AIAA JOURNAL
Vol. 32, No. 1, January 1994

Structural Damage Detection and Identification Using Neural Networks

Poyu Tsou* and M.-H. Herman Shen†
Ohio State University, Columbus, Ohio 43210

A novel methodology is presented for on-line damage identification of discrete structural systems. The damage characteristic (location and severity) of the system first can be detected and then identified from the change of its dynamic properties (eigenvalues and mode shapes) through a backward-propagation neural network. The neural network is constructed by three multilayer subnets that perform the tasks of input pattern generation, damage location identification, and damage severity determination, respectively. The methodology is demonstrated on two spring-mass systems. The effectiveness and limitations of the methodology are discussed.

Nomenclature

C = damping matrix of the discrete structural system
d_i = dynamic residual vector
K = stiffness matrix of the discrete structural system
M = mass matrix of the discrete structural system
λ_{d_i}, v_{d_i} = generalized eigenvalue and eigenvector of damaged system
λ_{h_i}, v_{h_i} = generalized eigenvalue and eigenvector of undamaged system

I. Introduction

THE construction of optimum designed load-carrying structural systems such as aircraft structures, turbines, and rotors, etc., has become more complicated. Although those systems are carefully designed for fatigue loading and are inspected before service as well as periodically during their operating life, there are instances of cracks or structural damages escaping inspection. Consequently, the development of structural integrity monitoring techniques has received increasing attention in recent years. Among these monitoring techniques, it is believed that the monitoring of the global dynamics of a structure offers a favorable alternative if the on-line damage detection is necessary.

In a number of studies,[1-6] global dynamic behavior of the damaged structures (e.g., natural frequencies, mode shapes, strain energy, etc.) were calculated based on the given properties such as damage location, damage severity, etc. The question of the estimation of such properties in the case where the behavior of the structure is known, i.e., the inverse problem, is discussed in this paper. In general, the inverse problem inherently involves the issues of comprehensive search, solvability, and uniqueness. Several investigators have developed the inverse procedure of identifying the engineering properties of structures from dynamic response information. For example, Gladwell et al.[7] successfully recovered the coefficients in the differential equation associated with a Bernoulli-Euler beam from knowledge of the eigenvalues or related spectral data. Shen and Taylor[8] presented an on-line nonintrusive damage identification technique of a vibrating beam. The idea of the procedure is related to methods of structural optimizations in which the damage was identified in a way to minimize one or another measure of the difference between measurements and the corresponding values for the dynamic response obtained by an analysis of a model for the damaged beam. Other noteworthy works presented in the literature that use modal data to detect structural damage are Kabe,[9] Zimmerman and Kaouk,[10] and Zimmerman and Widengren.[11]

In addition to the aforementioned approaches, a number of schemes have been developed based on the measurements of time responses; e.g., Agbabian et al.[12] used a time-domain identification procedure to detect structural changes on the basis of measurements of excitation and acceleration response. Cawley and Adams[2] have found that the change of stiffness, local or distributed, is the primary cause of decreasing natural frequencies. Later Cawley and Ray[13] confirmed the observation on the case of a beam with cracks and slots. Identification procedures have also been developed using frequency responses, e.g., Wu et al.[14] and Samman et al.[15]

In general, the goal of these studies was to recover the engineering or damage properties of structures from knowledge of a complete set or a subset of modal and spectral data. In recent years, the applications of neural networks have attracted increasing attention due to their capabilities such as pattern recognition, classification, function approximation, etc., and are well documented in the literature. The basic idea behind the proposed technique in this paper is to assess the structural integrity in an on-line mode through a neural-network-based identification process. A few researchers are applying neural networks in the field of damage assessment. Teboub and Hajela[16] employed the classification ability of a neural network to identify the damage in composite material beams. Wu et al.[14] used the pattern-matching capability of a neural network to recognize the location and the extent of individual member damage from the measured frequency spectrum of the damaged structure. Although their results look promising, there are some issues that need to be resolved and improved. The work presented in this paper draws on a similar idea but improvises on their work to deal with a situation of multiple damages. In addition, a new architecture for a neural network is proposed such that 1) the amount of input data is reduced, 2) the network provides better generalization estimations, and 3) it has the ability to detect minor damages.

As an illustration of this concept, the present paper examines two discrete multiple degree-of-freedom (DOF) linear systems using different information representations. In the

Received Oct. 8, 1992; presented as Paper 93-1708 at the AIAA/ASME/ASCE/AHS/ASC 34th Structures, Structural Dynamics, and Materials Conference, La Jolla, CA, April 19–21, 1993; revision received May 15, 1993; accepted for publication July 1, 1993.

*Graduate Research Associate, Department of Aeronautical and Astronautical Engineering. Student Member AIAA.
†Assistant Professor, Department of Aeronautical and Astronautical Engineering. Member AIAA.

example of a three-DOF spring-damper-mass system, the changes of eigenvalues of the damaged system are chosen to represent the damage information. An appropriate neural network is trained to recognize the features of damage information of the system in which spring members have sustained varying states of damages. With a limited amount of training data, the neural network is able to capture the general scope of the damage characteristics. However, this sort of data representation may not be a proper choice for real structural or mechanical systems that usually have closely spaced eigenvalues. To deal with this class of systems, a dynamic-residual vector proposed by Zimmerman and Kaouk[10] is adopted to represent the damage information. Also, a three subnet configuration of the neural network is proposed to realize the identification process. Kabe's eight-DOF spring-mass system[9] is demonstrated in the present idea. It will be shown that the network is capable of accomplishing such tasks and its performance is satisfactory for this specific model.

The next section of this paper presents a brief description of neural networks. Numerical results illustrating the use of neural networks in detecting and identifying damages are presented in Sec. III. The effectiveness and limitations of this approach, as well as the concluding remarks, are addressed in Sec. IV.

II. Brief Introduction to Neural Networks

An artificial neural network is a framework consisting of many numbers of interconnected neuronlike processing units. Each neuron unit is stimulated by the sum of the incoming weighted signals and transmits the activated response to the other connected neuron units. Such a network represents an efficient and parallel computational entity and will reflect the levels of simulations by different input signals. However, the process so far does not involve feedback and relaxation; in other words, it only has the ability to propagate the input information, which can be termed as a recognition process. Rosenblatt[17] developed the dynamic weight modification concept into the process of the neural network, which gave the neural networks the ability to "learn." Rumelhart and McClelland[18] further provided an excellent algorithm that allows the multilayer neural network to internally organize itself to be able to reconstruct the presented patterns. This method leads to the recent most popular neural network learning scheme called the back-propagation algorithm.

A typical architecture of a multilayer neural network is shown in Fig. 1. The input layer receives input patterns; it usually does not have processing units in this layer but simply transmits the signals faithfully to the next layer. The hidden layer or layers, residing between the input layer and the output layer, consist of a certain number of processing units. Each node in the preceding layer is fully connected to all processing units, and the connections are called the weights that represent a different weighting scale of the input signals. The processing unit sums up the weighted signals and activates a response transmitting to the next layer. The activation function can be monotonically increasing nonlinear (or linear) function. From

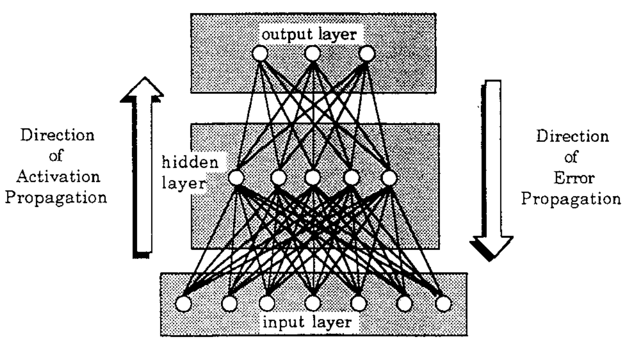

Fig. 1 Back-propagation neural network.

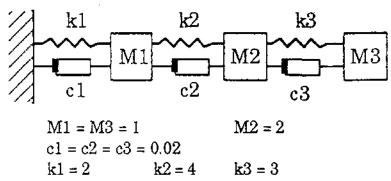

Fig. 2 Three-DOF mass-damper-spring system.

Kolmogorov's theorem[19] and Cybento's theorem,[20] it is known that any nonlinear mapping can be approximated by an appropriate combination of weights and activation functions and realized by the multilayer neural network. The input pattern is propagated forward, and calculated responses are obtained. The errors between the desired outputs and the calculated outputs then propagate backward through the network, providing vital information for weight adaptation. The back-propagation algorithm uses this information to adjust the weights such that a "mean-squared" error measure is minimized. This supervised learning algorithm, using a gradient descent optimization scheme, lets the network converge to a minimum in the weight space and completes the learning process.

In this paper, a back-propagation neural network simulation program called NETS (Version 2.0)[21] was used for the first example. This program was made available by NASA Johnson Space Center. The program was implemented on an APOLLO DN4500 workstation. The second example was done with an efficient alternative simulation code QUICKPROP written by Fahlman[22] and was implemented on a Cray Y-MP8/864 supercomputer.

III. Examples

Since most structural systems can be modeled by a discrete mass-damper-spring system, two spring-mass systems are used to explore the applicability of the proposed technique for structural damage identification problems.

A. Three-DOF Discrete System

For simplicity, a three-DOF model shown in Fig. 2 is first analyzed. The equation of motion of a three-DOF spring-damper-mass system is formulated as

$$M\ddot{x} + C\dot{x} + Kx = f \tag{1}$$

where

$$M = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \tag{2}$$

$$C = \begin{bmatrix} c_1 + c_2 & -c_2 & 0 \\ -c_2 & c_2 + c_3 & -c_3 \\ 0 & -c_3 & c_3 \end{bmatrix} \tag{3}$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \tag{4}$$

The task is to use the neural network to identify the locations and the severity if damage to the springs has occurred. That is, when the stiffness of the spring decreases because of the damage, say Δk_1, Δk_2, or Δk_3 (in percentage), the goal is to identify which spring and how much the spring stiffness changes from deducing the abnormal frequency responses of the discrete system.

The frequency responses of the three-DOF mass-damper-spring system have three spectrums in nature. An intuitive way of choosing the input pattern for the network learning is to use the magnitudes and/or phases of the sampling frequencies in sufficiently large spans of each or all spectrums. However, to characterize these spectral properties, a small sampling rate is required; in turn, a tremendous amount of sampling data will be needed, which significantly jeopardizes the efficiency and accuracy of the neural network training process. Thus, in this paper, the changes of the eigenvalues between the undamaged system and the damaged system are used as the input pattern. The choice was made based on the following advantages: 1) The length of the input pattern reduces to only three (compare with 200 nodes in Ref. 14). 2) The changes of the eigenvalues include two important pieces of information about frequency responses, i.e., natural frequencies and corresponding damping ratios.

The scenario of the proposed method is based on the assumptions that the masses and damping coefficients m_i and c_i are fixed and that the frequency responses of the systems are provided from the simulated measurements. The patterns are generated by calculating the absolute values of the differences between the eigenvalues of the undamaged system and those of the damaged system. Several cases are taken into account in our work. In general, these cases can be divided into two categories: single spring damage and multiple spring damages.

1. *Single Damage Identification*

In this category, the damage of the spring is simulated by losing its stiffness from 10 to 90% with 10% intervals. For every varying stiffness matrix, the generalized eigenvalues, which are slightly different from the referenced eigenvalues of the undamaged system, of Eq. (1) can be calculated by the standard generalized eigenvalue solver. The absolute values of the eigenvalue changes between the damaged and undamaged springs are inputs to the neural network, and its outputs are therefore the damaged springs and the corresponding stiffness losses. Then a three-layer neural network was trained by these training patterns with the back-propagation algorithm.[21] The training process continued until a prescribed convergent criterion was satisfied. Then a set of testing data corresponding to 5–95% with 10% interval damage states was propagated through the trained network to examine the generalization of the results.

Table 1 shows the final network outputs of the training process and the testing results for single spring damage cases. The convergent criterion for all three simulations is that the maximal pattern error should be less than 0.5%. The assumed damages are listed in columns 1 and 5 where column 1 indicates the training set and column 5 represents the testing set. The trained network outputs for each spring damage are tabulated in columns 2–4, respectively. Forty hidden neurons are used for each spring damage, and the training process takes about 80,000 iterations to learn the pattern representation. Although the results of training patterns show the precise estimation of the damage, it is not guaranteed that for other unknown damages the estimation would have the same accuracy. Therefore, it is necessary to test the generalization of the neural network. The middle points between every two consecutive training patterns, which reflect the possibly worst testing representation, are chosen to test the trained network. The testing results are shown in columns 6–8. In each entry, the number in the upper row is the network output and the number in parentheses is the relative percentage error. The results all lie in a satisfactory range of desired values. In other words, the neural network not only learns the training pairs but also establishes an accurate and unique input/output mapping for any possible damage. The largest discrepancies are those results of the 5 and 95% damage cases. This is because these two testing patterns are beyond the training representative range, which ranges from 10 to 90%. The estimation of both patterns could present a little distortion due to internal extrapolation of the trained mapping structure. However, the estimated results are still in an acceptable range (absolute error less than 0.02).

2. *Multiple Damages Identification*

In this category, the similar training process that applied in the single damage case is followed. Three ranges of assumed multiple damages were generated. They are as follows:

1) Case Kmix, wide-range variation of spring damages: Each spring has 0-5-10-20-40-80% damages, resulting in 216 combinations of spring damages. This case represents the broad scope of the identification problem.

2) Case Km01, short-range variation of spring damages: Each spring has 0-5-10% damages, resulting in 27 combinations. This case represents the minor damage identification problem.

3) Case Km45, short-range variation of spring damages: Each spring has 0-40-45-50% damages, resulting in 64 combinations. This case represents the identification resolution problem.

In each case, according to those damage combinations, the equation of motion was updated first and then generated a set of training patterns.

For cases of multiple spring damages, similar to the single spring damage, a three-layer back-propagation neural network was used. The numbers of input and output neurons are both three. The numbers of neurons in the hidden layer are 100, 60, and 40 for the Kmix, Km01, and Km45 cases, respectively. The convergent criteria for all three cases are that the

Table 1 Individual spring damage results

% damage	Δk_1	Δk_2	Δk_3	% damage	Δk_1	Δk_2	Δk_3
10	0.1026	0.1026	0.1034	5	0.0668	0.0659	0.0678
	(2.61)	(2.58)	(3.40)		(33.5)	(31.7)	(35.6)
20	0.1959	0.1962	0.1957	15	0.1465	0.1469	0.1468
	(2.05)	(1.90)	(2.15)		(2.24)	(2.07)	(2.16)
30	0.3008	0.3003	0.3004	25	0.2481	0.2480	0.2476
	(0.26)	(0.10)	(0.13)		(0.74)	(0.79)	(0.94)
40	0.4028	0.4024	0.4033	35	0.3526	0.3519	0.3526
	(0.70)	(0.65)	(0.82)		(0.73)	(0.55)	(0.73)
50	0.4999	0.5003	0.5009	45	0.4517	0.4517	0.4526
	(0.02)	(0.06)	(0.18)		(0.33)	(0.38)	(0.58)
60	0.5971	0.5977	0.5975	55	0.5480	0.5488	0.5489
	(0.48)	(0.38)	(0.43)		(0.42)	(0.23)	(0.19)
70	0.6995	0.6991	0.6991	65	0.6475	0.6477	0.6475
	(0.07)	(0.12)	(0.12)		(0.40)	(0.35)	(0.39)
80	0.8045	0.8037	0.8044	75	0.7523	0.7515	0.7519
	(0.56)	(0.46)	(0.50)		(0.32)	(0.20)	(0.24)
90	0.8968	0.8976	0.8976	85	0.8535	0.8534	0.8540
	(0.35)	(0.26)	(0.26)		(0.35)	(0.40)	(0.46)
				95	0.9321	0.9337	0.9326
Emax	0.004487	0.003892	0.004603		(2.08)	(1.72)	(1.83)

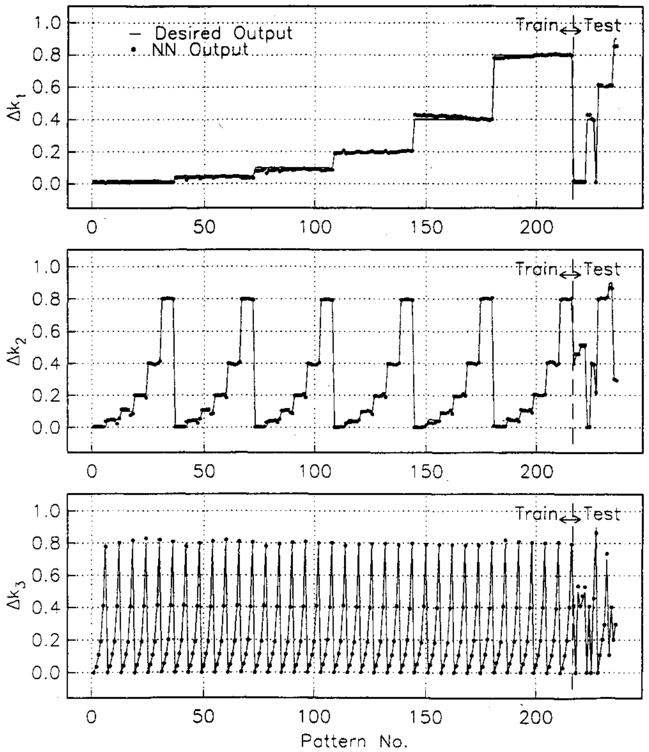

Fig. 3 Neural network outputs of case Kmix.

absolute pattern errors should be less than 3%. The training process for these cases takes about 60,000 to 70,000 iterations. After the network is trained, several patterns different from the training patterns are randomly chosen to test the performance of the network.

The simulation results for multiple damage cases are depicted in Figs. 3–5. Figure 3, which corresponds to the Kmix case, consists of 236 total patterns, including 216 training patterns followed by 20 testing patterns. For each testing pattern, the damage identification error is less than 0.03 except for those points that stand for 90% damage, at which the errors are still less than 0.05 (which corresponds to a relative percentage error of 5–5.5%). Nevertheless, this observation agrees with the result obtained in the single damage case because it anticipated the neural network's fair extrapolation ability to deal with the input data beyond the training range.

The result of the multiple damages identification for the Km01 case is shown in Fig. 4. One may observe that it seems to be less accurate than the previous case. The reason for the lag between the network output and the desired value is due to the 0.03 convergent criterion, which is about 30% of the training domain. However, the neural network has captured the feature of the damage information, and the results are in agreement with the desired tendency. There are a couple of remedies to improve the result: 1) toughen the convergent criterion, which will require a longer training time; and 2) increase the training patterns, which will be demonstrated on the next problem, the Km45 case. In Fig. 4, the last seven patterns are testing patterns. Parenthetically, the desired outputs are chosen to be the actual values in the training process. To compensate for the large relative error due to the coarse convergent criterion, the range of the desired outputs can be rescaled from [0, 0.1] to, say, [0.3, 0.7], which is more sensitive to the neuron's activation output. This adjustment is believed to be able to improve the performance of the neural network.

Since the preceding conjecture indicates that increasing the training patterns could improve the generalization capability of the neural network, the Km45 case was designed and tested for the multiple damage case. The results are very promising and are displayed in Fig. 5. The front portion of the curve in Fig. 5 is 64 training patterns and is followed by 26 testing patterns. One can observe that the neural network indeed produces very good approximations to the desired values, which confirms our conjecture. The pattern errors of testing data are all less than 0.0167 (relative error 3.55%).

From the preceding simulations of the three-DOF discrete system, it can be concluded that the application of the neural network to such a system is highly feasible. However, we should survey the frequency property of this example more carefully. It is understood that since the generalized eigenval-

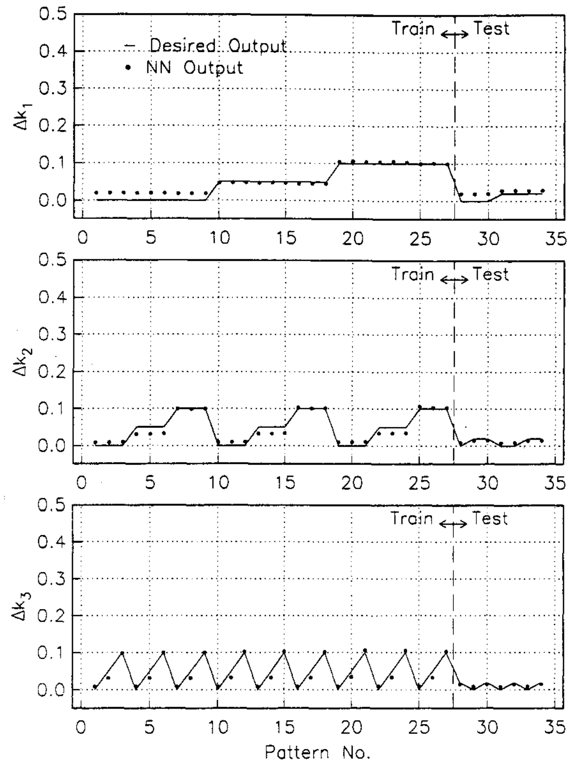

Fig. 4 Neural network outputs of case Km01.

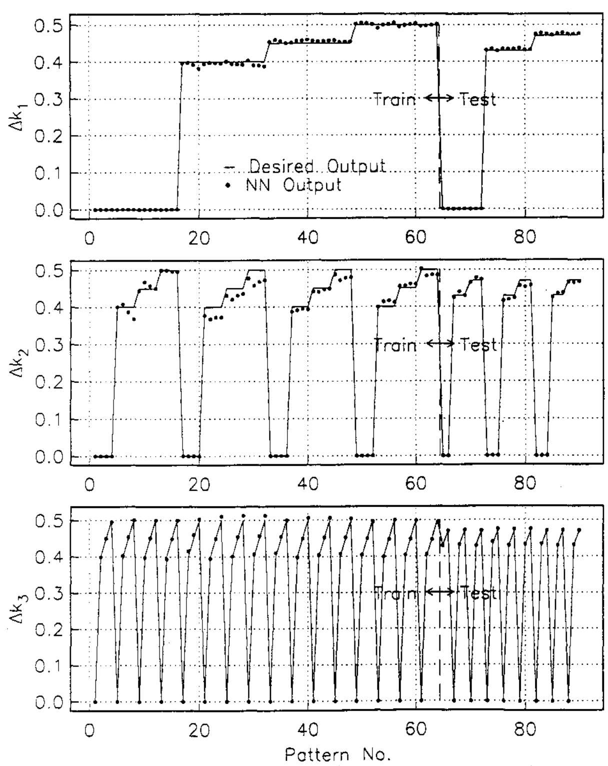

Fig. 5 Neural network outputs of case Km45.

ues of this system are sparsely distributed, the drifted eigenvalues due to decreasing spring stiffness will not cross over the lower eigenvalues. This phenomenon provides a simpler input/output mapping relationship and validates the proposed damage information representation. Conversely, for some physical systems that exist with a closely spaced eigenvalue distribution, the previous identification process may have difficulties in establishing proper interpolation of the damage patterns. For those kinds of physical systems, it is necessary to devise a new strategy of pattern representation. In the next subsection, Kabe's model,[9] a typical example of a closely spaced eigenvalue distribution physical system, will be investigated by a novel identification procedure.

B. Kabe's Problem

Kabe's eight-DOF spring-mass system is shown in Fig. 6. To simulate a challenging situation for the structural damage identification, the model was designed to generate the closely spaced frequencies in both local and global modes of vibration. This is achieved by using tremendous high stiffness and mass ratios as shown in Fig. 6 rather than the standard spring-mass system, which commonly used either very close stiffness or mass values.

1. *Subspace Rotation Algorithm*

Zimmerman and Kaouk[10] proposed a subspace rotation algorithm, which was derived from the generalized eigenvalue problem, to identify the location of a damaged spring in Kabe's system by only measuring the fundamental modal data. For a mass-spring model, the equation of motion can be expressed as

$$M\ddot{x} + Kx = 0 \tag{5}$$

The generalized eigenvalue problem of Eq. (5) is given by

$$K \cdot v_{h_i} = \lambda_{h_i} \cdot M \cdot v_{h_i} \tag{6}$$

where λ_{h_i} and v_{h_i} denote the ith eigenvalue and the corresponding eigenvector, respectively, of the undamaged structure system. Now consider that the ith eigenvalue λ_{d_i} and eigenvector v_{d_i} of the damaged structure are available. Then define a dynamic residual vector as follows:

$$d_i \equiv (K - \lambda_{d_i} \cdot M) \cdot v_{d_i} \tag{7}$$

Inspection of d_i in terms of the changed stiffness matrix reveals that the jth element of d_i will be zero if the jth degree of freedom is not affected by the damage. Conversely, a degree of freedom that is affected by the damage will result in a nonzero entry in the d_i vector. For instance, a spring connects the third and fourth degrees of freedom. The damage of this spring will cause the third and the fourth elements of the d_i vector to be nonzero. Thus, the location of damage can be determined by inspecting the affected degrees of freedom with nonzero entries in the d_i vector. Zimmerman and Kaouk's method provides a clever and simple way to detect the damage.

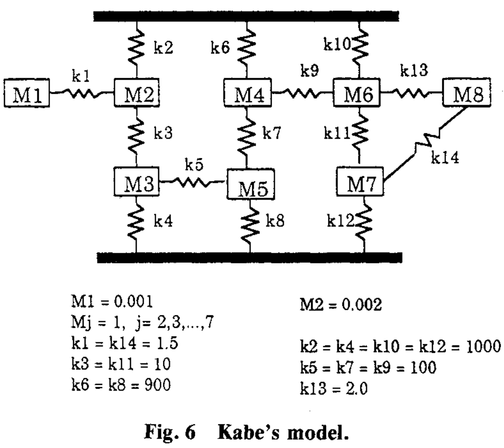

Fig. 6 Kabe's model.

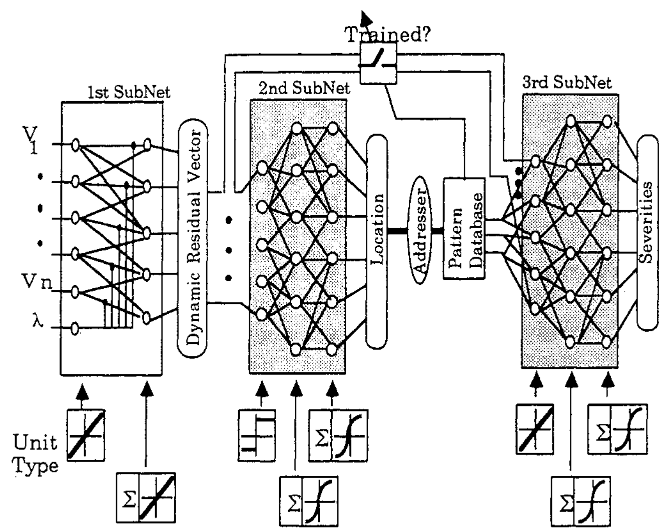

Fig. 7 Neural network architecture for structural damage identification.

2. *Application of Neural Networks to Kabe's System*

In this study, the preceding idea is employed to represent the damage information. The damage identification can be achieved by accomplishing the following tasks: 1) transforming the modal data (frequency and mode shape of the damaged system) to the dynamic residual vector, 2) identifying the locations of the damaged springs by applying the subspace rotation algorithm, and 3) determining the severity of the damage from measured data. The identification process exhibits two difficulties to be resolved: to translate the inspection of nonzero entries to numerical representation and to quantify the severity of damage from the given information. In light of the context of the problem at hand, a neural network presents itself as a logical tool. To realize these three operations, a neural network that consists of three subnets performing individual tasks is devised (see Fig. 7). As shown in Fig. 7, the first subnet is constructed of two layers whose inputs are the frequency and mode shape, and the weights are corresponding elements of the K and M matrices. Note here that the mode shape is normalized with respect to the M matrix such that $V^TMV = I$. The first subnet's output, the dynamic residual vector, is then evaluated based on the definition in Eq. (7).

Then the dynamic residual vector d_i is fed into the second subnet, which performs the learning process to identify the location of the damage. The input data are first transformed into the signed dynamic residual vector by posing the activation function on the input units of the second subnet, where the activation function is defined by

$$f(x) = \operatorname{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

The task is similar to a classification problem and is accomplished by a three-layer back-propagation neural network. The number of inputs to the second subnet is 8 in this example, and the output would be a 14-bit binary number in which each bit indicates a spring. The bit is 1 for a damaged spring and 0 for an undamaged spring. The subnet is pretrained by 105 training patterns that are generated by assuming any combination of two damaged springs, each with a 5% stiffness loss. This 5% is chosen such that it is desired that the proposed

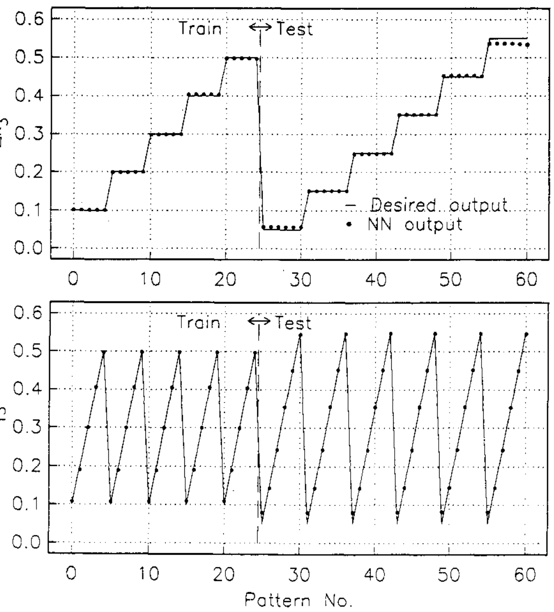

Fig. 8 NN outputs of Kabe's problem: two-spring damages.

neural network can identify at least 5% damage (which represents the resolution). Since the second subnet is pretrained, it can make a precise identification of the damage location at once. The output information would serve as an indicator for retrieving data for the third subnet. In addition, the outputs could be directed to a display device without further modification.

Finally, the third subnet utilizing the information given by the first two subnets calculates the severity of the damages. The process can be described in more detail as follows: at first, the 14-bit number obtained from the second subnet serves as an index of addresses in a pattern data base. The data base is prepared in advance by generating proper patterns corresponding to different combinations of assumed multiple damages. The number will trigger the third subnet to extract a set of training patterns in the data base at the indexed address. Then the subnet is trained on line by these training patterns. Since only a small number of the training patterns are associated with a damage, the on-line training process for the subnet is considered very fast. After the self-training, the dynamic residual vector d_i obtained from the first subnet is fed into the trained third subnet. Therefore, an estimation of the severity is obtained. It is important to note that this subnet performs an on-line learning first and then estimates the damage severity. This is different from the second subnet where it is trained in advance and identifies the damage location in real time.

A point should be mentioned in considering the learning process for the second subnet. If only the fundamental modal data are used, the patterns of the signed d vectors have over one-fifth repeatedness. More specifically, there are 11 pairs in these 105 training patterns that are identical patterns. This will cause ambiguous results of the damage location determination, and therefore the uniqueness of the damage identification is questionable. To reduce those indistinguishable patterns, it is suggested that the second modal data, or even the third modal data, should be included in the input data for training purposes. If the second modal data are also used, the identical pattern pairs reduce to two. Furthermore, if the first three modal data are used, all training patterns are distinguishable. Nevertheless, the tradeoff is the complexity of the complete network and the much longer training process for the second subnet. This observation agrees with the minimal modal data requirement suggested by Kabe.[9] A similar conclusion had been reached by Shen and Taylor.[8]

To demonstrate the results of the neural network, two simulations are presented. In the first simulation, the damages are assumed at k_3 and k_{13}. The first two modes are used as the input data (the pattern samples are listed in the Appendix). In the pattern data base, the damages of springs vary from 10 to 50% with 10% increments. Hence, for a case of two-spring damages, 25 patterns are retrieved for training the third subnet. To speed up the learning process, the number of hidden neurons of the third subnet is chosen to be 10. The on-line training was achieved at the stage that the absolute pattern error or the number of iterations are less than the specified values (0.5% and 500, respectively, were adopted in the present case). The input patterns corresponding to 5–55% with 10% increments of spring damage are generated and propagated through the trained network to test the performance of the network. The result is shown in Fig. 8 where the first 25 patterns are training patterns followed by 36 testing patterns. The maximal pattern errors for the testing set are 1.7 and 3.2% of k_3 and k_{13}, respectively. One can observe that the network is capable of providing an accurate identification of the damages.

To extend further, a case of three-spring damages is considered. It is assumed that the possible damages occurred at k_1, k_{13}, and k_{14}. As the number of damaged springs increases, the combinatorial problem will be dominant in generating the data base. To explore the compromise, we examined two exercises. First, the damage states are given at 0-15-30%. Thus, 27 patterns for each combination are generated. The simulation result is shown in Fig. 9, where the first 27 points are training patterns, followed by 64 testing patterns, which represent 2-10-20-28% damage possibilities. It is evident that the discrepancies on the testing results are outstanding. Hence, the damage states are chosen as 0-10-20-30%, and 64 patterns per combination are stored in the data base. Two sets of testing data, one corresponding to 5-15-25-35% damages for verifying the generalization and the other corresponding to 3-13-23-38% damages for checking the resolution, are examined. The result is shown in Fig. 10 where the first 64 points are the training set, followed by 64 patterns of the first testing set and 64 patterns of the second testing set. Except for the points representing 35% damage, the calculated results are relatively good. The abnormal phenomenon for the 35% damage case is expected because the damage severity is beyond the

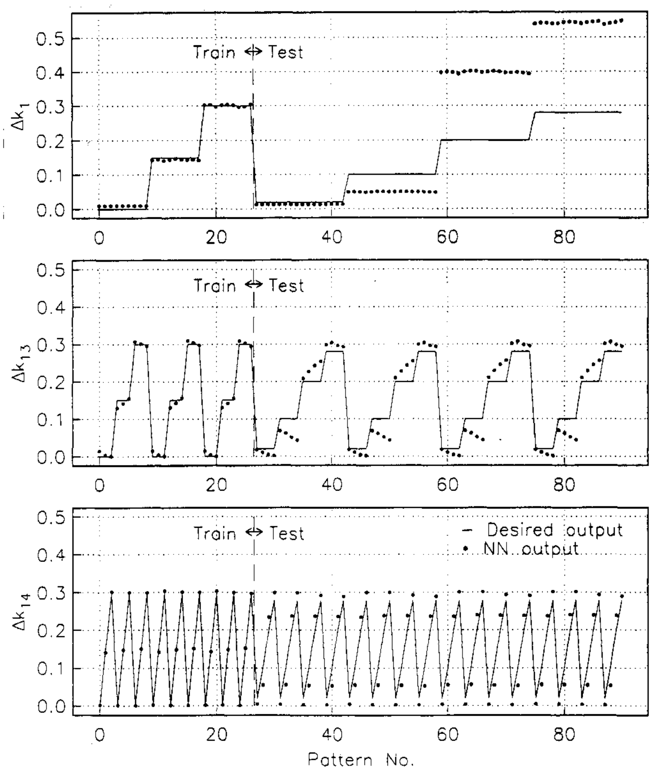

Fig. 9 NN outputs of Kabe's problem with 0-15-30% training.

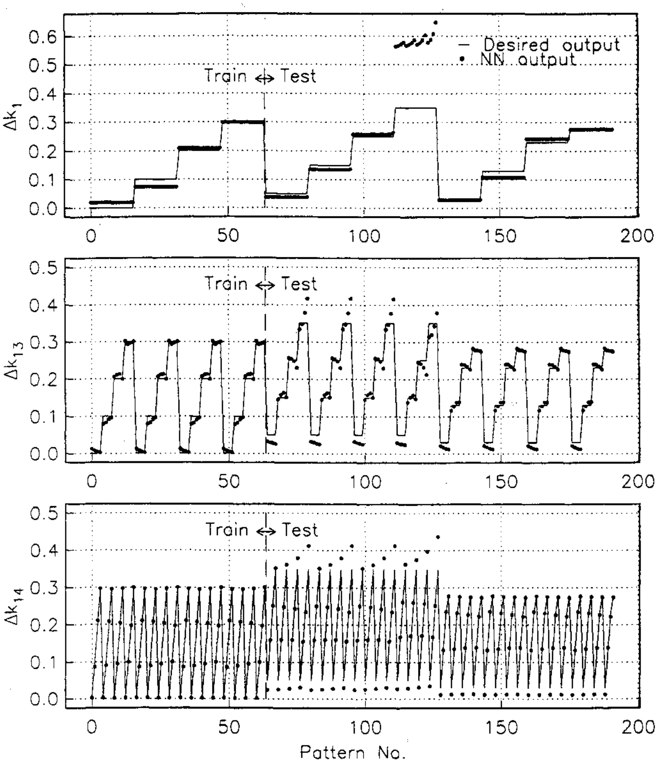

Fig. 10 NN outputs of Kabe's problem with 0-10-20-30% training.

representative domain of the neural network. Moreover, the result of the second testing set shows that the neural network did establish a good resolution of the damage representation. The maximal pattern errors for k_1, k_{13}, and k_{14} are 2.3, 2.8, and 2.5%, respectively (except for 35% points).

One point that should be addressed is the fact that the back-propagation neural network has no inherent ability to indicate when it is functioning outside the domain over which it was trained. Consequently, the extrapolation performance of the neural network may be fairly poor as shown in Figs. 9 and 10. This disadvantage may be overcome by using radial basis function networks,[23] which is beyond the scope of this paper.

IV. Concluding Remarks

The application of the artificial neural networks to the problem of structural damage detection and identification is explored. Since the neural network is capable of learning the features of the damage information, this study has shown that such an application is quite feasible.

With a limited number of training patterns, the neural network matches not only the training patterns but also the testing patterns. Since the testing patterns are selected on the midspan between every training pattern pair, it is concluded that the neural network generalizes the satisfactory results throughout the domain that the training data span. However, when the input testing pattern is beyond the representative domain, the neural network may fail to extrapolate such a pattern. Nevertheless, due to the separate network output, the damage location is still accurately identified when the testing data are within or outside the domain.

Two discrete models are demonstrated. For a three-DOF mass-damper-spring system, the three-layer back-propagation neural network is capable of identifying the single spring damage within 0.5% and the mixed spring damages within 3%. The neural network generalizes the mapping from the frequency changes to damage amounts very well. As a real structural system existing with a closely spaced eigenvalue spectrum, the sample data are no longer uniquely mappable. The conventional three-layer network unavoidably causes some unacceptable error. A new structural damage identification procedure constructed by an eight-layer network architecture is proposed and demonstrated on the well-known Kabe model. The location of the damaged springs were precisely identified, and the corresponding damage severities were also determined within 3% error.

The proposed approach provides the following advantages. 1) The neural network can identify the locations and the severity of the damages for either single spring damage or multiple spring damages. 2) Since the network is trained in the design stage, when the network is hardware implemented on the structural system, it can respond to the detection in real time, which makes the neural network very suitable for a real-time on-line structure health monitoring device. 3) The training process is computationally easy and can be implemented in parallel processors. 4) The network shows the potential use for structural damage identification.

Some issues remain to be resolved before this approach becomes a truly viable method for structural damage identification. First, for real-world complex structures, the degree of freedom of the system could be very large. It is believed that by performing several stages of model reduction the DOF of the final design model should be a relatively small number. Second, the approach using the dynamic residual vector may be applied to such reduced models. Third, the role of noise in actual modal data measurement should be investigated; where minor damage is concerned, the noise could cause some false estimations. Finally, what is the appropriate representation for damage information to reliably and precisely identify various states of damages requires further investigation. It is anticipated that a neural-network-based identification technique will be improved if a proper representation can be developed.

Appendix

For reference purpose, the input/output patterns for the simulation examples presented in the paper are sampled and listed next.

Example 1

For the single damage case, assuming k_3 is damaged, the numbers of the inputs and the output are 3 and 1, respectively. The training patterns are

$$\begin{array}{cccc} 0.024251 & 0.075986 & 0.001154 & 0.1 \\ 0.052197 & 0.144074 & 0.002107 & 0.2 \\ & \vdots & & \end{array}$$

The testing patterns would be

$$\begin{array}{cccc} 0.011685 & 0.038993 & 0.000606 & 0.05 \\ 0.037741 & 0.111006 & 0.001652 & 0.15 \\ & \vdots & & \end{array}$$

For multiple damages cases, the numbers of the inputs and the outputs are 3 and 3, respectively. The samples of the patterns are

$$\begin{array}{cccccc} 0.430977 & 0.190055 & 0.047652 & 0.10 & 0.40 & 0.20 \\ 0.469061 & 0.516377 & 0.018584 & 0.00 & 0.20 & 0.90 \end{array}$$

Example 2

Using $\Delta k_3 = 10\%$ and $\Delta k_{13} = 10\%$ as an illustration, the dynamic residual vector obtained from the first subnet looks like this:

$$d_1 = [0 \quad -0.311911 \quad 0.311911 \quad 0 \quad 0 \quad -0.075872 \quad 0 \quad 0.075872]$$

The training pattern for the second subnet would be like this:

Input: $0 \ -1 \ 1 \ 0 \ 0 \ -1 \ 0 \ 1$

Output: $0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0$

The training pattern stored in the data base has the following format:

0 − 0.636420 0.636420 0 0 − 0.032383 0 0.032383 0.2 0.1
0 − 0.636319 0.636319 0 0 − 0.075678 0 0.075678 0.2 0.2

Acknowledgments

The support of the Engineering Foundation through the 1991 Research Initiation Award is gratefully acknowledged. The authors also wish to acknowledge the support from the Flight Dynamics Directorate of the Wright Laboratory at Wright-Patterson Air Force Base.

References

[1]Mayes, I. W., and Davies, W. G. R., "Analysis of the Response of a Multi-Rotor-Bearing System Containing a Transverse Crack in a Rotor," *Journal of Vibration, Acoustics, Stress, and Reliability in Design*, Vol. 106, No. 1, 1984, pp. 139–145.

[2]Cawley, P., and Adams, R. D., "The Location of Defects in Structures from Measurements of Natural Frequencies," *Journal of Strain Analysis*, Vol. 14, No. 2, 1979, pp. 49–57.

[3]Dimarogonas, A. D., and Papadopoulos, C. A., "Vibration of Cracked Shafts in Bending," *Journal of Sound and Vibration*, Vol. 91, No. 4, 1983, pp. 583–593.

[4]Gudmundson, P., "The Dynamic Behaviour of Slender Structures with Cross-Sectional Cracks," *Journal of the Mechanics and Physics of Solids*, Vol. 31, 1983, pp. 329–345.

[5]Shen, M.-H. H., and Pierre, C., "Natural Modes of Bernoulli-Euler Beams with Symmetric Cracks," *Journal of Sound and Vibration*, Vol. 138, No. 1, 1990, pp. 1154–134.

[6]Shen, M.-H. H., and Pierre, C., "Natural Modes of Bernoulli-Euler Beams with a Single Edge Crack," *Journal of Sound and Vibration* (to be published).

[7]Gladwell, G. M. L., Englang, A. H., and Wang, D., "Examples of Reconstruction of an Euler-Bernoulli Beam from Spectral Data," *Journal of Sound and Vibration*, Vol. 119, No. 1, 1987, pp. 81–94.

[8]Shen, M.-H. H., and Taylor, J. E., "An Identification Problem for Vibrating Cracked Beams," *Journal of Sound and Vibration*, Vol. 150, No. 3, 1991, pp. 457–484.

[9]Kabe, A. M., "Stiffness Matrix Adjustment Using Mode Data," *AIAA Journal*, Vol. 23, No. 9, 1985, pp. 1431–1436.

[10]Zimmerman, D. C., and Kaouk, M., "Structural Damage Detection Using a Subspace Rotation Algorithm," *Proceedings of the AIAA/ASME/ASCE/AHS/ASC 33rd Structures, Structural Dynamics, and Materials Conference*, AIAA, Washington, DC, 1992, pp. 2341–2350.

[11]Zimmerman, D. C., and Widengren, W., "Finite Element Models Correction Using a Symmetric Eigenstructure Assignment Technique," *AIAA Journal*, Vol. 28, No. 9, 1990, pp. 1670–1676.

[12]Agbabian, M. S., Masri, S. F., Miller, R. K., and Caughey, T. K., "System Identification Approach to Detection of Structural Changes," *Journal of Engineering Mechanics*, Vol. 117, No. 2, 1991, pp. 370–390.

[13]Cawley, P., and Ray, R., "A Comparison of the Natural Frequency Changes Produced by Cracks and Slots," *Journal of Vibration, Acoustics, Stress, and Reliability in Design*, Vol. 10, No. 3, 1988, pp. 366–370.

[14]Wu, X., Ghaboussi, J., and Garrett, J. H., Jr., "Use of Neural Networks in Detection of Structural Damage," *Computers & Structures*, Vol. 42, No. 4, 1992, pp. 649–659.

[15]Samman, M. M., Biswas, M., and Pandey, A. K., "Employing Pattern Recognition for Detecting Cracks in a Bridge Model," *International Journal of Analytical and Experimental Model Analysis*, Vol. 6, No. 1, 1991, pp. 35–44.

[16]Teboub, Y., and Hajela, P., "A Neural Network Based Damage Analysis of Smart Composite Beams," AIAA Paper 92-4685, Sept. 1992.

[17]Rosenblatt, F., *Principles of Neurodynamics*, Spartan, New York, 1962.

[18]Rumelhart, D. E., and McClelland, J. L. (eds.), *Parallel Distributed Processing, Vol. 1: Foundations*, MIT Press, Cambridge, MA, 1986.

[19]Kolmogorov, A. N., "On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition," *Doklady Akademii Nauk SSR*, Vol. 114, 1957, pp. 953–956.

[20]Cybento, G., "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control Signals and Systems*, Springer-Verlag, New York, 1989.

[21]Baffes, P. T., *NETS User's Guide*, NASA Johnson Space Center, Sept. 1989.

[22]Fahlman, S. E., "Faster-Learning Variations on Back-Propagation: An Empirical Study," *Proceedings of 1988 Connectionist Models Summer School*, Morgan Kaufmann, San Mateo, CA, 1988, pp. 38–51.

[23]Leonard, J. A., Kramer, M. A., and Ungar, L. H., "Using Radial Basis Functions to Approximate a Function and Its Error Bounds," *IEEE Transactions on Neural Networks*, Vol. 3, No. 4, 1992, pp. 624–627.