Journal of Sound and Vibration (2001) **242**(1), 47–58
doi:10.1006/jsvi.2000.3348, available online at http://www.idealibrary.com on IDEAL®

IDENTIFICATION OF RESTORING FORCES IN NON-LINEAR VIBRATION SYSTEMS USING FUZZY ADAPTIVE NEURAL NETWORKS

Y. C. LIANG AND D. P. FENG

Department of Mathematics, Jilin University, Changchun, 130012, *People's Republic of China. E-mail: ycliang@public.cc.il.cn*

AND

J. E. COOPER

*Manchester School of Engineering, University of Manchester, Manchester, M*13 9*PL, U.K. E-mail: jecooper@fs*1*.eng.man.ac.uk*

(*Received* 1 *June* 1999, *and in final form* 14 *August* 2000)

The fuzzy adaptive back-propagation (FABP) algorithm which combines fuzzy theory with artificial neural network techniques is applied to the identification of restoring forces in non-linear vibration systems. Simulated results show that the FABP algorithm is effective for the identification of dynamic systems. The FABP algorithm not only increases the training speed of the network, but also decreases the artificial interference of network parameters to a certain extent. Based upon the FABP algorithm, an improved scheme with a mutation mechanism is presented in this paper. The improved fuzzy adaptive BP (IFABP) algorithm extends the effectiveness and adaptivity of the FABP algorithm still further. The successful estimation of simulated systems show that a feasible method of identification is provided, which can be used to estimate the restoring forces in non-linear vibrating systems quickly and effectively.

1. INTRODUCTION

Model identification of dynamic systems in the vibration engineering field has been followed with interest in recent years. A number of identification techniques on this topic are now available, such as parametric or non-parametric identification methods, time domain or frequency domain estimation approaches, etc. But there exist some unavoidable limitations in most of the methods, including that *a priori* information about the system under investigation is required, that the properties of the identified system is constrained and that the nature of the excitation source to be used is restricted, and so on. Therefore, novel models and methods need to be introduced to improve the estimation of dynamic systems.

It is well known that artificial neural network models have several inherent properties which distinguish them from traditional computational models, such as parallel architectures and computation, higher degree of robustness or fault tolerance, and the property of adaptation or learning, etc. The most outstanding characteristics of the neural network aided computation is that neither complicated programming nor rigid algorithms are needed. These properties make neural network methods the ideal choice in cases where

0022-460X/01/160047 + 12 $35.00/0

real-time adaptation and fast processing of large amounts of data are required. For this reason, a lot of attention has been paid to neural networks for system identification in the fields of vibration engineering and computational mechanics [1–15].

Masri *et al.* explored a procedure based on neural networks for the identification of non-linear dynamic systems [1]. It was a successful attempt to employ neural network techniques to physical systems in the applied mechanics field. Because of the abilities of learning and generalization of neural networks, no *a priori* information about the system under investigation is required and the nature of the excitation source to be used is not restricted in the procedure. The approach can be used effectively for the identification of the restoring forces of some typical non-linear structural systems, but it only dealt with the identification of single-degree-of-freedom systems in reference [1]. References [2, 3] extend the procedure to multi-degree-of-freedom non-linear vibration systems, and make it applicable in a wider range. In most current studies, it is usual to employ the conventional back-propagation (BP) algorithm to train the neural networks as was implemented in references [1–3]. Owing to the inherent shortcoming of requiring a longer training time and having difficulties in selecting the training parameters in the BP algorithm some limitations are unavoidable in the real applications of neural network methods. Liang *et al.* [4, 5] applied a fuzzy adaptive BP (FABP) algorithm which combines the fuzzy theory with the structural neural network techniques to the identification of non-linear characteristics in cushioning liners and attained the goal of increasing the training speed of the network.

This paper applies the FABP algorithm to the identification of restoring forces in multi-degree-of-freedom non-linear vibration systems and increases the training speed of the network to a certain extent. In order to increase the efficiency of the FABP algorithm still further this study makes some improvement to the algorithm. On the one hand, the mutation mechanism in the evolutionary computation is introduced into the FABP algorithm, which increases the training speed of the network still further, on the other, the artificial interference to some parameters of the network is decreased, which increases the adaptivity of the algorithm still further. The aim of the proposed method is to establish an equivalent neural network model for a non-linear system rather than identify any "parameters" of the physical systems. The model is validated by matching the projected output. Simulated results show that it is much more efficient to apply the improved fuzzy adaptive BP (IFABP) algorithm to the identification of restoring forces in non-linear vibration systems. The IFABP algorithm is also more applicable in practice because of its characteristics of using less artificial interference in the processing operation.

2. IDENTIFIED MODEL

A general mechanical structure can be discretized into a lumped parameter, n-degree-of-freedom system. The equations of motion of the system can be written as

$$\begin{cases} m_1\ddot{y}_1 + g_1(y_1, \dot{y}_1, \ldots, y_n, \dot{y}_n) = F_1(t), \\ m_2\ddot{y}_2 + g_2(y_1, \dot{y}_1, \ldots, y_n, \dot{y}_n) = F_2(t), \\ \quad\vdots \\ m_n\ddot{y}_n + g_n(y_1, \dot{y}_1, \ldots, y_n, \dot{y}_n) = F_n(t), \end{cases} \tag{1}$$

where $g_i(y_1, \dot{y}_1, \ldots, y_n, \dot{y}_n)$ and $F_i(t)$ represent the linear and non-linear restoring force and the input excitation acting at the mass m_i respectively.

It is assumed that the excitations $F_i(t)$ and the accelerations $\ddot{y}_i(t)$ $(i = 1, 2, \ldots, n)$ of the system are available from measurements, and that the mass m_i $(i = 1, 2, \ldots, n)$ are known or easily estimated. The non-linear characteristics of the system and the restoring forces $g_i(y_1, \dot{y}_1, \ldots, y_n, \dot{y}_n)$ $(i = 1, 2, \ldots, n)$ acting on the system are unknown. The purpose of the paper is to identify the restoring forces, which are the functions of the displacements and the velocities, using neural network methods.

From equation (1) the restoring forces can be written as

$$g_i(y_1, \dot{y}_1, \ldots, y_n, \dot{y}_n) = F_i(t) - m_i\ddot{y}_i(t) \quad (i = 1, 2, \ldots, n). \tag{2}$$

The estimation procedure described in this paper requires the velocity and displacement responses simultaneously at each response location. The displacements and the velocities of the system can be found by direct measurement, or through integration of $\ddot{y}_i(t)$ [3]. If the displacement, velocity, acceleration and the input excitation signals are taken at discrete times t_k,

$$y_{ik} = y_i(t_k), \quad \dot{y}_{ik} = \dot{y}_i(t_k), \quad \ddot{y}_{ik} = \ddot{y}_i(t_k), \quad F_{ik} = F_i(t_k) \quad (i = 1, 2, \ldots, n), \tag{3}$$

then the values of the resting forces at times t_k are

$$g_{ik} = F_{ik} - m_i\ddot{y}_{ik} \quad (i = 1, 2, \ldots, n). \tag{4}$$

3. REVIEW OF CONVENTIONAL AND FUZZY ADAPTIVE BP ALGORITHM

A three-layer feedforward neural network is used in this paper to illustrate the algorithms. The inputs to the net are the measured or calculated displacements and velocities $y_1(t), \dot{y}_1(t), \ldots, y_n(t), \dot{y}_n(t)$. The network outputs $g'_i(y_1, \dot{y}_1, \ldots, y_n, \dot{y}_n)$ $(i = 1, 2, \ldots, n)$ are the unknown restoring forces. The network topology is represented by the weight matrices $[W^i]$ and the threshold vectors $\{\theta^i\}$ $(i = 1, 2)$.

Let

$$\{y\} = \{y_1, \dot{y}_1, \ldots, y_n, \dot{y}_n\}^{\mathrm{T}} \tag{5}$$

be the input vector to the net,

$$\{g'\} = \{g'_1, g'_2, \ldots, g'_n\}^{\mathrm{T}} \tag{6}$$

be the output vector to the net and

$$f(x) = \tanh(\beta x) \quad (\beta > 0) \tag{7}$$

be a non-linear *activation* function.

The outputs of the network are computed according to the following equations:

$$\begin{aligned} \{\bar{v}\} &= [w^1]\{y\} + \{\theta^1\}, \qquad v_j = f(\bar{v}_j) \quad (j = 1, 2, \ldots, l), \\ \{\bar{g}\} &= [w^2]\{v\} + \{\theta^2\}, \qquad g'_j = f(\bar{g}_j) \quad (j = 1, 2, \ldots, n), \end{aligned} \tag{8}$$

where l is the number of the neurons in the hidden layer.

The identification approach consists of two phases: the network training (or learning) phase and the validation phase. During the training phase, the network is presented with the sequence of input vectors $\{y_{1k}, \dot{y}_{1k}, \ldots, y_{nk}, \dot{y}_{nk}\}^{\mathrm{T}}$ and the sequence of desired output vectors $\{g_{1k}, g_{2k}, \ldots, g_{nk}\}^{\mathrm{T}}$. Given a set of weights and threshold (which initially is chosen randomly), the input vector is propagated forward through the net and the network output $\{g'_{1k}, g'_{2k}, \ldots, g'_{nk}\}^{\mathrm{T}}$ is calculated according to equations (8).

The error between the actual system output and the desired output is defined as

$$E = \frac{1}{2} \sum_{k=1}^{p} \sum_{i=1}^{n} (g_{ik} - g'_{ik})^2, \tag{9}$$

where p is the number of patterns in the training set.

The purpose of the training phase is to adjust the weights and thresholds w^1_{ij}, w^2_{jk}, θ^1_j, θ^2_k ($i = 1, 2, \ldots, 2n; j = 1, 2, \ldots, l; k = 1, 2, \ldots, n$) which are the elements of $[w^1]$, $[w^2]$, $\{\theta^1\}$ and $\{\theta^2\}$, respectively, in the direction that will reduce the error. The training is performed by the back-propagation (BP) algorithm [15]. According to the conventional BP algorithm, the modified formulas of the network parameters are as follows:

$$\begin{cases} w^2_{ij}(t+1) = w^2_{ij}(t) - \eta \dfrac{\partial E(t+1)}{\partial w^2_{ij}} + \alpha \Delta w^2_{ij}, \\[2ex] \theta^2_j(t+1) = \theta^2_j(t) - \eta \dfrac{\partial E(t+1)}{\partial \theta^2_j} + \alpha \Delta \theta^2_j, \\[2ex] w^1_{jk}(t+1) = w^1_{jk}(t) - \eta \dfrac{\partial E(t+1)}{\partial w^1_{jk}} + \alpha \Delta w^1_{jk}, \\[2ex] \theta^1_k(t+1) = \theta^1_k(t) - \eta \dfrac{\partial E(t+1)}{\partial \theta^1_k} + \alpha \Delta \theta^1_k, \end{cases} \tag{10}$$

where η is the learning rate, α is the momentum factor, and Δw^2_{ij}, $\Delta\theta^2_j$, Δw^1_{jk} and $\Delta\theta^1_k$ are the increments of connection weights and thresholds respectively.

During the validation phase, the network is given other input vector sequences $\{y_{1\alpha}, \dot{y}_{1\alpha}, \ldots, y_{n\alpha}, \dot{y}_{n\alpha}\}^{\mathrm{T}}$ not among those used for training. If the training was successful and the network is a good identifier, it should produce an output sequence $\{g'_{1\alpha}, \ldots, g'_{n\alpha}\}^{\mathrm{T}}$ very close to the actual system output

$$\{g_{1\alpha}, \ldots, g_{n\alpha}\}^{\mathrm{T}} = \{g_1(y_{1\alpha}, \ldots, \dot{y}_{n\alpha}), \ldots, g_n(y_{1\alpha}, \ldots, \dot{y}_{n\alpha})\}^{\mathrm{T}}. \tag{11}$$

References [1–3] applied the conventional BP algorithm to the identification of restoring forces in non-linear vibration systems successfully. In general, however, the speed of convergence of the BP algorithm for training the feedforward multilayered neural networks is slower. Therefore, the BP algorithm needs to be improved in order to speed up the convergence. The learning rate η and the momentum factor α are the main parameters affecting the speed of convergence in the BP algorithm. References [4, 5] applied the FABP algorithm to the identification of non-linear characteristics in cushioning liners, realized the adaptive adjustment of the learning rate η and the momentum factor α, and attained the goal of increasing the training speed of the network.

In the fuzzy adaptive BP algorithm, the error function changes with the iteration number. The input variables of the fuzzy controller are defined as

$$CE(t+1) = CT \, \frac{E(t) - E(t+1)}{\max\{E(t), E(t+1)\}}, \tag{12}$$

$$CCE(t+1) = CCT\,[CE(t+1) - CE(t)], \tag{13}$$

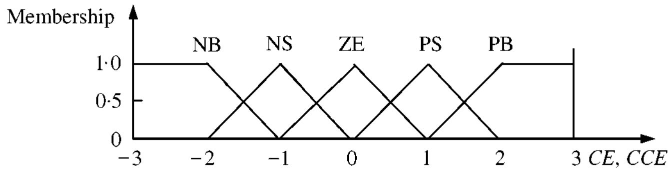

Figure 1. Membership function curves of fuzzy subsets for CE and CCE.

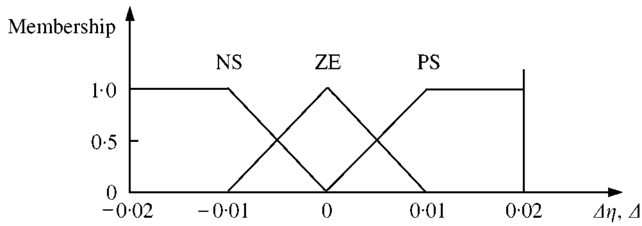

Figure 2. Membership function curves of fuzzy subsets for $\Delta\eta$ and $\Delta\alpha$.

respectively, where CT and CCT are scale factors which depend upon the changing condition of the error curves. The determination of the input variables of the fuzzy controller is based on the following considerations: (1) the value of the absolute changing rate of E is fairly small, therefore, the relative changing rate of E is fairly small, therefore, the relative changing rate of E is given by $CE(n+1)$; (2) the increase of CE indicates the decrease of the error E according to the definition of CE in equation (12), therefore, the definition of CE makes it easier to determine the control rules.

In the fuzzy BP algorithm the input variables of the fuzzy controller CE and CCE are divided into five fuzzy subsets: "Positive Big", "Positive Small", "Zero", "Negative Small" and "Negative Big", represented by PB, PS, ZE, NS and NB respectively. Their membership funtions are shown in Figure 1.

The curves of the membership functions of $\Delta\eta$ and $\Delta\alpha$ are shown in Figure 2. The fuzzy control variables $\eta(t+1)$ and $\alpha(t+1)$ are adjusted according to the following equations:

$$\begin{cases} \eta(t+1) = \eta(t)(1+\Delta\eta), \\ \alpha(t+1) = \alpha(t)(1+\Delta\alpha). \end{cases} \tag{14}$$

The fuzzy control rules of $\Delta\eta$ and $\Delta\alpha$ can be obtained and are given in Tables 1 and 2 respectively.

In order to simplify the calculation of fuzzy decision, the table matching method is used. Therefore, the values of $\Delta\eta$ and $\Delta\alpha$ can be obtained by looking up Tables 1 and 2 according to the values of CE and CCE. The adjustment values of $\Delta\eta$ and $\Delta\alpha$ are taken as $NS = -0{\cdot}01$, $ZE = 0{\cdot}0$, $PS = 0{\cdot}01$.

4. AN IMPROVED FUZZY ADAPTIVE BP ALGORITHM (IFABP)

In a large number of simulated experiments using the FABP algorithm we observe that the following case occurs quite often: after a certain phase of adjustment, parameters η and α stabilize at some value whereas the calculated error does not reach the desired value and

TABLE 1

Fuzzy control rules of $\Delta\eta$

	CE				
CCE	NB	NS	ZE	PS	PB
NB	NS	NS	NS	ZE	ZE
NS	NS	NS	ZE	ZE	PS
ZE	NS	ZE	ZE	PS	PS
PS	NS	ZE	ZE	PS	PS
PB	NS	PS	PS	PS	PS

TABLE 2

Fuzzy control rules of $\Delta\alpha$

	CE				
CCE	NB	NS	ZE	PS	PB
NB	NS	NS	ZE	NS	NS
NS	NS	NS	ZE	ZE	NS
ZE	NS	NS	PS	ZE	ZE
PS	NS	NS	ZE	ZE	ZE
PB	NS	NS	ZE	NS	NS

the speed of the decrease of the calculated error is very slow. The numerical analysis results show that the parameters η and α tending towards stability are not adjusted to the optima. A detailed analysis has revealed the following two reasons causing this phenomenon:

(1) The parameters being adjusted may get stuck at a local optimum. There, however, exist probably much better values in some far region. Owing to the limitation of the FABP algorithm, a slight adjustment on the parameters no longer has an effect once the parameters get stuck in a local optimum, i.e., a slight adjustment on the parameters cannot improve the results any further. The fuzzy BP algorithm does not possess the ability of escaping from the local optima, therefore, it fails to continue the global searching.

(2) The searching direction may not be optimal. It is found that a slight adjustment on parameters in one direction may result in better results than in other directions. Unfortunately, the FABP algorithm does not possess the ability of selecting the optimal searching direction. In fact, the algorithm often follows the original searching direction. As a result, the search may even run in the opposite direction of the optimum one and the adjusted parameters get further away from their global optima. For example, if we set the searching interval of parameter η to be $[\eta_1, \eta_2]$, then the search may end at either η_1 or η_2 and then stops at one end without searching through the other end. Such cases occurs quite often in the simulated experiments.

In view of these facts we adopt the idea of mutation operation widely used in the genetic algorithm for avoiding being trapped in a local optimum. Set a mutation probability p_m in advance. If the adjusted parameters always rest on some point then the mutation operation is implemented with a probability of p_m. The result of the mutation is to assign randomly a new value to the adjusted parameter.

The improved fuzzy adaptive BP algorithm added with a mutation operation possesses the following characteristics:

(1) It prevents the parameters from getting stuck at the local optima or a constant searching direction to a certain extent. The mutation gives the adjusted parameters chances of escaping from the local optima and adjusting directions.

(2) If the adjusted parameter stabilizes in the optimum point then the mutation may have a certain destructiveness. However, the algorithm has the ability to search for the optimum again. Because the mutation probability is very small, the destructiveness of the mutation to the algorithm is minor.

(3) The FABP algorithm can be considered as a special case of the improved algorithm with a zero mutation probability. The improved FABP can provide an efficient way of increasing the training speed, if the parameter p_m is set properly.

The FABP algorithm described above needs the adjustment of the parameters CT and CCT in equations (12) and (13) to calculate the input variables of the fuzzy controller $CE(t+1)$ and $CCE(t+1)$. The purpose of adjusting CT and CCT is to map reasonably CE and CCE to a desired interval and divide the fuzzy subsets. Note that CE and CCE are changeable and the changing region is very large (for example, in the initial phase of training, the decreasing speeds of CE and CCE are fast owing to the error itself is large, whereas as the error is small the decreasing speeds of CE and CCE are less several quantitative orders than those in the initial phase of training), therefore even if some suitable values of CT and CCT are chosen in the beginning, they will certainly lose their reasonableness finally as CE and CCE change. If CE and CCE cannot be mapped reasonably to the desired interval then the division of the fuzzy subsets is no longer scientific and the fuzzy control will lose its supported foundations.

In view of the above-mentioned reasons we cancel the parameters CT and CCT in the improved algorithm and adopt the following method to quantize CE and CCE.

First, let

$$CE^*(t+1) = \frac{E(t) - E(t+1)}{\max\{E(t),\ E(t+1)\}}, \tag{15}$$

$$CCE^*(t+1) = CE^*(t+1) - CE^*(t); \tag{16}$$

then define referenced quantized scales

$$\overline{CE^*}(t+1) = \tfrac{1}{2}\,(CE^*(t) + CE^*(t+1)), \tag{17}$$

$$\overline{CCE^*}(t+1) = \tfrac{1}{2}\,(CCE^*(t) + CCE^*(t+1)). \tag{18}$$

On the basis of these define the input variables of the fuzzy controller as

$$CE(t+1) = \begin{cases} a & \text{if } \overline{CE^*}(t+1) > a, \\ -a & \text{if } \overline{CE^*}(t+1) < -a, \\ \overline{CE^*}(t+1) & \text{if } |\overline{CE^*}(t+1)| \leqslant a, \end{cases} \tag{19}$$

$$CCE(t+1) = \begin{cases} a & \text{if } \overline{CCE^*}(t+1) > a, \\ -a & \text{if } \overline{CCE^*}(t+1) < -a, \\ \overline{CCE^*}(t+1) & \text{if } |\overline{CCE^*}(t+1)| \leqslant a. \end{cases} \tag{20}$$

In the present study we set $a = 3$.

5. SIMULATED EXAMPLE

In order to verify the efficiency of applying the FABP and IFABP algorithms to the identification of multi-degree-of-freedom non-linear vibration systems, the restoring force identification of a three-degree-of-freedom vibration system with hardening springs is examined.

The restoring forces in equations (1) are

$$\begin{cases} g_1(y_1, \dot{y}_1, y_2, \dot{y}_2, y_3, \dot{y}_3) = k_{11}y_1 + k_{12}(y_1 - y_2) + k_{13}(y_1 - y_3) + k_{11}^{(3)}y_1^3 \\ \qquad + c_{11}\dot{y}_1 + c_{12}(\dot{y}_1 - \dot{y}_2) + c_{13}(\dot{y}_1 - \dot{y}_3), \\ g_2(y_1, \dot{y}_1, y_2, \dot{y}_2, y_3, \dot{y}_3) = -k_{12}(y_1 - y_2) + k_{23}(y_2 - y_3) - c_{12}(\dot{y}_1 - \dot{y}_2) \\ \qquad + c_{23}(\dot{y}_2 - \dot{y}_3), \\ g_3(y_1, \dot{y}_1, y_2, \dot{y}_2, y_3, \dot{y}_3) = -k_{13}(y_1 - y_3) - k_{23}(y_2 - y_3) + k_{33}y_3 - c_{13}(\dot{y}_1 - \dot{y}_3) \\ \qquad - c_{23}(\dot{y}_2 - \dot{y}_3) + c_{33}\dot{y}_3, \end{cases} \tag{21}$$

where the values of the physical parameters are taken as

$$m_1 = 1\ \text{kg}, \quad m_2 = 1{\cdot}3\ \text{kg}, \quad m_3 = 2\ \text{kg},$$

$$k_{11} = 1000\ \text{N/m}, \quad k_{12} = 2000\ \text{N/m}, \quad k_{13} = 800\ \text{N/m},$$

$$k_{22} = 1200\ \text{N/m}, \quad k_{23} = 1500\ \text{N/m}, \quad k_{33} = 3000\ \text{N/m},$$

$$c_{11} = 20\ \text{N s/m}, \quad c_{12} = 15\ \text{N s/m}, \quad c_{13} = 10\ \text{N s/m},$$

$$c_{22} = 15\ \text{N s/m}, \quad c_{23} = 30\ \text{N s/m}, \quad c_{33} = 25\ \text{N s/m},$$

$$k_{11}^{(3)} = 1{,}000{,}000\ \text{N/m}^3.$$

The excitation used for training the neural network is a swept sine signal with amplitude 50 N and excitation frequency 2π. The excitation only exists at m_1. The net inputs $\{y_{1k}, \dot{y}_{1k}, y_{2k}, \dot{y}_{2k}, y_{3k}, \dot{y}_{3k}\}^T$ and the desired outputs $\{g_{1k}, g_{2k}, g_{3k}\}^T$ are sampled in the time interval [0·2, 20] s at intervals of 0·2 s. The number of patterns in the sample set is $p = 200$.

TABLE 3

Comparisons between measurements and identification of displacements, velocities and restoring forces

Measurements	Minima	Maxima	Estimates	Minima	Maxima	Errors of maxima (%)
y_1	– 0·01780	0·01780	y_1'	– 0·01721	0·01721	3.29
$\dot{y}_1$	– 0·10106	0·10115	$\dot{y}_1'$	– 0·09700	0·09708	4.02
y_2	– 0·00934	0·00934	y_2'	– 0·00903	0·00903	3.29
$\dot{y}_2$	– 0·05355	0·05360	$\dot{y}_2'$	– 0·05133	0·05138	4.14
y_3	– 0·00538	0·00538	y_3'	– 0·00520	0·00520	3.29
$\dot{y}_3$	– 0·03101	0·03104	$\dot{y}_3'$	– 0·02974	0·02977	4.09
g_1	– 50·29174	50·26534	g_1'	– 47·66461	48·17251	4.16
g_2	– 0·30951	0·31054	g_2'	– 0·28218	0·28318	8.81
g_3	– 0·46270	0·29123	g_3'	– 0·38029	0·27320	6.19

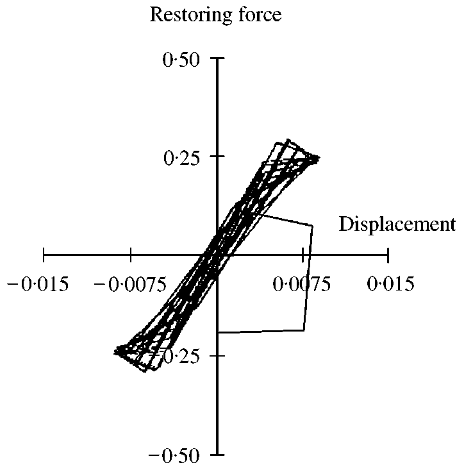

Figure 3. Measurements $g_2(y_2)$.

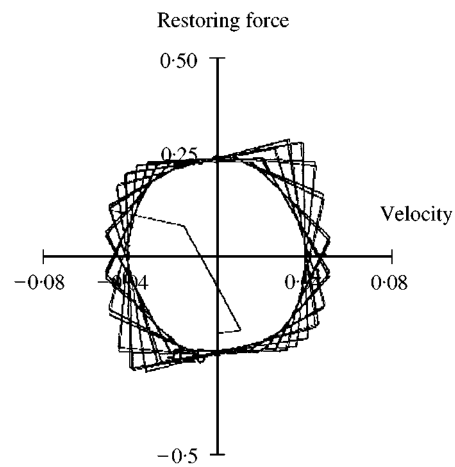

Figure 4. Measurements $g_2(\dot{y}_2)$.

The number of the hidden layer neurons is $l = 13$. The total number of the network parameters to be adjusted is 133 (117 weights and 16 thresholds terms). Both mutation probabilities of the learning rate η and the momentum factor α are taken as 0·01. The data used in the validation phase are obtained by employing the excitation at m_1 with amplitude 48 N.

For the sake of comparing the FABP and IFABP algorithms with the conventional BP algorithm, we set a desired error in the training process and examine the training speeds of the three algorithms. The simulated results show that the training speed of the IFABP algorithm is about 1·5 times faster than that of the FABP algorithm, and is 4 times or so faster than that of the conventional BP algorithm. It can be seen that the mutation operation plays an important role in the implementation process on the algorithm. Table 3 gives the minimum and maximum values of measurements and identifications of displacements, velocities and restoring forces, and the relative errors of the maximum values as the desired error is taken as 0·003.

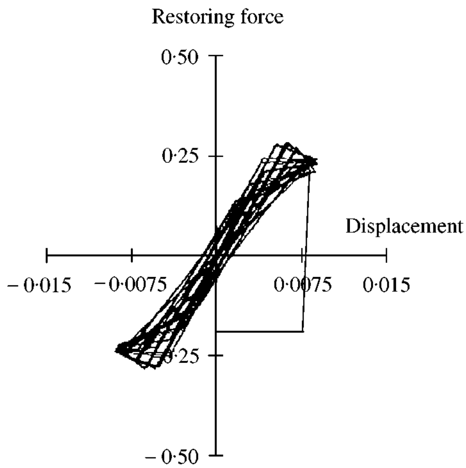

Figure 5. Identification results $g'_2(y'_2)$.

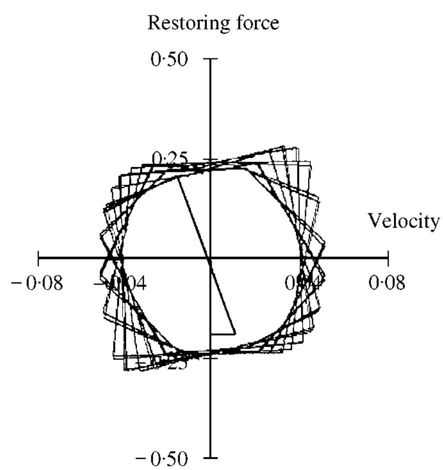

Figure 6. Identification results $g'_2(\dot{y}'_2)$.

In order to inspect and compare the identification results intuitively, Figures 3 and 4 show the measurements $g_2(y_2)$ and $g_2(\dot{y}_2)$, where y_2 and $\dot{y}_2$ are taken as independent variables, respectively; Figures 5 and 6 show the corresponding identifications $g'_2(y'_2)$ and $g'_2(\dot{y}'_2)$, respectively; Figure 7 shows the comparison of error curves from IFABP, FABP and conventional BP algorithms.

From Table 3 and Figures 3–7 it can be seen that both FABP and IFABP algorithms can be applied effectively to the identification of restoring forces in multi-degree-of-freedom vibration systems, whereas using the IFABP algorithm can obtain a big increase in the training speed of the network.

6. CONCLUSIONS

Simulated results show that the fuzzy adaptive BP (FABP) algorithm which combines the fuzzy theory with artificial neural network techniques is effective in solving the problems of

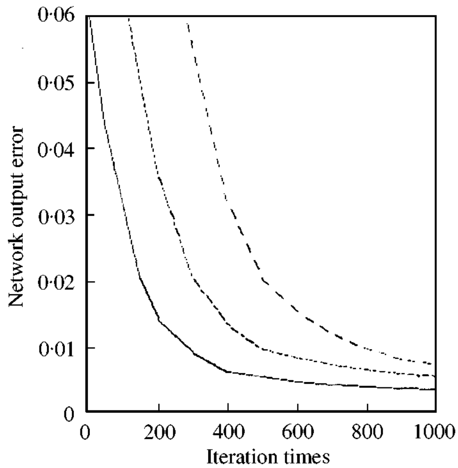

Figure 7. Comparison of error curves from IFABP, FABP and conventional BP algorithms: —— IFABP algorithm; - - - FABP algorithm; – – – conventional BP algorithm.

identifying restoring forces in non-linear vibration systems. The FABP algorithm increases the training speed of the network to a certain extent. The improved fuzzy adaptive BP (IFABP) algorithm which is based on the FABP algorithm and presented in this paper not only increases the effectiveness of the algorithm but also enhances the adaptivity of the algorithm still further and enables the algorithm to be more applicable in practice.

ACKNOWLEDGMENT

This work has been supported by the National Natural Science Foundation of China and the Open Research Laboratory of Symbol Computation and Knowledge Engineering of the Chinese National Education Ministry.

REFERENCES

1. S. F. Masri, A. G. Chassiakos and T. K. Caughey 1993 *American Society of Mechanical Engineers Journal of Applied Mechanics* **60**, 123–133. Identification of nonlinear dynamic systems using neural networks.
2. A. G. Chassiakos and S. F. Masri 1996 *Earthquake Engineering and Structural Dynamics* **25**, 117–128. Modeling unknown structural systems through the use of neural networks.
3. Y. C. Liang, C. G. Zhou and Z. S. Wang 1997 *Journal of Sound and Vibration* **206**, 103–108. Identification of restoring forces in non-linear vibration systems based on neural networks.
4. Y. C. Liang, Z. Wang, X. W. Yang and C. G. Zhou 1997 *Mechanics Research Communications* **24**, 447–455. Identification of non-linearities in cushioning packaging using neural networks with fuzzy adaptive control.
5. Y. C. Liang, Z. Wang, X. W. Yang and C. G. Zhou 1997 *Acta Mechanica Sinica* **29**, 497–500. A study of identification of nonlinear characteristics in cushioning packaging based on neural networks.
6. C. Dong, Z. N. Li, R. W. Xia and Q. Z. He 1995 *Advances in Mechanics* **25**, 186–196. Advances and problems in the study of multilayer feedforward neural networks.
7. J. G. Yang, D. C. Dai, W. Gao and Z. Q. Cao 1995 *Journal of Vibration Engineering* **8**, 62–66. Identification of nonlinear characteristics of vibration system using structural neural network.

8. G. YAGAWA and H. OKUDA 1996 *Archives of Computational Methods in Engineering — State of the Art Reviews* **3**, **4**, 435–512. Neural networks in computational mechanics.
9. Y. C. LIANG, X. W. YANG, C. G. ZHOU and Z. S. WANG 1996 *Mechanics Research Communications* **23**, 607–613. Application of neural networks to identification of nonlinear characteristics in cushioning packaging.
10. G. G. GREEN and S. MANCHANDA 1996 *Proceedings of the International Conference on Identification in Engineering Systems, Swansea, UK*, 741–750. System identification and minimal model realisation using recurrent neural networks.
11. A. DELGADO, C. KAMBHAMPATI and K. WARWICK 1996 *Proceedings of the International Conference on Identification in Engineering Systems, Swansea, UK*, 751–760. Approximation of non-linear systems using dynamics recurrent neural networks.
12. J. G. JUANG 1996 *Proceedings of the International Conference on Identification in Engineering Systems, Swansea, UK*, 761–770. Parameter estimation in non-linear system using recurrent networks.
13. J. X. XU, Y. H. CHEN and Y. L. JIANG 1998 *Advances in Mechanics* **28**, 145–162. Nonlinear dynamics of artificial neural networks and applications.
14. Y. C. LIANG, W. Y. GONG, X. W. YANG and C. G. ZHOU 1998 *Mechanics Research Communications* **25**, 395–403. Identification of nonlinear characteristics in cushioning packaging using genetic evolutionary neural networks.
15. D. E. RUMELHART and J. L. MCCLELLAND 1988 *Parallel Distributed Processing*. Explorations in the microstructure of cognition.