Characterization of Nonlinear Dynamic Systems Using Artificial Neural Networks

Angel Urbina
FAST Center for Structural Integrity
The University of Texas at El Paso
El Paso, Texas
(915) 747-7891
e-mail: aurbina333@aol.com

Norman F. Hunter
Engineering Science and Analysis Division
Los Alamos National Laboratory
Los Alamos, New Mexico
(505) 667-2099
e-mail: hunter@lanl.gov

Thomas L. Paez
Experimental Structural Dynamics Department
Sandia National Laboratories
Albuquerque, New Mexico
(505) 844-7052
e-mail: tlpaez@sandia.gov

The efficient characterization of nonlinear systems is an important goal of vibration and modal testing. We build a nonlinear system model based on the acceleration time series response of a single input, multiple output system. A series of local linear models are used as a template to train artificial neural networks (ANNs). The trained ANNs map measured time series responses into states of a nonlinear system. Another ANN propagates response states in time, and a third ANN inverts the original map, transforming states into acceleration predictions in the measurement domain. The technique is illustrated using a nonlinear oscillator, in which quadratic and cubic stiffness terms play a major part in the system's response. Reasonable maps are obtained for the states, and accurate, long-term response predictions are made for data outside the training data set.

INTRODUCTION

There has recently been an increase in the use of sophisticated computational schemes to solve traditional problems, such as the modeling of dynamic systems. An important tool underlying one such scheme is the artificial neural network (ANN). The ANN is an inductive tool that attempts to "learn" the behavior of a system through the repeated presentation of exemplars during the so-called training phase. This approach has been applied to the field of system modeling by Paez et al., (1997a, 1997b) and Lapedes et al., (1987). In each of these works, data used for training the ANNs has either been obtained experimentally or generated by means of numerical simulations.

A critical prerequisite for accurate system simulation with an ANN is a satisfactory training phase which, in turn, requires presentation of relevant data to the ANN-i.e., data that will provide enough information to allow the ANN to construct a map. In the case of dynamic systems, there are numerous measures that could be used to develop a model, such as excitation and response values, in the form of accelerations, velocities, or displacements. Ideally one

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

would like to use measured response (output) of a system and the applied forcing function (input) to train the ANN. Then one would like to use the trained ANN to predict a future response of the system. However, the map that relates measured input to the response of a nonlinear system can be extraordinarily complex and potentially over-parameterized. Furthermore, such a map does not yield additional information about the system being modeled. This fact prompted us to separate this complex, nonlinear map into a series of simpler maps, each of which provides some information about the system. This investigation decomposes the nonlinear system map into a series of simpler maps by means of a canonical variate analysis (CVA). CVA is a method that uses structural input and responses to establish a transformation into a state space. CVA propagates past state information to future states and, ultimately, calculates future waveforms in the measurement space from state responses (Hunter, 1992).

This paper presents a detailed description of a methodology that, uses data obtained from a CVA analysis of a dynamic system to train several ANNs. The ANNs are linked in series to accurately and efficiently simulate the CVA analysis and to predict future time series. In the following sections we describe the CVA method by emphasizing the basic algorithm. This is followed by a review of the basic composition of a multivariate linear spline neural network. A numerical example is introduced, and the development of three individual maps is described. The long-term (iterated) time series prediction is presented, and finally, some conclusions are offered.

CANONICAL VARIATE ANALYSIS

Canonical variate analysis (CVA) is a model reduction technique for time series data (Larimore, 1983). Initially developed by Hotelling (1936), CVA has been improved, and more robust methods have been developed where CVA is used to estimate state models of linear systems from time series data (Larimore, 1983). Reduced order, nonlinear models have also been constructed from time series data as well as prediction of future time series values, estimation-of-state rank, and computation of Lyapunov exponents. The equations of CVA are developed in this section.

Initially we assume that the state model form is potentially nonlinear and construct a number of local models using CVA. The idea behind CVA is that past waveforms are selected for their utility in predicting future waveforms, in contrast to auto-regressive, moving average (ARMA) models, which use past values to predict a future point. We define the past and future of a multivariate time series as:

$$\begin{aligned} \boldsymbol{p}(t_0) &= \{\boldsymbol{u}(t_0-\tau)\boldsymbol{u}(t_0-2\tau)\boldsymbol{u}(t_0-3\tau)\ldots\boldsymbol{u}(t_0-j\tau)\boldsymbol{y}(t_0-\tau)\boldsymbol{y}(t_0-2\tau)\ldots\boldsymbol{y}(t_0-k\tau)\} \\ \boldsymbol{f}(t_0) &= \{\boldsymbol{y}(t_0+\tau)\boldsymbol{y}(t_0+2\tau)\ldots\boldsymbol{y}(t_0+k\tau)\} \end{aligned} \tag{1}$$

The $\boldsymbol{y}$'s are sampled response time series values (potentially vectors, in the multiple response case), and the $\boldsymbol{u}$'s are sampled time series input values. The subscript on the time value is the time index-i.e., t_0 is the time at index 0. The variable τ is the time increment, j is the number of input lags, and k is the number of response lags. Matrices of the past and future behavior are constructed from sets of past and future waveforms. Minimization of the error in predicting the future $\boldsymbol{f}(t)$ from the past $\boldsymbol{p}(t)$ is accomplished using a series of singular value decompositions on $\boldsymbol{p}$ and $\boldsymbol{f}$, leading to a transformation matrix $\boldsymbol{T}$, which selects from the past the information critical to the prediction of the future (Larimore, 1983). The selection of the optimal "memory" of the past is described by:

$$\boldsymbol{m}(t) = \boldsymbol{Tp} \tag{2}$$

Once $\boldsymbol{m}(t)$ is evaluated at every time t, the state equations may be obtained in a least squares sense using:

$$\begin{aligned} \boldsymbol{m}(i+1) &= \boldsymbol{Am}(i) + \boldsymbol{Bu}(i) + \boldsymbol{w}(i) \\ \boldsymbol{y}(i) &= \boldsymbol{Cm}(i) + \boldsymbol{Du}(i) + \boldsymbol{Ew}(i) + \boldsymbol{v}(i) \end{aligned} \tag{3a,b}$$

In Eqs. 3a and 3b, $\boldsymbol{m}(i)$, $\boldsymbol{u}(i)$, and $\boldsymbol{y}(i)$ are known at any time t_i; $\boldsymbol{w}(i)$ and $\boldsymbol{v}(i)$ are white noise processes that result from errors in the solution; and $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, and $\boldsymbol{E}$ are determined using least squares. The term $\boldsymbol{Ew}(i)$ allows for a possible correlation between the state noise *w(i)* and the measurement noise *v(i)*. This process is described in detail in Larimore (1983).

Dealing with the nonlinear case requires the use of potentially nonlinear maps which: (1) transform the past and future matrices $\boldsymbol{p}$ and $\boldsymbol{f}$ into the estimated states $\boldsymbol{m}$. (Eq. 2)–the measurement-to-state transformation map; (2) map past states $\boldsymbol{m}$(i) into future states $\boldsymbol{m}$(i +1). (Eq. 3a)–the state-to-future state map; and (3) predict future time series values from future states. (Eq. 3b)–the state-to-measurement space prediction map.

In local modeling, in contrast to global modeling, we use a specially selected set of past and future waveforms to formulate the $\boldsymbol{p}$ and $\boldsymbol{f}$ matrices. In local modeling the current past waveform $\boldsymbol{p}(t_r)$ is considered a reference waveform. Each past waveform has a Cartesian distance from the reference waveform. In local modeling n waveforms that have the smallest distance from the reference waveform are used to formulate the matrices $\boldsymbol{p}$ and $\boldsymbol{f}$ of the past and future vectors. Intuitively this means that wave shapes (referred as nearest neighbors) that are similar to the reference waveform are used to formulate the local linear model.

The local linear formulation requires repeated evaluation of Eqs 2, 3a, and 3b for every time series point $\boldsymbol{p}$ for which a model formulation or prediction of the future $\boldsymbol{f}$ is required. Repeated local linear formulation is computationally expensive, especially in finding nearest neighbors from a large set of training data and in the singular value decomposition calculation.

It is this fact that motivates the implementation of an ANN paradigm for use in the solution of this problem. For this work data generated with CVA will be used to train several ANNs individually to perform the three maps that were described earlier. Once trained, the ANNs will be linked in series to obtain a map from past to future prediction in several stages. Before the actual development of the networks is presented, the following section provides a brief explanation of the type of ANN that was used in this work.

THE MULTIVARIATE LINEAR SPLINE NETWORK

The multivariate linear spline (MVLS) network is an ANN of the radial basis function type. The radial basis function ANN, which was developed by Moody and Darken (1989), simulates mappings via the superposition of radial basis functions. It is an accurate local approximator, and although it trains rapidly, it has the potential for size difficulties as the dimension of the input space grows. A generalization of the radial basis function ANN is the connectionist normalized linear spline (CNLS) network. This was developed by Jones et al., (1990) and seeks to simulate a mapping by using radial basis functions in a higher order approximation than the radial basis function network. The MVLS network generalizes the CNLS network to multiple output dimensions, and a schematic representation of this type of ANN is shown in Figure 1.

To commence development of the MVLS network, we let $\boldsymbol{x}$ be an n-dimensional input vector to the system being modeled, and let z = g($\boldsymbol{x}$) be its corresponding m-dimensional output vector. We assume that the function g($\boldsymbol{x}$) is deterministic but that its form and parameters are unknown. We can approximate the mapping from $\boldsymbol{x}$ to $\boldsymbol{z}$ in a region of the input/output space using the linear form:

$$z \cong \boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x}-\boldsymbol{c}) \tag{4}$$

where $\boldsymbol{y}$ is an approximation to z, $\boldsymbol{c}$ is a vector with the same dimension as $\boldsymbol{x}$ in the vicinity of which the approximation is made, and $\boldsymbol{A}$ is an mxn matrix of constants. This approximation needs to be optimized using least squares or weighted least squares and is accurate in the vicinity of the data used to develop it as long as the behavior of the mapping in the neighborhood is truly linear. The vector $\boldsymbol{c}$ is the "center" of the local linear approximation (also referred to as center vector or center). We can develop similar approximations in other neighborhoods of the input vector space. To account for this, we append the subscript j to the coefficient matrix and the center vector.

$$\begin{gathered} z \cong \boldsymbol{y} = \boldsymbol{A}_{\mathrm{j}}(\boldsymbol{x}-\boldsymbol{c}_{\mathrm{j}}) \\ \mathrm{j} = 0,\ldots \mathrm{N}-1 \end{gathered} \tag{5}$$

where N is the number of regions of approximations.

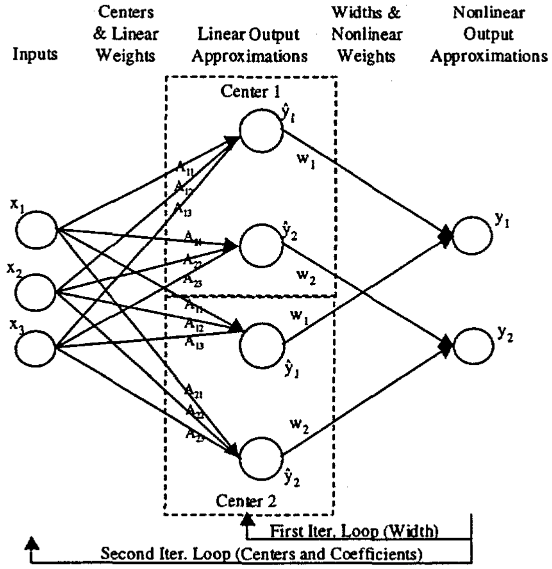

Figure 1: Graphical representation of the multvariate linear spline ANN

Having developed local linear approximations of the $\boldsymbol{x}$ to $\boldsymbol{z}$ mapping, we can now combine the local approximations to create an approximate, global map. The approximation takes an input vector $\boldsymbol{x}_0$ and maps into $\boldsymbol{y}_0$, an approximation to g($\boldsymbol{x}_0$). To accomplish this, we superimpose several of the linear approximations in a series. We weight each component in the series according to its distance from the input vector $\boldsymbol{x}_0$. Local linear models that are near $\boldsymbol{x}_0$ are weighted heavily, whereas those that are further away are weighted less. The series is:

$$\sum_j \boldsymbol{y}_0 \boldsymbol{w}_j = \sum_j \boldsymbol{A}_j(\boldsymbol{x}_0 - \boldsymbol{c}_j)\boldsymbol{w}_j \tag{6}$$

where the $\boldsymbol{w}_j$ are the weights attached to the local linear models. The output vector $\boldsymbol{y}_0$ on the left side is independent of the index j, so it can be removed from the sum, and the equation can be simplified to:

$$\boldsymbol{y}_0 = \frac{\sum_j \boldsymbol{A}_j(\boldsymbol{x}_0 - \boldsymbol{c}_j)\boldsymbol{w}_j}{\sum_j \boldsymbol{w}_j} \tag{7}$$

This is the parametric form of the MVLS network.

We choose the form of the multivariate Gaussian probability density function for the weighting expression. It is known as a radial basis function and has the form:

$$\boldsymbol{w}_j = \exp\left(-\frac{1}{2\beta^2}\left\|\boldsymbol{x}_0 - \boldsymbol{c}_j\right\|^2\right) \tag{8}$$

The quantity β is a network parameter related to the width of the radial basis functions; it requires optimization.

The MVLS network is used in the feed forward operation by specifying the input vector $\boldsymbol{x}_0$, evaluating the weights $\boldsymbol{w}_j$ using Eq. 8, substituting the weights and input vector into Eq. 7, and evaluating the output $\boldsymbol{y}_0$. This output should present an interpolation among the training outputs that corresponds to the input as an interpolation among the training inputs. Note that the range of the summation index j is not specified in Eq. 7. It is clear that the summation should be carried out over those linear models nearest the input vector $\boldsymbol{x}_0$. There are several ways to accomplish this. In the present code a user-defined number of models is used (Eq. 5) and the ones nearest, in Cartesian space, to the input vector are chosen to make each prediction.

There are two groups of quantities and one scalar that need to be identified in order to establish the MVLS network as an approximator to the relation $z = g(\boldsymbol{x})$. These are the linear coefficient matrices A_j, j = 0, ... N-1; the centers $\boldsymbol{c}_j$, j = 0, ... N-1; and the radial basis function width parameter β. There are clearly many ways in which the parameters could be identified. The following approach is used in this work.

First, use a self-organization procedure to optimally locate the centers in the space of input exemplars. Next start a loop to optimize the radial basis function width. Specify the width. For this width estimate the parameters of all local linear models. Compute the error. Iterate on the width until the error is satisfactorily minimized. For a detailed explanation of this procedure, see Paez and Hunter (1997b).

Next is a summary of the problem to be investigated followed by the ANN implementation and some results.

NUMERICAL EXAMPLE – APPLICATION TO A NONLINEAR, HARDENING OSCILLATOR

The system shown in Figure 2a will be used to test the ANN algorithm. The terms c and k in this figure indicate damping and stiffness, respectively.

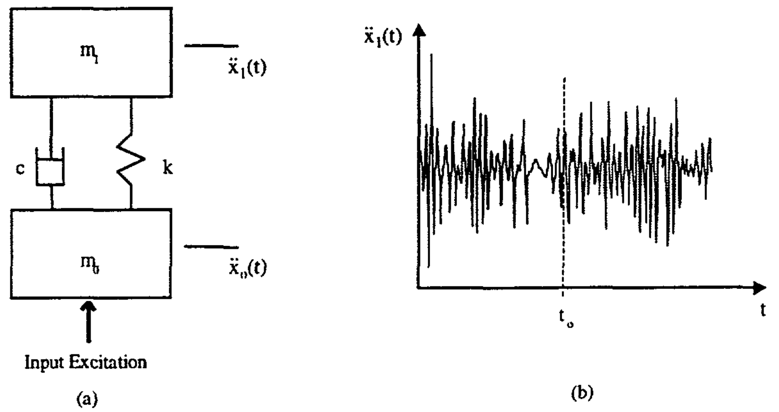

Figure 2: (a) Nonlinear hardening oscillator, (b)Time history

Equation 9 defines the system's dynamics in detail.

$$\ddot{x}_1 + 2\zeta\omega_n(\dot{x}_1 - \dot{x}_0) + \omega_n^2(x_1 - x_0) + \alpha\omega_n^2(x_1 - x_0)^2 + \beta\omega_n^2(x_1 - x_0)|x_1 - x_0| = 0 \tag{9}$$

where α=3000, β=3500, ϖ_n=2π(11.5), and ζ=0.04. In this experiment the level of the band-limited random drive $\ddot{x}_0$ is adjusted to excite about equal mean square (rms) responses in the linear and quadratic terms. The response acceleration $\ddot{x}_1$ and the drive acceleration $\ddot{x}_0$ were digitized at 150 samples/second. It is noted that a fundamental peak in the response spectrum occurred at 15.0 Hz with a significant third harmonic response at 45.0 Hz. A more detailed explanation of this system can be found in Hunter et al. (1992). CVA was run using eight lags to predict two states using 1,000 neighbors. With these parameters 8,000 points of data were generated and used to train the ANN model. These data represent:

1) A series of waveforms corresponding to the input excitation consisting of eight lagged values in the past, one current (or present) value, and one future value.
2) A series of waveforms corresponding to the response of the system consisting of eight lagged values in the past.

3) The CVA calculated past and future states.
4) A predicted future value.

As stated earlier, the approach we followed in this work was to create three maps: 1) the measurement-to-state transformation map, 2) the state-to-future state map, and 3) the state-to-measurement space prediction map. The desired objective is shown graphically in Figure 2b where the solid line represents the training cases in the past (at a time less than t_0), and the dashed line is the resulting predicted values (at a time greater than t_0). In the following sections we describe the development of these maps; present individual results; and, finally, the combination of the three maps to create an iterated procedure to predict future waveforms.

THE MEASUREMENT-TO-STATE TRANSFORMATION MAP

This map transforms the past into the states (or, more correctly, past states). The input to the ANN is a series of waveforms of the input-i.e., the forcing function-consisting of eight lagged values in the past, one current (or present) value, and one future value. In addition, eight lagged values in the past of the system's response are also used as input for a total of 18 inputs to the ANN. The output is two state values corresponding to this input. The network was trained with 5,000 waveforms and tested with 256 out-of-sample waveforms-i.e., waveforms that the network was not trained on. The MVLS network was trained with 500 center vectors, used 600 neighbors, and had a radial basis width, β, of 0.3. Training was done on a Pentium 200 PC, and the average training time was 20 minutes. Two models were combined to create the output. The MVLS network results are compared with the results given by the CVA analysis. Figure 3 shows the time history for State 1 for the ANN's predicted results and the one obtained from the CVA for the out-of-sample data. The corresponding graph that was shown for State 1 is also presented for State 2 (Figure 4). One aspect that is clear is that State 2 is noisier than State 1. The predictions were good in most intervals and poor in a few. The second map that will be presented is the past states to future states map.

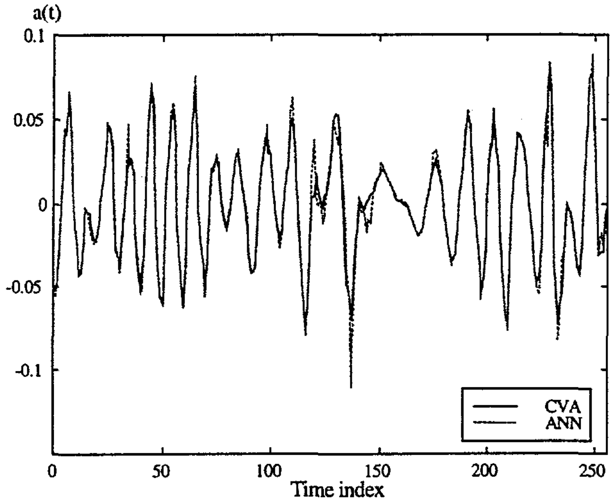

Figure 3: Measurement-to-state transformation map: State #1

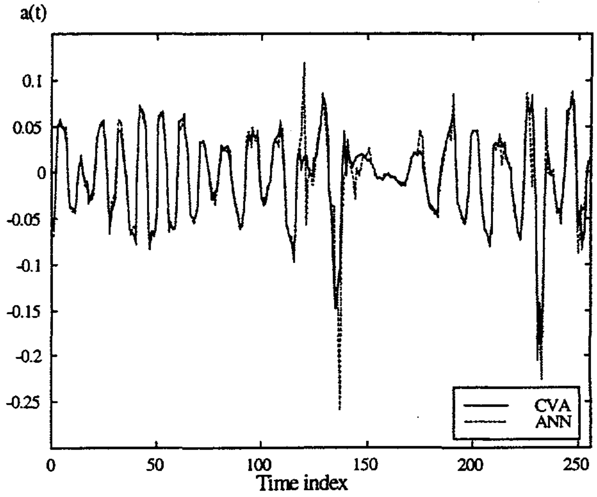

Figure 4: Measurement-to-state transformation map: State #2

THE STATE-TO-FUTURE STATE MAP

This map transforms the past states into future states. The input to the ANN is two values corresponding to the past states (State 1 and State 2). The output is two state values corresponding to the future states (future State 1 and future State 2). The network was trained with 5,000 waveforms and tested with 256 out-of-sample waveforms. The MVLS network had 405 center vectors, used 600 neighbors, and had a radial basis width, β, of 0.3. Two models were combined to create the output. Figure 5 shows the time history for State 1 for the ANN's predicted results and the one obtained from the CVA for the out-of-sample data. The corresponding graphs that were shown for State 1 are also presented for State 2 (Figure 6). In contrast to the transformation map where the second state prediction did not correlate very well with the estimated state, the state map produced a very good prediction of the second state as shown by Figures 5 and 6. The third map that will be presented is the future states to prediction map.

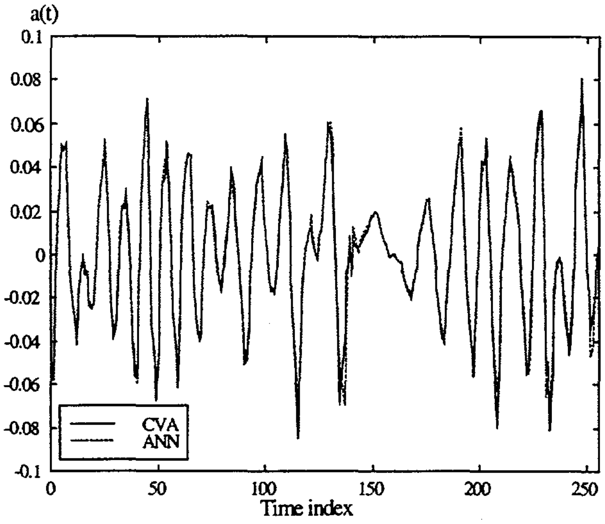

Figure 5: State-to-future state map: future State #1

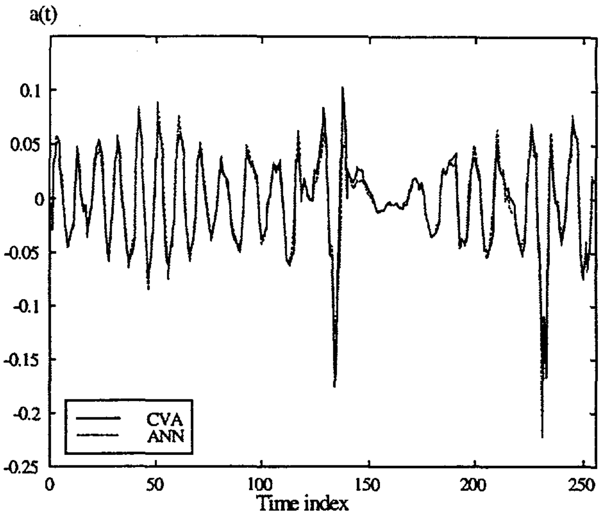

Figure 6: State-to-future state map: future State #2

THE PREDICTION MAP

This map transforms the future states into future prediction points. The input to the ANN is two values corresponding to the future states (future State 1 and future State 2). The output is one value corresponding to a future point in the time history. The network was trained with 5,000 waveforms and tested with 256 out-of-sample waveforms. The MVLS network was trained using 400 center vectors, used 600 neighbors, and had a radial basis width, β, of 0.3. Two models were combined to create the output. Figure 7 shows the time history, which represents the future evolution of the system for the out-of-sample data. The ANN produced a rather good map of the future states to the predicted values.

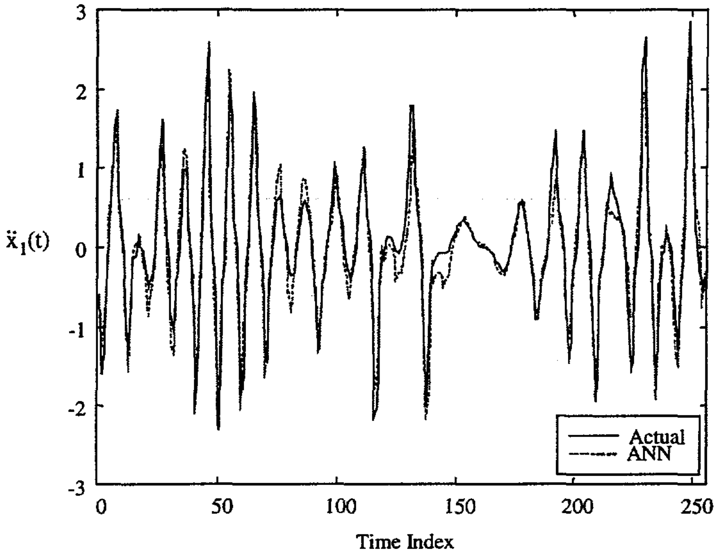

Figure 7: State-to-measurement space prediction map: future points

With these maps identified, the final objective is to use them in series in order to obtain an ANN model that maps the past into the future. A strict test of this algorithm is its ability to predict the future in an iterative manner-that is, use the output of a previous map and use it as input to the following map. The procedure followed to do this is described in the following section.

THE ITERATIVE PREDICTION

The initial objective of this work was to develop an algorithm using ANNs to identify a map from past values of structural response to future values. The proposed map would accept as inputs past waveforms and, through a series of intermediate maps, obtain a future value in the time history. The previous sections described the development of three maps. As shown by the results, these maps are able to perform their individual tasks fairly well. We now link them in series to predict future values in the time history. Graphically this is shown in Figure 8.

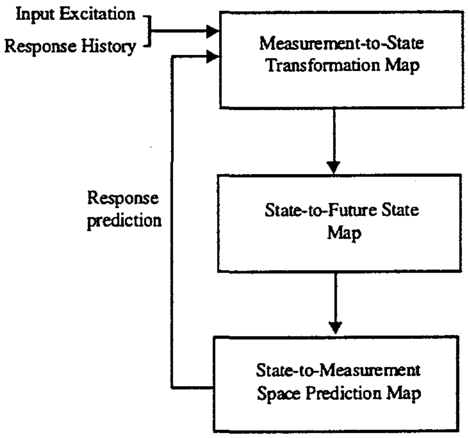
Input Excitation
Response History
Measurement-to-State Transformation Map
Response prediction
State-to-Future State Map
State-to-Measurement Space Prediction Map

Figure 8: Iterative prediction algorithm

The three maps described in the previous sections were linked, and the algorithm was tested with 256 out-of-sample data points. The results of this prediction are shown in the final sequence of graphs. Figure 9 shows the iterated time history predicted with the ANN and the corresponding waveform obtained from the actual data. The correlation between the two curves is approximately ninety two percent.

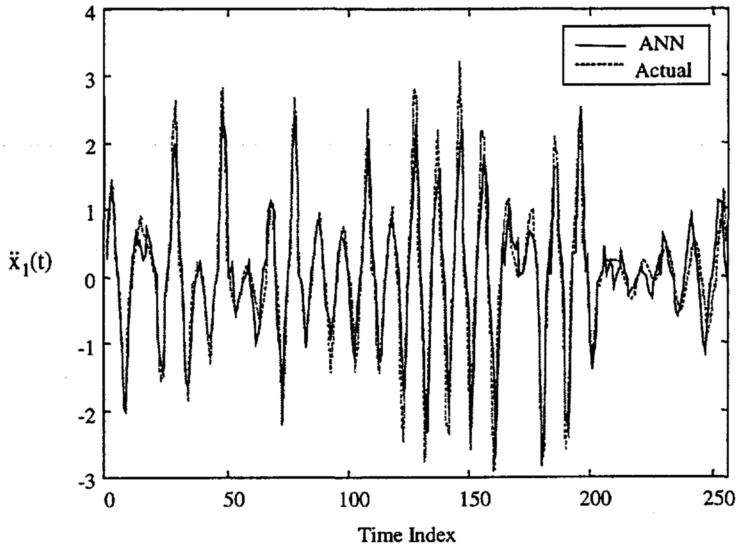

Figure 9: Iterated prediction using composite MVLS net: time history

With the high correlation in the time history, the spectral density should be very accurate, particularly in the detection of the two primary features: the fundamental frequency and the third harmonic. This is shown in Figure 10.

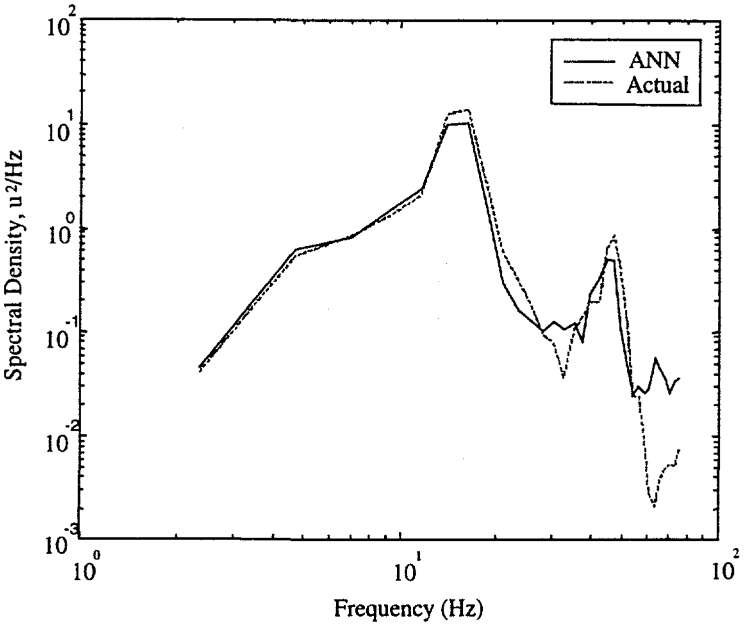

Figure 10: Iterated prediction using composite MVLS net: spectral density

The fundamental frequency is detected at 16.3 Hz with an error of twenty six percent in the peak value. The third harmonic occurred at 46.6 Hz with an error of forty three percent in the peak value. The final plot shows that the network underpredicted both the positive and negative peaks, and it slightly overpredicted some of the points in the middle range (Figure 11).

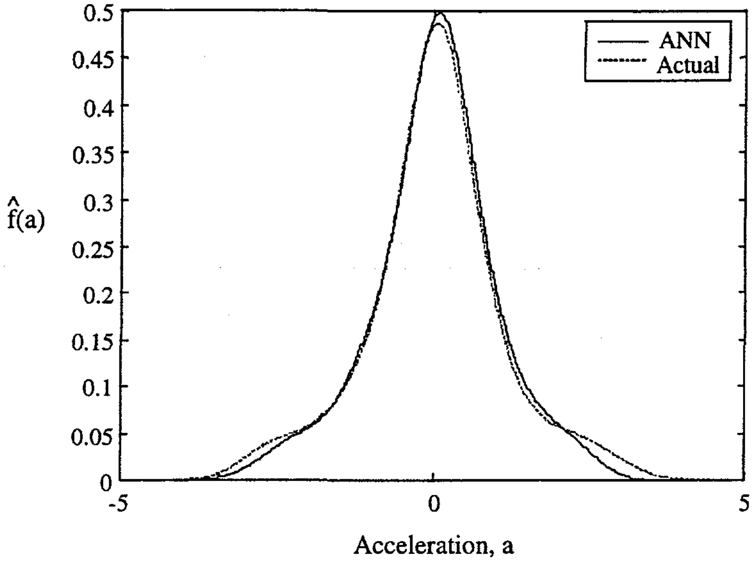

Figure 11: Iterated prediction using composite MVLS net: kernal density estimation

CONCLUSIONS AND RECOMMENDATIONS

A neural network-based algorithm was developed for the prediction of future time histories based on past information. To simplify this development, a canonical variate analysis was used to separate the initial map into a series of simpler maps: the measurement-to-state transition map, the state-to-future state map, and the state-to-measurement space prediction map. Each map was individually trained and tested to an accuracy that was deemed sufficient. The correlation between the ANN predicted values and the CVA-computed, or the actual values was used as a metric of ANN quality. If the correlation was above ninety percent for each of the individual maps, the ANN was deemed sufficiently accurate. Once this level was achieved, the three maps were linked, and an iterative prediction was obtained. A comparison with the actual data and the ANN algorithm shows that the ANN is capable of simulating the testing data set very closely. In addition, the resulting ANN simulation contains sufficient information to obtain the fundamental peak and the third harmonic response.

An advantage to using the ANN is the speed of training and the need for less data for training. Each map trained in approximately 20 minutes, and the prediction of the 256 testing points was done in approximately 2.5 minutes. The CVA technique required approximately ten minutes to perform the same prediction.

Additional work is suggested to improve the predictive capability of the ANN model. One aspect would involve enhancement of the learning algorithm-i.e., improving the width parameter optimization routine-and/or improvement of the data used to train the ANN-i.e., improve the calculated states. Another area where improvement is required is the stabilization of the ANN algorithm under certain conditions.

ACKNOWLEDGMENTS

This work was sponsored by the United States Department of Energy under contracts with Sandia National Laboratories (Contract DE-AC04-94AL85000) and Los Alamos National Laboratory.

Sandia is a multiprogram laboratory
operated by Sandia Corporation, a
Lockheed Martin Company, for the
United States Department of Energy
under contract DE-AC04-94AL85000.

REFERENCES

Hotelling, H., (1936), "Relations Between Two Sets of Variates", *Biometrika*, Volume 28, pp. 321-377.

Hunter, N., and Theiler, J., (1992), "Nonlinear Signal Processing: The Time Series Analysis of Driven Nonlinear Systems", Los Alamos National Laboratory Report, Los Alamos, NM., May 1992.

Jones, R. D., et al., (1990), "Nonlinear Adaptive Networks: A Little Theory, A Few Applications", Cognitive Modeling in System Control, The Santa Fe Institute.

Lapedes, A., and Farber, R., (1987), "Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling", *IEEE-Neural Networks*, San Diego, Ca., June 1987.

Larimore, W., (1983), "System Identification, Reduced Order Filtering and Modeling Via Canonical Variate Analysis", *Proceedings of the 1983 American Control Conference*, H. S. Rao and P Dorata, Eds., 1982, pp. 445-451.

Moody, J., and Darken, C., (1989), "Fast Learning in Networks of Locally-Tuned Processing Units", Neural Computation, Volume 1, pp. 281-294.

Paez, T., O'Gorman, C., and Tucker, S., (1997a), "Simulation of Nonlinear Random Vibrations Using Artificial Neural Networks", *Proceedings of the Sixth International Conference on Recent Advances in Structural Dynamics*, Institute of Sound and Vibration Research, Southampton, United Kingdom.

Paez, T., and Hunter, N., (1997b), "Dynamical System Modeling Via Signal Reduction and Neural Network Simulation", Shock and Vibration Symposium, 1997.