b \qquad \ldots (18)$$

where $Wffc_{ba}$ are the weights between the hidden neuron gc_a and the output neurons. Then the sum ($netco_b$) will be passed through a linear activation function of slope 1; another slope can be used to scale the output, as:

$$Oc_b = L(netco_b) \qquad \ldots (19)$$

The outputs of the feedforward neural network controller represent control action.

The training of the feedforward neural controller is performed off-line as shown in figure (8). And adaptive weights are adapted on-line. It depends on the posture neural network identifier to find the wing Jacobian through the neural identifier model. This approach is currently considered as one of the better approaches that can be followed to overcome the lack of initial knowledge.

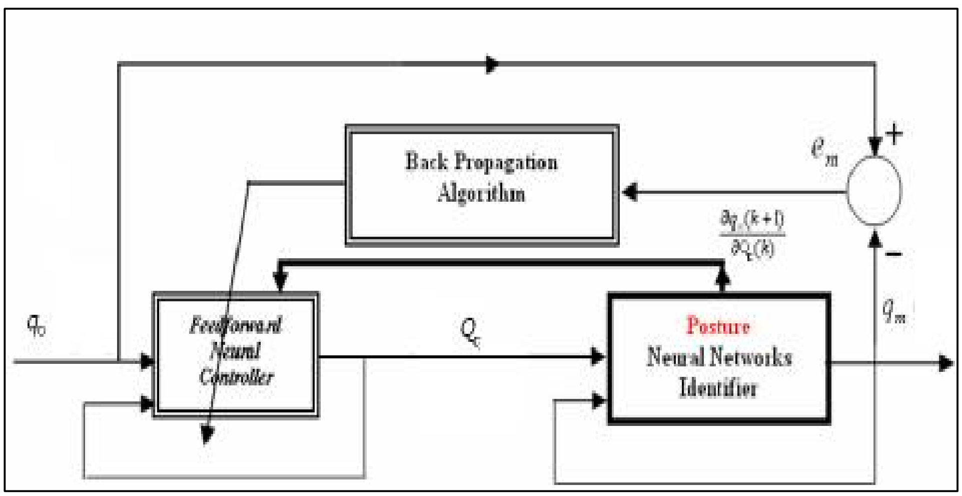

Figure (8): The feedforward neural controller structure for wing model.

The dynamic back propagation algorithm is employed to realize the training the weights of the feedforward neural controller. The sum of the square of the differences between the desired posture q_o and neural network posture q_m is:

$$Ec = \frac{1}{2}\sum_{i=1}^{npc}(q_o - q_m)^2 \quad \dots (20)$$

Where: *npc* is number of patterns.

To achieve equation (20) a modified Elman neural network will be used as posture identifier. This task is carried out using an identification technique based on series-parallel and parallel configuration with two stages to learn the posture identifier. The first stage is an off-line identification, while the second stage is an on-line modification of the weights of the obtained wing neural identifier. The on-line modifications are necessary to keep tracking any possible variation in the dynamic parameters of the wing system. Back Propagation Algorithm (BPA) is used to adjust the weights of the posture neural identifier to learn dynamic of the flexible wing system by applying a simple gradient decent rule.

The connection matrix between hidden layer and output layer is $Wcont_{ba}$.

$$\Delta Wffc_{ba}(k+1) = -h\frac{\partial Ec}{\partial Wffc_{ba}} = -h\frac{\partial Ec}{\partial q_m(k+1)}\frac{\partial q_m(k+1)}{\partial Q_c(k)}\frac{\partial Q_c(k)}{\partial oc_b}\frac{\partial oc_b}{\partial netc_b}\frac{\partial netc_b}{\partial Wffc_{ba}} \quad \dots (21)$$

$$\frac{\partial Ec}{\partial q_m(k+1)} = \frac{\partial \frac{1}{2}\sum(q_o - q_m)^2}{\partial q_m(k+1)} \quad \dots (22)$$

$$Jacobian = \frac{\partial q_m(k+1)}{\partial Q_c(k)} = \frac{\partial q_m(k+1)}{\partial o_k(k)}\frac{\partial o_k(k)}{\partial net_k}\frac{\partial net_k}{\partial g_j}\frac{\partial g_j}{\partial net_j}\frac{\partial net_j}{\partial Q_c(k)} \quad \dots (23)$$

For linear activation function in the outputs layer and nonlinear activation functions in the hidden layer for neural network identifier the equation (23) becomes as follows:

$$\frac{\partial q_m(k+1)}{\partial Q_c(k)} = \sum_{j=1}^{nh} f(net_j)' VH_{jb}\sum_{k=1}^{K} W_{kj} \quad \dots (24)$$

Substituting equations (22 and 24) into equation (21), $\Delta Wffc_{ba}(k+1)$ becomes:

$$\Delta Wffc_{ba}(k+1) = hgc_a \times \sum_{j=1}^{nh} f(net_j)' VH_{jb}((eq_m(k+1)W_{1j})) \quad \dots (25)$$

$$Wffc_{ba}(k+1) = Wffc_{ba}(k) + \Delta Wffc_{ba}(k+1) \quad \dots (26)$$

The connection matrix between input layer and hidden layer is $Vffc_{an}$.

$$\Delta Vffc_{an}(k+1) = -h\frac{\partial Ec}{\partial Vffc_{an}} = -h\frac{\partial Ec}{\partial q_m(k+1)}\frac{\partial q_m(k+1)}{\partial Q_c(k)}\frac{\partial Q_c(k)}{\partial oc_b}\frac{\partial oc_b}{\partial netc_b}\frac{\partial netc_b}{\partial gc_a}\frac{\partial gc_a}{\partial netc_a}\frac{\partial netc_a}{\partial Vffc_{an}} \quad ...(27)$$

$$-h\frac{\partial Ec}{\partial Vffc_{an}} = -h\frac{\partial Ec}{\partial q_m(k+1)}\frac{\partial q_m(k+1)}{\partial Q_c(k)}\times\sum_{b=1}^{B}Wffc_{ba}\times f(netc_a)'\times Z_n \quad ...(28)$$

Substituting equations (22 and 24) into equation (28), $\Delta Vffc_{an}(k+1)$ becomes:

$$\Delta Vffc_{an}(k+1) = hZ_n f(netc_a)'\sum_{b=1}^{B}Wffc_{ba}\sum_{j=1}^{nh}f(net_j)'\sum_{i=1}^{I}VH_{ji}((eq_m(k+1)W_{1j})) \quad ...(29)$$

The B and I are equal to one because there is one output in the feedforward neural controller.

$$Vffct_{an}(k+1) = Vffc_{an}(k) + \Delta Vffc_{an}(k+1) \quad ...(30)$$

Once the feedforward neural controller has learned, it generates the control action to keep the output of the wing at reference value and to overcome any external disturbances during motion.

RESLUTS AND DISCUSSION

The proposed controller is verified with computer simulation using Matlab program. Because of the vast number of the recorded data that resulting from the solution structure -fluid interaction and to learn the neural network algorithm these date in easy way and reasonable time in personal computer that has limited memory, first generalized displacement is used only to modulate the wing system.

Also the first generalized displacement has predominate effects on aeroelastic wing behaviour than others generalized displacement.

The simulation is carried out by tracking a desired plunging before, through and after flutter condition.

The first stage of operation is to set the position (plunging motion) neural network identifier. This task is performed using series-parallel and parallel identification technique configuration with modified Elman recurrent neural networks model. The identification scheme of the wing system is needed to input-output training data pattern to provide enough information about dynamics wing model to be modelled. This can be achieved by injecting a sufficiently rich input signal to excite all process modes of interest while also ensuring that the training patterns adequately covers the specified operating region. A hybrid excitation signal has been used for the wing model.

The training set is generated by feeding a pseudo random binary sequence (PRBS) signals, with a sampling time of 0.0005 second, to the model and measuring its corresponding outputs, position (plunging motion) . Back propagation learning algorithm is used with the modified Elman recurrent neural network of the structure 6-8-8-1. The number of nodes in the input, hidden, context and output layers are 6, 8, 8 and 1 respectively as shown in figure (6).

A training set of 2000 patterns has been used with a learning rate of 0.1 and variable speed inputs U= [135 153, 160] m/sec. After 5439 epochs, the identifier output of the neural network, plunge motion is approximated to the actual outputs as shown in figure (9).

The testing set is generated by difference feeding a PRBS signals as shown in figure (10), and it is applied to the system. Figure (10) compare the time response of the parallel mode output with the actual plant output, and there is excellent identification.

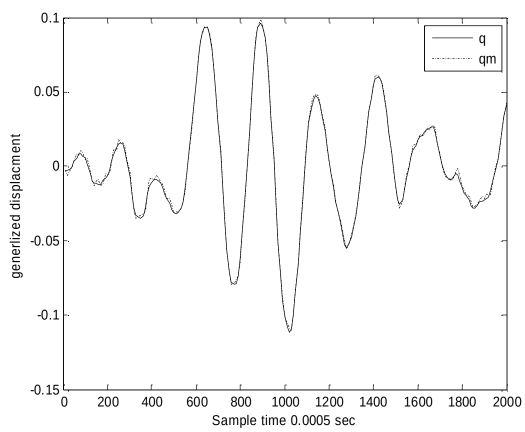

Figure (9): The response of the neural network Identifier with the actual flexible wing Model output for the learning set.

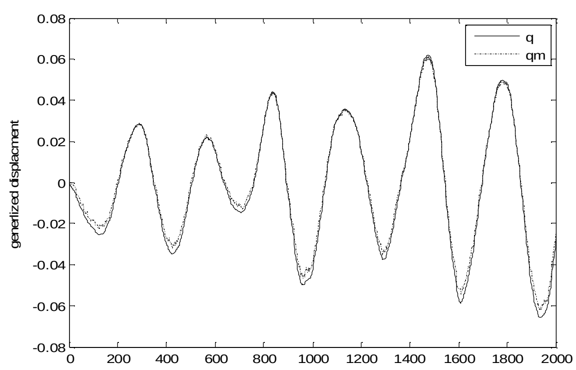

Figure (10): The response of the neural network Identifier with the actual flexible wing model output for the testing set.

The controller performance is simulated at three values of the flight speed (135,153,160 m/sec) which in unstable region and at different initial conditions of plunging motion. Figure (11) shows the closed loop responses for the controlled wing system. The controller reach the requirements and the closed-loop simulation obtained is stable .The over shoot and settling time increase slightly with increasing of the velocity. Also the oscillation during the transient period appears with increasing of the velocities, but its amplitude is small and converge to desired condition very quickly with settling time 0.7sec at high velocity U=153m/sec.

Also in the figure (11) can be seen the responses of the controller action (function of additional lift) at U=135m/sec, 153 m/sec and 160m/sec.The controller action in real may represent any external controller device like control surface, jets in the walls through which there is a small mass flow, either by way of blowing or suction and smart wing that change the shape of the airfoil of the wing to change the lift.

The values of the initial conditions are varying to make the stable or unstable, so the present controller performance is tested at different initial conditions as shown in figure (12). When the initial values of plunging increase the over shoot, the oscillation and setting time increase during the transient period. The present controller can give acceptable performance and reaches to desired condition at very short setting time about 1sec at large initial condition.

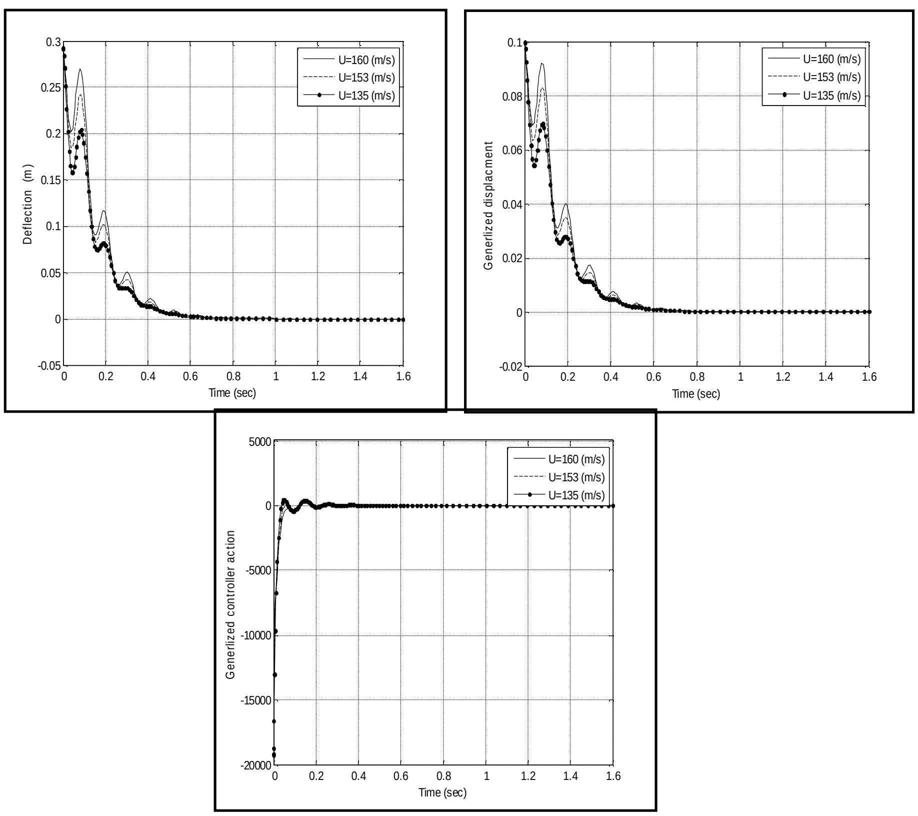

Figure (11): System responses with controller at different speed.

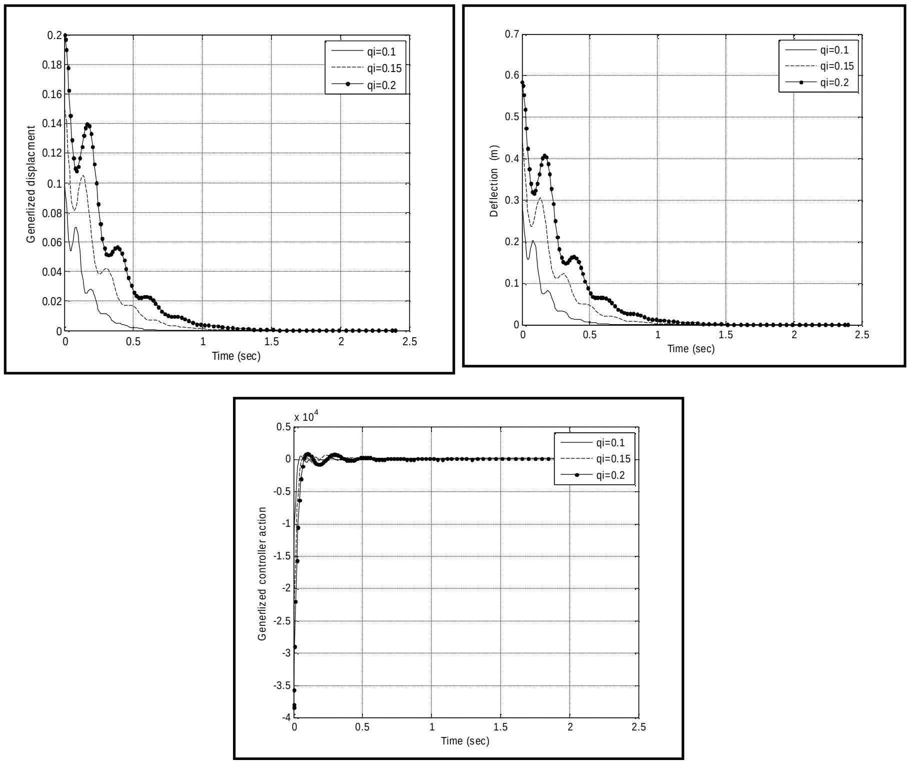

Figure (12): System responses with controller at different initial condition

CONCLUSIONS

Time domain simulations are used to predict the dynamic aeroelastic instabilities, find the flutter speed and LCO for model of a flexible wing with fully coupled fluid-structural interaction. The adaptive neural control methodology for nonlinear flutter wing is presented in this paper. The proposed controller consists of two parts: (plunging motion) neural network identifier and feedforward neural controller. The control scheme minimizes the cost function of tracking errors. It uses two models of neural networks in the structure of the controller, multi-layer Perceptron and modified Elman neural network. They are trained off-line and adapted on-line using back propagation algorithm with series-parallel and parallel configurations to guarantee that the model outputs of the neural network match those of the wing model outputs.

The simulation results show that the proposed controller has the capability to generate smooth and suitable controller action commands without sharp spikes. Moreover, it has the capability of compensating any different velocities and from any initial conditions sudden change of the aeroelastic system .Therefore, the proposed adaptive neural control methodology can be considered capable of effectively eradicating the tracking errors for the flexible wing model.

Simulation results show that the proposed controller is robust and effective in comparison with the controller in [2] in terms of fast response with minimum settling times and minimum tracking error until in flutter condition until when it use in unstable region.

REFERENCES

[1]. Palaniappan K, Sahu P.and Alonso J.J " Design of Adjoint Based Laws for wing flutter control",American Institute of Aeronautics and Astronautics Paper AIAA-(2009)-148.

[2]. Afkhami S. and. Alighanbari H. "Nonlinear control design of an airfoil with active flutter suppression in the presence of disturbance", IET Control Theory Appl., Vol. 1, No. 6, November (2007)

[3]. Haiwei, Y. and Jinglong H. " Robust flutter analysis of a nonlinear aeroelastic system with parametric uncertainties", Aerospace Science and Technology , Vol 13 ,(2009) pp.139–149

[4]. Melin P.and Castillo O." Adaptive intelligent control of aircraft systems with a hybrid approach combining neural networks, fuzzy logic and fractal theory", Applied Soft Computing vol3, (2003) pp.353–362

[5]. Chen C., Wu J.and Chen J. " Prediction of flutter derivatives by artificial neural networks", Journal of Wind Engineering and Industrial Aerodynamics ,Vol. 96 ,(2008), pp.1925–1937

[6]. Marques F.D,. Gruppioni E. M,and. De Marqui Jr. C. " Active aeroelastic response control of a flexible wing with structural nonlinearity", Brazilian conference on dynamics ,control and application,may,2008

[7]. Tawfik M. A. , Abu-Tabikh,M.I and. Abd Al-Amir H. S "Influence of Flap Angle on the Aeroelastic Behavior of Wing-Flap Configuration Using Fully Coupled Structure-Fluid Interaction Model", Eng. & Tech. Journal, Vol. 29, No.7, 2011

[8]. Hodges D. H. "Introduction to structural dynamics and aeroelasticity" Cambridge University press, 2002.

[9]. Giles, Gary L., "Equivalent Plate Modeling for Conceptual Design of Aircraft Wing Structures," AIAA-1995-3945, Presented at 1st AIAA Aircraft Engineering, Technology and Operations Congress, Sept. 1995.

[10]. Abbas1 L K, Chen1 Q. and Milanese A. " Non-linear aeroelastic investigations of store(s)-induced limit cycle oscillations", J. Aerospace Engineering Vol. 222 Part G (2008).

[11]. Pham D. T. and Xing L. " Neural Networks for Identification, Prediction and Control", Springer, (1995).

[12]. Omatu, S. M. Khalid, and R. Yusof "Neuro-Control and its Applications" London: Springer-Velag, 1995.

[13]. Zurada J. M., Introduction to Artificial Neural Systems. Jaico Publishing House, Pws Pub Co. (1992).

[14]. Narendra, K. S. and K. parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Trans. Neural Networks, vol. 1,pp. 4-27, 1990.

[15]. Narendra, K. S. and K. parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," IEEE Trans. Neural Networks, vol. 2 No. 2, pp. 252-262, 1991.