# USE OF NEURAL NETWORKS FOR MANOEUVRE LOAD CONTROL

D McLean
Department of Aeronautics and Astronautics, University of Southampton
Southampton SO17 1BJ, England

## Abstract

The use of neural network controllers to provide structural load alleviation for a large transport aircraft which exhibits a significant number of bending modes is a new application. In this paper some details of a particular model of the aircraft dynamics are given followed by information relating to a linear, continuous optimal LQR controller (the baseline controller ). The results obtained from digital simulations of the optimal manoeuvre load alleviation control system then follow. These results and many others obtained from the same simulation were used as the training data for the neural network controllers used in the design. A particular type of neural network was studied and information about its corresponding training performance and control effectiveness is presented. The dynamic performance achieved by using the neural controller is compared with that obtained using the baseline LQR controller. The paper concludes with a number of suggestions in respect of neural networks as controllers in a manoeuvre load control system.

## Introduction

Structural load alleviation is one of the chief flight control modes of what has been called "Active Control Technology". Such technology allows aircraft to be more efficient by permitting them to be lighter in structure, yet still capable of sustaining the high loads experienced in operational flight.

The usual method of manoeuvre load control (MLC) is to sense at appropriate points on the fuselage and wing the displacements and rates associated with any bending modes and to use these values as feedback signals in a control system which has been designed to drive the available control surface actuators to reduce bending effects.

The use of continuous, or discrete, feedback control systems designed by means of methods such as $H_\infty$, linear quadratic regulator (LQR), or other quantitative methods to produce an effective MLC, has had only limited success (Harvey and Pope[6], McLean and Prasad [9], Ali[2]).

The application of neural networks to aircraft flight control systems has been limited to somewhat elementary cases (Jorgensen and Schley (8), Ahmed-Zaid et al (1)) , Sadhukhan and Feteih (11). Such applications have not involved aircraft dynamics of much complexity. The work in this paper is intended to demonstrate the effectiveness of neural controllers when used as MLC controllers.

The design of a MLC requires a detailed mathematical model of the aircraft dynamics, including its rigid body motion and the significant flexibility modes. It is hoped to show, using the results of some digital simulations, that the dynamic performance achieved by using neural networks is as good as the performance achieved by a base-line controller.

## Aircraft Dynamics

The mathematical model of the flexible transport aircraft used in this research is given in detail in McLean and Prasad (9). It was used in this work to carry out initial simulation studies: a form of "test-bed" to generate training data of adequate quality (in terms of frequency and scale content) and in sufficient quantity for the networks to learn.

The subject aircraft chosen for this research work was the Lockheed C-5a Galaxy, a large, four engined, subsonic transport aircraft. Much of the data can be found in Harvey and Pope (6) and McLean and Prasad (9). The data related only to longitudinal motion for a single flight condition.

In McLean and Prasad (9) there are presented six models. Each model had state, control and output vectors; the dimensions of five of these models used for this work are listed in Table 1.

| Name of Model | Dimension of Vectors | | |
|---|---|---|---|
| | State | Control | Output |
| Arne | 79 | 2 | 56 |
| Bach | 42 | 2 | 56 |
| Clementi | 24 | 2 | 38 |
| Handel | 5 | 2 | 5 |
| Vivaldi | 17 | 2 | 28 |

TABLE 1 - Dimensions of Vectors used in Mathematical Models

ARNE was the greatest, being 79. Besides including the first fifteen bending modes, ARNE includes both the Küssner and Wagner lift growth functions and the time delays involved before the wing and tail encounter any gust. Handel represented only rigid-body motion and actuator dynamics.

For this work the bending and torsional moments of five separate stations on the wing, denoted as WS1 to WS5, were included in the output vector. The locations were chosen such that WS1 is at the root of the wing, WS3 is at mid-span and WS5 is at the wing tip. WS2 and WS4 are located equidistant between the wing root and mid-span and between mid-span and the wing tip respectively.

## Baseline Controllers

Designing and training neural network control schemes suitable for use in a MLC requires that there are available sets of representative training data and also a baseline controller against which to measure the performance of the neural control. Linear optimal state feedback and output feedback controllers were used for these tasks. A block diagram of the MLC is shown in Figure 1. BACH was used to obtain full state feedback controller designs. Several test situations were devised in which both deterministic and stochastic excitation signals were applied. Table 2 lists the deterministic test signals used to generate transient responses which were later used to provide initial training strategies for the neural controllers.

In those test cases where turbulence was introduced the assessment of the effectiveness of the control achieved in turbulence was carried out by calculating the rms values of the bending moments .

## Uncontrolled Aircraft Response

The first assessment of the aircraft response was made on the basis of the uncontrolled aircraft ie open-loop system. Responses were obtained by using the mathematical model BACH when subjected to test cases ALPHA or BETA . Some of these responses are shown in Figures 2 - 4. The natural frequency of the short-period rigid body mode is only separated from the fundamental bending mode by a factor of 4. Bending mode number 2 was not subsequently much affected by control action.

| Test Case | Initial Values of State Variables | Commanded Aileron Deflection $\delta_{A_e}$ | Commanded Inboard Elevator Deflection $\delta_{Ei_c}$ |
|---|---|---|---|
| ALPHA | All zero except $w(t_0) = 1.0$ | 0 | 0 |
| BETA | All zero | 1.0 | 0 |
| GAMMA | All zero | 0.0 | 1.0 |

TABLE 2 Training Data Test Signals

## Controlled Aircraft

Full State Feedback In Figures 5 and 6 are shown some typical bending and torsional moment responses obtained with the use of optimal full state feedback controllers. The selection of the weighting matrices used in the design of the optimal LQR controllers was not unique ie several combinations of different weighting matrices could give rise to near identical control laws. Some methods which can assist the selection process have been proposed (Bryson and Ho (3), Johnson (7) It was found to be as effective and quick to make the choices empirically, based on observation of the responses.

Wing Moments It was decided to use the bending moment responses occurring at the wing root (WS1) and mid -span (WS3) as measures of the SLACS performance. It was found that bending moments could be substantially reduced by means of the action of the MLC. A reduction in any oscillatory motion of the bending moment responses to reduce the cumulative fatigue of the wing was also desirable, but it usually required several attempts at controller design to achieve this.

Figures 5 and 6 indicate how much alleviation of both bending and torsional moments can be achieved with good controllers. Responses obtained from successful controllers were retained as training data for the neural networks. Some systems could achieve good WRBM (Wing Root Bending Moment) reduction but did not reduce WRTM (Wing Root Torsional Moment) and vice versa. This situation is illustrated in Figures 7 and 8. Under certain circumstance it may be necessary to sacrifice some reduction in WRBM to secure some alleviation in the torsional moment at the wing root (WRTM), or even at the wing tip, WS5. The key objective is to reduce the bending moments at the wing tip and then inboard all along the wing without increasing the torsional moment. Figure 9 shows the control surface activity required by the MLC to achieve load alleviation.

## Neural Networks

In this paper the use of the multi-layered perceptron (MLP) for structural load alleviation is considered. The choice of a particular type of NN depends on the nature of the problem being solved, but several issues relating to the number of nodes required, the number of layers to be used, and their inter-connexion are non-trivial. Some rules-of-thumb can help in the initial decisions eg it is known that a network with a single hidden layer can produce the same results as a network with several, although it may require more training. The hidden layer must contain at least as many neurons as there are inputs.

The MLP is a non-linear feed forward network which is usually trained by using an algorithm based upon back-propagation of the errors. The designer can choose the activation function (or the transfer function) of the neural network. Among the most usual choices are sigmoid, hyperbolic tangent and sine. In this work. use of the sigmoid function in the hidden layer of the MLP resulted in poor performance in terms of training time. Both the sine and hyperbolic tangent functions provided much better performance, with the use of the sine function resulting in the lowest training times. With the sine activation function the aircraft dynamics containing the most complete description of the flexibility effects provided the best results. However, when using different data sets, the difference in performance from the other two activation functions was not marked.

A number of tests were used to ensure that the data obtained from the simulation were representative. Responses to gust and deterministic inputs, as well as responses to initial conditions, were used in addition to those obtained when the aircraft dynamics were augmented by unsteady aerodynamics and excited by simulated atmospheric gust disturbances. The need for many different test-situations is reinforced by the requirement for assessment of how well the neural network has learned its control strategy from the training data presented. The system adopted for assessing the effectiveness of training networks - was to test the network's capacity for reproducing the training data presented and then to adjust to a new set of training data.

The available sets of data were split in a random way into training and test data sets. A further source of training data was obtained from the aircraft simulation using responses to excitation inputs not used earlier. Thus, gusts with different levels of rms intensity were used to excite the system and responses obtained from the neurally-controlled system with training completed using only deterministic data but with the system subjected to gusts, were also studied .

An issue which caused some difficulty was to include/exclude the unsteady aerodynamic effects. Generation of appropriate training data based upon the mathematical model BACH produced redundant data and caused, by its complexity, training to be slow and convergence could not be guaranteed. As a measure of how useful any training data were for training the neural controllers, the condition number of the autocorrelation matrix (based upon the network inputs) was used. The rate of convergence depended upon the smallest non-zero eigenvalue of the autocorrelation matrix, and the parameter chosen for the learning rate of the hidden layer(s).

BP networks suffer from a number of problems which cause difficulty during training. The weight vector can be updated using an algorithm which depends on the learning rate parameter but convergence is usually slow. To speed this up an empirical procedure was used to adaptively adjust the learning rates in the hidden layers: after a specified number of training cycles the learning rate was incremented by a specified amount. Too large increments, however, could result in a total failure to converge, whereas values which were too small caused very slow convergence. Using standard BP, even with special enhancements, sometimes resulted in the network training taking place over several days.

Network paralysis was commonly encountered. When the weights in the hidden layer were too large the output nodes produced large values. When activation functions such as sigmoid or tanh were used, such large outputs caused saturation and the BP error became small causing training to cease prematurely. To check for network paralysis required that the nodal activity and weight changes were checked throughout the training period. When more than four bending modes were included in the mathematical model of the aircraft dynamics network paralysis was avoided.

Sometimes learning interference occurred between the network weights which had already been trained, and those weights which remained only partially trained , for then the rms output error from the network was back-propagated to cause those areas of the network which were already partially trained to become untrained as a consequence of the network's efforts to train areas which were initially untrained. Wherever this happened the training times for dynamical models with more than four bending modes (ie the first four) increased by about four times. A practical solution was achieved by using a series of training experiments referred to here as "sequential build (SB)". These experiments attempted to decouple the control problem involving the control of rigid and flexible modes by using with the rigid body modes a deterministic LQR controller and with the bending modes a neural controller.

## Improvement of MLP Learning Times

Several techniques were studied as a means of speeding up the MLP learning times.

### Momentum Term

This term modified the extent to which the old values of the network weights were changed. Usually the value of this momentum term lay between 0.1 and 1.0; when models of low order were used, it was found that a value of 0.7 was satisfactory. However, for models of greater order, the appropriate value was less and was determined by experiment.

### Cumulative Update of Weight Vector

This involves updating the network weights only after a fixed number of training data sets have been presented to the network. By using the cumulative error over the epoch a more effective updating of the network weights was established. The epoch size lay between 16 - 70 but this approach required more computing.

### Fast B.P. (Samad [10])

This involved the error at the (s - 1)th layer being added to the output before the connection weight was established. An amplification factor, between 0.5 and 2.0,was applied, but, experiments did not show any appreciable improvement in learning times.

### BIAS Term

A bias term allows the origin of the activation function to be offset, thereby sometimes providing more rapid convergence. When models involving only the first 2 - 4 bending modes were considered, the neural networks, using bias terms, achieved significant improvements in training times. The same result was noted when the complete model, using all fifteen bending modes, was employed. The bias firing value was reduced from 1.0 to 0.786 when the aircraft dynamics involving the first fifteen modes were considered.

Quickprop (Fahlman's Methods) (5).

These involved the following three heuristic techniques.

Biasing the Derivative of the Activation Function Normally this was considered only when the activation function used was the sigmoid. It was then referred to as the F' offset, viz F'. The advantage of using this was that learning occurred at all times: the technique prevented the occurrence of saturation for the aircraft mathematical model, BACH. When an $F^1$ offset of 0.279 was used with a hidden layer of 15 nodes, with a sine activation function,a considerable improvement in the training times was observed.

Changed Error Function in the Output Layer When an hyperbolic tangent (tanh) function was used as the activation function for the output layer, the large errors in the network were further amplified. Using a cubic error function as the activation function in the output layer reduced training times by almost an order of magnitude.

Self-Adaptive Learning Rate for Hidden Layer DBD (Adaptive Learning and Momentum Rate) networks increase their learning rate linearly, but the weight decrease is geometric, to prevent the learning rate from decreasing too quickly and to ensure that, in regions of the weight space where the error curve is greatest, the connection learning rate is rapidly decreased. A parameter, λ, was used to control the network recovery phase such that, if the error vector after the $i^{th}$ training epoch is denoted $E_i$, then

$$E_{i_t} > \lambda E_{i_{t-1}}$$

All the connection weights reverted to the best weights of the preceding epoch which were retained. The learning and momentum rates were also increased. This technique provided better training and performance when used with the model BACH than did the MLP with BP.

## Network Training Features

A beneficial feature for network training was found to be the technique of separating the actuator dynamics from the rigid body and flexible dynamics, and controlling them separately. For realism, rate and displacement limits were placed on the appropriate control surface, as shown in Table 3.

| Control Surface | Deflection Limit (degrees) | | Rate Limit deg/s |
|---|---|---|---|
| | Max | Min | |
| Ailerons | 14.5 | 14.5 | 25 |
| Inboard Elevators | 20 | 30 | 15 |

TABLE 3 - Control Surface Deflection and Rate Limits

Networks trained on data obtained from the responses to deterministic inputs performed better when the system was tested with simulated atmospheric turbulence than networks trained using response data obtained from turbulence input tests. The resulting controllers, however trained, were almost as robust as the baseline controllers from which the training data were obtained.

As more bending modes were included in the model of the aircraft dynamics the training time and the number of nodes which were required to be used in the hidden layer(s) increased. The effect of including the Küssner dynamics in the mathematical model was not very great in terms of the observed closed-loop performance, although the increased complexity of the associated mathematical model representing the aircraft dynamics resulted in greatly complicating the MLP. Attempts at retaining the complete set of inputs to the network, but with a restricted number of nodes in the hidden layer, did not result in satisfactory control. It was found to be difficult to design well-trained MLP networks for large systems. To overcome this difficulty a series of smaller sub-models were constructed and used sequentially to build a larger network.

## Sequential Build (SB) Networks

These were obtained by finding several LQR baseline controllers, based upon the mathematical model BACH, and collecting the responses to several types of input from the SLACS which used these LQR controllers. Data relating to the flexibility effects ie the bending mode displacements, the associated rates, and the moments (both bending and torsional) at the five designated wing stations, were collected separately. A feedback control system,using only the rigid body motion variables, and the corresponding feedback gains of an LQR controller, was implemented with a multi-purpose neural network block in parallel. This NN block was trained using the data based on flexibility effects. Collecting data relating to bending displacements and rates of individual bending modes for training a NN to control just that mode was also tried. It was found that, in general, using such a sequential build (SB) technique was more efficient in terms of training time: the smaller networks were easier to train. Moreover, it was possible to control the rigid body motion independently: it was then easier to evaluate the load alleviation performance of particular neural controllers, for the effect of that controller could be compared directly with the alleviation achieved by the baseline LQR controllers which had already provided the training data. It was found to be important to ensure that the response data used for training truly represented all the mode contributions and that the corresponding mode variables were satisfactory from a scale and frequency viewpoint. Of course, such response data could be poorly conditioned, with a bad effect on the network error, causing it to converge to its minimum only very slowly, if at all. Data scaling can be effective in overcoming this problem.

Initially each mode was controlled by a separate neural controller, but, by experiment, it was established that modes could be gathered into controllable groups,allowing the system to be re-modelled , further training data collected, and larger neural networks developed. As a check on the appropriateness of including or excluding a particular mode in a cluster, the condition number of the scaled data matrix was examined. If it was too large,either a mode was removed, or fewer inputs were used. In Table 4 are presented some data relating to mode 4 which proved to be particularly troublesome. Whenever mode 4 was included in a cluster the condition number of the data matrix and consequently, the network training time, increased. The solution was to exclude mode 4 from a cluster and to use another neural controller for just that single mode.

| Cluster of Modes | Condition No of Data Matrix | No of Network Inputs | No of Network Outputs |
|---|---|---|---|
| 3, 4, 5, 6 | 2414 | 8 | 2 |
| 3, 4, 5 | $9.53 \times 10^{15}$ | 6 | 2 |
| 3, 4 | $9.5 \times 10^{15}$ | 4 | 2 |
| 3 | 2.24 | 2 | 2 |
| 3, 5 | 34.79 | 4 | 2 |

TABLE 4 - Sequential build; Variation of condition number with Mode 4 included or excluded

Simply neglecting a mode from the control scheme is inappropriate, for modal interaction can give rise to effects which would be unobserved by the control scheme. Modal interaction can be significant, too, in relation to the training data. If modes are clustered and the response data re-obtained for particular types of excitation inputs, then the condition number of the autocorrelation matrix can increase significantly if there is modal interaction within the cluster. This feature can be demonstrated by means of Table 5.

One interesting result of the study using the sequential build technique related to whether it was significant if higher bending modes were neglected when using neural control: the steady-state values of the bending and torsional moments were almost the same whether the full number of bending modes were included or not. See Figure 10. When training the associated neural network it was necessary to ensure that sufficient number of nodes were used in the hidden layer, otherwise the quality of learning was much impaired. An illustration of using too few nodes in the hidden layer of a neural network which had been trained on data relating to a controlled aircraft with only a few lower bending modes is shown in Figure 11. The responses shown were obtained from using the neural controller with a simulation based upon CLEMENTI. When the number of nodes in the hidden layer was increased the response improved, but the training took longer. The

| Cluster of Modes | Condition No of Data Matrix | No of Network Inputs | No of Network Outputs | Input Excitation Type |
|---|---|---|---|---|
| 5, 6 | 5.3505 | 4 | 2 | square wave |
| 5, 6 | 719 | 4 | 2 | sine wave |
| 5 | 3.1987 | 2 | 2 | sine wave |
| 6 | 5.2141 | 2 | 2 | sine wave |

TABLE 5 - Sequential build; Variation of condition number with different system inputs

response produced was less rapid than the corresponding response provided by the parent LQR controller. It is believed that the neural controller was driving the control surface actuators at rates close to the frequencies of the lower bending modes which are those with the greatest energy. These modes were excited, therefore, causing increased surface activity - a situation which could result in instability, although this possibility could not be inferred from the training or validation data. Large scale MLP networks, with back-propagation, seem to be inappropriate for on-line, real-time learning on account of the large number of computations required per training cycle and the large number of iterations needed before convergence. By adopting the technique of updating in real-time the weights in only those networks which require updating, the SB method overcame the problem of learning interference in large MLP networks. However, the technique of SB could not overcome two other problems related to the use of the MLP with BP viz network paralysis and an inability to produce weights incrementally ie if there is a finite time for training and reasonable convergence can be achieved, then all the inputs have to be seen before any weight change can take place.

Figure 12 shows the responses of the system when a neural controller was employed which controlled only the rigid body and first bending mode; Figure 13 shows the responses when the controller controls the rigid-body, and first and second bending modes. These indicate small reductions compared to the LQR control in both wing root bending moment and wing root torsion moment and in all the moments up to wing station 4. The bending moment at the wing tip is lower, however, than the corresponding LQR controller. Whenever neural control was used for the purpose of controlling just the first bending mode, it was observed that modal interaction, particularly at WS3 and 5, occurred and was especially noticeable in the wing torsional moment.

## Conclusions

The use of neural controllers has been shown to be effective for use in MLC. Individual networks can be made to learn to optimize the control of particular bending or torsional moments by changing the response of individual bending modes,

The choice of LQR-based responses for training data was influenced by two factors: LQR-based MLC had already been shown to be particularly effective; and the guaranteed closed-loop stability of the LQR controlled systems meant that neural networks learning such mappings initially would also possess this desirable property.

The non-linearities which existed in the simulation related to the control surfaces deflection and rate limits, but the principal advantages expected from applying neural control to linear dynamics were the expected lower-order controller capable of providing acceptable dynamic performance even when there were significant errors in the mathematical model and, producing stability and robustness.

The use of smaller sub-networks applied to individual (or selected clusters of) bending modes has been proposed to overcome the difficulties observed in the early attempts to train MLP based networks using a large number of inputs. Difficulties attending the use of large MLP manifested themselves in increased training times caused by the severity of the phenomenon of learning interference.

Consequently,it was decided to sub-divide the large MLP network into small sub-networks which could control specific bending modes individually. Such smaller networks facilitated the problems when studying the bending behaviour along the wing stations.

Neural control produced dynamic load alleviation equal to that produced by the LQR controller from which the training data were obtained; its use did not significantly improve upon that performance, and required substantial amounts of training data and time.

## Acknowledgements

The generous support of the Engineering and Physical Sciences Research Council of UK under Reserach Grant GR/G52340, is gratefully acknowledged.

## References

1. Ahmed-Zaid f., Ioannou P.A., Polycarpou M.M., & Youssef H.M. Identification and Control of Aircraft Dynamics using Radial Basis Function. *Proc 2nd IEEE Conf. Cont. appl. 1993,* 567-72, Vancouver, BC, Canada.

2. Ali Q.R. Aircraft Load Alleviation by Specifying its closed Loop eignestructurel *PhD thesis,* 1987, Loughborough University, England.

3. Bryson A.E. and Ho Y.C. Applied Optimal Control. *J. Wiley and Son,* 1975, New York.

4. Chalk C.R. et al. Background Information and User's Guide for MIL-8785B(ASD): Military Specification - Flying Qualities of Piloted Aircraft. AFFDL-TR-69-72, August 1969.

5. Fahlman S.E. An Empirical Study of Learning Speed in Back-Propagation Networks. *CMU Technical Report*, CMU-CS-88-162, June 1988.

6. Harvey C.A. and Pope R.E. Study of the Synthesis Techniques for Insensitive Aircraft Control Systems. *Nasa CR 2803,* April 1977.

7. Johnson S.H. Aircarft Model Protoypes which have Specified Handling Qualities Time Histories. *NASA CR-142848,* 1978.

8. Jorgenson C.C. and Schley C. A Neural Network Baseline Problem for Control of Aircraft Flare and Touchdown. *Neural Network for Control, Ed W.T. Miller, Sutton R.S. and R.S. Werbos.* MCT Press, 1990, 401-425, Mass. USA.

9. McLean D. and Prasad R.A. A structural Load Alleviation Control System for a Large Transport Aircraft. *TT 8002,* July 1980, Loughborough University, England.

10. Samad T. Back-Propagation Extensions. *Honeywell SSDC Technical Report,* Golden Valley, MN 55427, USA. 1989.

11. Sadhukhan D. and Feteih S. F8 Neurocontroller Based on Dynamic Inversion. *J.Guid.Cont. Dyn,* 1996, 19(1), 150-55.

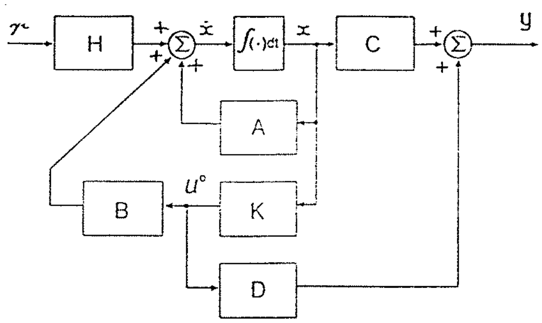



**Figure 1. Structural Load Alleviation Control System**

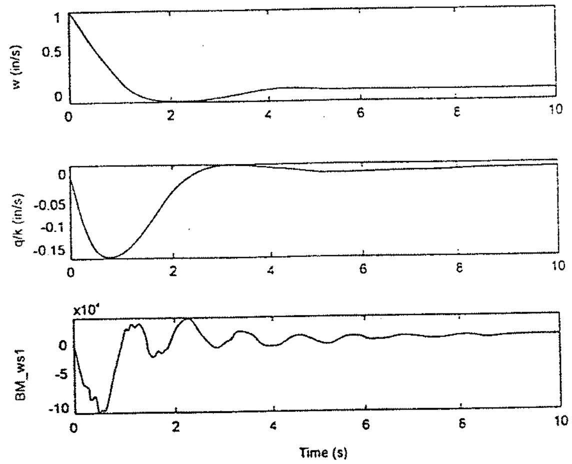



**Figure 2. Uncontrolled Aircraft Response**

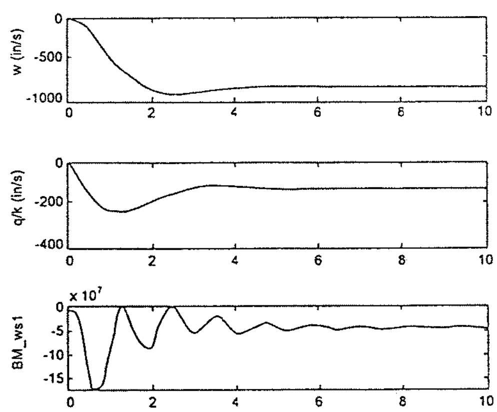



**Figure 3. Uncontrolled Aircraft Response (model BACH, test BETA)**

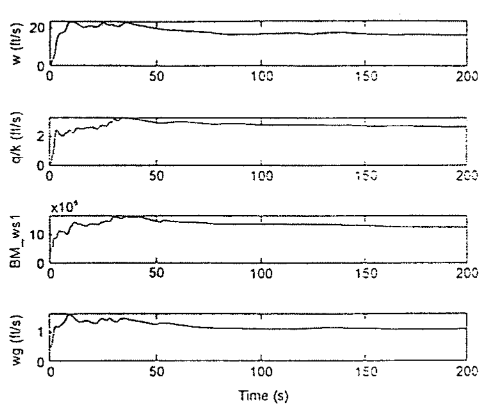



Figure 4. Uncontrolled Aircraft R.M.S. Response (model BACH, test DELTA)

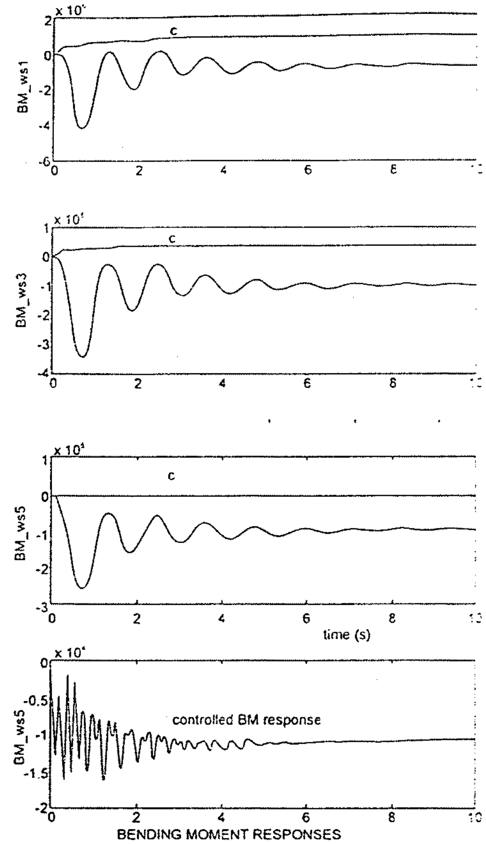



**Figure 5. Wing Bending Moment Responses c denotes controlled response**

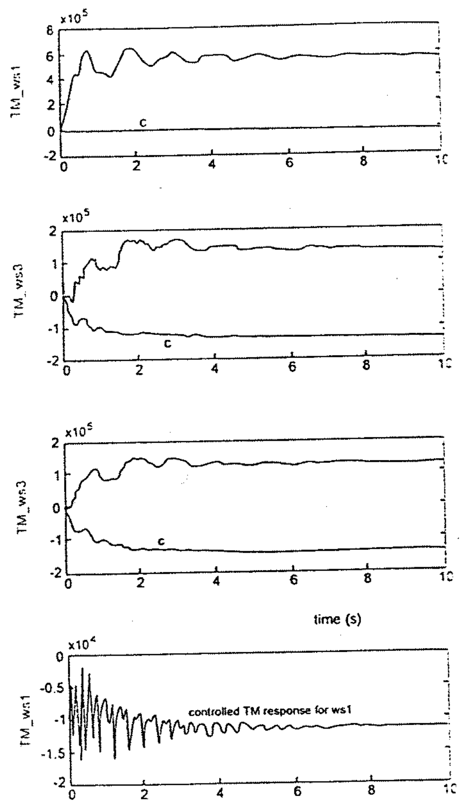



**Figure 6. Wing Torsional Moment Responses. c denotes controlled response**

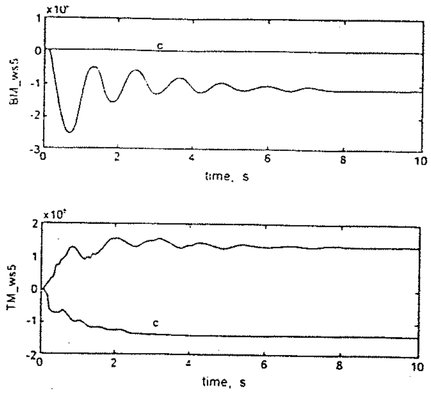



Figure 7. Good Wing Root Bending Moment Reduction

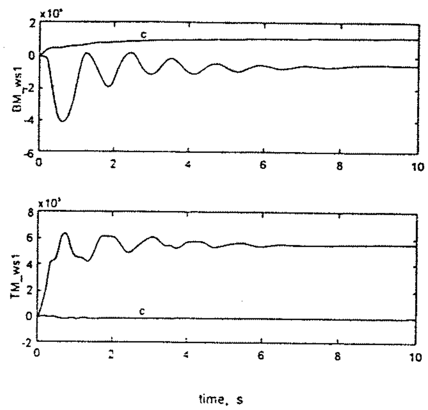



Figure 8. Good Wing Root Torsional Moment Reduction

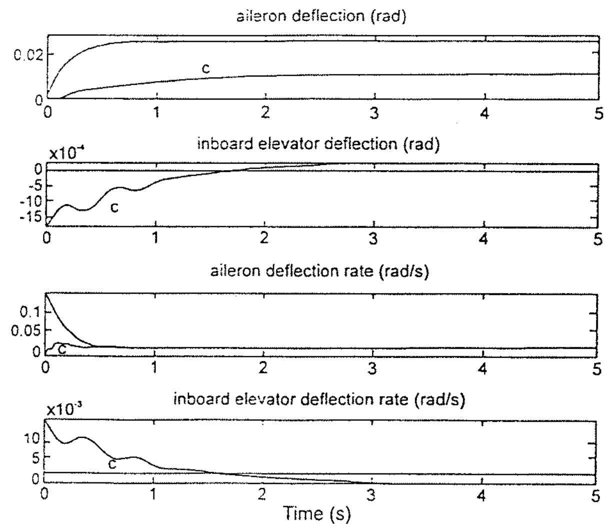
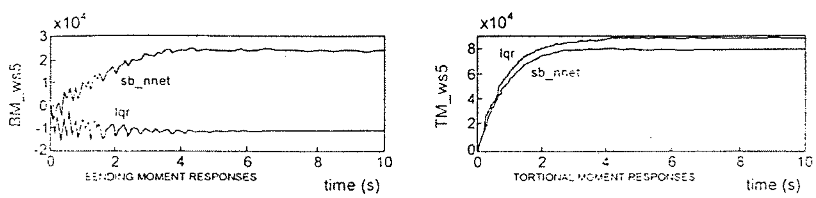
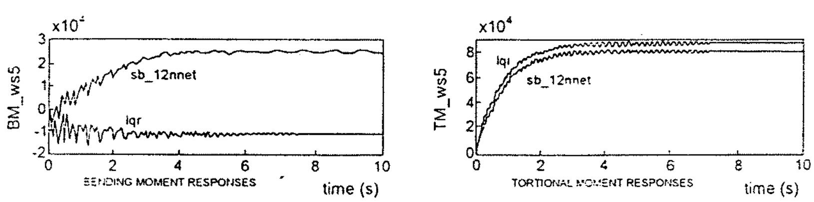



Figure 9 Control Surface Activity (model BACH, test BETA)
c denotes controlled response




S.B. NN Controller (Controlling all modes)




S.B. NN Controller (Controlling 1st & 2nd modes)

Figure 10. Moment Responses for Different S.B. NN Controllers

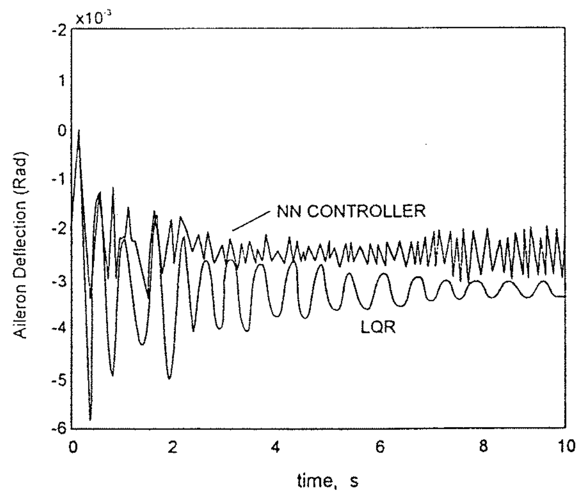



Figure 11. Example of insufficient nodes in the hiddenlayer to learn 15 mode model

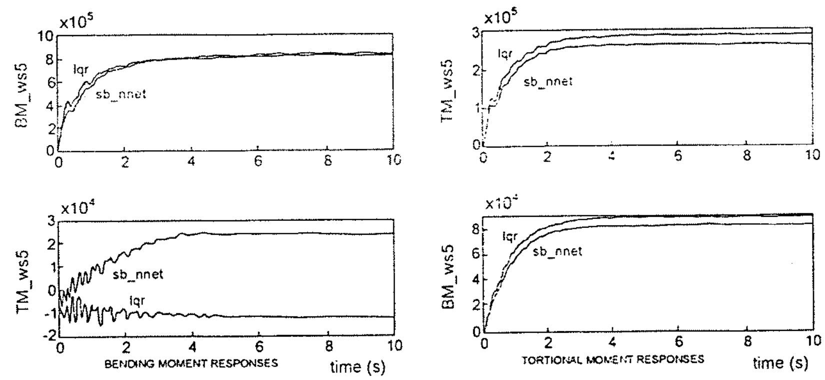



Figure 12. S.B. NN Controller - Rigid and 1st Bending Mode

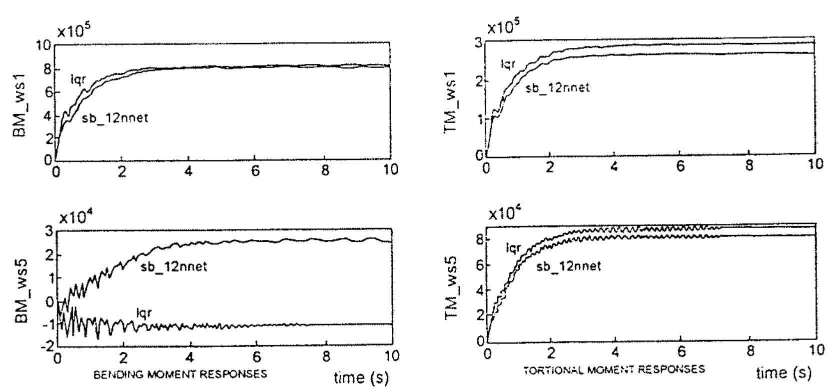



Figure 13. S.B. NN Controller - Rigid, 1st and 2nd Bending Modes