Using SDM to Train Neural Networks for Solving Modal Sensitivity Problems

Brian J. Schwarz, Patrick L. McHargue, & Mark H. Richardson
Vibrant Technology, Inc.
18141 Main Street
Jamestown, California 95327

ABSTRACT

*The Structural Dynamics Modification (SDM) algorithm is very useful for solving the so-called **forward variational problem** for structures. That is, given changes in a structure's mass, stiffness, or damping properties, SDM efficiently yields the corresponding changes in its modal properties.*

*There are several important classes of problems, however, that require solutions to the **inverse variational problem**, or modal sensitivity problem. That is, given changes in a structure's modal properties, what corresponding changes in its mass, stiffness, and damping properties have taken place. Applications such as structural damage detection, finite element model updating using test data, and vibration suppression or control through structural modification all require solutions to the modal sensitivity problem.*

Unlike the forward variational problem, the modal sensitivity problem cannot be solved in a straightforward manner. For most practical cases, its solution requires the inversion of a rank deficient matrix, which creates numerical difficulties.

Neural networks offer promise for solving the modal sensitivity problem, because of their pattern recognition and interpolation capabilities. In order to solve an inverse variational problem, however, a neural network must be "trained" using a set of solutions to its corresponding forward variational problem. Training a neural network typically requires hundreds, even thousands of solution sets. In this paper we show how SDM can be used to train a neural network for solving the modal sensitivity problem. Because SDM only requires the modal parameters of the structure, which can be obtained from a modal test or a finite element model, this method can be applied in a wide variety of experimental and analytical cases.

NOMENCLATURE

t = time variable (seconds).

jω = frequency variable (radians/second).

n = number of measured DOFs.

[M] = (n by n) mass matrix (force/unit of acceleration).

[C] = (n by n) damping matrix (force/unit of velocity).

[K] = (n by n) stiffness matrix(force/unit of displacement).

[$\underline{M}$] = [M] + [ΔM] = (n by n) mass matrix of modified structure.

[$\underline{C}$] = [C] + [ΔC] = (n by n) damping matrix of modified structure.

[$\underline{K}$] = [K] + [ΔK] = (n by n) stiffness matrix of modified structure.

[ΔM] = (n by n) matrix of mass changes.

[ΔC] = (n by n) matrix of damping changes.

[ΔK] = (n by n) matrix of stiffness changes.

$\{\ddot{x}(t)\}$ = acceleration response n-vector.

$\{\dot{x}(t)\}$ = velocity response n-vector.

$\{x(t)\}$ = displacement response n-vector.

$\{f(t)\}$ = excitation force n-vector.

{X(jω)} = discrete Fourier transform of the displacement response n-vector.

INTRODUCTION

The underlying assumption of the modal sensitivity problem is that changes in the vibration characteristics (modal properties) of a structure are *strongly coupled* to changes in its physical properties. Modal testing itself assumes that the structure remains in a stationary condition throughout the test. That is, its modal properties must not change. Many times, this condition is difficult to maintain.

Most experimentalists have encountered mass loading effects during a modal test. The apparently insignificant mass of the measurement transducers (e.g. accelerometers) causes the modal frequencies to shift as transducers are moved from one point to another on the structure. Temperature changes during the course of an all day modal test can also cause modal frequencies to shift.

For all but the simplest of cases, matching boundary conditions between finite element analysis and modal test presents major difficulties. When the boundary conditions prescribed in the finite element analysis cannot be duplicated in the laboratory, the experimental modes don't match the analytical modes. More complex physical changes, such as the gravitational effects on joint stiffnesses in spacecraft, or the complex aerodynamic interactions of flight flutter in aircraft, also cause changes in the structural modes of vibration.

In recent years, numerous researchers have proposed solutions to the modal sensitivity problem. Entire conferences have been held of the subject of structure damage detection using vibrational changes [11]. As modal testing has become more widespread, finite element model updating using modal test data has also become popular.

Vibration problems are often caused when one or more modes are located too near to an operating frequency or excitation frequency of a machine or structure. The difficulty with problems of this type is that moving one mode, (by adding a stiffener, for instance), also causes other modes to shift in frequency, thus causing a different, perhaps greater vibration problem. One of our major expectations for this trained neural network is that it will find stiffness solutions that change the frequency of a problem mode without also causing the frequencies of other modes to shift to other problem areas.

The authors have tried to solve modal sensitivity problem in the past, using pseudo-inverse [3]-[5], and other approximation methods [6], but with limited success. Neural networks have also been applied to this problem [7] - [9], but in a different manner than is described here.

THEORETICAL BACKGROUND

To re-state the two problems under consideration:

Forward Variational Problem Given changes in a structure's mass, stiffness, and damping properties, find the corresponding changes it its modal properties.

Inverse Variational Problem Give changes a structure's modal properties, find the corresponding changes in its mass, stiffness, and damping properties.

If the equations of motion of the structure are stated as the usual statement of Newton's second law,

$$[\mathbf{M}]\{\ddot{\mathbf{x}}(\mathbf{t})\}+[\mathbf{C}]\{\dot{\mathbf{x}}(\mathbf{t})\}+[\mathbf{K}]\{\mathbf{x}(\mathbf{t})\}=\{\mathbf{f}(\mathbf{t})\} \qquad (1)$$

then the solution to the forward variational problem can be stated as the *unique solution* to a matrix eigenvalue problem,

$$\Big[[\mathbf{M}](\mathbf{j}\omega)^2+[\mathbf{C}](\mathbf{j}\omega)+[\mathbf{K}]\Big]\{\mathbf{X}(\mathbf{j}\omega)\}=\{0\} \qquad (2)$$

Notice that the above equation contains the matrices of the modified structure, and that the equations are homogeneous (external forces are zero). The above equation can be solved, either in physical space or modal space, for the modes of the modified structure. Most finite element analysis software packages solve this problem in physical space. The SDM algorithm solves it in modal space, using the modes of the unmodified structure instead of the unmodified mass, stiffness, and damping matrices [2].

The primary advantage, and assumption, of the SDM approach is that the dynamics of the unmodified structure can be adequately represented by relatively few of its fundamental (lowest frequency) modes. *The solution of the forward variational problem can also be considered as a transformation from physical space to modal space.*

Representation of structural dynamics in terms of mass, stiffness, and damping matrices typically requires much more data than a modal representation. For example, to represent the dynamics of a structure with a 1000 DOF model using its first ten (normal) modes would require,

(10 modes) X (1000 DOFs per shape + frequency + damping) = 10,020 numbers.

(The use of complex modes would approximately double this number). To represent the same structure dynamically using real symmetric mass, stiffness, and damping matrices would require,

3 matrices X (1000 X (1000 + 1) / 2) = 1,501,500 numbers.

In other words, *the physical model requires 150 times as much data as the modal model!* This comparison also illustrates the difficulty of solving the inverse variational problem, or *of transforming from modal space back to physical space*.

Real world structures have an infinite number of degrees of freedom. Therefore, to perfectly match the dynamics of a real structure, equation (2) would require infinite dimensional matrices, and would yield an infinite number of modes. In practice, of course, we *approximate* the dynamics of infinite structures using finite dimensional matrices and finite numbers of modes.

Nevertheless, solving the inverse variational problem, requires transforming from modal coordinates (with relatively little data), to physical coordinates (with large matrix representations), even for simple cases.

Any direct solution of the inverse variational problem requires the inversion of the flexibility matrix (or an equivalent operation), to obtain the stiffness matrix [1]. Since only a relatively small number of the modes of a real structure are usually known (are measured), the flexibility matrix will be rank deficient, and matrix inversion is impossible. So, we seek a solution to the inverse variational problem which preserves the unique relationship between changes in modal and physical properties, and provides discrete mass, stiffness, and damping changes, instead of the matrices themselves.

In the past few years, "standardized" neural network software has become available that can be used "right out of the box". For this work, the NeuralWindows software by

Ward Systems Group [10] was used in a Visual Basic program that performed the rest of the processing necessary to train the network.

To quote the NeuralWindows operating manual, "Neural networks excel at problem diagnosis, decision making, prediction, and other classifying problems where pattern recognition is important and precise computational answers are not required."

Neural networks are made up of "neurons" arranged in layers. A neuron is a simple input-output device with a built-in transfer function. In a simple feed forward network, the output of each neuron in one layer in linked to the input on each neuron in the succeeding layer. Inputs to the entire network are fed into the neurons in the first (input) layer. Outputs (answers) from the neural network are the outputs of the last (output) layer.

For this application, we used a three layer network, with an input, an output, and a hidden layer. Network training was done using the Backpropagation of errors method.

MODAL SENSITIVITY PROBLEM

In structures, stiffness modifications cause modal frequency changes, mass modifications cause modal frequency and damping changes, and damping modifications cause modal frequency and damping changes. In all cases, mode shapes may also change.

To simplify this analysis, we will focus only on stiffness changes, although the method is equally valid for mass and damping changes. When a local stiffness modification is applied to a structure, the mode shapes dictate which modes will "absorb" the modification, and hence be most sensitive to the modification. Therefore, realizing that mode shapes also play a significant role in stiffness modifications, we can define a modal sensitivity problem as follows:

Stiffness Sensitivity Problem

Given a set of modal frequencies different from those of a baseline (unmodified) structure, find the stiffness changes required to yield the new modal frequencies.

This definition indicates how the neural network should be setup in order to solve the problem. Each input neuron represents a modal frequency of the modified structure. Each output neuron represents a DOF pair and will yield an amount of stiffness change.

The number of input neurons should always be within a reasonable range, (1 to 50). On the other hand, for large structure models (with hundreds of modal test points or finite element nodes), output neurons cannot represent all of the possible DOF pairs, which could range into the thousands. A practical neural network size should contain perhaps hundreds of outputs, of DOF pairs. This means that a subset of DOFs must be chosen for the neural network. However, this requirement is often a benefit, since most realistic stiffness changes can only be made between a subset of DOFs.

ELIMINATING CONFLICTING TRAINING SETS

Since we are only training the neural network with modal frequencies and stiffness changes, it is possible that two or more sets of different stiffness changes will yield the *same set* of modal frequencies. This will *always* be the case when modification DOFs are chosen that correspond to nodal points (zero magnitudes) of the mode shapes involved. If two training sets have different stiffness changes but the same modal frequencies, one of them must be eliminated.

We used the Modal Assurance Criterion (MAC) to determine whether or not two sets of modal frequencies are the same. MAC was developed for comparing two mode shapes., and is essentially a projection of one vector onto the other. In this case, we simply assembled the modal frequencies of each training set into a vector, and applied MAC to all pairs of training set vectors. Any pair of training sets with a MAC value above 99% was considered to have the same frequencies.

Once a pair of training sets with the same frequencies is found, one set has to be eliminated in order to avoid *confusing* the neural network. Another criterion must be used to determine which of the two sets is *more desirable*. We chose the set with the *minimum* stiffness metric, defined by:

Stiffness Metric = SUM (Stiffness Change Magnitudes)

This metric will eliminate all training sets that cause no changes in the modal frequencies, When one of them is compared to the baseline training set, which has a Stiffness Metric = 0, it will always be less desirable than the baseline set.

AN EXAMPLE

To demonstrate this method, the 2 DOF lumped parameter model shown in Figure 1 was used. This model has a closed form solution for modal frequencies [12].

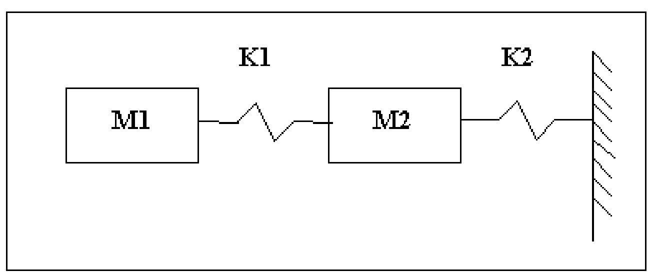

Figure 1.

Using this model allowed us to create training sets independently of SDM, and compare the training sets with those given by SDM.

This model has modal frequencies:

$$F1 = 12.355 \text{ Hz}$$

$$F2 = 52.171 \text{ Hz}$$

Training Data To train the neural network, the spring stiffnesses were varied over the ranges:

$$\Delta K1 \in [-12348, 12154]$$

$$\Delta K2 \in [-30240, 29760]$$

where each stiffness change is relative to the baseline stiffnesses. The neural network was trained using a total of 10000 training sets. Each stiffness change was varied in 100 increments between its minimum and maximum values. The stiffness changes yielded frequencies in the ranges:

$$F1 \in [\ 2.51, 17.08\ ] \text{ Hz}$$

$$F2 \in [\ 31.93, 66.28\ ] \text{ Hz}$$

Neural Network Predictions After the neural network was trained, its was fed 10000 pairs of modal frequencies, and it output stiffness changes. The percentage error between the actual stiffness changes and the neural network predictions are plotted in Figure 2., where the error was computed as the ratio of stiffnesses:

Error = Predicted Change / Actual Change

The worst error for the K1 stiffnesses is about 6.5 %, for the K2 stiffnesses about 9%.

CONCLUSIONS

This approach offers much promise for solving realistic modal sensitivity problems in practical situations. The main difficulty with solving this problem is having sufficient modal data to train a network. In a testing situation, a few sufficiently accurate modal frequencies can usually be obtained by curve fitting measurement data. Sufficiently accurate mode shape data can usually be obtained as well. Modal damping is the most difficult parameter to accurately estimate from test data.

We have shown with the use of a simple example that a neural network can be trained to give back usable stiffness results by only using modal frequencies and stiffness changes for training. We have also pointed out that conflicting training sets can result, where two different stiffness changes yield the same set of modal frequencies. When this occurs, one of the two conflicting sets must be eliminated to avoid confusing the neural network.

Another way to resolve this ambiguity in training sets is to use mode shape data also as inputs. Since the shape values will often be different even when the frequencies are not, this will add uniqueness to each training set.

The difficulty with using mode shapes as inputs is that the size of the network will quickly grow larger, lengthening the training time. To minimize this impact, only those shape values corresponding to the stiffness DOF pairs (output neurons) should be considered first. In the case of the 2 DOF example used here, adding the mode shapes as inputs would only increase the number of input neurons from 2 to 6.

In summary, neural networks appear to do a good job of solving the modal sensitivity problem, if properly trained. The SDM algorithm provides a straightforward way of providing training data sets for a neural network. The combination of these two methods should provide a practical tool for solving a variety of vibration trouble shooting, finite element model updating, and structural fault detection problems.

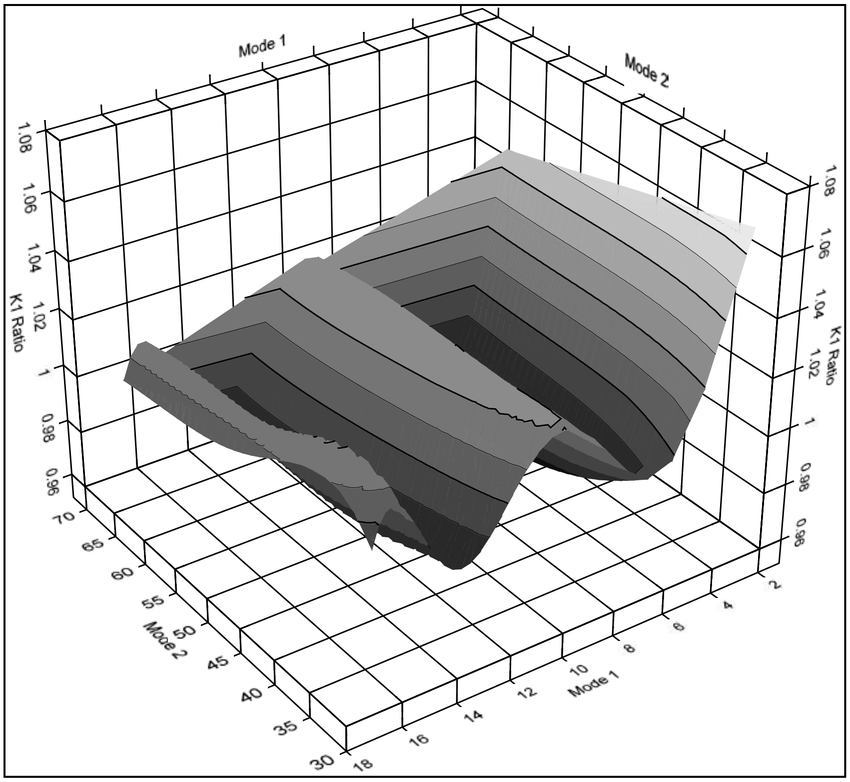

Figure 2.A K1 Ratio (Predicted/Exact)

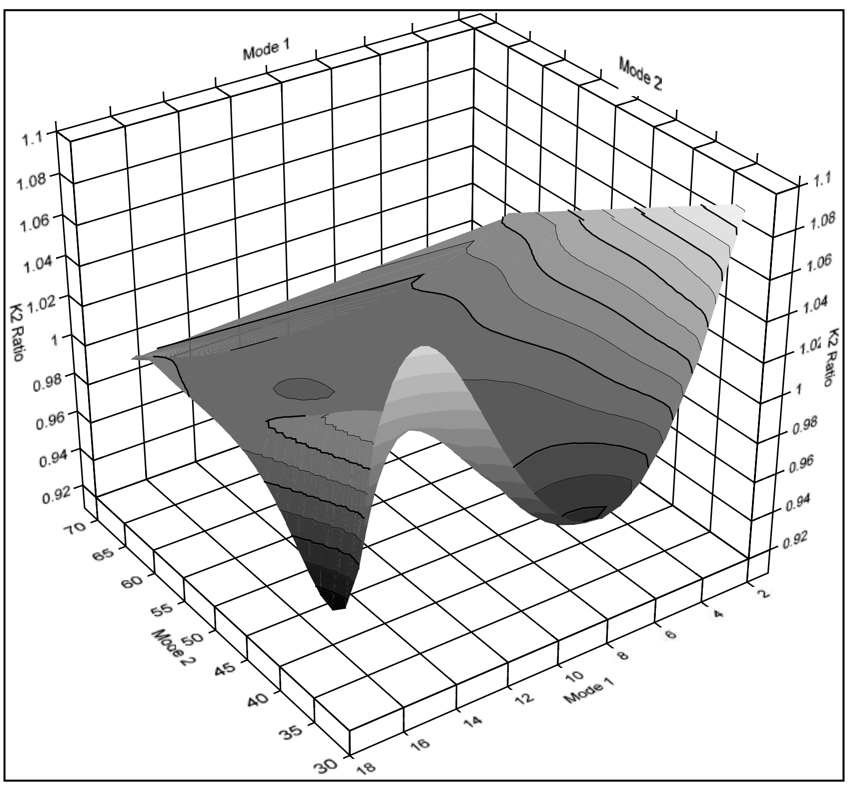

Figure 2.B K2 Ratio (Predicted/Exact)

REFERENCES

[1] Potter, R. and Richardson, M.H. **"Mass, Stiffness and Damping Matrices from Measured Modal Parameters"**, I.S.A. International Instrumentation - Automation Conference, New York, New York, October 1974.

[2] Wallack, P., Skoog, P., and Richardson, M. H. **"Simultaneous Structural Dynamics Modification (S^2DM)"** 6th IMAC Proceedings, Kissimmee, Florida, February, 1988.

[3] Mannan, M.A. and Richardson, M.H. **"Detection and Location of Structural Cracks using FRF Measurements"** 8th IMAC Proceedings, Kissimmee Florida, February, 1990.

[4] Mannan, M.A. and Richardson, M.H. **"Determination of Modal Sensitivity Functions for Location of Structural Faults"** 9th IMAC Proceedings, Florence, Italy, April, 1991.

[5] Richardson, M. and Mannan, M.A. **"Remote Detection and Location of Structural Faults using Modal Parameters"** 10th IMAC Proceedings, San Diego, California, February, 1992.

[6] Richardson, M. and Mannan, M.A. **"Correlating Minute Structural Faults with Changes in Modal Parameters"** 11th IMAC Proceedings, Kissimmee, Florida, February, 1993.

[7] Kudva, J., Munir, N. and Tan, P.W. **"Damage Detection in Smart Structures using Neural Networks"** Smart Materials and Structures, Vol.1, p100-112, 1992.

[8] Worden, K. Ball, A.D. and Tomlinson, G.R. **"Fault Location in a Framework Structure using Neural Networks"** Smart Materials and Structures Vol.2, p189-200, 1993.

[9] Worden, K. and Tomlinson, G.R. **"Damage Location and Quantification using Neural Networks"** Engineering Integrity Assessment ed. J.H.Edwards, J.Kerr and P.Stanley p11-31, 1994.

[10] **"NeuroWindows Neural Network Dynamic Link Library"** Ward Systems Group, Inc., Frederick, Maryland, 21702 (301) 662-7950

[11] **"Structural Damage ID Workshop"** Charles Farrar, coordinator., Los Alamos National Laboratory, Los Alamos, New Mexico, September, 1995

[12] Blevins, Robert D. Robert E. **"Formulas for Natural Frequency and Mode Shape"** Kreiger Publishing Co., Malabar, FL. 1984