AUTOMATED DIAGNOSIS AND CLASSIFICATION OF STEAM GENERATOR TUBE DEFECTS

Mechanical Engineering Department

New Mexico State University

Box 30001, MSC 3450

Las Cruces, NM 88003

Telephone: 505-646-3501

Fax: 505-646-6111

E-mail:

Website: me.nmsu.edu

Principal Investigator: Dr. Gabe V. Garcia

EXECUTIVE SUMMARY

A major cause of failure in nuclear steam generators is tube degradation. Tube defects are divided into seven categories, one of which is intergranular attack/stress corrosion cracking (IGA/SCC). Defects of this type usually begin on the outer surface of the tubes and propagate both inward and laterally. In many cases these defects occur at or near the tube support plates. Several different methods exist for the nondestructive evaluation of nuclear steam generator tubes for defect characterization. At present, multifrequency eddy current testing has emerged as the dominant method in industry but requires significant human expertise to accurately identify and classify tube defects. The research described in this final report includes a focus on defects characterized as intergranular attack/stress corrosion cracking (IGA/SCC) occurring at tube support plates. The automated detection software developed as a result of this grant uses a sophisticated set of transform algorithms to enable data visualization from the raw eddy current data. In addition, the software parameterizes the eddy current data for a second stage defect analysis by an artificial neural network architecture. Although our initial plans called for the use of a fuzzy ART neural architecture, after experimentation with data sets, we found that a Learning Vector Quantization (LVQ) architecture provided better predictive power for the given data sets.

This project involved the collection of representative Eddy Current data and design of damage detection software for the purpose of automated diagnosis and classification of tube defects. Artificial neural networks have proven to be a successful method for complex pattern detection especially under circumstances of noisy input data. Two neural network approaches were investigated in this project: a traditional supervised neural network and building on that work, an advanced fuzzy unsupervised neural network. Although a fuzzy unsupervised network represents a novel approach for the inspection of steam generator tubes, it is an approach that is well tested in many areas of pattern recognition.

To accomplish the design of this new method of diagnosis and classification of steam generator tube defects, a complete articulation of many types and forms of tube defects were modeled. This step created a training set for the artificial neural network.

We have developed new software for extraction of feature vectors from raw eddy current data files. The data used was provided to the provided by the Electric Power Research Institute and contained data used for training of human experts in reading eddy-current signatures.

Several new techniques were developed for extraction of defect signatures from noisy raw data generated by bobbin type eddy current probes. Damage detection software was developed in MATLAB and was tested against data sets for Westinghouse U-Tube Steam Generator defect data related to Stress Corrosion Cracking in the area of tube supports. The damage detection software automates the process of extraction of damage signatures. The reduction of the noisy data to feature vectors allows for training of an artificial neural network. However, a limitation was encountered in the amount of data available for complete training of the artificial neural network.

Results from this work formed the basis for a follow on project to develop a more advanced analysis technique based on least square vectors.

BACKGROUND AND PROBLEM DEFINITION

The nuclear power industry would greatly benefit from technological advances that decrease the downtime of nuclear plants particularly for maintenance and repairs. Detection and mitigation of team generator tube failures can contribute significantly to the efficiency and economy of nuclear power. The principal problems with steam generators have been reported by EPRI to include: denting and tube support corrosion, denting and tubesheet corrosion, tubing corrosion (wastage, pitting, ID cracking, OD stress corrosion cracking, intergranular attack), mechanical damage (fretting, fatigue cracking, impingement). A significant number of the pressurized water reactors (PWRs) are operating with tubing defects near or beyond the limits established by regulatory guidelines. Over ten steam generator tube ruptures have occurred over the past 25 years at a rate of one rupture every two years. In the US and in recent years, tube ruptures have occurred at a rate of one per year. "Five different tube degradation mechanisms caused ten rupture: three ruptures were caused by outside diameter stress corrosion cracking (ODSCC), two ruptures were caused by high-cycle fatigue, two ruptures were caused by loose part wear, two ruptures were caused by primary water stress corrosion cracking (PWSCC).

The probability of steam generator tube failures can be reduced through timely and effective inspection, diagnosis, and appropriate acceptance (fitness-for-service) criteria. Eddy-current testing has emerged as a standard procedure for inspecting thin-walled steam generator tubes because it offers high scanning speed and good accuracy. However, eddy-current detection threshold is high; deep cracks can be missed and sizing and resolution are not accurate. Furthermore, eddy-current methods are weak for sizing the length and depth of circumferential cracks, intergranular attack (IGA) damage, pitting, high-cycle fatigue cracking, fretting, and wear.

Ultrasonic inspections methods have had some success in sizing and resolution of ODSCC detected during an eddy-current inspection. Innovative new methods are also development and offer the potential for significantly improvements over eddy-current methods or complementing eddy-current techniques. These include a laser imaging system for inside surfaces of the tubing.

Nassersharif, et. al are researching and developing a new technique based on a continuous wave radar that offers great potential for significant improvements in non-destructive testing of steam generator tubes by providing a volumetric scan of the tube wall at higher resolution.

Eddy-current as well as and the continuous-wave radar technique require both calibration and personnel training for successful detection of damage in the tubes. The EPRI NDE center in Charlotte, North Carolina is a national center of excellence for calibrations and personnel training in the use of eddy-current devices. Effective and accurate use of the inspection devices requires significant personnel training with calibrated samples. While calibrated samples provide reference frame for training personnel and comparative data points, accurate,

effective, and consistent diagnosis of the defect type in the field remains a significant challenge.

RESEARCH OBJECTIVES

The overall objective of this research is to develop a method and a software prototype for automation of the diagnosis of the generated signals that will provide a more consistent and faster method for diagnosis of defects based on generated signals from either eddy-current or continuous-wave radar.

Our research and development objectives include:

I   SIGNAL CHARACTERIZATION.

The first objective of this research is to characterize the signal generated by the continuous-wave radar based on the already established methods for eddy-current technology. Signal characterization will be extremely important in the diagnosis step.

II   NEURAL NETWORK DESIGN.

The design of the neural network system will have to be determined as to the appropriateness of either a supervised back-propagation network or and unsupervised network.

III   NEURAL NETWORK TRAINING.

The neural network will have to be trained by signal data generated from calibrated sample signals. The EPRI NDE center has a comprehensive calibration set for various types of defects. The principal investigators are currently collaborating with the EPRI NDE center on the continuous-wave radar project. The training process may generate additional needs for calibration samples which will be documented and communicated back to EPRI.

IV   PROOF OF PRINCIPLE.

The prototype system will have to be tested against the current methods with results documented and reported to DOE and EPRI as well as published in the open literature.

SCIENTIFIC SIGNIFICANCE

The concept for this project arose from the need to classify and "invert" the observed signal data in the case of the continuous-wave radar project which is funded under the nuclear energy research initiative (NERI) under Department of Energy Grant Number (DE-FG-3-99SF21986) [14, 15]. Additionally, after

investigating the methods currently used for analysis of the eddy-current generated signatures revealed that similar issues exist in the diagnosis of damage based on an eddy-current signal. This problem has led to our preliminary research on the topic of automation of the detection of damage based on a noisy signal generated by either the eddy-current or continuous-wave radar probe system. Several new contributions are expected as a result of this work that have scientific significance well beyond the needs of the project itself, namely:

Design, development, and selection of neural network software to diagnose the electronic signal generated from the probe. The new contribution of this work will be to focus both on the static signal image as well as the temporal signal generated from the probe.

Develop a classification scheme to identify and diagnose various defect types or lack of a diagnosis. The classification scheme should also allow for characterization of the defect as to sizing of length and depth as well as type.

Identification and inclusion of signal features that can improve the speed and accuracy of the diagnosis.

Technical Approach

DEVELOPMENT OF A BASIS FOR SIGNAL CHARACTERIZATION AND CLASSIFICATION.

Over the past ten years, a great deal of work has been initiated in studying the viability of automated testing and diagnosis systems for nuclear steam generator tubes. One commonality in this research has been the use of neural network architectures for the diagnosis portion of the systems. A sampling of these studies is presented next. The Babcock & Wilcox Owners' Group funded a study that focused on the use of probabilistic neural networks (PNN) with generalized regression methods (Yang, 2001). This study demonstrated the feasibility of using different training algorithms coupled with the PNN for defect detection. Results of this study were very promising. German et al. (2001) compared the performance of multilayer perceptron neural networks with PNNs. They found that the PNN relied less on training data than did the multilayer perceptron architecture. This suggests that the PNN is a better selection in those cases where training data is limited. Using the Fourier descriptor method to parameterize the eddy current input data, Udpa and Udpa (1990) employed a two-layered feed forward architecture with five hidden layers with a back propagation algorithm to create their defect detection/classification model. In a later work by Udpa and Udpa (1991), they added a feature to the network architecture that allowed the network to filter the internal representation of the signal based on scale, rotation and translation. This work demonstrated the superiority of this method over classical clustering algorithms.

Those researchers designing complete analysis and characterization systems include Spanner (2000). He focused his work on detecting intergranular attack/stress corrosion cracking in steam generator tubes with the application of

a frequency-independent automated system using principal components analysis and discrete wavelet transforms. This research used an error back propagation training algorithm with a neural net classifier.

Another fully developed automated system by Xiang et. al. (2001) used a four step algorithm consisting of a: 1) data preprocessing stage; 2) decision tree system; 3) multilayer perceptron (MLP) neural network; and 4) Richardson Lucy blind deconvolution algorithm. Simone and Morabito (2001) focused on addressing signal-to-noise ratio including lift off, probe angle errors, and sensor drift. Their neural architecture involved a radial basis function design where several different training algorithms were applied. Results showed that the Levenberg-Marquardt training algorithm performed especially well.

Each pattern recognition method requires that a calibration base set be available for "inverting" the signal back to a defect type. The empirical data must be correlated with actual defect types and the networks or the algorithm must be trained with the calibration results. The new method must then be compared to existing approaches for diagnosis of defects with respect to speed, accuracy, robustness, and consistency.

The study of pattern detection has been, and will continue to be a well-researched area. Interest in this type of problem is due to the inherent complexity associated with the various model inputs, the variety of realistic applications in industry, and the value of successful results as measured by cost, performance, reliability and maintenance.

One technique that has been used to study pattern detection problems is artificial neural networks (ANN). Researchers have shown that ANN's are a good method for solving a range of pattern detection problems including the identification and classification of defects.

Classical approaches to pattern detection using neural networks involve a process of back-propagation. These supervised nets require providing all possible detectable patterns to the architecture for training purposes. The drawbacks of using supervised neural nets most often involve a very lengthy training time and the assumption that all defect patterns are in fact known. Furthermore, with ANN's alone, sometimes a close to optimal solution is unreachable. In other scenarios, the ANN training period may require a great deal of computing time making the technique unrealistic for dynamic industrial applications.

To overcome the challenges with classical approaches, more recent hybrid approaches involve integrating parallel techniques with the ANN implementation. These techniques include Lagrangian relaxation, simulated annealing, simulation and genetic algorithms. Hybrid approaches have demonstrated a reduction in ANN training time and/or the production of more accurate pattern detection. Although using these hybrid approaches suggests a promising area for research, they still do not guarantee that for large problems especially, training time can be reduced or that an optimal solution will be reached.

In application fields such as control or forecasting, researchers have proven that the use of fuzzy logic combined with neural networks has found better solutions than by using an ANN alone [Kuo 2001]. The use of fuzzy logic incorporated with an ANN architecture seems to improve a model's performance, especially those

models with noisy input patterns. This point is particularly appropriate for pattern detection problems due to the input parameters.

ANN's utilizing fuzzy logic most often make use of a feed-forward architecture that reduces training time. This type of neural network is self-organizing resulting in an architecture that permits both supervised and unsupervised learning. In an unsupervised mode, the ANN has the ability to learn patterns. Yet the architecture also permits supervised learning for the purpose of learning and predicting patterns that fit the statistics of the input-output environment. The advantage of this type of architecture is its capability to learn complex mappings and create new pattern defect categories when new information is presented. This eliminates the need for a time consuming training process typical in ANN's with only a supervised learning process.

Therefore, the scope of work proposed for this project is presented in a two-year time frame. Year 1 will focus on creating a baseline classical neural network architecture of in-tube defect detection using back propagation under a supervised mode of learning. Year two will focus on moving to a fuzzy ARTMAP neural architecture having the capability to perform in a supervised and unsupervised environment. Then, the relative advantages and defect detection power of both approaches will be analyzed according to cost, reliability, performance and maintenance metrics in conjunction with the Department of Energy and EPRI.

Identification and Documentation of Known Types of Tube Defects

The table below indicates the primary data files that were investigated in this research effort. The damage mechanism and location are the same for all tubes as well as the outside diameter and inside diameter of the tube.

Tape	Real	Plant	Tube	Damage Mechanism	Location
tape34Wcal01c	727	D.C. Cook 2	R18C77		
tape34W.cal10	728	Farley 1	R20C26		
tape34W.cal15	736		R12C08		
tape34W.cal17	737		R16C74		
			R20C66	IGA/SCC	Tube Support Plate
			R08C66		
tape34W.cal18	738	Trojan	R30C64		
tape34W.cal19	739		R29C70		
tape34W.cal20	740		R08C69		
			R12C70		

Mathematical Modeling/Algorithms

During the past year, we have been analyzing the defect data to identify characteristic features in the data that constitutes a defect. Here we have focused on the following forms of the data: (1) 400/10 frequency mixed data for slope data, (2) 400/10 frequency mixed data for r and θ, (3) 400/10 frequency mixed data for the slope of r and θ data, (4) 400 Hz slope data, (5) 400 Hz r and θ data, (6) 400 Hz slope of r and θ data, (7) 200/10 frequency mixed data for slope data, (8) 200/10 frequency mixed data for r and θ, (9) 200/10 frequency mixed data for the slope of r and θ data, (10) 200 Hz slope data, (11) 200 Hz r and θ data, and (12) 200 Hz slope of r and θ data. The figure below shows what this data looks like for a typical defect.
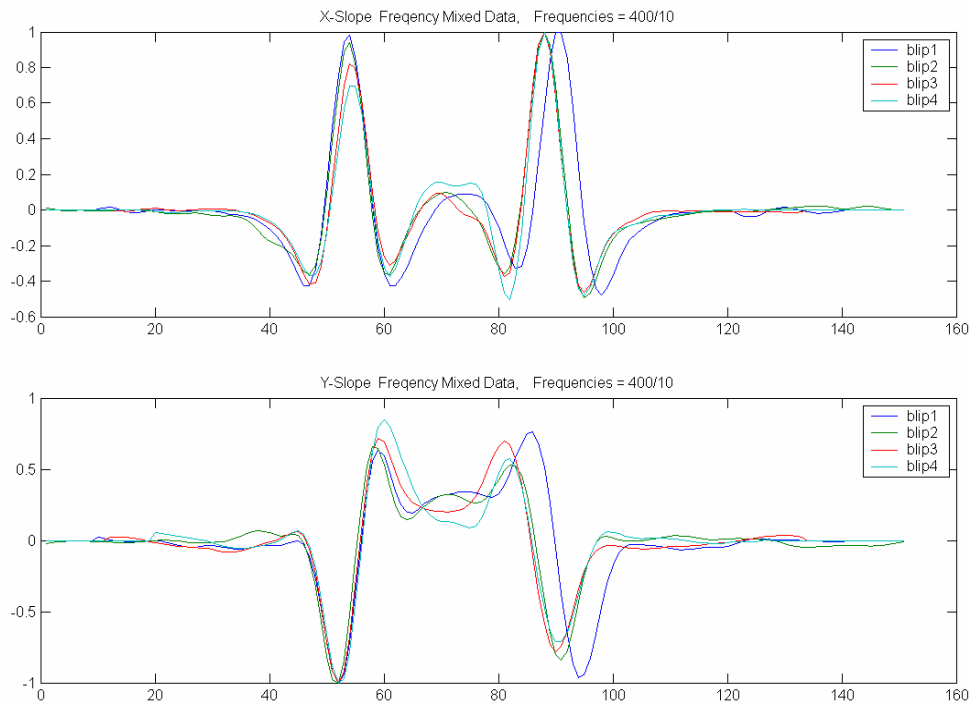
These figures have been generated for all the IGA/SCC types of defects at the tube support plates. We have analyzed these figures to identify characteristics that that can be utilized in our neural network algorithm. So far we have noticed that the defects generally occur at the following locations: (1) peak, (2) valley, (3) inflection point, or (4) situated within one-to-two points from these points. To some degree we have been able to mathematically model these defects; however, the inconsistency of the data has limited the success of these efforts. The idea here was too identified and model features that are indicative of damage and to utilize these features in our neural network algorithms to identify actual defects using data from eddy current tests.

In addition to this work, we identified the portion of the signal generated by the tube support plate. The idea here was to take the signal for a damaged support plate and subtract out the signal generated by an undamaged support plate. Thus, the resulting signal would only contain the portion of the signal generated by the damage. This procedure was performed on a number of the data files to determine the successes of

the method. After comparing the results with previous methods it was determined that the new method did not perform any better than any of the previous methods that had been investigated.

We have also investigated various approaches, including a reflected energy integral to analyze the data for automation. Analysis of the data obtained from EPRI for the MIZ-18 probe proved to be difficult because of the resolution of the data particularly when the defect resides in the tube region inside the tube support plates.

Over the past year we have investigated a number of methods and algorithms to isolate the signal generated by the tube support plate in an effort to identify features that could be used in an ANN to identify and localize damage. Each of these attempts has work to some degree. It is believed that the inconsistency in the resolution of the data is the primary reason for our inability to identify a method which works for all damage cases.

After some time we finally settled on the following approach for the extraction of feature vectors. The signal resulting from the tube support plate was first extracted for each of the tubes. The differential and absolute data was then shifted such that the mean of the signal was centered along the x-axis. Next the slope of the two dimensional curve (this is the figure eight shaped curve for the differential data) was calculated for the differential and absolute data. It was determined the slope data for the 200 Hz signal for the differential and absolute data had a characteristic shape to it. As a result we used the error associated with a nonparametric curve fitting technique as feature vectors. In addition to this we also used the negative peak associated with the 100 Hz signal for the differential data. It was observed that a negative peak existed in the 100 Hz signal and that its location usually indicated the location of the defect. The figures and the program below was used to construct a three dimensional feature vector to used by the neural network for the detection of defects.
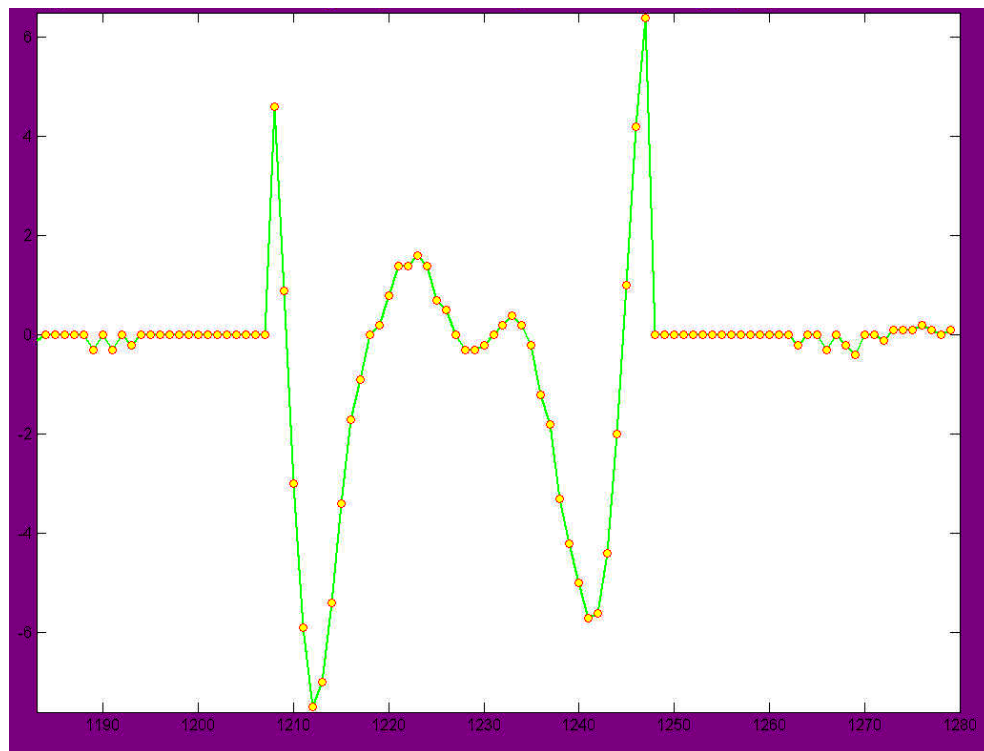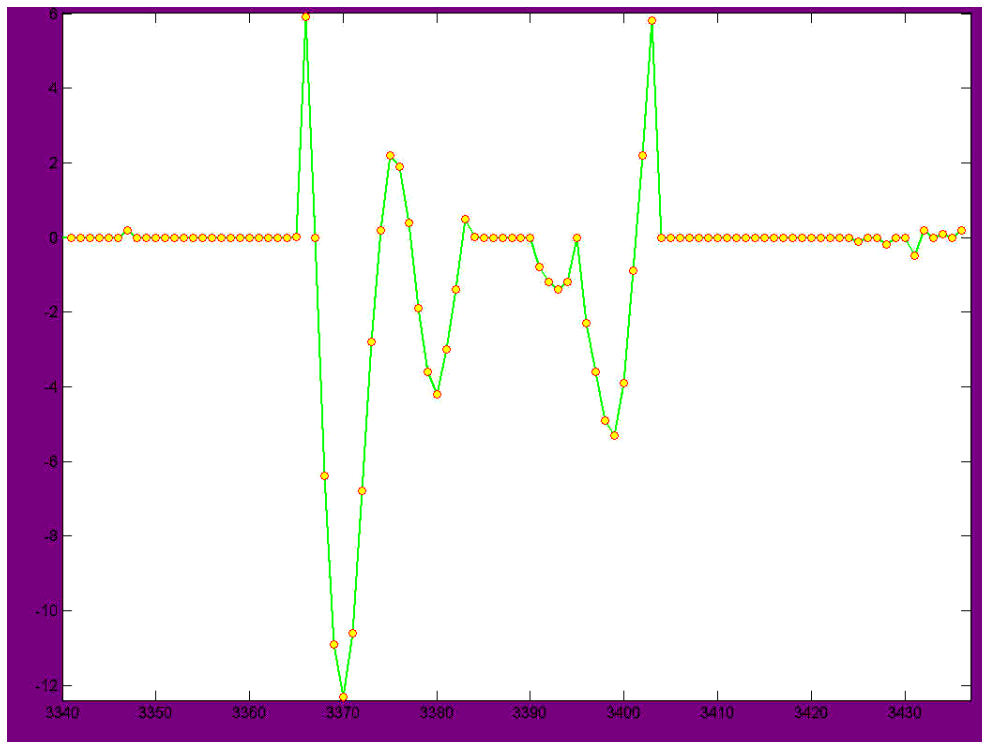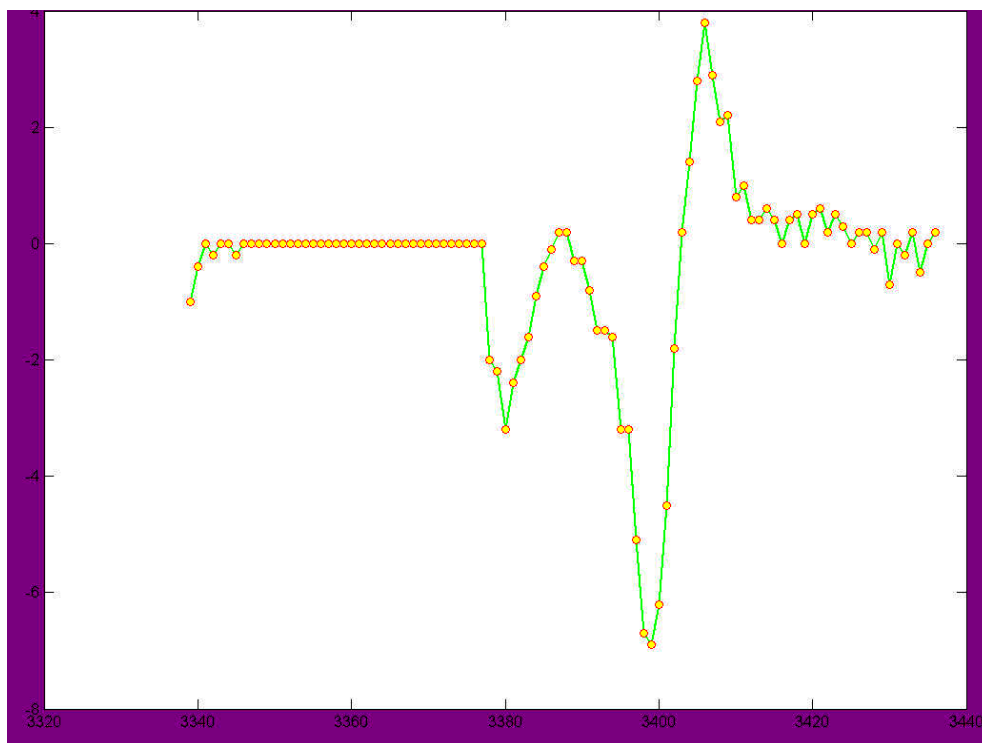

Differential Data, 200 Hz, No damage
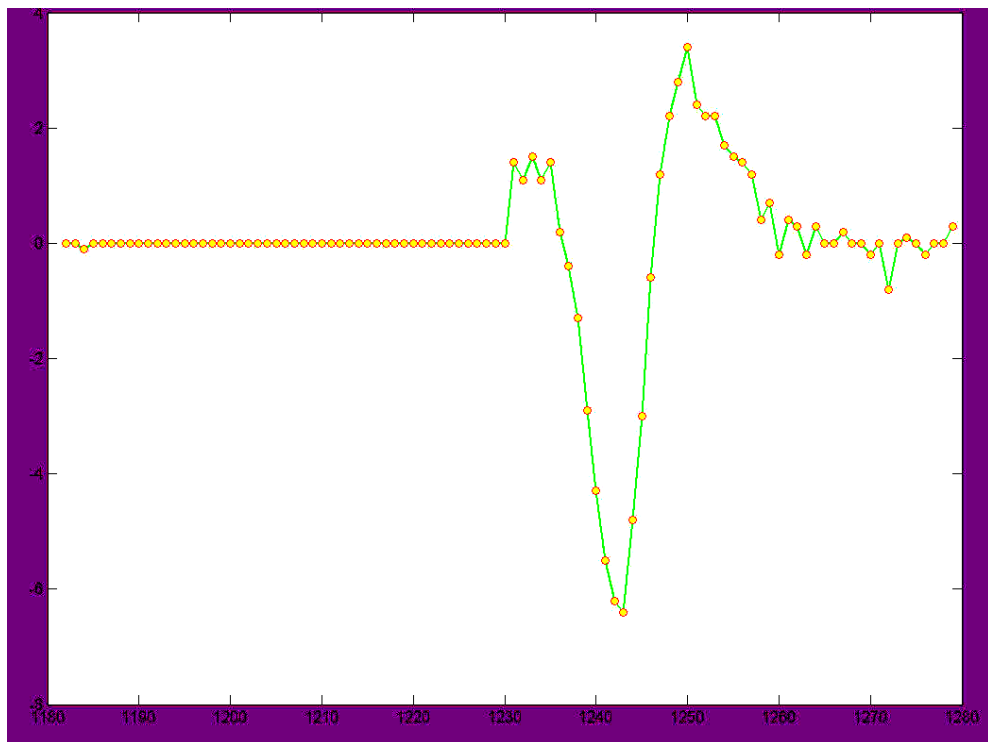
Absolute Data, 200 Hz, No damage
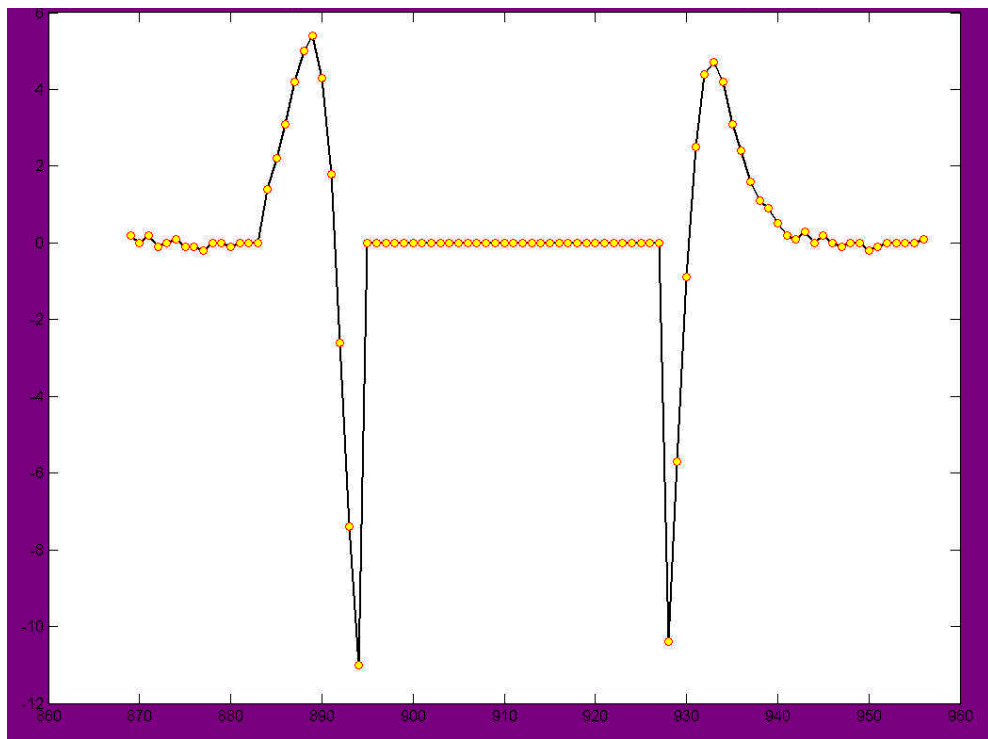
Differential Data, 100 Hz, No damage

Differential Data, 200 Hz, Damage

Absolute Data, 200 Hz, Damage
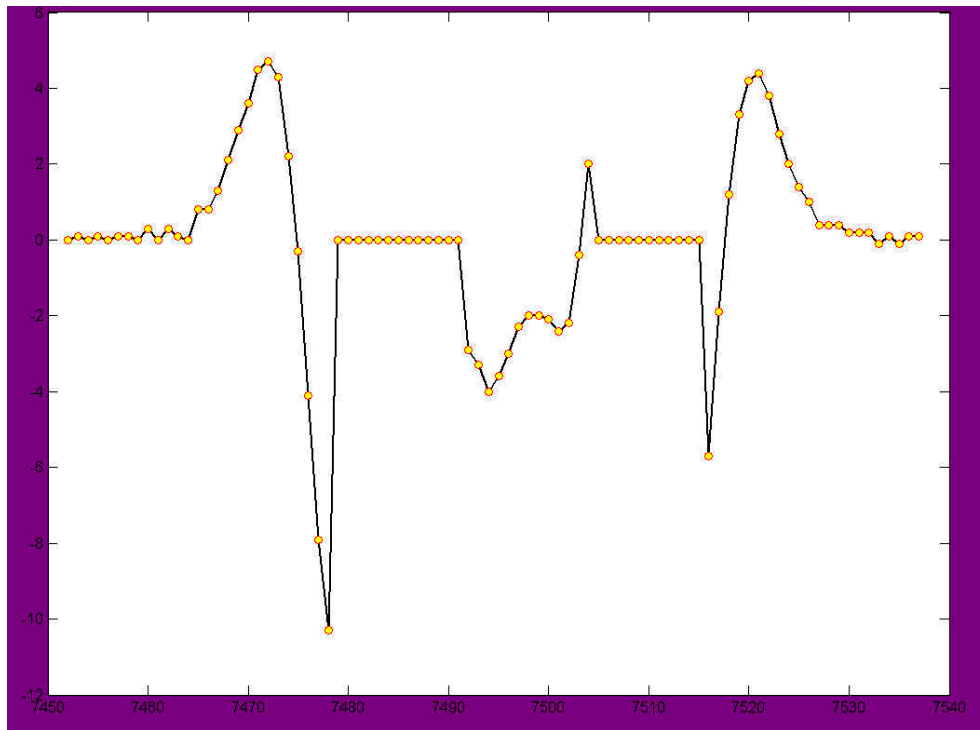
Differential Data, 100 Hz, Damage

CODE USED TO EXTRACT FEATURE VECTORS

```
% This program is used to read in raw data and generate feature vectors
for
% the data file.  The raw data is the unfilterd figure 8 data.  Note this
% is the orginal data unaltered.

% File created: filename.feature_sp_error type.txt => feature vector file
% #'s are listed below and show the differences in the feature vectors

% sp => smoothing parameter
% error type =>  see ext1 - ext4 below
clc
clear

% Read-in Data File filename.fig8.raw.txt
openfile

% sp => smoothing parameter
sp = [0.1 0.3 0.5]';
ssp = [1 3 5]';

for iikk = 1:3

%open file to write figure 8 data to file
ext1=['.feature_' int2str(ssp(iikk)) '_sse.txt']; % data file for sum of
squares due to error
ext2=['.feature_' int2str(ssp(iikk)) '-rsq.txt']; % data file for R-square
ext3=['.feature_' int2str(ssp(iikk)) '_adr.txt']; % data file for Adjusted
R-square
ext4=['.feature_' int2str(ssp(iikk)) '_rms.txt']; % data file for Root
mean squared error

name1=[name ext1];
name2=[name ext2];
name3=[name ext3];
name4=[name ext4];

fid1=fopen(name1,'w');
fid2=fopen(name2,'w');
fid3=fopen(name3,'w');
```

```
        fid4=fopen(name4,'w');

% Read in raw data file containing blips
blpc=0;  %used to count the number of blips read in
for ii=1:100
    blpc=ii;
    [dum,count] = fscanf(fid,'%f',[16,1]); % 16 is the number of columns
of data (there is only one row)
    if count==0
        break
    end

    % The first element in the feature vector indicates if the blip is
    % damaged or undamaged.  0 => no damage     1 => damage
    blp=int2str(ii);
    dam1=['Enter ''1'' if Blip ' blp ' is damaged or ''0'' for no damage:
'];
    clc
    dam=input(dam1);

    if dam == 1
        fv1(1,ii)  = 1;
        fv2(1,ii)  = 1;
        fv3(1,ii)  = 1;
        fv4(1,ii)  = 1;
    else
        fv1(1,ii)  = 0;
        fv2(1,ii)  = 0;
        fv3(1,ii)  = 0;
        fv4(1,ii)  = 0;
    end

    dum = dum';
    lp = dum(1); %this is the start of the peak (digitized data point)
    rp = dum(2); %this is the end of the peak (digitized data point)
    clear dum
    line = fgetl(fid);
    nump = rp-lp+1;  %number of points
    dum = fscanf(fid,'%f',[16,nump]); % 16 is the number of columns of
data (there is nump digitized data points)
    dum = dum';

% Seperate data into differential and absolute sets
    for i = 1:4 %4 because there are 4 frequencies for the data
        xdif(:,i)=dum(:,(i-1)*4+1);  %differntial x data
        ydif(:,i)=dum(:,(i-1)*4+2);  %differntial y data
        xabs(:,i)=dum(:,(i-1)*4+3);  %absolute x data
        yabs(:,i)=dum(:,(i-1)*4+4);  %absolute y data

        %Here we find the mean of the signal and adjust the signal such
that
        %the signal is centered around 0 in the vertical axis
        xdifm=mean(xdif(:,i));
        ydifm=mean(ydif(:,i));
        xabsm=mean(xabs(:,i));
        yabsm=mean(yabs(:,i));

        if xdifm < 0
            xdif(:,i)=xdif(:,i)+abs(xdifm);
        else
            xdif(:,i)=xdif(:,i)-xdifm;
        end

        if ydifm < 0
            ydif(:,i)=ydif(:,i)+abs(ydifm);
        else
            ydif(:,i)=ydif(:,i)-ydifm;
        end

        if xabsm < 0
            xabs(:,i)=xabs(:,i)+abs(xabsm);
        else
            xabs(:,i)=xabs(:,i)-xabsm;
        end

        if yabsm < 0
            yabs(:,i)=yabs(:,i)+abs(yabsm);
        else
            yabs(:,i)=yabs(:,i)-yabsm;
        end
    end
```

```matlab
      ddp=[lp:rp-1]';  % digitized data point vector one less point because
we calculate the slope
%   ddp=[lp:rp]';  % digitized data point vector

% here we calculate the slope associated with the figure 8's.  This is the
% slope of the figure 8's and not the slope of the x-data or the y-data
    for i=1:4  %there are 4 frequncies of data
        for kk=1:nump-1
            delfxdif=xdif(kk+1,i)-xdif(kk,i);
            delfxabs=xabs(kk+1,i)-xabs(kk,i);
            if delfxdif < 0.000001
                delfxdif=0.001;
            end
            if delfxabs < 0.000001
                delfxabs=0.001;
            end
            slfdif(kk,i)=(ydif(kk+1,i)-ydif(kk,i))/(delfxdif);
            slfabs(kk,i)=(yabs(kk+1,i)-yabs(kk,i))/(delfxabs);
        end


        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % here we extract information and construct the feature vectors %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


        if i==2 % here we use a Smoothing Spline to fit the data
            % for i = 2 we are looking at the second frequancy.
            % We are using the Smoothing Spline on the differential and
            % absolute data

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%       Differential Data Picking beginning and ending positive peak
%
%    A goodness of fit value will be used as the second feature vector
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
            fig=figure('Position',[20 20 1250 900],'Color',[0.5 0.25 .5]);
            plot(ddp,slfdif(:,i),'g','LineWidth',1.5)
            axis([min(ddp)+1  max(ddp)+1 min(slfdif(:,i))-1000
max(slfdif(:,i))+1000])
            ti=['Select Begining and Ending Positive Peaks (Blip ' blp
')'];
            title(ti,'FontSize',15,...
                'FontWeight','bold','Color',[1 1 0])
            hold

plot(ddp,slfdif(:,i),'bo','MarkerFaceColor','y','MarkerEdgeColor','r',...
                'MarkerSize',5,'LineWidth',.2)
            h=gca;
            set(h,'Color',[.7 .7 .7])
            dcm_obj = datacursormode(fig);
            set(dcm_obj,'Enable','on','SnapToDataVertex','on')
            pause(.5)
            waitforbuttonpress
            cursor_info = getCursorInfo(dcm_obj);
            a=cursor_info(1).Position;
            pause(.5)
            waitforbuttonpress
            cursor_info = getCursorInfo(dcm_obj);
            b=cursor_info(1).Position;
            close all
            nn=length(slfabs(:,i));
            % Find right end data point
            for nl=1:nn
                if ddp(nl)==a(1)
                    break
                end
            end
            % Find left end data point
            for nr=1:nn
                if ddp(nr)==b(1)
                    break
                end
            end

            % Apply Smoothing Spline
            x=ddp(nl:nr);
```

```matlab
            y=slfdif(nl:nr,i);

            fo_ =
fitoptions('method','SmoothingSpline','SmoothingParam',sp(iikk));
            ft_ = fittype('smoothingspline' );
            % Fit this model using new data
            [fresult,gof] = fit(x,y,ft_ ,fo_);
            figure('Position',[20 20 1250 900],'Color',[0.5 0.25 .5]);
            plot(fresult);
            hold
            plot(x,y,'b*')
            title('Curve Fit (Screen will regenerate in 2
seconds)','FontSize',15,...
                'FontWeight','bold','Color',[1 1 0])
            pause(2)
            close

            fv1(2,ii) = gof.sse;        %sum of squares due to error
            fv2(2,ii) = gof.rsquare;    %R-square
            fv3(2,ii) = gof.adjrsquare; %Adjusted R-square
            fv4(2,ii) = gof.rmse;       %Root mean squared error


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%          Absolute Data Picking beginning and ending of curve
%
%     A goodness of fit value will be used as the third feature vector
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
            fig=figure('Position',[20 20 1250 900],'Color',[0.5 0.25 .5]);
            plot(ddp,slfabs(:,i),'g','LineWidth',1.5)
            ti=['Select Begining and Ending of Curve (Blip ' blp ')'];
            title(ti,'FontSize',15,...
                'FontWeight','bold','Color',[1 1 0])
            hold

plot(ddp,slfabs(:,i),'bo','MarkerFaceColor','y','MarkerEdgeColor','r',...
                'MarkerSize',5,'LineWidth',.2)
            h=gca;
            set(h,'Color',[.7 .7 .7])
            dcm_obj = datacursormode(fig);
            set(dcm_obj,'Enable','on','SnapToDataVertex','on')
            pause(.5)
            waitforbuttonpress
            cursor_info = getCursorInfo(dcm_obj);
            a=cursor_info(1).Position;
            pause(.5)
            waitforbuttonpress
            cursor_info = getCursorInfo(dcm_obj);
            b=cursor_info(1).Position;
            close all
            nn=length(slfabs(:,i));
            % Find right end data point
            for nl=1:nn
                if ddp(nl)==a(1)
                    break
                end
            end
            % Find left end data point
            for nr=1:nn
                if ddp(nr)==b(1)
                    break
                end
            end

            % Apply Smoothing Spline
            x=ddp(nl:nr);
            y=slfabs(nl:nr,i);

            fo_ =
fitoptions('method','SmoothingSpline','SmoothingParam',sp(iikk));
            ft_ = fittype('smoothingspline' );
            % Fit this model using new data
            [fresult,gof] = fit(x,y,ft_ ,fo_);
            figure('Position',[20 20 1250 900],'Color',[0.5 0.25 .5]);
            plot(fresult);
            hold
            plot(x,y,'b*')
            title('Curve Fit (Screen will regenerate in 2
```

```
seconds)','FontSize',15,...
            'FontWeight','bold','Color',[1 1 0])
        pause(2)
        close

        fv1(3,ii) = gof.sse ;        %sum of squares due to error
        fv2(3,ii) = gof.rsquare;     %R-square
        fv3(3,ii) = gof.adjrsquare;  %Adjusted R-square
        fv4(3,ii) = gof.rmse;        %Root mean squared error
    end


    if i==3 % here we select the the most negative point between
        % beginning and ending negative peaks
        % for i = 3 we are looking at the third frequancy.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%   Differential Data Picking Most Negative Point  Between Negative Peaks
%
% The value od the negative point will be used as the fourth feature
vector %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
        fig=figure('Position',[20 20 1250 900],'Color',[0.5 0.25 .5]);
        plot(ddp,slfdif(:,i),'k','LineWidth',1.5)
        ti=['Select Negative Peak Between First and Last Peaks (Blip '
blp ')'];
        title(ti,'FontSize',15,...
            'FontWeight','bold','Color',[1 1 0])
        hold

plot(ddp,slfdif(:,i),'bo','MarkerFaceColor','y','MarkerEdgeColor','r',...
        'MarkerSize',5,'LineWidth',.2)
        h=gca;
        set(h,'Color',[.7 .7 .7])
        dcm_obj = datacursormode(fig);
        set(dcm_obj,'Enable','on','SnapToDataVertex','on')
        pause(.5)
        waitforbuttonpress
        cursor_info = getCursorInfo(dcm_obj);
        a=cursor_info(1).Position;
        close all
        fv1(4,ii) = a(2);   %Value of negative peak
    end
  end
end

fvv=min(fv1(4,:));

fv1(4,:)=fv1(4,:)/fvv;
fv2(4,:)=fv1(4,:);
fv3(4,:)=fv1(4,:);
fv4(4,:)=fv1(4,:);

fprintf(fid1,'  %12.4e  \n',fv1);
fprintf(fid2,'  %12.4e  \n',fv2);
fprintf(fid3,'  %12.4e  \n',fv3);
fprintf(fid4,'  %12.4e  \n',fv4);

fclose(fid1);
fclose(fid2);
fclose(fid3);
fclose(fid4);

frewind(fid)

clear fv1 fv2 fv3 fv4

end
```

Neural architecture

A great deal of background research was completed to understand what types of neural architectures have been used to analyze signal data. This literature search indicated that

among the most widely used neural approaches recently are those with less dependence on the availability of training data. However, almost all researchers working with neural architectures stressed the importance of having an adequate amount of training data for the networks. In several studies, researchers noted that the results of their efforts could be further improved with more training data. These findings led us to investigate candidate architectures that would overcome the drawbacks of networks more heavily reliant on training data such as the multi-layer perception architectures. In addition, we experimented with simulated data to test candidate networks. After researching and testing fuzzy ART , we considered radial basis networks and self-organizing networks using learning vector quantization (LVQ). Ultimately we moved forward with the LVQ network architecture.

The LVQ architecture is a member of the class of self-organizing models (Kohonen, 1995). In the learning vector quantization method, a first-step hidden, competitive layer is used to refine weights according to a supervised learning approach. Training algorithms may vary according to the problem type and desires of the designer. Since the LVQ in its basic form uses a training algorithm where only a single weight is refined based on the vector input, a number of alternative training algorithms have been introduced for the LVQ. These algorithms work to update multiple weights, rather than only the winning weight during each training epoch. As a result, the prediction accuracy increases with the introduction of more complex training algorithms.

The second step of the LVQ method involves classifying the input vector to an output target or class. For a detailed background on LVQs see Karayiannis (1997), Hollmen et. al (2000) and Kohonen et. al (1996). In our project, as with other researchers' projects, one of the greatest challenges lies in the training phase. Because of the nature of defects we are considering, pulled tube training data is limited since we initially focused on IGA/SCC at tube support plates for a specific Westinghouse nuclear steam generator.

Over 100 different computer programs were written for the manipulation, testing and analysis of data. Three different programs were used in the analysis stage of this project: Excel, SPSS 9.0 and MatLab 7.0.

Excel was used primarily for basic data analysis, transformation and visualization at the beginning stages of the project.

The fuzzy and neural toolboxes were both used in MatLab for the design and testing of defect vector inputs. Several different programs were written using the radial basis, probabilistic and learning vector quantification network architectures. Program 1 below is an example of an LVQ network program that called vector sets consisting of binary inputs based on four lissajous characteristics.

```
Program 1

%===========================================================
% Program for creating, training, simulating and testing the damage
% detection learning vector quantization neural network.
%
% Prepared by Automated Diagnosis of Eddy Current Signature Data team
% Linda Ann Riley on version 1.1
%
% Version 1.5
% September 1, 2003
%
% Inputs
%
dummy=importdata('vectorbinary.m');
```

```
rows=size(dummy);
datavects=dummy(2:rows,:);  % second-last rows comprise input vectors
classes=dummy(1,:);  % first row in file is classes
%
%=========================================================
% Parameters for creating the LVQ damage detection network.  Passed to newlvq.
%
PR = P              % PR = R-by-2 matrix of min. and max. values for R input elements
S1 = 2              % S1 is the number of first layer hidden neurons.
PC = [.5 .5]    % PC is an S2 element vector of class percentages.
LR = .01            % is the learning rate of .01.
LF = 1    % is the learning function learnlv1 = 1.
%
%=========================================================%
% Conversion and display of the target matrix
%
% Conversion of the target matrix to target vectors.
Targets=full(T)        % Displays matrix
%
%=========================================================
%
% Create the damage detection network.  See above for parameters...damagenet =
newlvq(PR,S1,PC,LR,LF)
%
damagednet = newlvq(minmax(P), 2, [.50 .50], .01); % Creation of the damagenet


W = midpoint(2,[0 0; .5 .5])      % Select and display the midpoint of the weights
damagednet.IW{1,1}                % Check and display the first layer weights
damagednet.LW{2,1}                % Check and display the second layer weights

%
%=========================================================
%
% Simulate the network to check for initial weight assignments
%
Y = sim(damagednet,P);
%Y = vec2ind(Yb4t)
%
%=========================================================
%
% Train the damagenet using the LVQ1 1 algorithm
%
damagednet.trainParam.epochs = 500;   % Stopping criteria
damagednet = train(damagednet,P,T);   % Calls newly created damagenet, input vectors and
target vectors
%
%=========================================================%
% Check and display the first-layer weights
%
damagednet.IW{1,1}
%
%=========================================================%
% Simulate the trained network to check for targets
%
Y = sim(damagednet,P)   % Simulation of the net to check for target accuracy
Yc = vec2ind(Y)
%
% Vector check
%
pchk1 = [1 0 0 1];                   % Input vector close to a target
Y = sim(damagednet,pchk1);     % Simulate the check on the input vector
Yc1 = vec2ind(Y)                     % Request and display second layer target
%
%=========================================================
```

A great deal of experimentation in this project occurred varying two basic parameters: 1)
the vector inputs; and 2) the neural architecture. Within the neural architecture, several
different approaches were used to vary the weights, hidden layers and training
algorithms.

Validation:

SPSS was used to further analyze and validate the results of the neural networks. The
best performing neural network, (the LVQ), possessed a predictive capacity of 91%

using binary vector inputs describing the damage incidences. Applying binary regression on the same vector sets by designating the damaged or undamaged vector element as the dependent variable, the identical prediction level (91%) was achieved using forced stepwise regression. Thus, no better performance was achieved using binary vector inputs with a neural architecture versus a regression approach.

Project Management and Key Personnel

The principal investigators for this project were, Dr. Gabe Garcia, Dr. Linda Ann Riley, and Dr. Bahram Nassersharif. Dr. Garcia is a professor in the Mechanical Engineering Department who has expertise in pattern recognition as related to instrumentation and sensor data. He was also a co-investigator on the In Tube Radar project mentioned earlier. Dr. Riley has experience in application of neural networks and genetic algorithms. Dr. Nassersharif has an extensive track record in research and publications related to steam generator tube ruptures, use of expert systems and neural networks.

John Schaub, a student in Mechanical Engineering and Engineering Physics, performed much of the data manipulation, analysis, and laboratory testing of the software. He has also been the main project archivist and data manager.

Facilities and Resources

The computational facilities, equipment, and basic software were provided in the Mechanical Engineering Department at New Mexico State University. The proposed project was leveraged with the In Tube Radar project to provide the necessary test equipment and data. We acknowledge the support provided by the EPRI NDE center in Charlotte, North Carolina on the generation of training data and calibration and test samples with respect to eddy-current data. EPRI support was instrumental to our progress and success in obtaining appropriate data sets.

Relationship to Existing Projects and Other Proposals

This project will interact closely with a collaborative NERI project between New Mexico State University (NMSU), Sandia National Laboratories, and EPRI. The NERI project is a three year project funded by DOE under grant number DE-FG-3-99SF21986. The In-Tube Radar (ITR) project resulted in the creation of a prototype probe, however, the project scope and funding did not allow for collection of sample training sets to be used in this project.

Publications and Presentations

L. A. Riley, B. Nassersharif, G. Garcia and J. Schaub. "An Automated Testing and Classification System For Identifying Defects in Nuclear Steam Generator Tubes Using a Learning Vector Quantization Neural Architecture," Proceedings of the 2003 Advanced Simulation Technologies Conference, Society for Computer Simulation International, Orlando, Florida (April 2003).

References

Aditya, R. and S. Ghosal. 2000. "An Integrated Segmentation Technique for Interactive Image Retrieval." Proceedings of IEEE Conference on Image Processing (ICIP 2000), Vancouver, Canada, September.

Au-Yang, M. K. and J. C. Griffith. 2001. "Analysis of Eddy Current Data Using a Probabilistic Neural Network (PNN)." ASME Pressure Vessel and Piping Conference, Atlanta, Georgia - July 22-26.

Bell, J. W. 1988. "An Engineer's Guide to Eddy Current Testing", Section 1 of the Fifth EPRI Balance-of-Plant Heat Exchanger NDE Symposium, June 15-17, Incline Village, Nevada.

Brooks, S. 2000. "Continuous Wave In-Tube Radar: Development of an Experimental Prototype," M.S.M.E. thesis, New Mexico State University.

Caffey, T. W. H. 1997a "A Tool to Detect External Cracks from within a Metal Tube," Technical Report SAND97-0170, Sandia National Laboratories, August, Albuquerque, NM.

Caffey, T. W. H. 1997b "Elements of a Continuous-Wave Borehole Radar," Technical Report SAND97-1995, Sandia National Laboratories, August, Albuquerque, NM.

Caffey, T. W. H. 1998. "A Range-to-Target Algorithm for a Continuous-Wave Ground Penetrating Radar", Proceedings of the 7th International Conference on Ground-Penetrating Radar, May 27-30, The University of Kansas, Lawrence, Kansas.

Caffey, T. W. H., Nassersharif, B., Garcia, G. V., Smith, P. R., Jedlicka, R. P., Hensel, E. C. 2000. "An In-Tube Radar for Detecting Cracks in Metal Tubing," Proceedings of Sixth EPRI Balance-of-Plant Heat Exchanger NDE Symposium, June.

Carpenter, G.A. and S. Grossberg. 1994. "Fuzzy ARTMAP: A Synthesis of Neural Networks and Fuzzy Logic for Supervised Categorization and Nonstationary Prediction." In Fuzzy   Sets, Neural Networks, and Soft Computing, R.R. Yager and L.A. Zadeh, eds., New York.

Diercks, D. R., Shack, W. J., Muscara, J. 1996. "Overview of Steam Generator Tube Degradation and Integrity Issues", Proceedings of the U. S. Nuclear Regulatory Commission, Twenty-Fourth Water Reactor Safety Information Meeting, Bethesda, Maryland, October 21-23, Vol. 2, 297-312.

Fortemps, P. 1997. "Jobshop Scheduling with Imprecise Durations: A Fuzzy Approach." IEEE transactions on Fuzzy Systems 5, No. 4, November: 557-569.

Fukunaga, K. 1990. "Introduction to Statistical Pattern Recognition," Academic Press, New York.

Glorieux, C.; J. Moulder; J. Basart; J. Thoen. 1999. "The determination of electrical conductivity profiles using neural network inversion of multi-frequency eddy-current data." Journal of Physics. D, Applied Physics, Vol. 32, No. 5, March 7: 616-622.

Gorman, J.; R. Ravas; L. Syrova. 2001. "Application of Neural Networks in Multifrequency Eddy-Current Testing." Measurement Science Review, Vol. 1, No. 1: 25-28.

Hollmen, J; Tresp, V.; Simula, O. 2000. "A learning vector quantization algorithm for

probabilistic models." Proceedings of EUSIPCO 2000. Vol. 2: 721-724.

James, M. 1985. "Classification Algorithms," John Wiley & Sons Inc., U.S.A.

Jedlicka, R.P., Castillo, S.P., and Warne, L.K. 2000. "Coupling Through Tortuous Path Narrow Slot Apertures into Complex Cavities," IEEE Transactions on Antennas and Propagation, March, Vol. 48, No. 3, 456 - 466.

Karayiannis, N. 1997. "Learning vector quantization: A review." International Journal of Smart Engineering Design, Vol. 1, No. 1: 33-58.

Katragadda, G.; D. Lewis; J. Wallace; J. Si. 2000. "Swept Frequency Eddy Current Material Profiling Using Radial Basis Function Neural Networks for Inversion." Materials Evaluation, Vol. 48, March: 342-347.

Klimasauskas, C. 2002. "Taking Backpropagation to the Extreme: A Master Shares His Secrets." PCAI, January/February: 31-33.

Kohonen, T.; Hynninen, J.; Kangas, J.; Laaksonen, J.; and Tokkola, K.. 1996. "LVQ_PAK: The learning vector quantization package." Technical Report A30, Helsinki University of Technology, Laboratory of Computer and Information Science.

Kohonen, T. 1995. Self Organizing Maps Springer-Verlag. Popescul, A.; L. H. Ungar; D. M. Pennock; S. Lawrence. 2001. "Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments." In Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence (UAI-2001), UAI: 437-444.

Kuo, R.J. 2001. "A Sales Forecasting System Based on a Fuzzy Neural Network with Initial Weights Generated By Genetic Algorithms." European Journal of Operational Research 129: 496-517.

Lazarev, Y. N. et al. 2000. "Three-Dimensional Unified Codes for Solving the Forward Problem of Single well Electromagnetic Logging", Report on Task 4A of Contract AS-5553, Sandia National Laboratories, March, Albuquerque, NM.

Lin, C.T. and G. Lee. 1996. Neural Fuzzy systems, A Neuro-Fuzzy Synergism to Intelligent Systems, Prentice Hall, Upper Saddle River, NJ.

Luh, P.B.; X. Zhao; Y. Wang; L.S. Thakur. 2000. "Lagrangian Relaxation on Neural Networks for Job Shop Scheduling." IEEE Transactions on robotics and Automation 16, No. 1, February: 78-88.Add interest Select an idea or topic for your report.

MacDonald, P. E., Shah, V. N., Ward, L. W., Ellison, P. G. 1995. "Steam Generator Tube Failures", NUREG/CR-6365, INEL-95/0383.

Nassersharif, B., Caffey, T. W. H., Garcia, G. V., Smith, P. R., Jedlicka, R. P., Hensel, E. C. 2001. "An In-Tube Radar for Detecting Defects in Thin-Walled Metal Tubes," 16th International Conference on Structural Mechanics in Reactor Technology (SMiRT 16), August.

Nari, B. 1997 "Steam Generator Replacement and Repair Technology Continues to Advance", Power Engineering Magazine, April: 22-37.

Rao, B. P. C.; B. Raj; T. Jayakumar; P. Kalyanasundaram. 2002. "Using Artificial Neural Networks to Quantify Discontinuities in Eddy Current Testing." Materials Evaluation, Vol. 60, No. 1, January: 84-88.

Siegel, M. W. 1994. "Automation for Nondestructive Inspection of Aircraft." The Robotics Institute, School of Computer Science, Carnegie Mellon University. Signals and Systems

Group, Uppsala Unviersity. 1999.

"Neural Network Classifier for Eddy Current Signals."
www.signal.uu.se/Research/CANDIA/CANDIA.html.

Simone, G. 2001. "RBFNN-Based Hole Identification System in Conducting Plates." IEEE Transactions on Neural Networks, Vol. 12, No. 6, November: 1445-1454.

Smith, A. 1993. "Predicting Product Quality with Backpropagation: A Thermoplastic Injection Molding Case Study." International Journal of Advanced Manufacturing Technology, Vol. 8, April: pp. 252-257.

Spanner, J.; L. Udpa; R. Polikar; P. Ramuhalli. 2000. "Neural Networks for Ultrasonic Detection of Intergranular Stress Corrosion Cracking," The e-Journal of Nondestructive Testing And Ultrasonics, Vol. 5, No. 7, July.

Upda, L. and S. S. Upda. 1990. "Eddy Current Defect Characterization Using Neural Networks." Materials Evaluation, Vol. 48, March: 342-347.

Upda, L. and S. S. Upda. 1991. "Neural Networks for the Classification of Nondestructive Evaluation Signals." IEE Proceedings-F, Vol. 138, No. 1, February: 41-45.

Wait, J. R. and Hill, D. A., 1977, "Electromagnetic Fields of a Dipole Source in a Circular Tunnel Containing a Surface Wave Line", International Journal of Electronics, Vol. 42, No. 4,377-391.

Xiang, P.; S. Ramakrishnan; X. Cali; P. Ramuhalli; R. Polikar; S. S. Upda; L. Upda. 2000. "Automated Analysis of Rotating Probe Multi-Frequency Eddy Current Data from Steam Generator Tubes." International Journal of Applied Electromagnetics and Mechanics, 12: 151-164.