

An Inverse Vibration Problem Solved by an Artificial Neural Network

E.H. SHIGUEMORI, L.D. CHIWIACOWSKY, H.F. de CAMPOS VELHO, J.D.S. da SILVA, Laboratório Associado de Computação e Matemática Aplicada, LAC, INPE, São José dos Campos, SP, Brasil.

Abstract. Inverse problems in vibration is a process of determining parameters based on numerical analysis from a comparison between measured vibration data and its predicted values provided by a mathematical model. In this work the displacement data have been chosen in order to identify the stiffness matrix which will cause a changing in the time-history of the system displacement. This is an inverse problem, since the stiffness matrix evaluation is obtained through the determination of the modified stiffness coefficients. In this work, the artificial neural network technique is applied to the inverse vibration problem where the goal is to estimate the unknown time-dependent stiffness coefficients simultaneously in a two degree-of-freedom structure, using a Multilayer Perceptron Neural Network model. Numerical experiments have been carried out with synthetic experimental data considering a noise level of 1%. Good recoveries have been achieved with this methodology.

1. Introduction

In vibration science, the solution of direct forced vibration problems is concerned with the determination of the system displacement, velocity and acceleration at time t when the initial conditions, external forces and time-dependent system parameters, such as stiffness and damping coefficients, are specified. On the other hand, the solution of inverse vibration problems is concerned with the estimation of such quantities: stiffness or damping coefficients, external forces, from the measured vibration data, such as frequency and/or mode shape measurements, or displacement measurements at different time t .

Considering the structural damage detection problem, it is displayed as an inverse vibration problem, since the damage evaluation is achieved through the determination of the stiffness coefficient variation, or the stiffness coefficient by itself. A variety of experimental, numerical and analytical techniques has already been proposed to solve the damage identification problem, and it has received considerable attention due to its practical applications [5]. These methods are usually classified under different categories, such as frequency and time domain methods, and deterministic and stochastic approaches [3]. Among the deterministic approaches the Conjugate Gradient method with the Adjoint Equation has been successfully used for different damaged lumped parameter systems [9, 8, 4]. Among the stochastic

methods, the use of the artificial neural networks (ANN), which has already been used successfully in thermal sciences [12], has also been presented as a satisfactory choice to deal with the damage identification problem [5, 13, 14, 2]. The fault tolerance, generalization capabilities of ANN's make them attractive to approach inverse problems.

Mathematically the inverse problems are classified as ill-posed problems. A problem is considered well-posed if the following requirements are satisfied: the problem solution exists, it is unique and stable related to the input data [6]. For the inverse problems, in general, this conditions are not satisfied because small variations in the input data, such as random errors inherent to the measurements used in the analysis, can cause large oscillations on the solution. Usually the inverse problem, i.e. the ill-posed problem, is presented as a well-posed functional form, whose solution is obtained through the use of optimization procedures. It should be pointed out that knowing the behaviour of the stiffness coefficients in the time domain, a prediction of the service life of the structure could be assigned.

2. The Direct Problem

The N -DOF damped system considered in this work is presented in the Figure 1 and the mathematical formulation of this forced vibration system is given by

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}(t)\dot{\mathbf{x}}(t) + \mathbf{K}(t)\mathbf{x}(t) = \mathbf{f}(t), \quad t > 0, \quad (2.1)$$

with initial conditions

$$\mathbf{x}(0) = \mathbf{x}_0 \quad \text{and} \quad \dot{\mathbf{x}}(0) = \dot{\mathbf{x}}_0, \quad (2.2)$$

where \mathbf{M} represents the system mass matrix, $\mathbf{K}(t)$ the time-dependent stiffness matrix, $\mathbf{C}(t)$ the time-dependent damping matrix, $\mathbf{f}(t)$ the external forces vector, and $\mathbf{x}(t)$ the displacements vector. There exists no analytical solution for system (2.1)-(2.2) for any arbitrary functions of $\mathbf{K}(t)$, $\mathbf{C}(t)$, and $\mathbf{f}(t)$. For this reason the numerical solution with the *Newmark* method [11] is applied to solve the direct problem. This method calculates the system displacement vector $\mathbf{x}(t)$, if initial

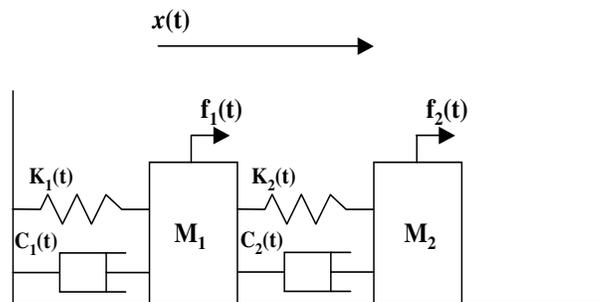


Figure 1: The forced vibration system.

conditions, system parameters \mathbf{M} , $\mathbf{K}(t)$ and $\mathbf{C}(t)$, and the time-dependent external forces $\mathbf{f}(t)$ are known.

3. Multilayer Perceptron Neural Network

Artificial neural networks (ANN's) are made of arrangements of processing elements called neurons. The artificial neuron model basically consists of a linear combiner followed by an activation function, Figure 2(a), given by:

$$y_k = \varphi \left(\sum_{j=1}^n w_{kj} x_j + b_k \right), \quad (3.1)$$

where w_{kj} are the connections weights and b_k is a threshold parameter. Arrangements of such units form the ANN's that are characterized by:

- Very simple neuron-like processing elements;
- Weighted connections between the processing elements;
- Highly parallel processing and distributed control;
- Automatic learning of internal representations.

ANN's aim to explore the massively parallel network of simple elements in order to yield a result in a very short time slice and, at the same time, with insensitivity to loss and failure of some of the elements of the network. These properties make artificial neural networks appropriate for application in pattern recognition, signal processing, image processing, financing, computer vision, engineering, etc.

There exist different architectures of ANN that are dependent upon the learning strategy adopted. This paper briefly describes the Multilayer Perceptron (MLP) with backpropagation learning. Detailed introduction on ANN's can be found in [7] and [10]. MLP with backpropagation learning algorithm, are feedforward networks composed of an input layer, an output layer, and a number of hidden layers, whose aim is to extract high order statistics from the input data [7]. Figure 2(b) depicts a backpropagation neural network with a hidden layer.

Functions $\varphi(\cdot)$ provide the activation for the neuron. Neural networks will solve nonlinear problems, if nonlinear activation functions are used for the hidden and/or the output layers. From several activation functions, the sigmoid are commonly used:

$$\text{logistic function : } \quad \varphi(v) = \frac{1}{1 + \exp(-av)}; \quad (3.2)$$

$$\text{bipolar function : } \quad \varphi(v) = \frac{1 - \exp(-av)}{1 + \exp(-av)}. \quad (3.3)$$

A feedforward network is a non-linear mapping to compute the output vector from an input vector. The connections among the several neurons (Figure 2(b)) have

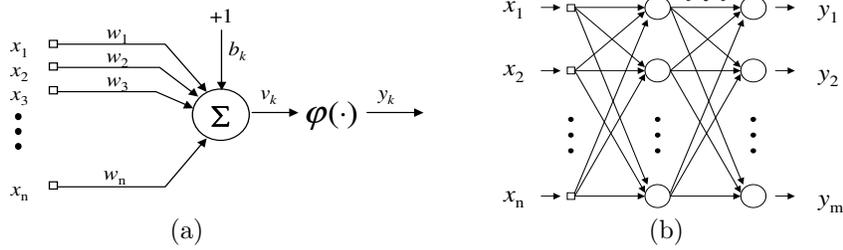


Figure 2: (a) Single Neuron, (b) Multilayer Neural Network.

associated weights that are adjusted during the learning process, thus changing the performance of the network. Two distinct phases can be devised while using an ANN: the training phase (learning process) and the run phase (activation of the network). The training phase consists of adjusting the weights for the best performance of the network in establishing the mapping of many input/output vector pairs. Once trained, the weights are fixed and the network can be presented to new inputs for which it calculates the corresponding outputs, based on what it has learned.

The backpropagation training is a supervised learning algorithm that requires both input and output (desired) data. Such pairs permit the calculation of the error of the network as the difference between the calculated output and the desired vector. The weight adjustments are conducted by backpropagating such error to the network, governed by a change rule. The weights are changed by an amount proportional to the error at that unit, times the output of the unit feeding into the weight. Eq. (3.4) shows the general weight correction according to the so-called the delta rule

$$\Delta w_{kj} = \eta \delta_k y_j, \quad (3.4)$$

where, δ_k is the local gradient, y_j is the input signal of neuron k (for hidden layers, or even the out-put layer), and η is the learning rate parameter that controls the strength of change.

4. Inverse Problem Solution by ANN

The stiffness estimation problem has already been solved employing others methods [8, 9, 4]. In this work a MLP Neural Network is employed to solve the same problem presented in [9], a 2-DOF dynamical system. The following parameters have been used in the forward problem, given by system (2.1)-(2.2): $M_1 = 1.0$ and $M_2 = 3.0$; $C_1(t) = 8.0$ and $C_2(t) = 5.0$; $f_1(t) = 50.0$ and $f_2(t) = 60.0$; and initial conditions $\mathbf{x}(0) = 0$ and $\dot{\mathbf{x}}(0) = 0$. The unknown transient stiffness coefficients have been assumed as:

$$\mathcal{K}_1(t) = A_1 - B_1 \sin(\omega_1 t); \quad \mathcal{K}_2(t) = A_2 + B_2 \cos(\omega_2 t). \quad (4.1)$$

In order to improve the time information to the neural network, a history of 15 time steps is used, resulting of 30 input vector and 2 desired output vector. This

means that we are using a time-history with 15 time-steps for predicting the stiffness matrix at time-step 16, and so on. This strategy is called as "time-delay ANN", see for example [1]. Only one hidden layer was sufficient to obtain satisfactory results. The sigmoid activation function has been used in both hidden and output layers, Eq. (3.3).

The experimental data, which intrinsically contains errors in the real world, have been simulated by adding a random perturbation to the exact solution of the direct problem, such that

$$x^{exp}(t) = x(t) + x(t)[\sigma \times \mathcal{R}], \quad (4.2)$$

where σ is the standard deviation of the noise and \mathcal{R} is a random variable taken from a Gaussian distribution, with zero mean and unitary variance. For numerical purposes, it has been adopted the whole time period $t_f = 36$ s, and the time step $\Delta t = 0.1$ s.

In order to evaluate the accuracy of this procedure, the error between the solution obtained by the neural network and the exact solution is defined by

$$E(\hat{\mathcal{K}}) = \left\| \frac{\mathcal{K}_{exact}(t) - \hat{\mathcal{K}}(t)}{\mathcal{K}_{exact}(t)} \right\|_2^2, \quad (4.3)$$

where $\|\cdot\|_2$ is the 2-norm.

4.1. Training

The training set is built up from the solution of the direct model, system (2.1)-(2.2), assuming several different variations of the stiffness functions, Eqs. (4.1), generating the respective displacements. For the training phase the desired output of the ANN are the stiffness coefficients, while the input of the ANN are the corresponding measured displacements. For each assumed stiffness coefficient function, the corresponding displacement is computed to adjust the weight and bias which will be used in the activation phase. The training set was composed by 12 different functions for the stiffness coefficients. These functions are defined through the expressions (4.1) where the values for the parameters A_i , B_i and ω_i ($i = 1, 2$), have been assumed different from those used in the generalization phase.

Furthermore, according to the noise level, three different sets of the displacement functions have been used. The first one free of noise (noiseless) – ANN-1, the second one corrupted by 1% of noise – ANN-2, and third data set using 3% of noise – ANN-3. Typical displacement functions are presented in Figures 3 and 4 for the noiseless and 1% of noise cases, respectively.

5. Numerical Results

In the activation phase the inverse problem is solved by using the weights and bias obtained during the training phase. The robustness of the trained MLP is evaluated employing displacement functions not used in the training phase. Displacement

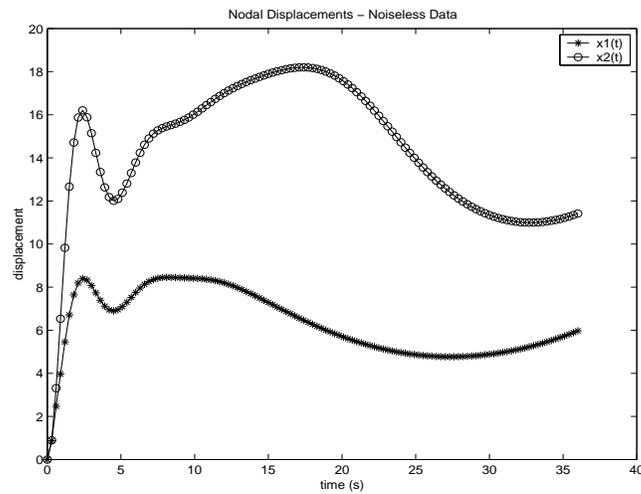


Figure 3: Noiseless data.

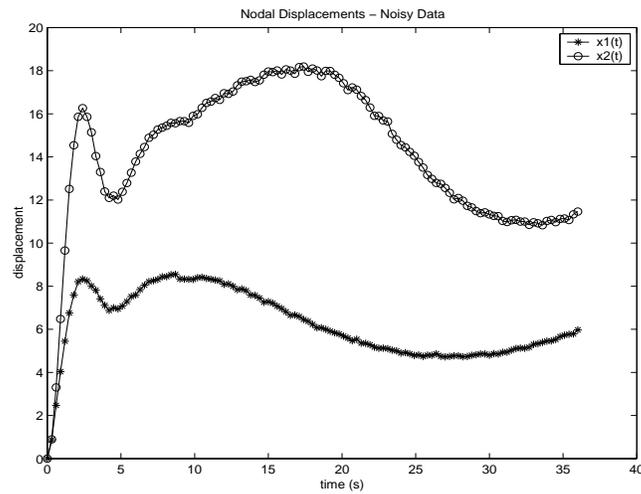


Figure 4: Noisy data - 1%.

functions corrupted by different levels of noise have been used to simulate a more realistic situation.

The generalization capacity of the MLP is verified considering 18 different functions. The first 14 functions are defined through the expressions (4.1) where the parameters A_i , B_i and ω_i , for $i = 1, 2$, are presented in Table 1.

Other 4 functions were used in the generalization phase which are defined as:

Table 1: Parameters used in the training functions.

Function	A_1	A_2	B_1	B_2	ω_1	ω_2
1	18	8	5	3	$2\pi/36$	$2\pi/36$
2	18	8	7	4	$2\pi/36$	$2\pi/36$
3	18	8	3	2	$2\pi/36$	$2\pi/36$
4	11	5	5	3	$2\pi/36$	$2\pi/36$
5	25	11	5	3	$2\pi/36$	$2\pi/36$
6	18	8	5	3	$2\pi/52$	$2\pi/52$
7	18	8	5	3	$2\pi/18$	$2\pi/18$
8	18	8	7	4	$2\pi/52$	$2\pi/52$
9	18	8	3	2	$2\pi/18$	$2\pi/18$
10	11	5	3	2	$2\pi/18$	$2\pi/18$
11	18	8	0	0	0	0
12	18	8	5	3	$5\pi/9$	$5\pi/9$
13	18	8	5	3	$2\pi/36$	$5\pi/9$
14	18	8	5	3	$5\pi/9$	$2\pi/36$

- Function 15

$$\mathcal{K}_1 = \begin{cases} 4/5 t + 12 & 0 \leq t < 18, \\ 4/5(36 - t) + 12 & 18 \leq t \leq 36. \end{cases} \quad (5.1)$$

$$\mathcal{K}_2 = \begin{cases} 2/5 t + 4 & 0 \leq t < 18, \\ 2/5(36 - t) + 4 & 18 \leq t \leq 36. \end{cases} \quad (5.2)$$

- Function 16

$$\mathcal{K}_1 = \begin{cases} 4/5 t + 12 & 0 \leq t < 18, \\ 4/5(36 - t) + 12 & 18 \leq t \leq 36. \end{cases} \quad (5.3)$$

$$\mathcal{K}_2 = \begin{cases} 12 - 2/5 t & 0 \leq t < 18, \\ 12 - 2/5(36 - t) & 18 \leq t \leq 36. \end{cases} \quad (5.4)$$

- Function 17

$$\mathcal{K}_1 = \begin{cases} 4/5 t + 10 & 0 \leq t < 18, \\ 4/5(t - 18) + 10 & 18 \leq t \leq 36. \end{cases} \quad (5.5)$$

$$\mathcal{K}_2 = \begin{cases} 12 - 2/5 t & 0 \leq t < 18, \\ 12 - 2/5(36 - t) & 18 \leq t \leq 36. \end{cases} \quad (5.6)$$

- Function 18

$$\mathcal{K}_1 = \{10 - 5 \sin(2\pi t/36)\}. \quad (5.7)$$

$$\mathcal{K}_2 = \begin{cases} 12 - 1/5 t & 0 \leq t < 18, \\ 12 - 1/5(t - 18) & 18 \leq t \leq 36. \end{cases} \quad (5.8)$$

The average error, Eq. (4.3), is computed considering the recovered functions for each stiffness coefficient \mathcal{K}_1 and \mathcal{K}_2 .

$$\overline{E(\hat{\mathcal{K}}_j)} = \sum_{i=1}^N \frac{E(\hat{\mathcal{K}}_j)_i}{N}, \quad \text{where } N = 18, \quad j = 1, 2. \quad (5.9)$$

The standard deviation of the estimation errors for the recovered stiffness coefficients \mathcal{K}_1 and \mathcal{K}_2 has been calculated using the expression

$$E(\hat{\mathcal{K}}_j)_{std} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(E(\hat{\mathcal{K}}_j)_i - \overline{E(\hat{\mathcal{K}}_j)} \right)^2}, \quad \text{where } N = 18, \quad j = 1, 2. \quad (5.10)$$

The activation was performed using noiseless data, and data with 1% and 3% of noise, see Eq. (4.2). Table 2 presents the statistic values, defined by the Eqs. (5.9)-(5.10), when the inverse problem has been solved for 50 different “seeds” used in the generation of the random noise for each noise level.

Table 2: Errors - First test: training for noiseless data.

noise	$E(\hat{\mathcal{K}}_1)_{mean}$	$E(\hat{\mathcal{K}}_2)_{mean}$	$E(\hat{\mathcal{K}}_1)_{std}$	$E(\hat{\mathcal{K}}_2)_{std}$
0%	2.6226	1.6313	5.5789	1.7493
1%	2.8722	1.8303	5.6167	1.7425
3%	4.8850	3.4548	5.6910	1.7925

In the second test, the MLP has been trained with displacement functions corrupted by a 1% noise in training set. Similar to previous test, the estimations for the 50 different cases were used to generate the statistic values presented in Table 3.

Table 3: Errors - Second test: training for data with 1% of noise.

noise	$E(\hat{\mathcal{K}}_1)_{mean}$	$E(\hat{\mathcal{K}}_2)_{mean}$	$E(\hat{\mathcal{K}}_1)_{std}$	$E(\hat{\mathcal{K}}_2)_{std}$
0%	2.3745	1.4745	5.7822	1.7652
1%	2.5552	1.6472	5.7824	1.7567
3%	4.0020	3.0084	5.7480	1.7727

Tables 2 and 3 present qualitative results of the stiffness coefficients estimation. Furthermore, it has been showed that when a higher noise level is considered the mean of the estimation errors present a considerable increase. A sensible improvement of the mean errors has been noticed when noisy data are used in the neural network training, rendering the MLP slightly more robust in relation to the noisy experimental data. Following, for a quantitative evaluation of the stiffness coefficients estimation, the best and worst estimation cases, picked up from the complete generalization test set, will be presented.

Figures 5-7 show the best estimation among the 18 test-functions. These figures present the estimation results for the ANN-2 with three different levels of noise in the measured data. From the values presented in Table 4, it can be noticed that the stiffness estimations are strongly affected by the quality of the experimental data. However, the estimation results obtained when the experimental data are corrupted by a 1% noise are similar to that presented in [9].

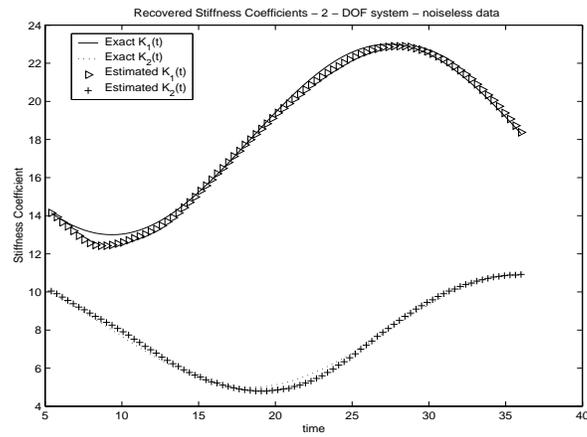


Figure 5: Estimated stiffness coefficients for ANN-2: noiseless data.

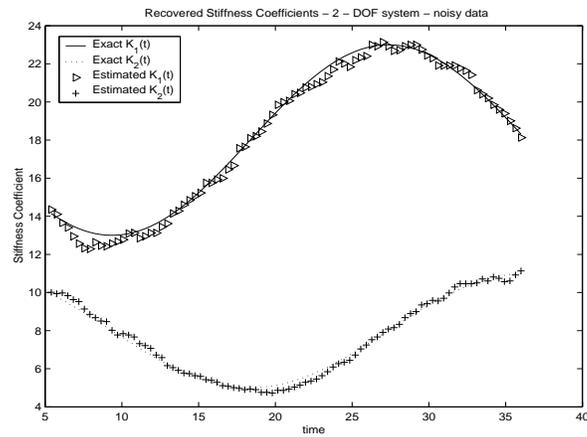


Figure 6: Estimated stiffness coefficients for ANN-2: data corrupted by 1% of noise.

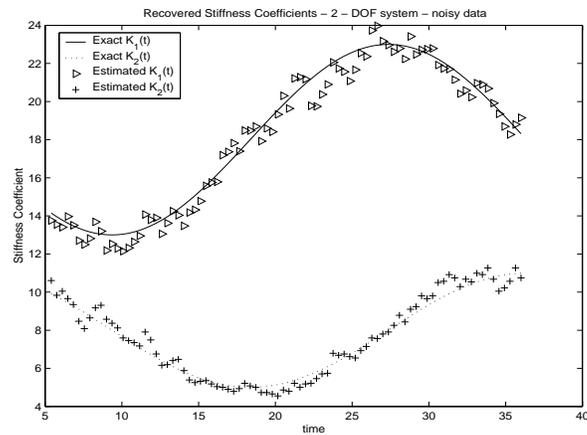


Figure 7: Estimated stiffness coefficients for ANN-2: data corrupted by 3% of noise.

Table 4: Errors - Stiffness 1.

noise	$E(\hat{\mathcal{K}}_1)_{mean}$	$E(\hat{\mathcal{K}}_2)_{mean}$	$E(\hat{\mathcal{K}}_1)_{std}$	$E(\hat{\mathcal{K}}_2)_{std}$
0%	0.3979	0.4729	—	—
1%	0.5459	0.6550	0.0573	0.0867
3%	2.0010	1.9959	0.2859	0.3343

The worst estimation has been achieved considering discontinuous stiffness coefficients functions. Figures 8-10 present the estimation results for ANN-2. It should be pointed out that the quality of the estimation does not present the same performance of those obtained previously. It can be confirmed through the statistical values presented in Table 5. Moreover, it should be noticed that even with the significant discontinuity presented in the first derivative, its influence is less pronounced for \mathcal{K}_2 coefficient.

Table 5: Errors - Stiffness 2.

noise	$E(\hat{\mathcal{K}}_1)_{mean}$	$E(\hat{\mathcal{K}}_2)_{mean}$	$E(\hat{\mathcal{K}}_1)_{std}$	$E(\hat{\mathcal{K}}_2)_{std}$
0%	25.1971	0.7345	—	—
1%	25.6438	0.9395	0.6404	0.0745
3%	26.9233	2.3354	1.5664	0.1956

6. Final Remarks

The inverse vibration problem of estimating the unknown time-dependent stiffness coefficients of a 2-DOF forced dynamic system has been solved using a MultiLayer Perceptron Neural Network. For structural engineering, the estimation of the stiffness matrix is cited as *damage identification*.

The numerical results presented in this work are concerned to the noiseless and noisy experimental data represented by the system displacements. When noiseless experimental data are considered, perfect estimations of the time-dependent stiffness coefficients have been achieved. Considering noisy experimental data, good (not perfect) estimations of the time-dependent stiffness coefficients have been achieved. Through the estimation of the time-history of the stiffness coefficients one could estimate the operating life time of the specific structure.

It should be pointed out the generalization capacity of the trained MLP employed in this work. Even the training set being composed only by continuous stiffness functions, also discontinuous functions have been satisfactorily estimated. Moreover, good results have been obtained only when using a range of 15 time steps of the displacement time-history in the input data of the neural network. This strategy has been used for the training and activation phases. The use of this

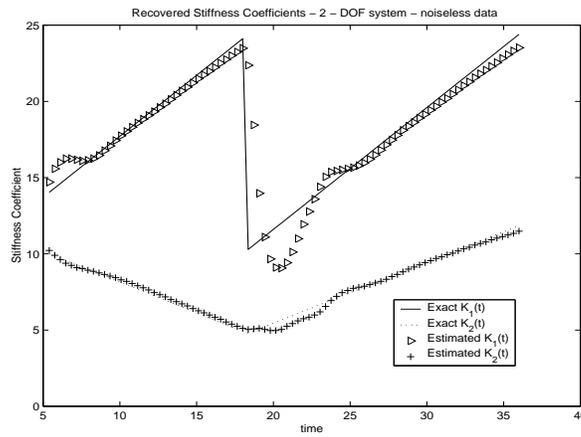


Figure 8: Estimated stiffness coefficients for ANN-2: noiseless data.

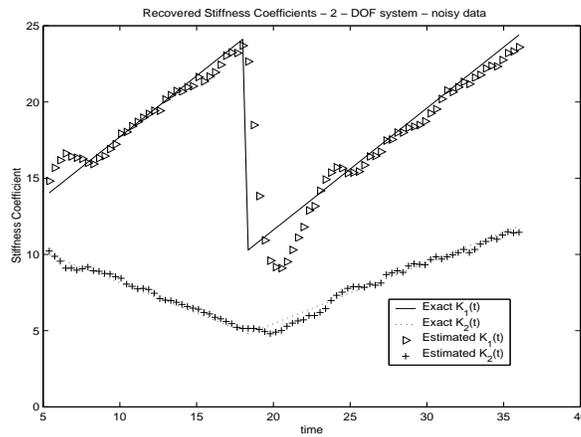


Figure 9: Estimated stiffness coefficients for ANN-2: data with 1% of noise.

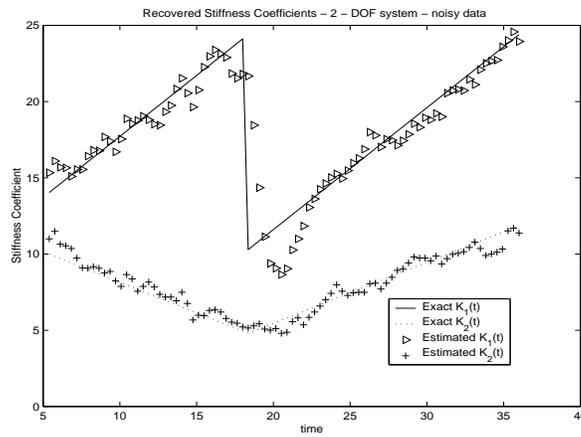


Figure 10: Estimated stiffness coefficients for ANN-2: data with 3% of noise.

extended time range has been shown adequate, since more information regarding the behaviour of the dynamic of the system is presented to the ANN.

Numerical experiments were carried out using three data set for training: (i) noiseless data (ANN-1); (ii) data with 1% of noise (ANN-2); and (iii) data with 3% of noise (ANN-3). During the learning phase two stopping criteria were used, a desired error of 10^{-5} or 5×10^4 maximum number of epoches. In all cases, the training phase was stopped with 5×10^4 epoches. Best results were obtained using ANN-2, see Figures 5-10.

Inverse solutions obtained are similar to those presented by C. H. Huang [9]. However, after training process, ANN's are much faster than the variational approach, additionally, ANN's are intrinsically parallel algorithms. Finally, ANN's can be implemented by hardware devices.

Acknowledgements

The authors would like to thank the support from FAPESP and CNPq, Brazilian agencies for research support.

References

- [1] S.V. Barai and P.C. Pandey, Time-delay neural networks in damage detection of railway bridges, *Advances in Engineering Software*, **28** (1997), 1-10.
- [2] E.P. Carden and P. Fanning, Vibration Based Condition Monitoring: A Review, *Structural Health Monitoring*, **3**, No. 4 (2004), 355-377.
- [3] L.D. Chiwiacowsky and H.F. Campos Velho, Different Approaches for the Solution of a Backward Heat Conduction Problem, *Inverse Problems in Engineering*, **11**, No. 6 (2003), 471-494.
- [4] L.D. Chiwiacowsky, H.F. Campos Velho and P. Gasbarri, The Damage Identification Problem: A Hybrid Approach, em "Anais do II Congresso Temático de Dinâmica, Controle e Aplicações (DINCON)" (J.M. Balthazar, G.N. da Silva, M. Tsuchida, M. Boaventura, L.C.S. Góes e J.D.S. Silva, eds.), pp. 1393-1402, São José dos Campos, SP, Brazil, 2003.
- [5] S.W. Doebling, C.R. Farrar, M.B. Prime and D.W. Shevitz, "Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review", Los Alamos National Laboratory, Technical Report LA-13070-MS, USA, 1996.
- [6] J. Hadamard, "Lectures on the Cauchy problem in linear partial differential equations", Yale University Press, New Haven, 1923.
- [7] S. Haykin, "Neural Networks: A Comprehensive Foundation", Macmillan, New York, 1993.
- [8] C. H. Huang, A non-linear inverse vibration problem of estimating the time-dependent stiffness coefficients by conjugate gradient method, *International Journal of Numerical Methods in Engineering*, **50** (2001), 1545-1558.

- [9] C.H. Huang, An inverse vibration problem for simultaneously estimating the time-dependent stiffness coefficients, in "Proceedings of the 4th International Conference on Inverse Problems in Engineering: Theory and Practice", pp. 123-131, Rio de Janeiro, RJ, Brazil, 2002.
- [10] M. Nadler and E.P. Smith, "Pattern Recognition Engineering", John Wiley & Sons, New York, 1993.
- [11] N.M. Newmark, A method of computation for structural dynamics, *ASCE - Journal of the Engineering Mechanics Division*, **85** (1959), 67-94.
- [12] E.H. Shiguemori, H.F. de Campos Velho, J.D.S. da Silva, Estimation of Initial Condition in Heat Conduction by Neural Network, *Inverse Problems in Science and Engineering*, **12**, No. 3 (2004), 317-328.
- [13] H. Sohn, C.R. Farrar, F.M. Hemez, D.D. Shunk, D.W. Stinemat, B.R. Nadler, "A Review of Structural Health Monitoring Literature: 1996-2001", Los Alamos National Laboratory, Technical Report LA-13976-MS, USA, 2003.
- [14] K. Worden and J.M. Dulieu-Barton, An Overview of Intelligent Fault Detection in Systems and Structures, *Structural Health Monitoring*, **3**, No. 1 (2004), 85-98.

