5-2004

Neural Network Fatigue Life Prediction in 7075-T6 Aluminum from Acoustic Emission Data

Emeka Chigozie Ibekwe
Embry-Riddle Aeronautical University - Daytona Beach

NEURAL NETWORK FATIGUE LIFE PREDICTION IN 7075-T6 ALUMINUM FROM ACOUSTIC EMISSION DATA

by

Emeka C. Ibekwe

A Thesis Submitted to the Graduate Studies Office in Partial Fulfillment of the Requirements for the Degree of Master of Science in Aerospace Engineering

Embry-Riddle Aeronautical University
Daytona Beach, Florida
May 2004

NEURAL NETWORK FATIGUE LIFE PREDICTION IN 7075-T6 ALUMINUM FROM ACOUSTIC EMISSION DATA

by

Emeka C. Ibekwe

This thesis was prepared under the direction of the candidate's thesis committee chairman, Dr. Eric v. K. Hill, Department of Aerospace Engineering, and has been approved by the members of his thesis committee. It was submitted to the School of Graduate Studies and Research and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE:

Eric v. K. Hill

Dr. Eric v. K. Hill
Chairman

Frank J Radosta

Dr. Frank J. Radosta
Member

Dhainaut

Dr. Jean-Michel Dhainaut
Member

K L Narayanaswami

Dr. Lakshman Narayanaswami
Graduate Program Coordinator

6/2/04
Date

Habib Eslami

Dr. Habib Eslami
Department Chair, Aerospace Engineering

6/2/04
Date

ACKNOWLEDGEMENTS

I would like to thank, above all, my parents, who have always encouraged me and been there to support me through my college career. I would also like to thank Dr. Eric v. K. Hill, my thesis advisor, whose tutelage was the greatest factor in achieving more than I ever expected. His teaching and insight have helped me to focus my professional direction. I would also like to thank Dr. Frank J. Radosta for teaching me so much about strength and fatigue and Dr. Jean-Michel Dhainaut for his time and effort in serving on my thesis committee. A special thanks must also go to those fellow students who helped me with this research project: Andrew White, Chris Hess, Ban Seng Keng, Matthew Stubbe, Caroline Liron and Takashi Osako. Their help was invaluable throughout.

ABSTRACT

Author: Emeka Chigozie Ibekwe

Title: Neural Network Fatigue Life Prediction in 7075-T6 Aluminum from Acoustic Emission Data

Institution: Embry-Riddle Aeronautical University

Degree: Master of Science in Aerospace Engineering

Year: 2004

The objective of this research was to classify acoustic emission (AE) data associated with fatigue cracks in aluminum fatigue specimens and to use early cycle life AE data to predict failure in such members. An AE data acquisition system coupled with a Kohonen self organizing map and a back propagation neural network were used to perform the analysis. AE waveforms were recorded during fatigue cycling of twenty-four notched 7075-T6 aluminum specimens using broad-band piezoelectric transducers. A Kohonen self organizing map was used to classify the AE flaw growth signals. The signals were classified into three categories based on their AE parameters: plastic deformation, plane strain fracture and mixed mode (plane strain and plane stress) fracture.

Acoustic emission amplitude data from the twenty-four low cycle fatigue tests were used to train and test a back propagation neural network for prediction of cycles to failure. The input data consisted of amplitude frequency histograms (30-100 dB) and the actual cycle lives. The output was the predicted cycles to failure or fatigue life. A network capable of predicting cycles to failure with a worst case error of - 9.30% was the final result.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

1.1 Problem Statement

The scope of the project was to identify fatigue crack growth and predict failure in 7075-T6 aluminum by classifying acoustic emission (AE) failure mechanism data. The data were categorized using a Kohonen self organizing map (SOM) neural network, and prediction was made possible using a back propagation neural network (BPNN). The aluminum test specimens were notched to promote low cycle fatigue crack growth. They were axially cycled between 0 and 4,000 lb, resulting in 50 ksi mean cyclic stresses and a stress ratio R = 0.0.

1.2 Previous Research

The suitability of acoustic emission for monitoring fatigue crack growth has been established in the past. Acoustic emission in conjunction with neural networks has proven to be an effective approach for detecting and analyzing fatigue crack growth. Almeida and Hill [1,2] used acoustic emission and a SOM neural network to classify signals from a riveted double lap joint during an axial cyclic loading. The riveted joint was constructed using 7075-T6 aluminum. They were then able to properly classify a minimum of 94 percent of crack growth signals and 99 percent of the rivet rubbing signals using a BPNN on power spectral density data.

Thornton and Hill [3] and Marsden and Hill [4,5] used acoustic emission and neural networks to monitor crack growth in a simulated aircraft fuselage. They employed a pressure vessel constructed from 2024-T3 aluminum. This simulated fuselage contained a notched round hole which was patched and cyclically pressurized to stress the notch. Marsden and Hill correctly classified time domain AE quantification parameters (rise time, counts, energy, duration, and amplitude) from this test into cracking, metal rubbing and rivet fretting using a SOM neural network. Thornton and Hill used a SOM and frequency domain (power spectral density) data, to accomplish the same objective.

Subsequently, Vaughn and Hill [6] used acoustic emission to monitor in-flight fatigue crack growth. Employing a SOM neural network they were able to distinguish crack growth, plastic deformation and rubbing signals in a Piper PA-28 aircraft engine cowling during flight. Rovik and Hill [7] monitored the growth of a fatigue crack in the vertical tail of a Cessna T-303 while performing various in-flight maneuvers. This was accomplished using a SOM neural network on AE quantification parameter data. Both projects are summarized in reference [8].

Finally, Ballard and Hill [9,10] predicted fatigue lives in inconel and stainless steel bellows from early cycle AE data. Back propagation neural networks were used initially. Later multivariate statistical analysis was employed as well [11]. The present research effort is an extension of this earlier work in fatigue life prediction.

2.0 BACKGROUND THEORY

2.1 Acoustic Emission

Acoustic emission (AE) is a nondestructive testing (NDT) method wherein data from stress waves generated by the sudden release of energy during flaw growth are collected and analyzed. One advantage of AE testing is that it is a noninvasive form of NDT and that it does not require removal of the structure or specimen from service for testing. A test article may be analyzed while it is in service under its normal operating conditions. Because AE is a passive testing procedure, the specimen requires some type of loading to generate the stress waves. Other forms of NDT, such as radiographic or dye penetrant testing, require an active method of probing the specimen and analyzing the response. One drawback to AE testing is that it is typically necessary to destroy at least one sample to acquire a reference set of AE data in order to properly adjust the test setup and ensure optimum data collection.

2.1.1 Waveform Parameters

Five AE waveform quantification parameters were utilized or collected during the cyclic testing (Figure 2.1). The three parameters used herein were amplitude, counts, and duration. Amplitude, measured in decibels [dB], is the maximum height of the waveform. Counts are the number of recorded peaks above the threshold, which is an artificial filter value below which no data is recorded. Duration, measured in microseconds [μs], is the time the waveform signal lasts.

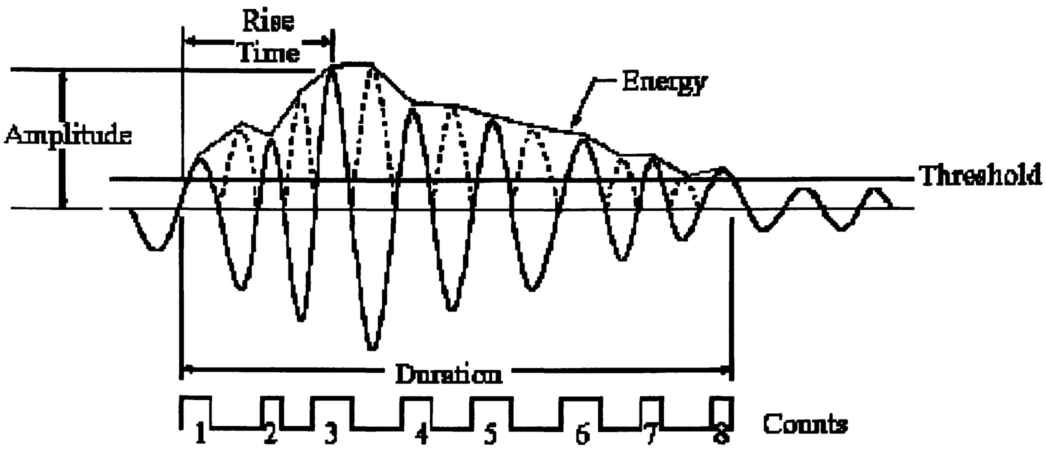


Figure 2.1 Waveform signal

The other two AE parameters are rise time (microseconds [μs]), which is the time it takes to reach the peak amplitude, and energy which is a function of both amplitude and duration. Energy is measured as the area under the rectified signal waveform.

2.1.2 Wave Speed Calculations

Given below are the calculated Lamé constants, λ and μ, which are used to determine the longitudinal wave speed (c_l), transverse wave speed (c_2), plate wave speed (c_p), Raleigh wave speed (c_r) and group wave speed (c_g) for the aluminum specimens.

$$\lambda = \frac{E\upsilon}{(1+\upsilon)(1-2\upsilon)} = \frac{10.4*10^6(0.32)}{1.32(0.36)} = 7.00*10^6 psi$$

$$\mu = \frac{E}{2(1+\upsilon)} = \frac{10.4*10^6}{2(1.32)} = 3.94*10^6 psi$$

$$c_1 = \sqrt{\frac{\lambda + 2\mu}{\rho}} = \sqrt{\frac{7.00*10^6 + 2(3.94*10^6)}{0.0975}} = 12,353 \ in/s$$

$$c_2 = \sqrt{\frac{\mu}{\rho}} = \sqrt{\frac{3.94*10^6}{0.0975}} = 6,357 \ in/s$$

$$c_p = c_2\sqrt{\frac{2}{(1-\upsilon)}} = 6,357\sqrt{\frac{2}{0.68}} = 10,902 \ in/s$$

$$c_r = c_2\sqrt{\frac{0.87 + 1.12\upsilon}{1+\upsilon}} = 6,357\sqrt{\frac{0.87 + 1.12(0.32)}{1.32}} = 6,132 \ in/s$$

$$c_g = 0.91c_p = 0.91(10,902) = 9,232 \ in/s$$

The group velocity c_g was used in conjunction with the data acquisition software to generate a location plot of AE activity. Plots of events versus position can be seen in Appendix A. Note that events could not have been accurately plotted at the notch location if the wave speed calculations were not correct.

2.2 Neural Networks

An artificial neural network (ANN) is a massively parallel system used for data processing comprised of artificial neurons or processing elements that allow a series processor (digital computer) to function as a parallel processing system which more closely resembles the human brain in efficiency and capability. This increase in efficiency allows the network to analyze very complex and noisy data more effectively than by other means such as statistical curve fitting. Two types of ANNs are used here, a self organizing map (SOM) used for classifying data into categories and a back propagation neural network (BPNN) used for predicting cyclic lives based on early cycle

AE failure mechanism data. Details of the operation of these two networks are provided in a tutorial by Walker and Hill [12].

2.2.1 SOM Network Architecture

A Kohonen self organizing map (SOM) neural network is a neural network that is capable of classifying a set of input data into different categories. The SOM architecture shown in figure 2.2 consists of three layers:

- Input layer – the number of nodes for the input layer depends on the number of inputs that are going to be used in the network each time.
- Kohonen layer – the number of nodes for the classification layer depends on the number of different categories that are desired.
- Output layer – step activation functions produce 0 or 1 output to identify the classification category.

Five input neurons were used to input the AE waveform parameters (amplitude, counts, duration, energy and rise time) for each hit. Three Kohonen neurons were then used to classify the AE hits into three categories: localized plastic deformation (0 0 1), plane strain fracture (0 1 0), and mixed mode fracture (1 0 0).

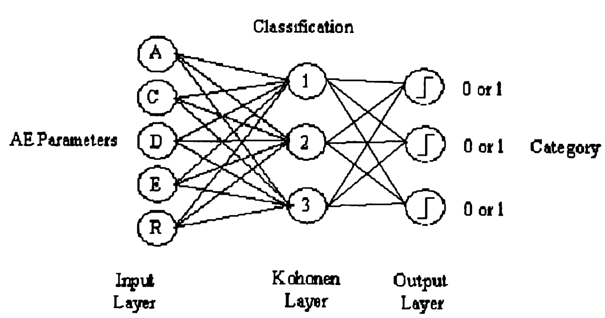


Figure 2.2 SOM network architecture

2.2.2 Back Propagation Network Architecture

Figure 2.3 shows a schematic of the back propagation network architecture. The network consisted of three layers: an input layer, a hidden layer, and an output layer. Seventy-one input neurons were used to input the amplitude frequency histogram from each specimen. The middle or hidden layer was the processing layer, mapping the input into the output. The single output neuron was used to predict cycles to failure.

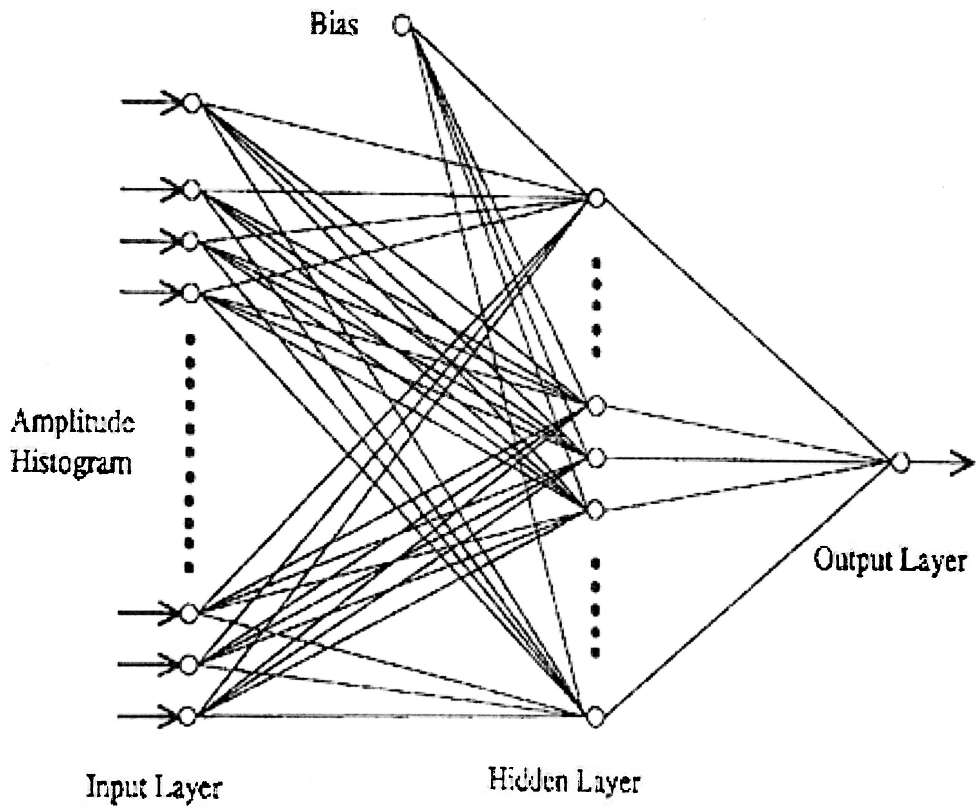


Figure 2.3 Back propagation network architecture

3.0 EXPERIMENTAL SETUP

3.1 Test Specimens

Twenty-four 7075-T6 aluminum test specimens per AMS 4045 were used for the cyclic testing. Figure 3.1 is a sketch of the samples. A 0.200 inch deep, 45 degree angle notch was added to the samples to initiate crack growth.

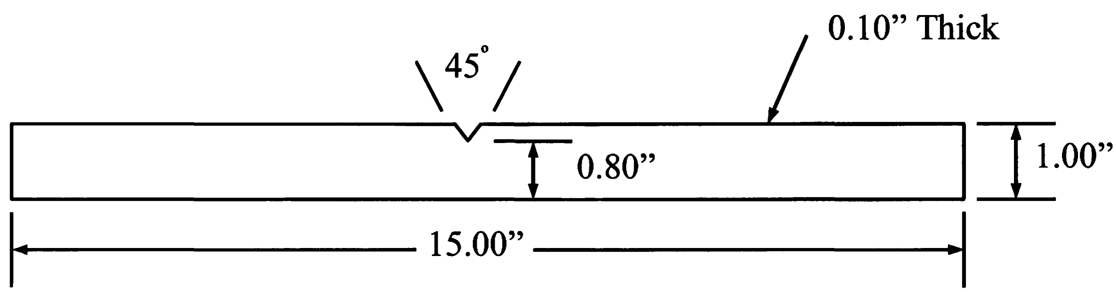


Figure 3.1 Test specimen

3.2 Fatigue Life Calculation

The test specimens were loaded axially to induce mean cyclic stresses of 50 ksi in the material. Shown below are the calculations used to arrive at the desired mean cyclic stress.

$$Area_{cross-section} = 0.08\ in^2$$

$$P_{cyclic} = 4{,}000\ lb$$

$$\sigma_{cyclic} = \frac{P_{cyclic}}{A} = \frac{4{,}000}{0.08} = 50{,}000\ psi$$

To make sure that the load would be sufficient to break the part in a reasonable length of time, a fatigue life chart was consulted. Looking at the fatigue life chart below (Figure 3.2) [13], it can be seen that for a 50 ksi cyclic axial load and a stress ratio R= 0.0 the specimen will experience fatigue at approximately 1600 cycles.

MIL-HDBK-5H
1 December 1998

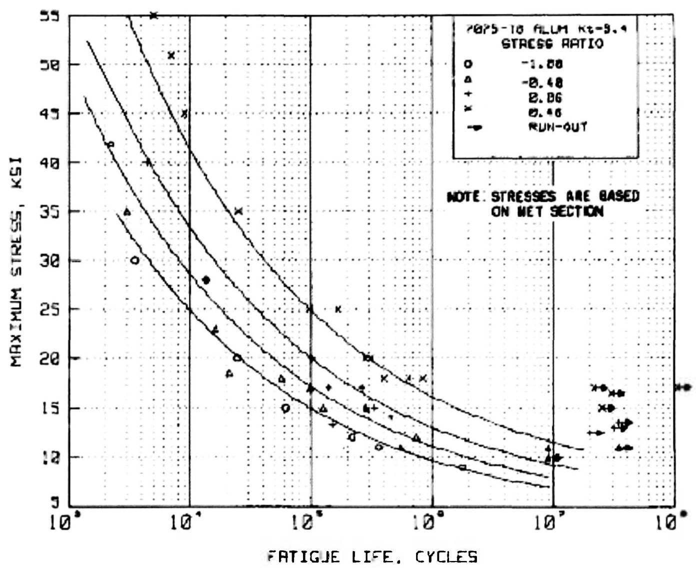


Figure 3.7.4.1.8(c). Best-fit S/N curves for notched, K_t = 3.4, 7075-T6 aluminum alloy rolled bar, longitudinal direction.

Figure 3.2 Fatigue life chart

3.3 Test Equipment Overview

The test equipment consisted of an MTS Machine to provide the desired cyclic loading of the specimens, an MTS Controller and Display units to provide adequate control of the MTS Machine, and a Data Acquisition System (DAS) or computer to record the AE data as they were generated. Figure 3.3 shows a schematic diagram of the data flow within the test system.

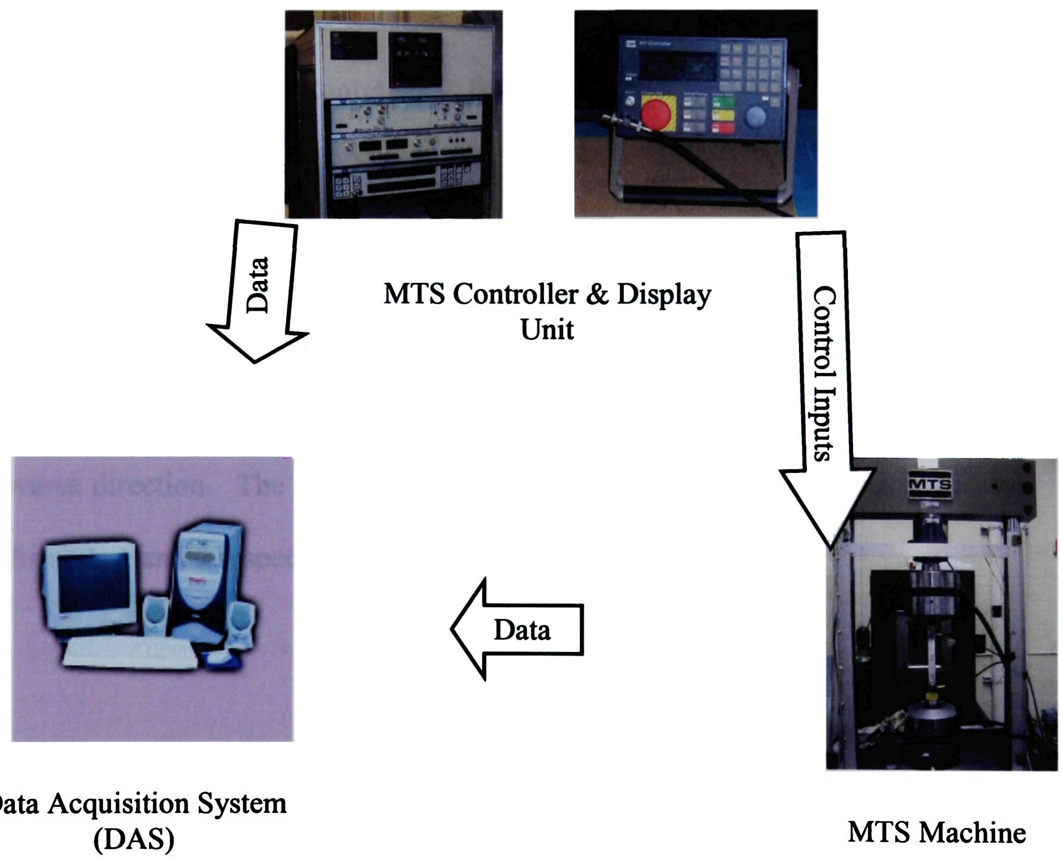


Figure 3.3 Equipment schematic

3.4 MTS Machine and Controller Setup

The process to set up the MTS Machine and Controller was as follows:

1. Disable the interlock.
2. Adjust the level of the lower grip to allow for cycling.
3. Set the Controller to DC to configure the units to pounds (lb).
4. Set the workload of the Controller for the desired cycling load(s) (Setpoint = 2,000 and Span = 2,000 for 0 to 4,000 lb cyclic loading).
5. Connect the transducers and Controller to the DAS.

3.5 Preparation of Specimens

The test specimens were prepared by attaching the AE transducers asymmetrically on both sides of the notch to facilitate location of the fatigue cracks (Figure 3.4). The specimens were then clamped vertically in the MTS Machine to ensure loading in the long transverse direction. The lower and upper transducers were then connected to the DAS as Channels 1 and 2 respectively.

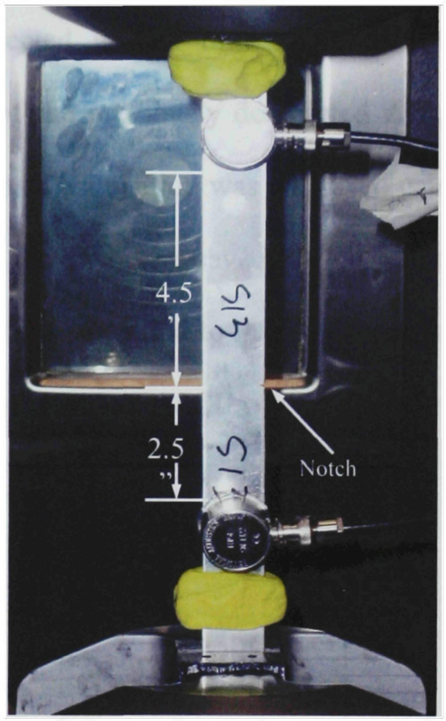


Figure 3.4 Test specimen setup

3.6 Preparation of DAS Software (MISTRAS)

The preparation of the DAS software is outlined as follows:

1. Generate the configuration file.

2. Set: threshold = 30 dB; PDT = 300 μsec; HDT = 600 μsec; HLT = 150μsec.

3. Define the available channels: 1 & 2.

4. Set the distance between the transducers: 7 in.

5. Set the desired output graphs.

6. Set the monitoring and data recording process to standby.

7. Start the monitoring and data recording at the beginning of cycling.

4.0 RESULTS

The data were collected in real time using the MISTRAS data acquisition software. They consisted of the previously designated AE parameters for each signal waveform (hit). Real time data acquisition was displayed using the line dump feature in MISTRAS. Once collected, the data were saved to an electronic file using the autodump function. Saved data were then converted to ASCII format for the analysis described in section 5.0. Figure 4.1 shows a sample ASCII file for thirty-nine AE hits.

```
"(FILE: C:\MISTRAS\IMPACT\SAMPLE~1.DTA)"

"(TEST START DATE & TIME)"
"(Mon Mar 10 14:53:08 2003)"

"(COMMENTS)"
"(MISTRAS-2001 DATA ACQUISITION TEST)"

"(ACTIVE AE DATA SET PARAMETERS)"
"(DDD""HH""MM""SS.mmmuuun)" "(PARA1)"    "(CH)"  "(RISE)""(COUNTS)""(ENERGY)""(DURATION)"   "(AMP)"
0  0 00:00:02.8678513         -10.10         1        1          1         0          5       37
 0 00:00:02.8679213          -10.10         2        1          1         0          1       30
 0 00:00:14.5541655          -10.09         2      116          1         0        117       31
 0 00:00:43.1972753          -10.09         1        1          1         0          1       30
 0 00:01:32.3917240          -10.10         1        1          1         0          1       34
 0 00:02:01.0341610          -10.09         1        1          1         0          1       32
 0 00:02:09.2514013          -10.10         1        3          1         0          4       30
 0 00:02:54.2882037          -10.10         1        1          1         0          1       30
 0 00:03:31.1887135          -10.10         1        1          1         0          1       32
 0 00:03:43.4329365          -10.09         1        1          1         0          1       30
 0 00:04:53.1147675          -10.10         1       10          1         0         46       31
 0 00:04:53.1148527          -10.10         2        1          1         0          1       31
 0 00:05:29.9843787          -10.10         1        1          1         0          1       33
 0 00:05:34.0616217          -10.09         1        1          1         0          1       31
 0 00:05:34.0616627          -10.09         2        1          1         0          1       30
 0 00:05:57.8889270          -10.09         2       53        882        38      13859       42
 0 00:06:40.5880227          -10.09         2       40          4         0         88       33
 0 00:06:40.9235285          -10.09         2        1          1         0          1       31
 0 00:06:40.9285773          -10.10         2        1          1         0          4       32
 0 00:06:40.9596495          -10.09         2       15         12         0        186       35
 0 00:06:41.6582663          -10.09         2       11          2         0         27       31
 0 00:06:41.6839127          -10.10         2        8          3         0         70       34
 0 00:06:41.6865065          -10.10         2       88         15         0        314       37
 0 00:06:41.8729167          -10.10         2        2          2         0        120       33
 0 00:06:41.8808855          -10.09         2        1          1         0          1       30
 0 00:06:41.8849440          -10.10         2        1          7         0        102       36
 0 00:06:41.9321060          -10.09         2       37         11         0        156       36
 0 00:06:41.9506455          -10.09         2       98         14         0        224       35
 0 00:06:41.9543200          -10.10         2        1          1         0          1       31
 0 00:06:41.9747137          -10.09         2        3          1         0          4       31
 0 00:06:41.9770655          -10.10         2      342          7         0        413       34
 0 00:06:41.9910625          -10.10         2       86          8         0        647       36
 0 00:06:41.9947813          -10.10         2      175          5         0        282       33
 0 00:06:41.9987417          -10.10         2       71         11         0        378       33
 0 00:06:42.0008597          -10.10         2       23          2         0        221       33
 0 00:06:42.0179035          -10.09         2       45         25         1        460       41
 0 00:06:42.0220150          -10.10         2        5          5         0        316       34
 0 00:06:42.0254485          -10.09         2        1          1         0          1       33
 0 00:06:42.0274860          -10.10         2        1          1         0          1       33
```

Figure 4.1 Sample ASCII file

Figure 4.2 shows screen plots of the AE parameters. Refer to Appendix A for plots of duration vs. amplitude and events vs. position for the first thirteen samples.

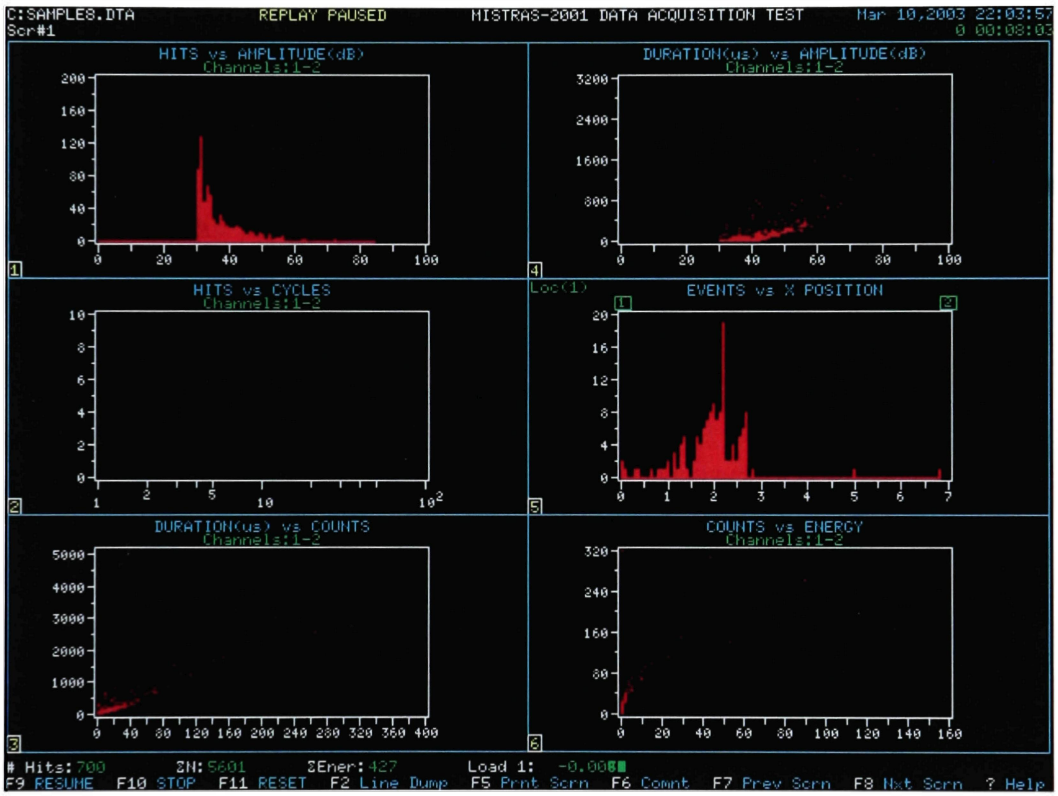


Figure 4.2 AE parameter plots

Note that the hits vs. cycles plot had no output because the plot setup was incorrect for this particular test. This was corrected for later tests.

the flashing conditions, or white in the auditory conditions when instead an auditory alert sounded simultaneously with the appearance of the dialog box. One alert (whether visual or auditory) was present for each condition except in baseline control conditions. The alert did not cease until the participant selected an option in management by consent (MBC) conditions or ran out of time in the management by exception (MBE) conditions (after six seconds had passed). Figure 3 provides an example of an image used in the experiment and a sample dialog box for when the automation stated that a threat existed while Figure 4 provides an example of when automation stated no threat existed. Please note that both Figure 3 and Figure 4 are examples of management by consent (MBC) because an option to accept or reject automation's response is provided.

Figure 3. Example of Management by Consent (MBC) Image with Threat Text

A duration vs. counts plot of the same data demonstrates the overall quality of the data. Correlation of this data is quite high at 96.52 percent, meaning that there were very few noisy hits.

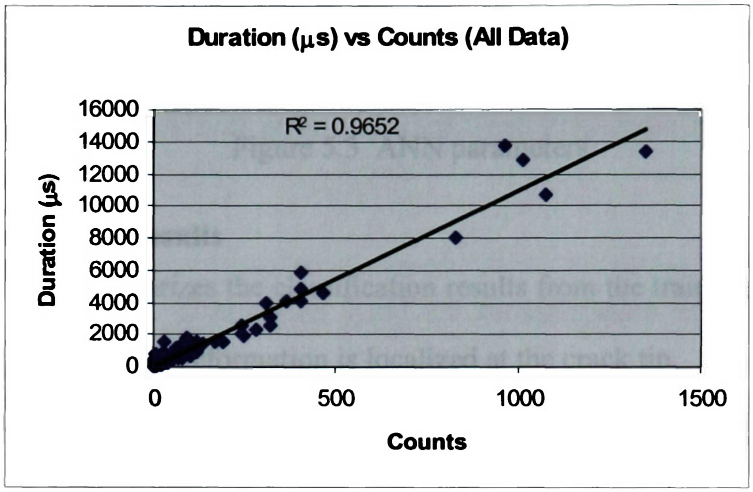


Figure 5.2 Duration vs. counts

5.1.1 SOM NeuralWorks Professional II/Plus Setup

The analysis in this section incorporates the concepts introduced earlier in section 2.2.1. The SOM NeuralWorks Professional II/Plus software has the capability to classify data if configured properly. Detailed instructions on configuring the SOM NeuralWorks Professional II/Plus software are located in the Appendix B. Also reference "Training and Testing A Self-Organizing Map Neural Network using NeuralWorks Professional II/PLUS" [Bibliography] for more details. Figure 5.3 shows the network settings for the analysis.

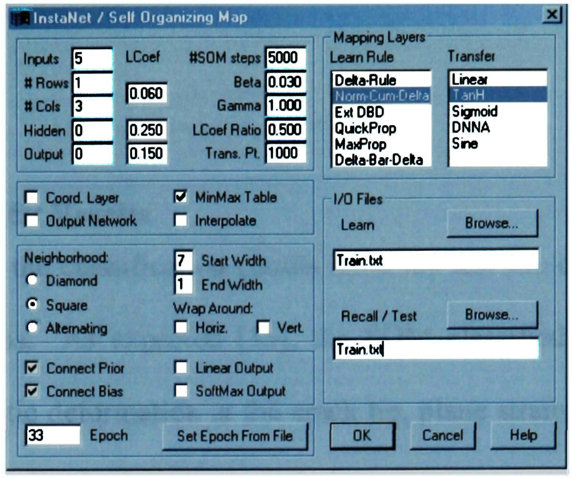


Figure 5.3 ANN parameters

5.1.2 Training File Results

Table 5.1 summarizes the classification results from the training file. It should be noted here that the plastic deformation is localized at the crack tip.

Table 5.1 Training file results

Rise Time	Counts	Energy	Duration	Amplitude	Category			Failure Mechanism
6	23	0	200	43	[illegible]	0	0	[illegible]
15	23	1	197	45	[illegible]	0	0	[illegible]
18	17	0	182	45	[illegible]	0	0	[illegible]
18	17	0	172	46	1	0	0	[illegible]
19	16	0	180	42	1	0	0	[illegible]
20	15	0	157	45	1	0	0	[illegible]
18	14	0	170	45	1	0	0	[illegible]
17	13	0	144	45	1	0	0	[illegible]
17	13	0	144	45	1	0	0	[illegible]
16	12	0	156	43	1	0	0	[illegible]
35	11	0	208	39	1	0	0	[illegible]
1214	843	16796	9606	98	0	0	[illegible]	Plane Strain Fracture
32	775	1479	10096	94	0	0	[illegible]	Plane Strain Fracture
45	390	388	5827	91	0	0	[illegible]	Plane Strain Fracture
88	375	771	4499	91	0	0	[illegible]	Plane Strain Fracture
42	167	112	1783	85	0	0	1	Plane Strain Fracture
124	150	48	1785	76	0	0	1	Plane Strain Fracture
691	146	19	1549	66	0	0	1	Plane Strain Fracture
17	110	31	1422	71	0	0	1	Plane Strain Fracture
12	95	18	1256	68	0	0	1	Plane Strain Fracture
19	55	4	567	58	0	0	1	Plane Strain Fracture
18	50	4	519	58	0	0	1	Plane Strain Fracture
44	5926	4190	48932	96	0	[illegible]	0	[illegible]
48	1094	201	12670	77	0	[illegible]	0	Mixed Mode Fracture
20	643	27	7045	48	0	[illegible]	0	Mixed Mode Fracture
272	533	29	5892	56	0	[illegible]	0	Mixed Mode Fracture
276	426	28	3295	57	0	[illegible]	0	Mixed Mode Fracture
4	320	36	6267	62	0	[illegible]	0	Mixed Mode Fracture
37	211	8	1109	49	0	[illegible]	0	Mixed Mode Fracture
172	195	9	1716	48	0	[illegible]	0	Mixed Mode Fracture
24	161	7	1073	53	0	[illegible]	0	Mixed Mode Fracture
391	149	17	2989	60	0	[illegible]	0	Mixed Mode Fracture
13	131	5	1006	46	0	[illegible]	0	[illegible]

5.1.3 SOM Network Test Results

Figure 5.4 shows the classification results from the test file containing data from all twenty-four specimens. As expected [15], the SOM classified the data into three categories: localized plastic deformation at the crack tip, plane strain fracture, and mixed fracture mode consisting of both plane strain and plane stress components.

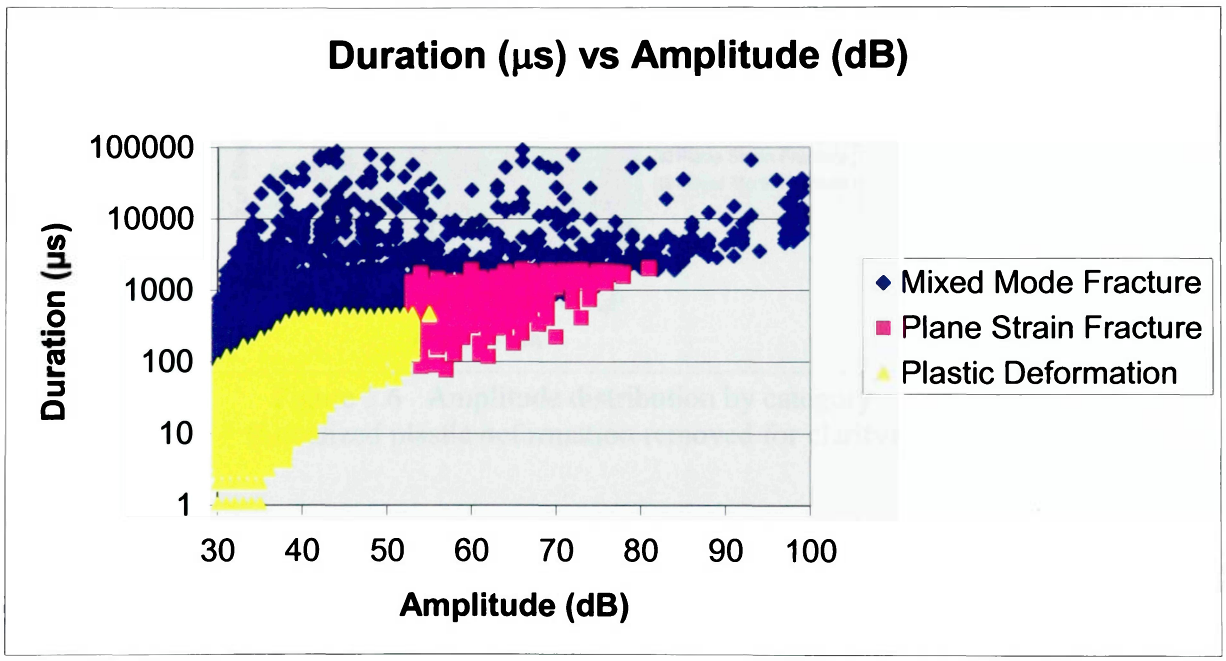


Figure 5.4 Test file results, duration vs. amplitude

Figure 5.5 through 5.7 show the amplitude distribution histograms with the three failure mechanisms broken down into categories and then combined.

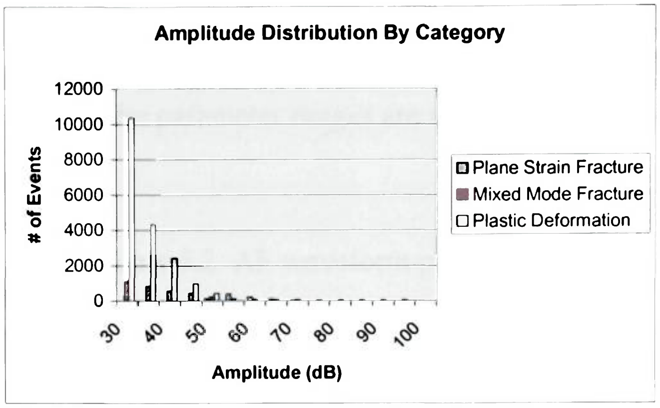


Figure 5.5 Amplitude distribution by category

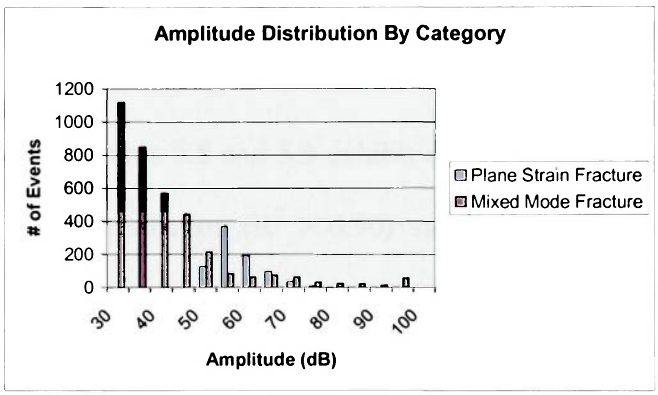


Figure 5.6 Amplitude distribution by category
(Localized plastic deformation removed for clarity)

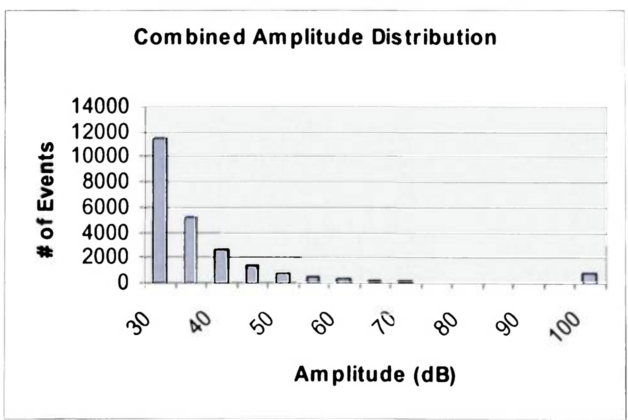


Figure 5.7 Combined amplitude distribution

The approximate waveform parameter ranges are summarized in Table 5.2:

Table 5.2 AE waveform parameter ranges

Failure Mechanism	Energy (counts)	Duration (μs)	Amplitude (dB)
Plastic Deformation	0 - 5	1 - 461	30 - 55
Plane Strain Fracture	0 82	76 - 1,973	53 - 81
Mixed Mode Fracture	0 - 21,846	91 115,201	30 - 99

Furthermore, Figures 5.8 and 5.9 exhibit a high level of data quality due to their high coefficient of determination (R^2 > 0.90) values. This indicates that the data for localized plastic deformation at the crack tip and plane strain fracture signals contain little noise. Mixed mode fracture (Figure 5.10), on the other hand, appears to contain a lot of noise as indicated by the scatter around the trend line and the R^2 value being less than 0.90.

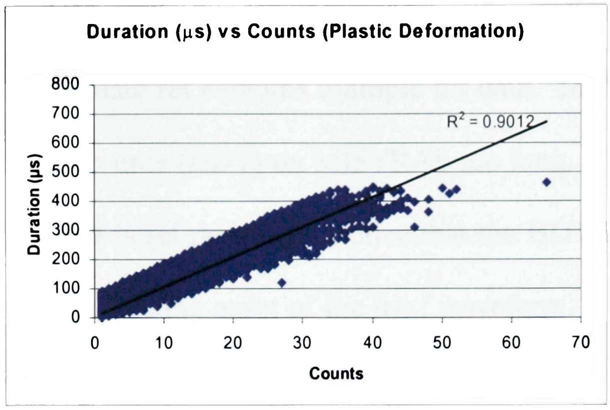


Figure 5.8 Duration vs. counts (plastic deformation)

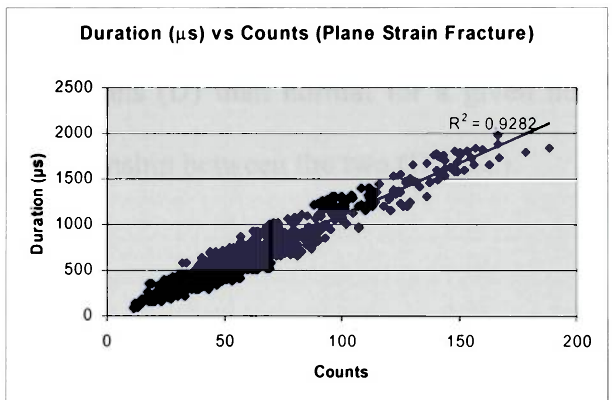


Figure 5.9 Duration vs. counts (plane strain fracture)

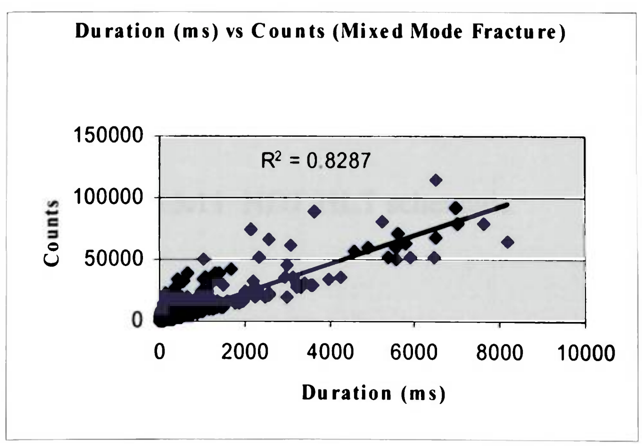


Figure 5.10 Duration vs. counts (mixed mode fracture)

The reason the mixed mode fracture data set has a lower R^2 value than the other two mechanisms is that this data set contains multiple hit data. Multiple hit data comes from setting the hit definition time (HDT) on MISTRAS too long. The waveform below (Figure 5.11) has a HDT that is set properly. Notice that the HDT is the right length to define the first wave-form before the onset of the next waveform. If the HDT is set too long the second waveform will be considered a part of the first, i.e., two hits will be combined. As such, the multiple hit data will appear on the duration vs. counts plots in the upper left hand portion of the graph above the trend line. This occurs because the software records longer durations (D) than normal for a given number of counts (C), ruining the natural linear relationship between the two (D = kC).

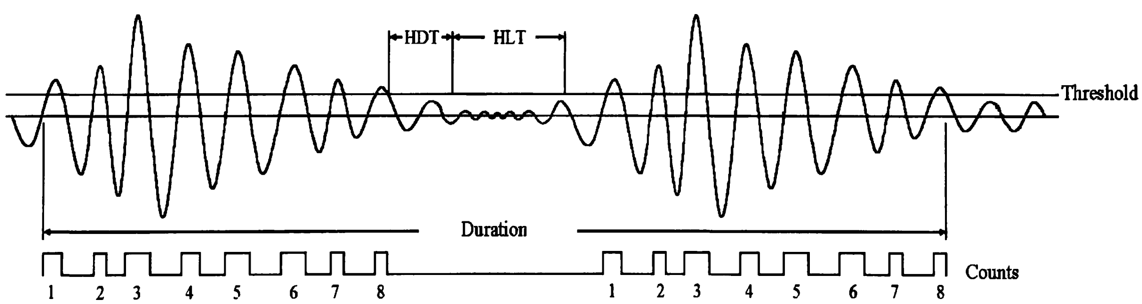


Figure 5.11 HDT/HLT schematic

When looking at the statistics related to each mechanism, the poor quality of the data for mixed mode fracture can also be seen. From the kurtosis [14] of the mixed mode fracture, which would be between 2 and -2 for normally distributed data, it can be seen that the counts and duration are far in excess of these values. Note also that the kurtosis values are higher in general for the mixed mode fracture statistics than either the plastic deformation or plane strain fracture modes.

Table 5.3 Statistics for fracture modes

Plastic Deformation		n = 18,158		
	Mean	Standard Deviation	Skewness	Kurtosis
Counts	5.01	6.76	2.58	7.49
Amplitude	35.40	5.17	1.24	1.03
Duration	53.99	73.37	2.25	5.50
Energy	0.07	0.32	4.99	29.50
Rise Time	12.07	17.85	3.11	16.57
Plane Strain Fracture		n = 834		
	Mean	Standard Deviation	Skewness	Kurtosis
Counts	58.91	31.18	1.30	1.66
Amplitude	59.46	5.16	1.02	0.81
Duration	637.31	368.61	1.28	1.34
Energy	7.67	8.36	3.46	17.80
Rise Time	54.34	85.39	1.30	24.72
Mixed Mode Fracture		n = 3,631		
	Mean	Standard Deviation	Skewness	Kurtosis
Counts	115.02	507.81	9.39	105.33
Amplitude	42.68	13.33	2.13	4.96
Duration	1852.08	6445.35	8.76	96.09
Energy	121.21	979.11	11.31	153.84
Rise Time	156.89	247.50	3.91	28.19

5.2 Back Propagation Network Test Results

Using a back propagation neural network it was possible to predict the cycles to failure with a worst case error of -9.3 percent. This network was trained on AE amplitude data up to fifty percent of the total cycles to failure. The training data are shown in Table 5.4 followed by the testing data in Table 5.5.

Table 5.4 Training data

Specimen #	Cycles To Failure	Amplitude Distribution Data (30 - 100 dB)
1	822.50	939 913 537 878 651 373 284 242 185 173 138 78 92 85 28 14 4 5 4 3 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2	1,063.50	1861 1019 630 1155 840 289 155 172 164 130 88 33 25 10 10 7 5 3 2 2 2 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3	582.00	1689 1035 545 1072 902 393 230 245 191 140 159 92 72 43 39 88 193 60 17 13 7 4 4 0 0 01 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4	588.25	2004 1104 564 1052 902 346 154 129 104 115 86 47 37 27 36 23 16 7 7 10 8 4 3 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5	495.00	429 52 11 12 6 3 2 0 1 3 0 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6	553.00	335 141 87 125 78 33 24 10 9 3 1 1 0 1 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7	456.50	1318 607 263 470 359 121 93 118 84 28 9 13 6 0 3 3 2 2 2 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8	765.50	1647 630 295 591 635 308 141 79 41 29 23 13 5 4 8 4 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9	499.25	861 338 161 325 370 167 68 64 13 18 35 27 16 30 29 20 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10	841.50	758 216 120 271 257 102 48 41 27 51 147 26 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11	472.75	424 110 41 69 46 20 5 4 2 2 5 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12	559.75	465 89 35 37 12 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13	545.75	688 387 205 375 286 114 89 143 102 37 10 6 6 4 5 2 6 6 2 3 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14	765.50	630 189 64 51 10 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15	719.25	600 183 86 161 122 27 4 1 0 2 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
16	516.50	786 372 185 277 285 149 78 96 93 55 7 5 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
17	376.25	253 311 132 241 265 161 145 189 156 96 102 19 14 4 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
18	454.00	103 194 130 247 168 75 55 98 86 46 41 44 26 40 32 8 6 2 1 0 2 1 1 3 2 0 2 1 1 0 1 1 0 1 0 0 0 1 1 1 0 2 2 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 5 0 0
19	310.75	129 159 85 140 127 63 55 74 56 61 74 62 40 24 20 10 9 7 8 10 3 4 6 6 2 3 1 4 2 2 3 0 0 0 1 3 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
20	422.25	43 60 17 42 33 18 20 11 8 18 20 9 14 10 7 8 3 9 3 6 5 4 6 4 4 0 5 4 4 1 0 1 1 1 2 1 1 1 1 0 3 2 0 0 2 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Table 5.5 Testing Data

Specimen #	Cycles To Failure	Amplitude Distribution Data (30 - 100 dB)
21	419.5	121 167 87 118 86 40 34 26 23 10 10 11 8 5 8 4 4 2 1 4 1 4 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22	492.0	121 207 100 136 42 6 10 6 3 8 10 7 6 5 3 3 4 1 4 3 5 2 5 1 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
23	452.0	91 81 42 67 33 33 24 34 31 20 23 15 10 12 16 7 6 7 9 8 4 8 1 7 5 1 1 3 1 1 1 0 1 1 1 1 2 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
24	423.0	64 46 30 48 40 17 13 12 18 9 20 13 13 15 7 5 10 4 6 3 3 6 6 1 2 3 2 1 3 1 3 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0

5.2.1 Back Propagation NeuralWorks Professional II/Plus Setup

Reference the "Tutorial for Optimizing a Back Propagation Neural Network Using NeuralWorks Professional II/Plus" [Bibliography] for detailed instructions on configuring the back propagation software. Detailed instructions on configuring the Back Propagation NeuralWorks Professional II/Plus software are provided in the Appendix C. Figure 5.12 shows the network settings for training the network to obtain the desired results.

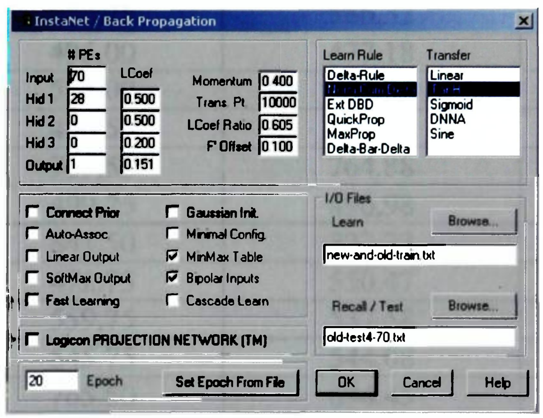


Figure 5.12 Back propagation network settings

5.2.2 Back Propagation Neural Network Results

The training set was compared to the predicted values after training. These values gave the following output (Table 5.6). Note that the worst case training error is 16.44 percent.

Table 5.6 Back propagation network training results

Specimen #	Actual Results (cycles)	Predicted Results (cycles)	% Error
1	822.50	821.32	-0.14
2	1063.50	1058.38	-0.48
3	582.00	583.88	0.32
4	588.25	586.32	-0.33
5	495.00	477.18	-3.60
6	553.00	554.65	0.30
7	456.50	463.94	1.63
8	765.50	764.88	-0.08
9	499.25	496.98	-0.45
10	841.50	837.73	-0.45
11	472.75	550.47	**16.44**
12	559.75	537.58	-3.96
13	545.75	544.08	-0.31
14	765.50	694.45	-9.28
15	719.25	747.68	3.95
16	516.50	507.79	-1.69
17	376.25	371.74	-1.20
18	454.00	452.85	-0.25
19	310.75	310.72	-0.01
20	422.25	421.71	-0.13

The above settings resulted in a network that when tested gave the following output (Table 5.7), where the worst case testing error was -9.30%.

Table 5.7 Back propagation network predicted results and percent errors

Specimen #	Actual Results (cycles)	Predicted Results (cycles)	% Error
21	419.50	380.50	**- 9.30**
22	492.00	474.65	- 3.53
23	452.00	482.21	+ 6.68
24	423.00	399.92	- 5.46

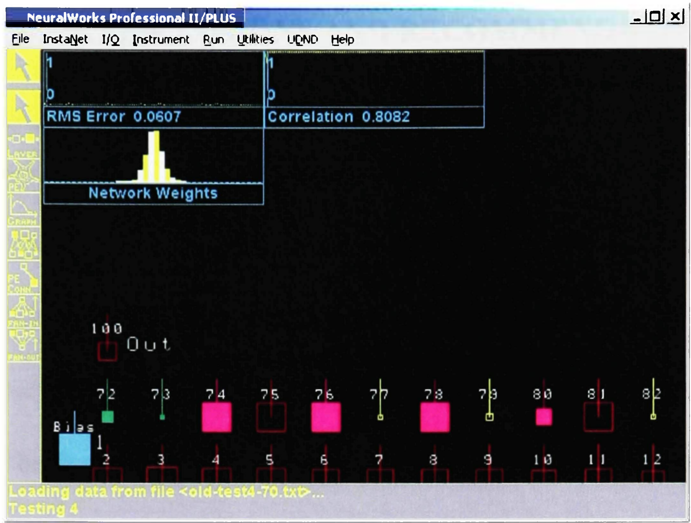


Figure 5.13 Screen shot of BPNN results showing network weights

Figure 5.13 shows the RMS error, correlation, and the final network weights assigned to the amplitude frequency distribution input. A closer look at the normal distribution of the weights shows that non-zero weights were assigned to the range of amplitudes (53 – 81), which corresponds to plane strain fracture. The amplitudes at the tail ends of the distribution had zero weights. This shows that the plane strain fracture mechanism is highly correlated with failure of the specimens, and thus is essential for prediction of cycles to failure. Figure 5.14 illustrates this correlation.

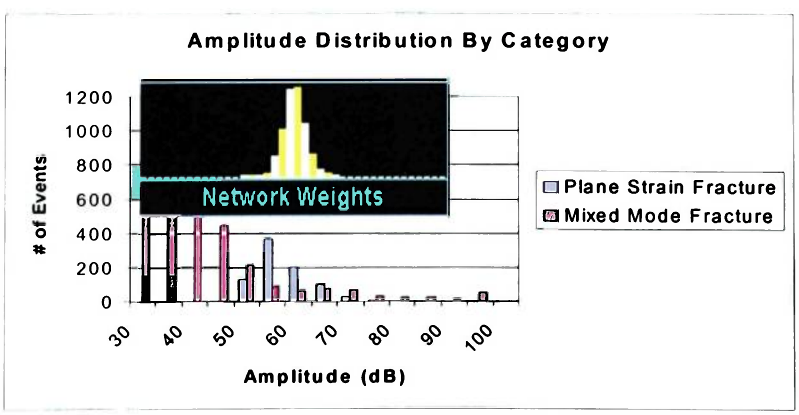


Figure 5.14 Network weights distribution by failure mechanism category

6.0 CONCLUSIONS AND RECOMMENDATIONS

6.1 SOM neural network

It was concluded that the SOM network was able to successfully sort acoustic emission data into the three expected failure mechanisms [15]: plastic deformation, plane strain fracture, and mixed mode (plane strain plus plane stress) fracture. It is recommended that additional samples be tested using a revised hit definition time (HDT) setting. This may help eliminate the multiple hit data that were collected. Furthermore, additional tests should be conducted using different cyclic loads to examine the network capability towards predicting fatigue life under different circumstances. Lower cyclic loads should be utilized to induce more fatigue crack emissions.

6.2 Back Propagation Neural Network

NeuralWorks Professional II/Plus software appears to have successfully created a back propagation neural network which has the ability to predict fatigue life to failure in these specimens. From the BPNN weights the plane strain fracture mechanism was found to correlate most highly with the fatigue life predictions. Thus, prediction might also have been accomplished with the plane strain fracture AE data alone if the data had not been so sparse. Sparseness of the data could be avoided in the future by reducing the cyclic load in order to promote more fatigue crack growth emissions. Prediction errors in either the training or testing results greater than 5 percent were not desirable. Minimizing the multiple hit data would probably provide the desired ± 5 percent prediction accuracy.

7.0 REFERENCES

1. Almeida, A.F., and Hill, E.v.K., "Neural Network Detection of Fatigue Crack Growth in Riveted Joints Using Acoustic Emission," ASNT 1994 Spring Conference, American Society for Nondestructive Testing, Columbus, OH, 1994, pp. 93-95.

2. Almeida, A.F., "Neural Network Detection of Fatigue Crack Growth in Riveted Joints Using Acoustic Emission," *Materials Evaluation*, Vol. 53, No. 1, 1995, pp. 76-82.

3. Thornton, W.P., and Hill, E.v.K., "Identification of Acoustic Emission Signals from Fatigue Crack Growth in Thin Aluminum Pressure Vessels," ASNT 1995 Spring Conference, American Society for Nondestructive Testing, Columbus, OH, 1995, pp. 112-114.

4. Marsden, M.L., and Hill, E.v.K., "Classification of Fatigue Cracks in Thin Aluminum Pressure Vessels Using Acoustic Emission Parameter Data," ASNT 1995 Fall Conference, American Society for Nondestructive Testing, Columbus, OH, 1995, pp. 42-44.

5. Marsden, M.L., and Hill, E.v.K., "Classification of Fatigue Cracking Data in a Simulated Aircraft Fuselage Using a Self Organizing Map," First International Conference on Nonlinear Problems in Aviation and Aerospace, S. Sivasundaram, Editor, Embry-Riddle Aeronautical University Press, Daytona Beach, FL, 1997, pp. 405-410.

6. Vaughn III, S.G., and Hill, E.v.K., "In-Flight Fatigue Crack Monitoring of an Aircraft Engine Cowling," ASNT Fall Conference and Quality Testing Show Paper Summaries, American Society for Nondestructive Testing, Columbus, OH, 1998, pp. 140-142.

7. Rovik, C.L., and Hill, E.v.K., "Classification of In-Flight Fatigue Cracks in Aircraft Structures using Acoustic Emission and Neural Networks," ASNT Fall Conference and Quality Testing Show Paper Summaries, American Society for Nondestructive Testing, Columbus, OH, 1998, pp. 143-145.

8. Rovik, C.L., Vaughn III, S.G., Demeski, R.J., and Hill, E.v.K., "Fatigue Crack Monitoring of Aircraft Structures Using Acoustic Emission," Second International Conference on Nonlinear Problems in Aviation and Aerospace, Vol. 2, S. Sivasundaram, Editor, European Conference Publications, Cambridge, 1999, pp. 657-664.

9. Ballard, D.L.W., and Hill, E.v.K., "Fatigue Life Prediction of Edge-Welded Metal Bellows using Neural Networks and Acoustic Emission Amplitude Data," ASNT

Fall Conference and Quality Testing Show Paper Summaries, American Society for Nondestructive Testing, Columbus, OH, 1998, pp. 146-149.

10. Ballard, D.L.W., and Hill, E.v.K., "Detection of Fatigue Crack Inception in Edge-Welded Metal Bellows Using Acoustic Emission," The 9th Asia-Pacific Conference on Nondestructive Testing in conjunction with ASNT's 1998 Spring Conference and 7th Annual Research Symposium, American Society for Nondestructive Testing, Columbus, OH, 1998, pp. 228-231.

11. Ballard, D.L.W., Allen, J.B., and Hill, E.v.K., "Acoustic Emission Detection of Fatigue Crack Growth in Edge-Welded Metal Bellows," Second International Conference on Nonlinear Problems in Aviation and Aerospace, Vol. 1, S. Sivasundaram, Editor, European Conference Publications, Cambridge, 1999, pp. 73-78.

12. Walker, J.L., and Hill, E.v.K., "An Introduction to Neural Networks: a Tutorial," First International Conference on Nonlinear Problems in Aviation and Aerospace, S. Sivasundaram, Editor, Embry-Riddle Aeronautical University Press, Daytona Beach, FL, 1997, pp. 667-672.

13. MIL-HDBK-5H, Metallic Materials and Elements For Aerospace Vehicle Structures, December 1, 1998, p. 384.

14. Walpole, R.E., Myers, R.H., and Myers, S.L., Probability and Statistics for Engineers and Scientists. Upper Saddle River, NJ: Prentice-Hall, Inc., 1998, pp 114-178.

15. Hartbower, C. E.; Gerberich, W. W.; and Liebowitz, H. "Investigation of Crack Growth Stress-Wave Relationships." *Engineering Fracture Mechanics*, vol. 1, no. 2, 1968, pp. 291-308.

8.0 BIBLIOGRAPHY

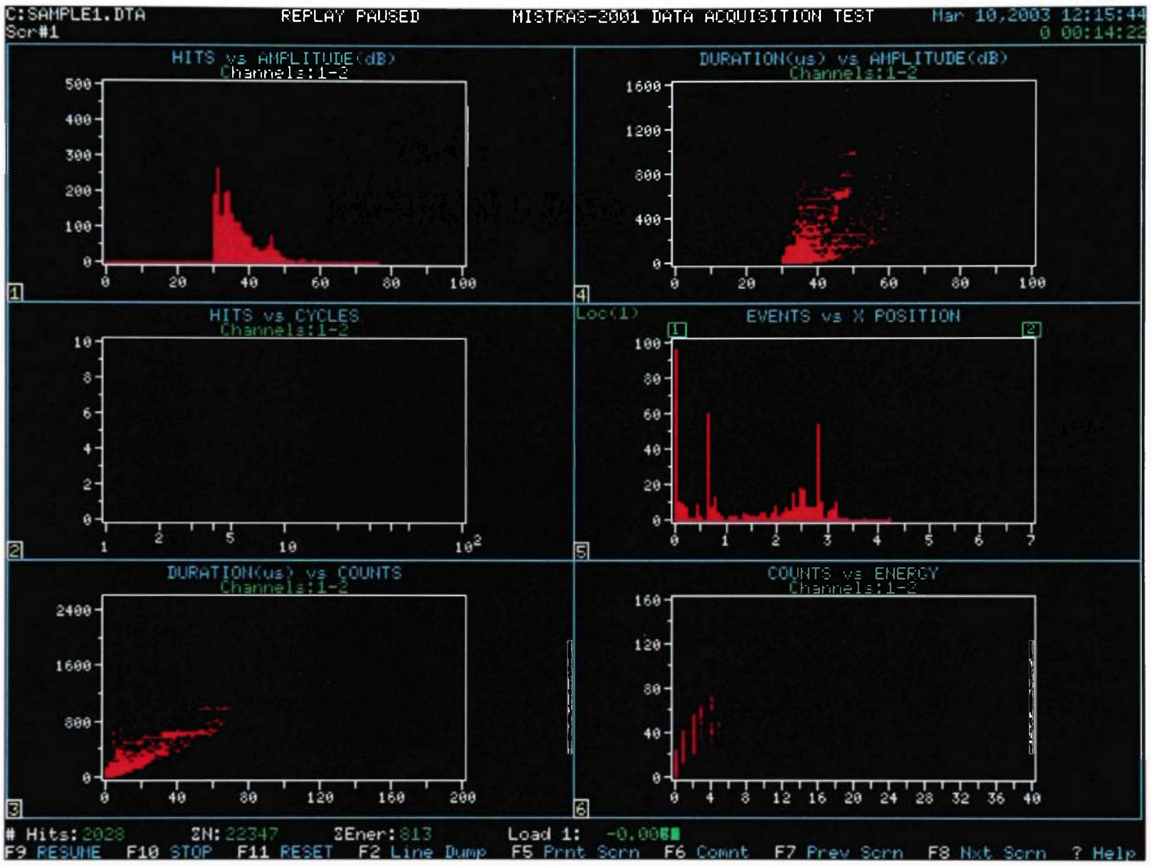
1. White, A., Keng, B.S., and Ibekwe, E.C., "Training and Testing A Self-Organizing Map Neural Network using NeuralWorks Professional II/PLUS," April 2003.

2. Osako, T., Liron, C., Parente, P., and Stubbe, M., "Tutorial for Optimizing a Back Propagation Neural Network Using NeuralWorks Professional II/Plus," April 2004.

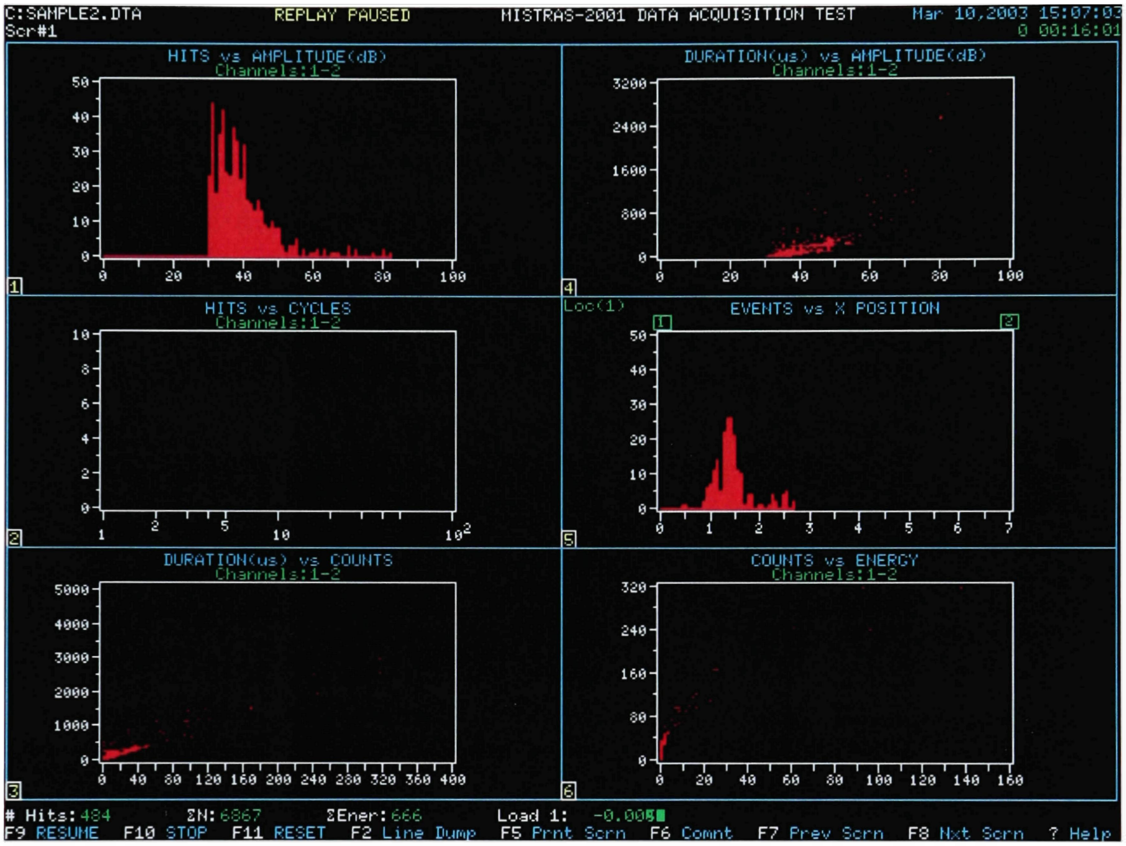
3. White, A., Ibekwe, E.C., "Classification of Fatigue Cracks in Aircraft Structures Using Acoustic Emission and Neural Networks," December 2003.

Appendix A – Data Plots

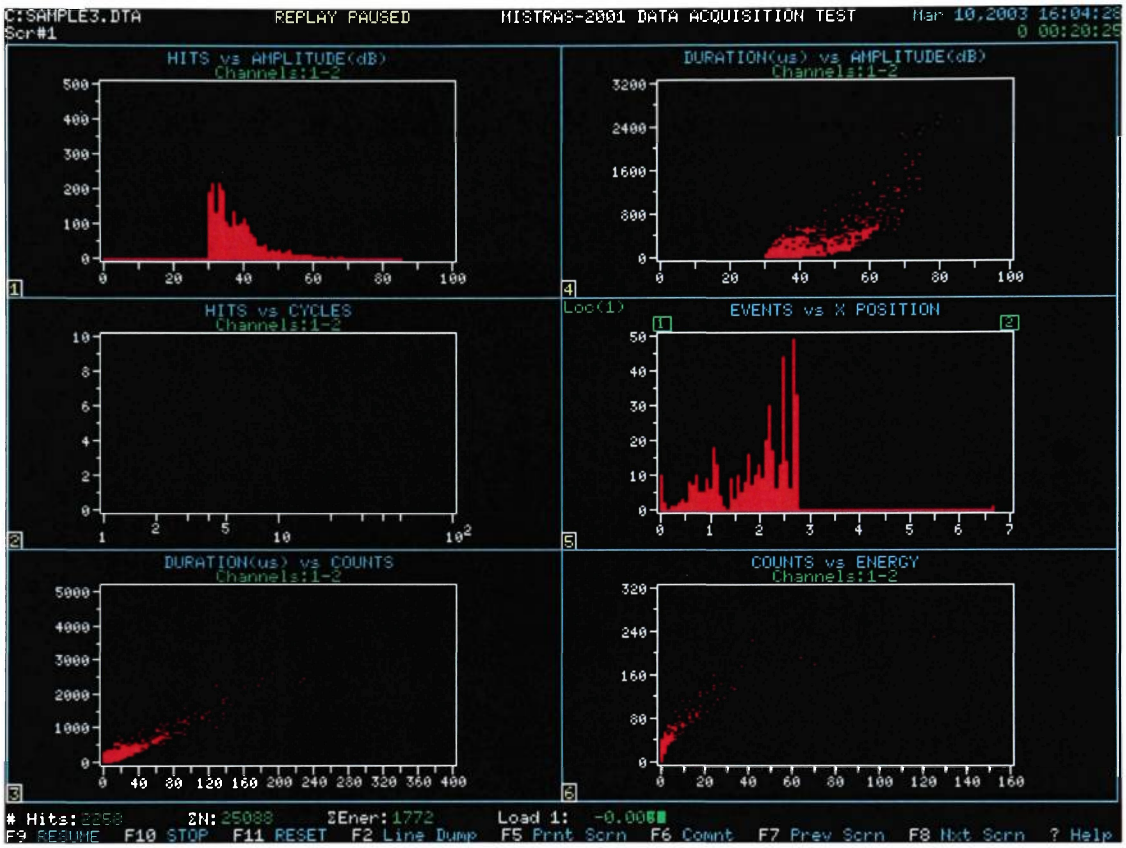
Sample 1 Data



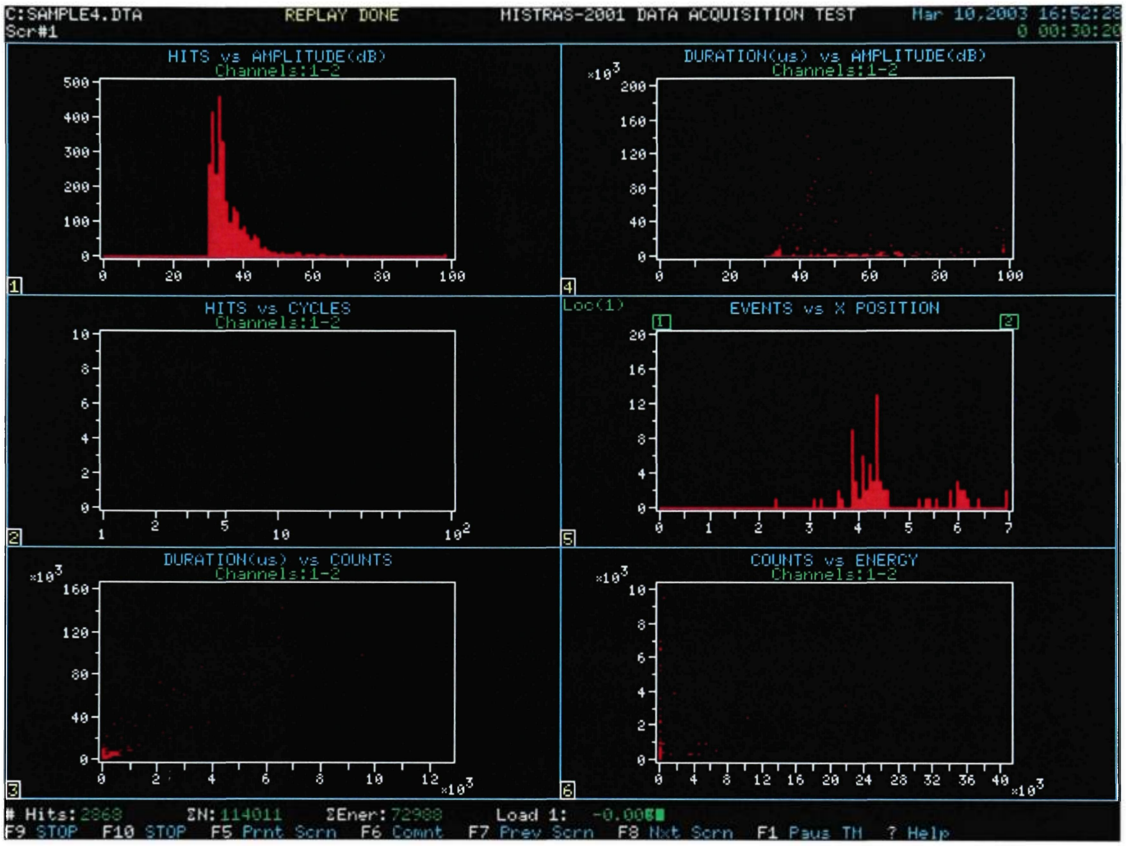
Sample 2 Data



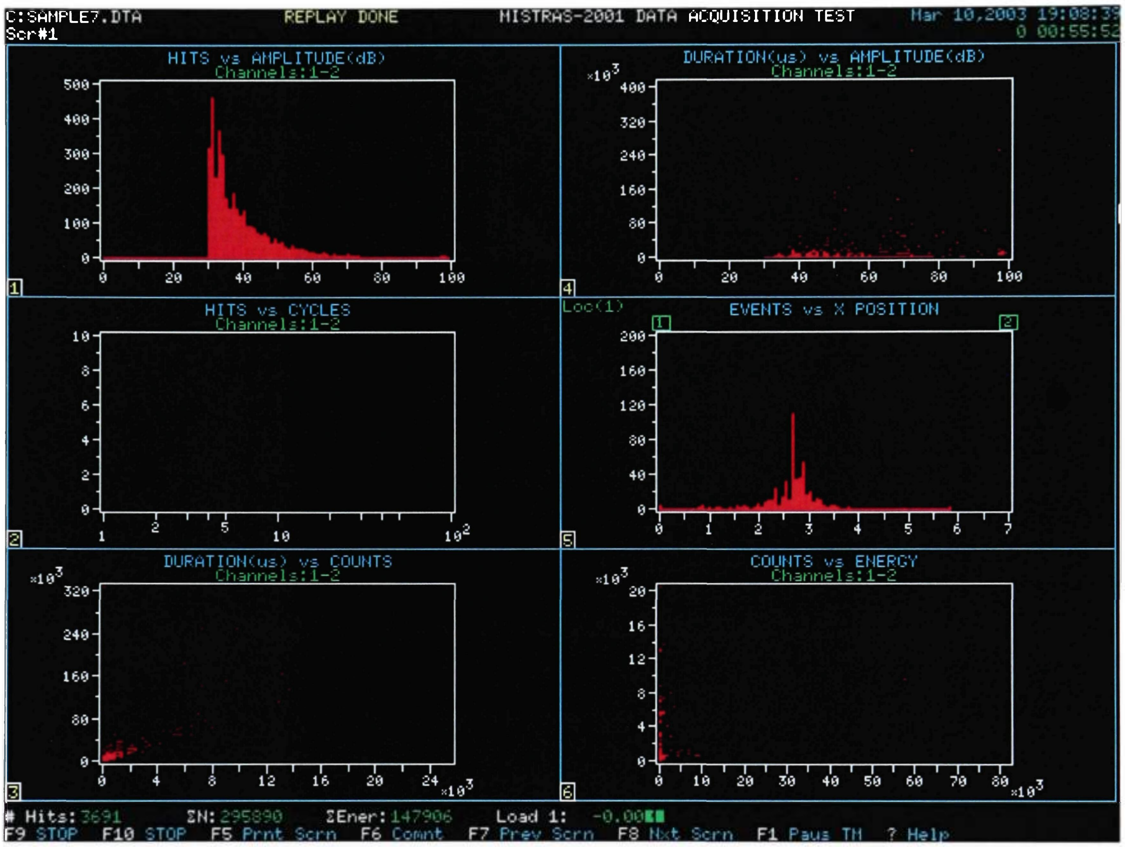
Sample 3 Data



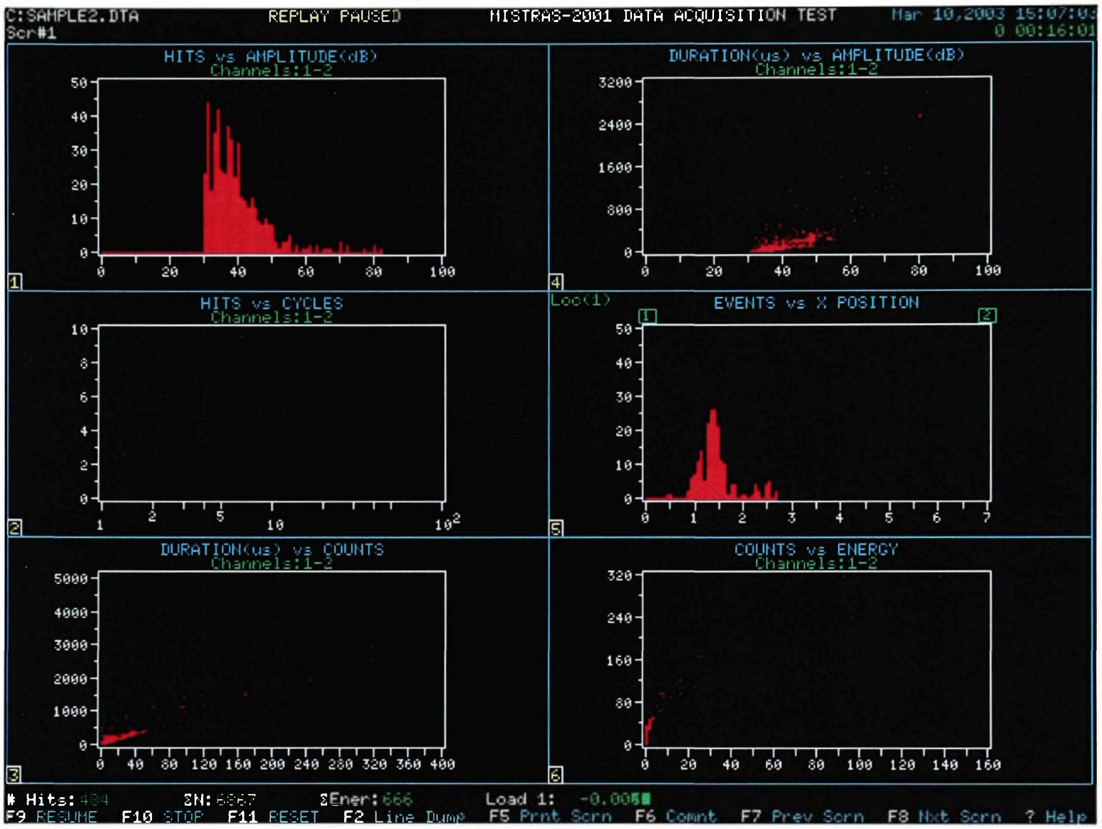
Sample 4 Data



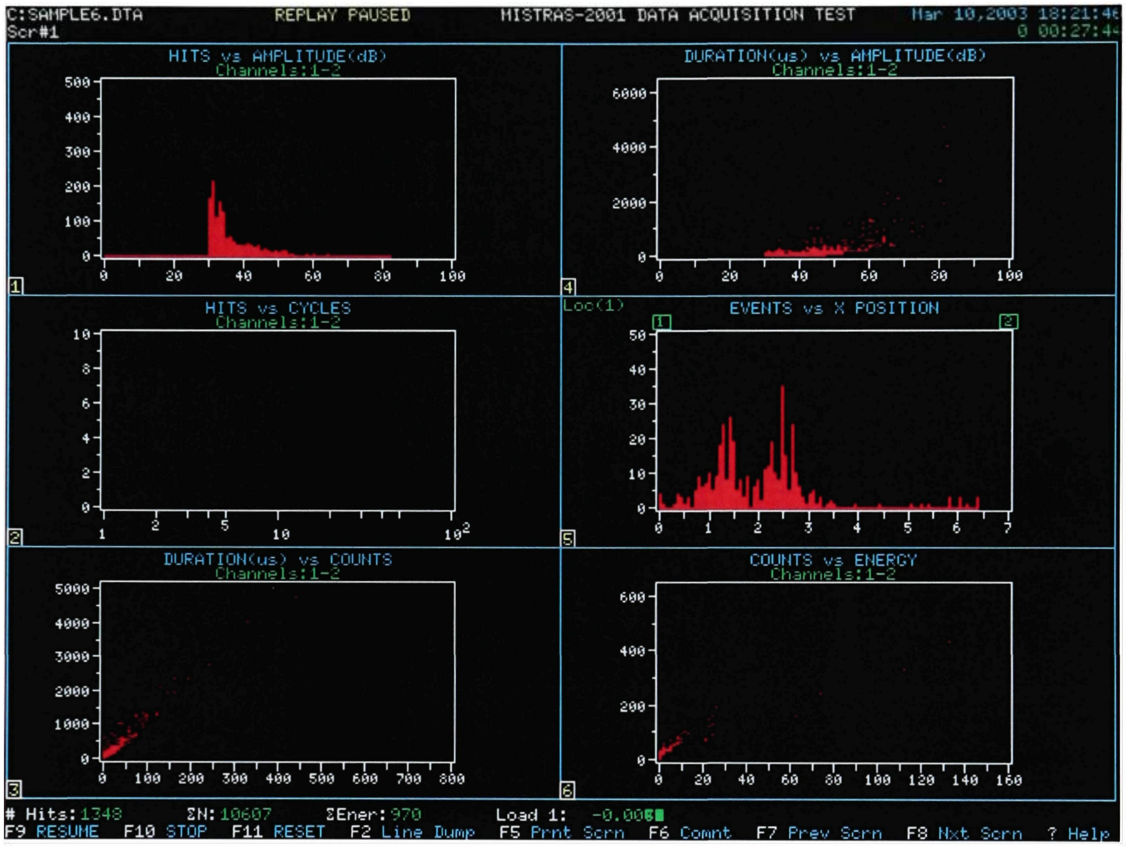
Sample 5 Data



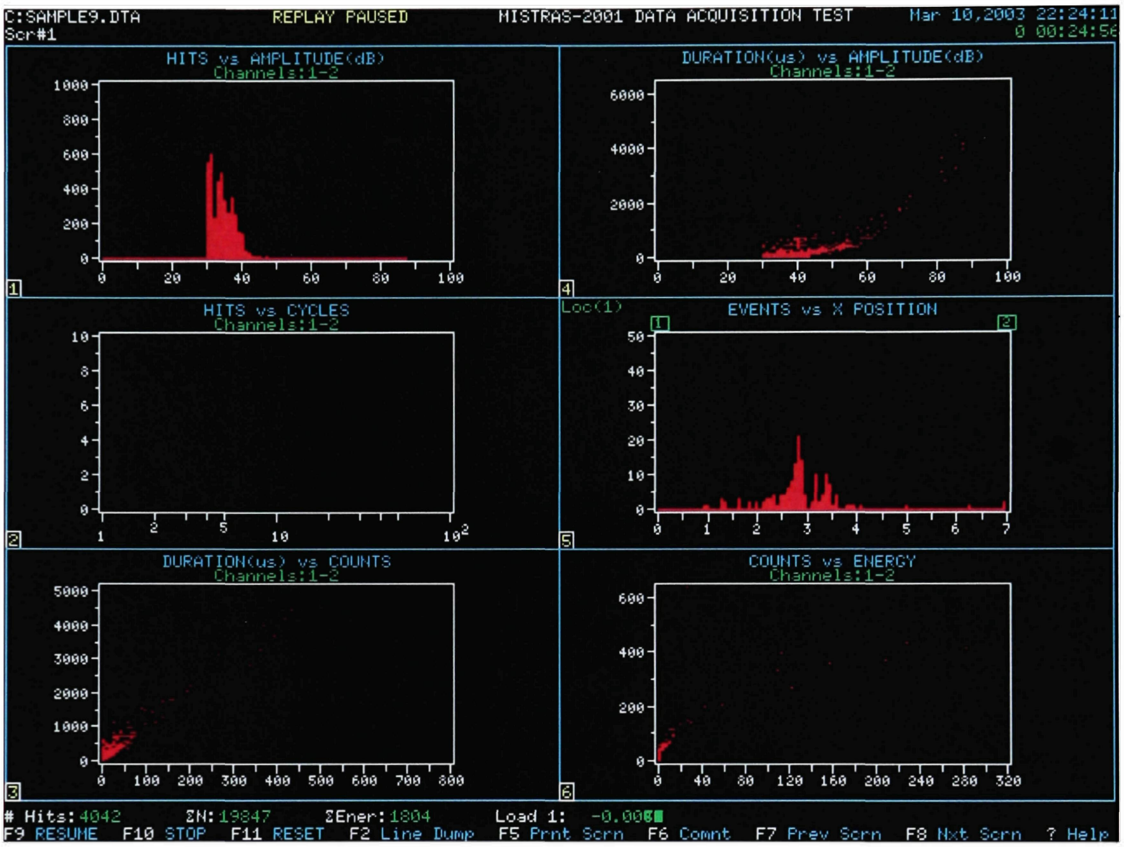
Sample 6 Data



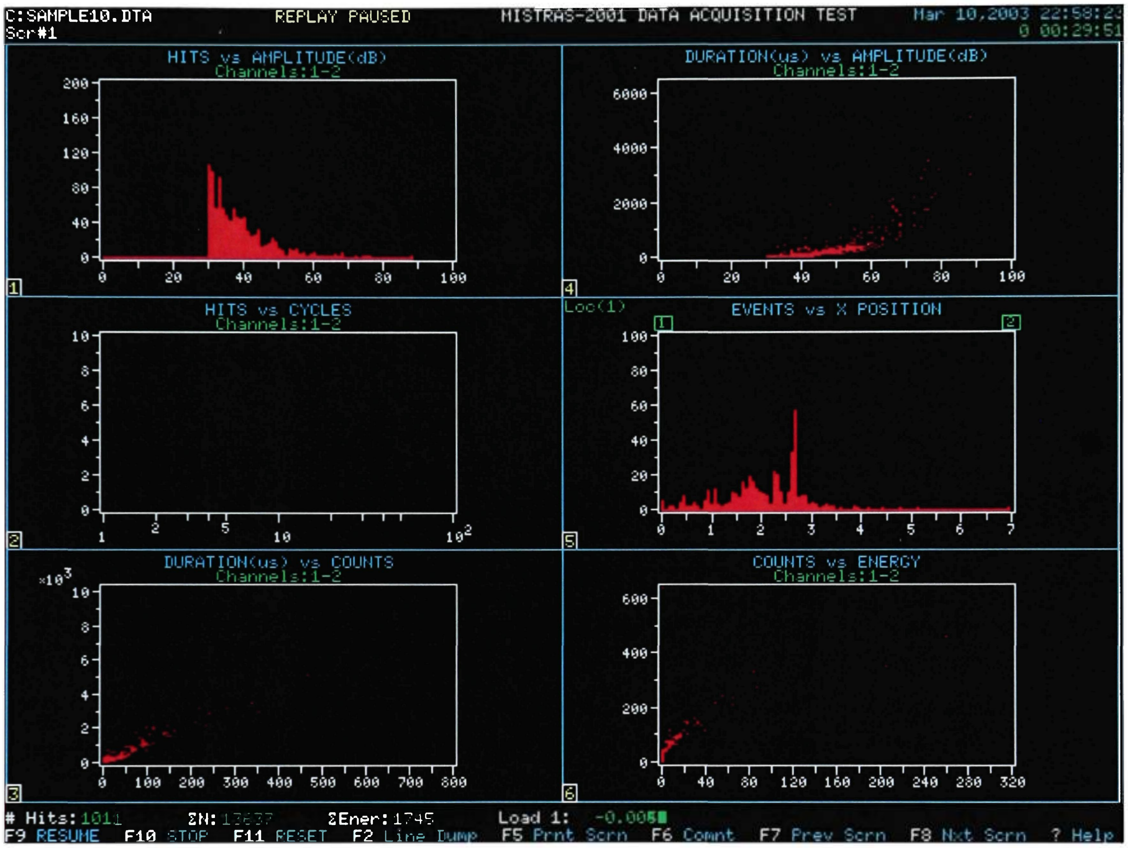
Sample 7 Data



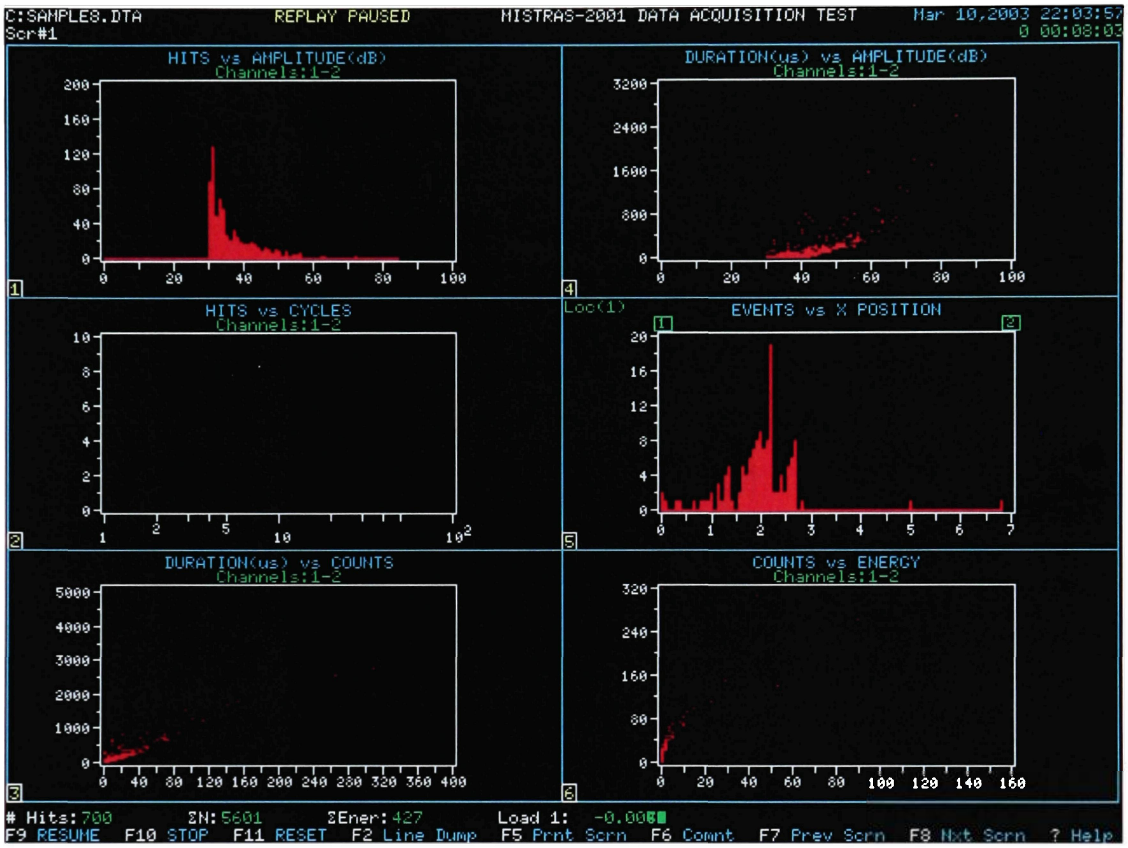
Sample 8 Data



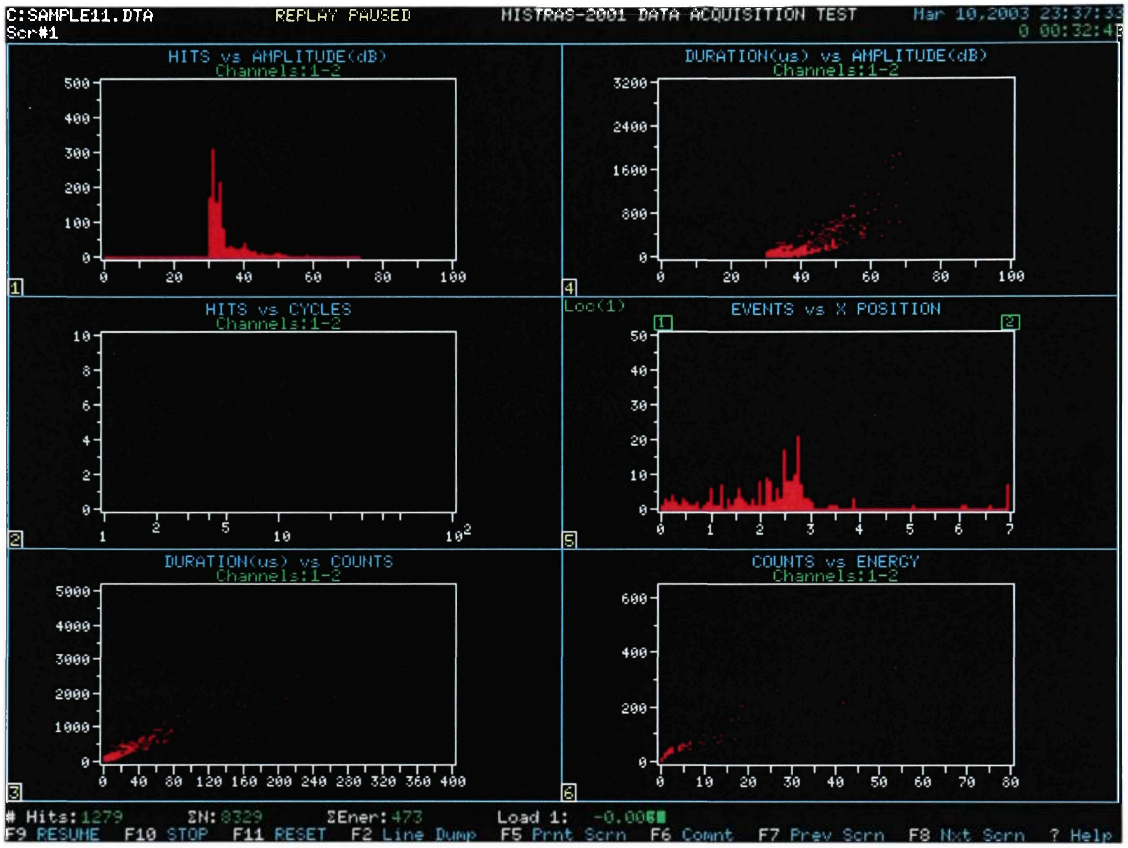
Sample 9 Data



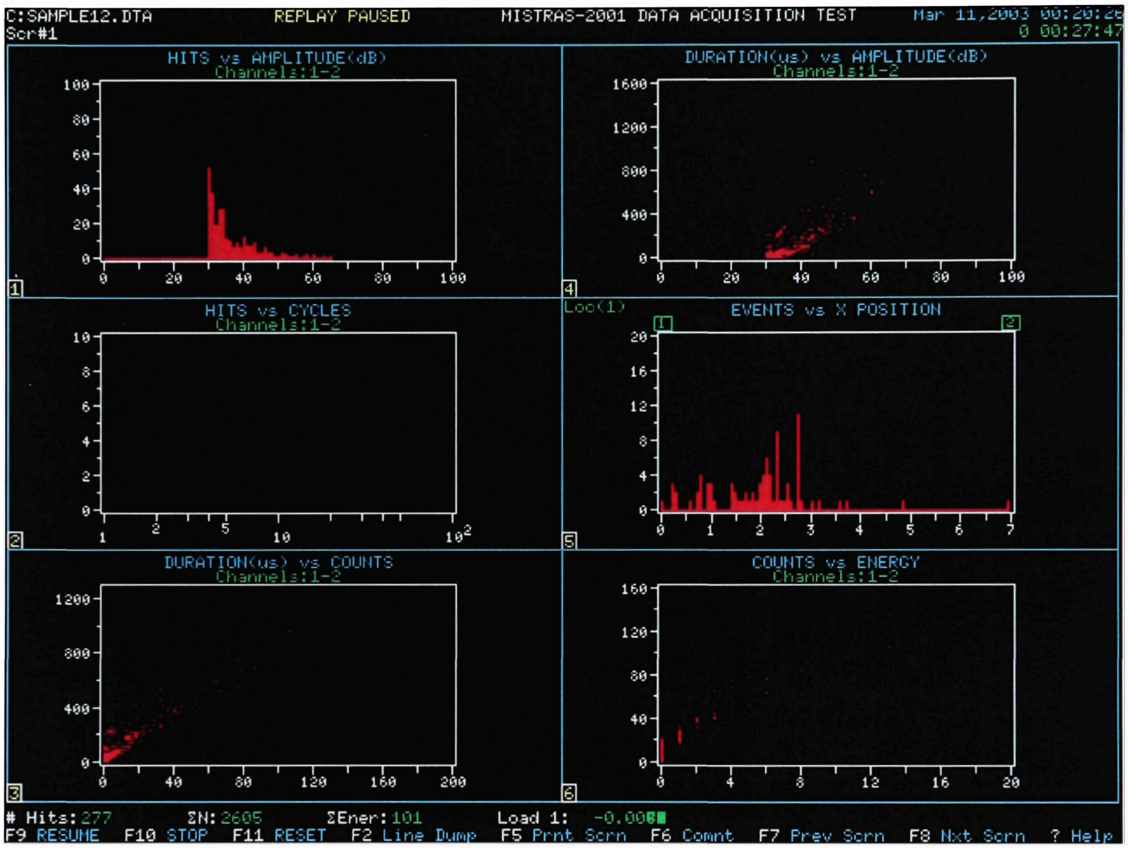
Sample 10 Data



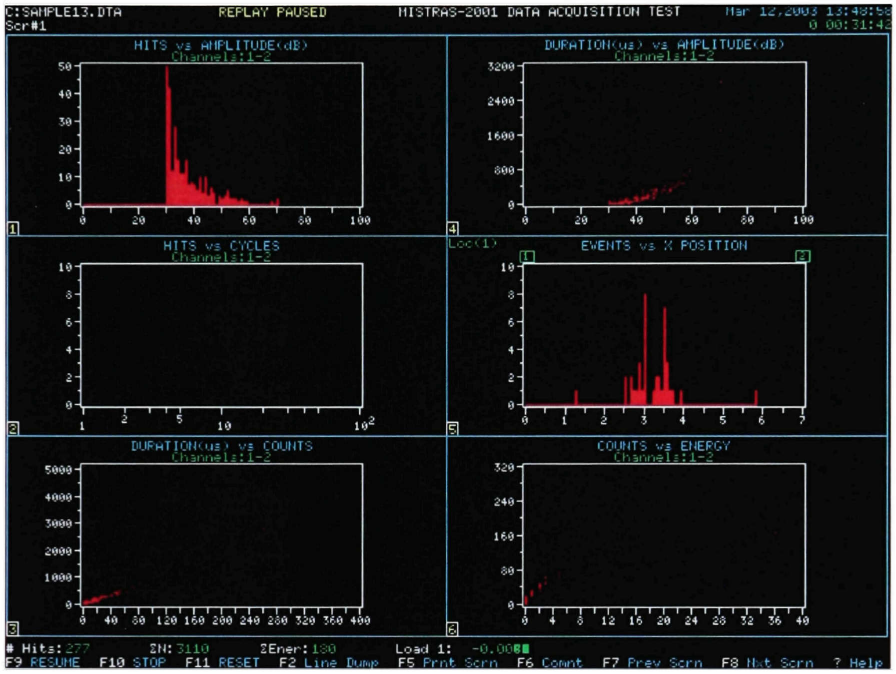
Sample 11 Data



Sample 12 Data



Sample 13 Data



Appendix B

Example of Kohonen Self-Organizing Neural Network by Hand Calculation

Objective

Determine new weights for self-organizing network with two inputs (X_1, X_2), and five cluster elements (D_1, D_2, D_3, D_4, D_5). Use a learning coefficient (LC) of .25 and a neighborhood factor of 1.

Network Schematic

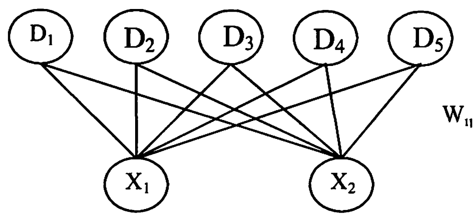


Figure B1 – Network Schematic

Initial Data

$$W_{ij} = \begin{vmatrix} 0.3 & 0.6 & 0.1 & 0.4 & 0.8 \\ 0.7 & 0.9 & 0.5 & 0.3 & 0.2 \end{vmatrix}$$

$$X_i = \begin{bmatrix} 0.5 & 0.2 \end{bmatrix}$$

Process

First compute Euclidean distances:

$$D_j = \sqrt{\left(W_{ij} - X_i\right)^2}$$

Next, determine the minimum Euclidean distance and update the surrounding weights within the neighborhood factor using:

$$W_{ij(new)} = W_{ij(old)} + LC \times \left(X_i - W_{ij(old)}\right)$$

For example, if D_2 is the minimum Euclidean distance, update W_{11}, W_{21}, W_{12}, W_{22}, W_{13}, & W_{23}.

Repeat process iteratively until the weight column being revised has reach the value of the input vector to within a desired error interval.

Summary of Results

Table B1 lists the iterative weight values for the given initial data set. Iterations were stopped upon reaching the input to within .001 (19 iterations). Iteration 20 is shown for reference.

Iteration:	0					**Iteration:**	5				
Wij:	0.3	0.6	0.1	0.4	0.8	**Wij:**	0.3	0.6	0.369	0.467	0.598
	0.7	0.9	0.5	0.3	0.2		0.7	0.9	0.298	0.233	0.200
Iteration:	1					**Iteration:**	10				
Wij:	0.3	0.6	0.180	0.420	0.740	**Wij:**	0.3	0.6	0.457	0.489	0.532
	0.7	0.9	0.440	0.280	0.200		0.7	0.9	0.232	0.211	0.200
Iteration:	2					**Iteration:**	15				
Wij:	0.3	0.6	0.244	0.436	0.692	**Wij:**	0.3	0.6	0.486	0.496	0.511
	0.7	0.9	0.392	0.264	0.200		0.7	0.9	0.211	0.204	0.200
Iteration:	3					**Iteration:**	19				
Wij:	0.3	0.6	0.295	0.449	0.654	**Wij:**	0.3	0.6	0.494	0.499	0.504
	0.7	0.9	0.354	0.251	0.200		0.7	0.9	0.204	0.201	0.200
Iteration:	4					**Iteration:**	20				
Wij:	0.3	0.6	0.336	0.459	0.623	**Wij:**	0.3	0.6	0.495	0.499	0.503
	0.7	0.9	0.323	0.241	0.200		0.7	0.9	0.203	0.201	0.200

Table B1 – Summary of Iterative Solutions

Introduction to Kohonen Self-Organizing Neural Network Software

A Self Organizing Map (SOM) Neural Network is a neural network that is capable of categorizing a set of input data into different categories. A SOM architecture consists of two layers:

- Input Layer – the number of nodes for the input layer depends on the number of inputs that are going to be used in the network each time.
- Output Layer – the number of nodes for the output layer depends on the number of different categories that is desired.

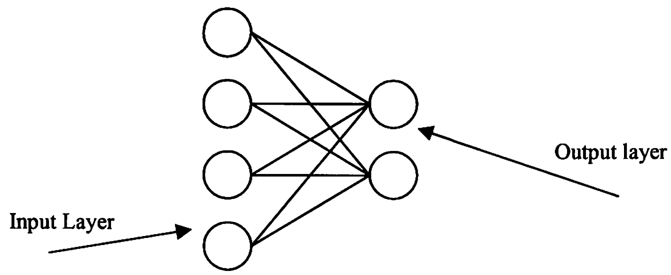


Figure B1.1 – Network Schematic

As in Figure B1.1, the SOM has 4 nodes in the input layer and 2 nodes in the output layer. All of the nodes are connected to each other, and each connection is assigned a weight. Initially these weights are assigned random values, and during the learning process the weights are updated. After the network is trained, the network will be able to map a given set of inputs to one of the categories.

In this tutorial, two examples are used to demonstrate how to train and test a SOM.

Objective

The objective of this tutorial is to:

- Demonstrate the capability of the SOM.
- Familiarize the reader with the software of NeuralWorks Professional II/PLUS through the training and testing of a SOM.

Data Preparation

This section discusses how to prepare the data that will be used by the software. If the user is familiar with the process of preparing the data, this section can be skipped and the user may move on to the next section.

The following are step by step instructions on how to prepare the data:

1. Start Microsoft Excel.
2. Enter the following data into 2 separate columns:

1	1
5	5
9	9

3. Then select ***>File>Save As…*** under the ***File*** pull down menu. The following window will appear:

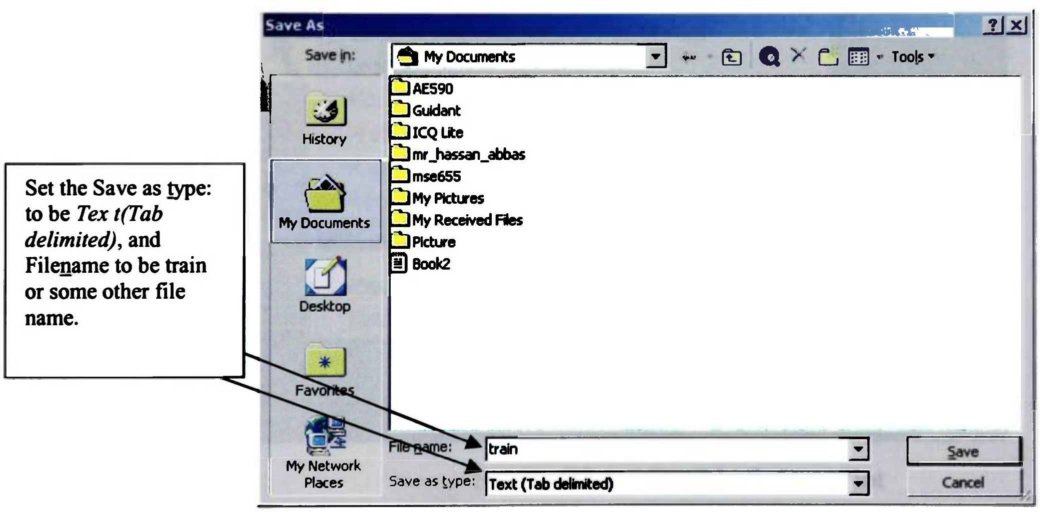


Figure B2.1 – Sample Screen plot

4. After saving the file, the training data that is going to be used to train the network is prepared. If a different set of data is to be used for testing, repeat steps 2 and 3 with the testing data using a different filename.

Note: In this tutorial, both the training and test data are going to be the same, so no other test data needs to be prepared in this case.

Program Usage and Data Analysis

In this section, two examples are used. The first example is simple and is used to inform the user as to how to construct and train a SOM. The second example is more practical and shows how the SOM may be applied in real life situations.

SOM Network Construction

A set of training data has already been saved in a file named *train.txt*. The following is a step by step process on how to create a SOM:

1. Start the NeuralWare Professional II/Plus (NWP2+) software.
2. Select ***>InstaNet>Self Organizing Map…*** as shown in the Figure B3.1.

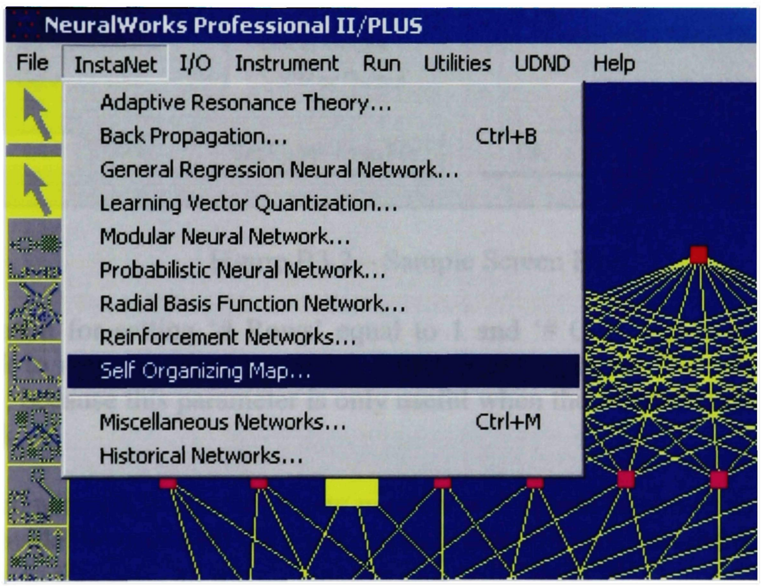


Figure B3.1 – Sample Screen Plot

3. A new window will appear as in Figure B3.2. Enter the data as shown and then click on the OK button.

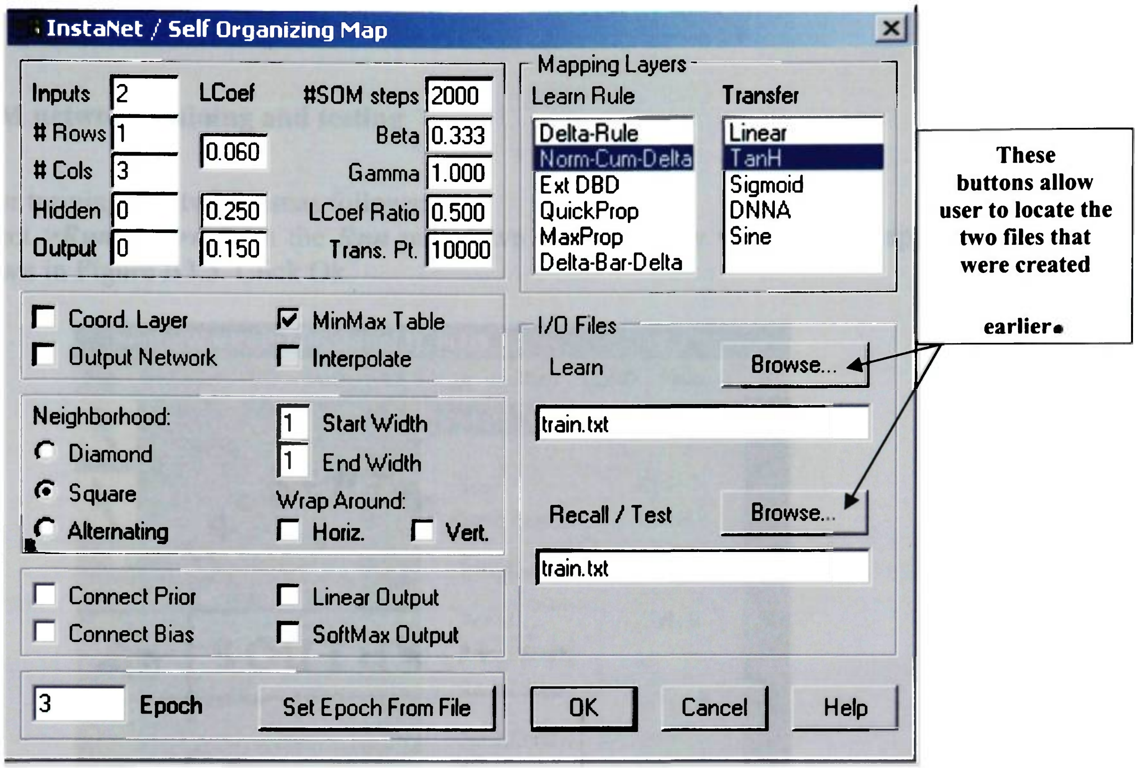


Figure B3.2 – Sample Screen Plot

The reason for setting '# Rows' equal to 1 and '# Cols' equal to 3 is to train the network to map the input into a 1 x 3 matrix output layer. The value of "Output" is set to 0, because this parameter is only useful when the SOM is linked to other neural network.

4. After the network is created, a new window will appear as shown in Figure B3.3 and will provide the user with the choice of data that will be displayed during the training process. In this case, all the options are selected. Click OK.

Figure B3.3 – Sample Screen Plot

SOM network training and testing

The steps for training a network areas follows:

1. Select ***>Run>Learn*** from the ***Run*** pull down menu. A new window will appear as shown in Figure B3.5. Click Ok.

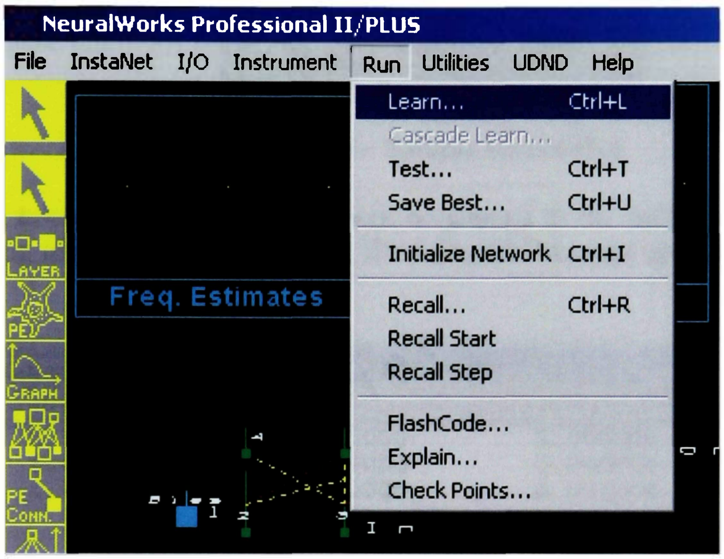


Figure B3.4 – Sample Screen Plot

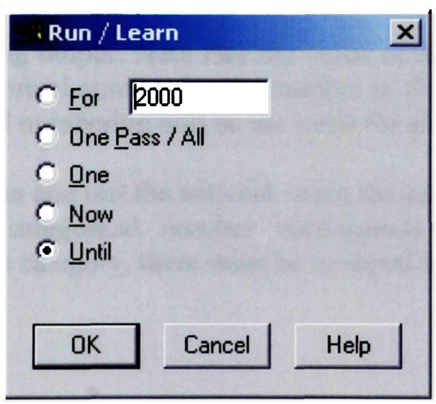


Figure B3.5 – Sample Screen Plot

Note: Based on experience, a larger number will train the network more closely.

2. The network will complete the training in seconds. Then select ***>Run>Test*** from the ***Run*** pull down menu. A new window will appear (see Figure B3.6). Click OK.

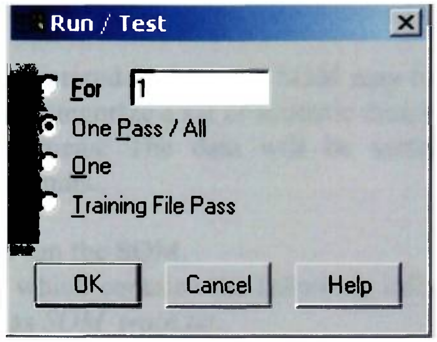


Figure B3.6 – Sample Screen Plot

3. After the testing a file *train_txt.nnr* will be created in the folder where the *train.txt* (the testing file) is located. Open that file using the Notepad application or other text editor.

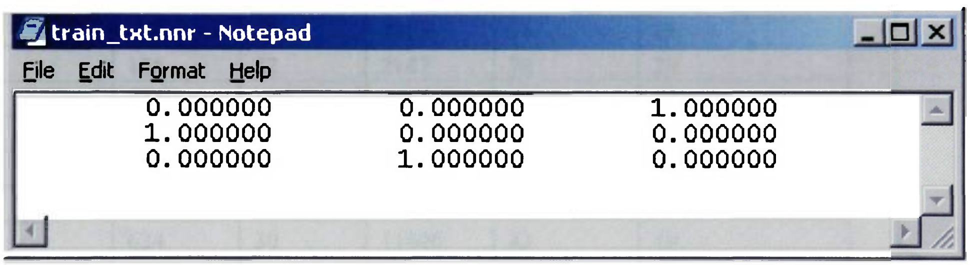


Figure B3.7 – Sample Screen plot

Figure B3.7 shows the training output. Note that the value of the first row in the output file consists of the first categorical number for the number in the first row of the training file (1 and 1). This categorical numbering will be the same for all future test files.

It is always a good idea to train and test the network using the same set of data so the user will have an idea which categorical number corresponds to which input. Most importantly, for each different category, there must be an equal number of input data used when training the network.

Application of SOM

In this section, the user will be familiarized on how the SOM may be applied to a real-world situation. A SOM will be created to categorize a set of acoustic emission data that is collected from cycling aluminum test specimens. The data will be sorted into rubbing, plastic deformation, and fatigue cracking signals.

The following steps will be used to run the SOM.

1. First, create a training file which contains the following information. Do not include the header section. Save it as *SOM_train.txt*.

Rise Time	Counts	Energy	Duration	Amplitude	Counts to Peak	
41	223	44	1751	72	13	
6	202	89	2918	75	1	
118	210	94	2284	71	22	
324	300	176	2886	77	32	Plane Strain Fracture
212	256	168	2778	80	23	
323	171	76	2175	73	23	
120	186	82	2147	70	20	
382	234	127	2996	75	37	
228	1341	97	20727	71	27	
42	799	37	12651	47	3	
126	909	43	16026	44	8	
221	624	30	11446	43	10	Mixed Mode Fracture
16	949	47	17037	50	1	
8	811	41	13497	55	1	
380	804	39	14795	45	9	
61	643	33	13144	45	7	
56	12	0	390	37	5	
37	16	0	191	39	4	
1	1	0	1	30	1	
161	1	0	162	31	1	Plastic deformation
99	1	0	100	30	1	
115	3	0	116	34	2	
6	1	0	7	32	1	
85	10	0	518	36	6	

2. Then create a SOM using the same settings as shown in Figure B3.8 and then proceed to train the network.

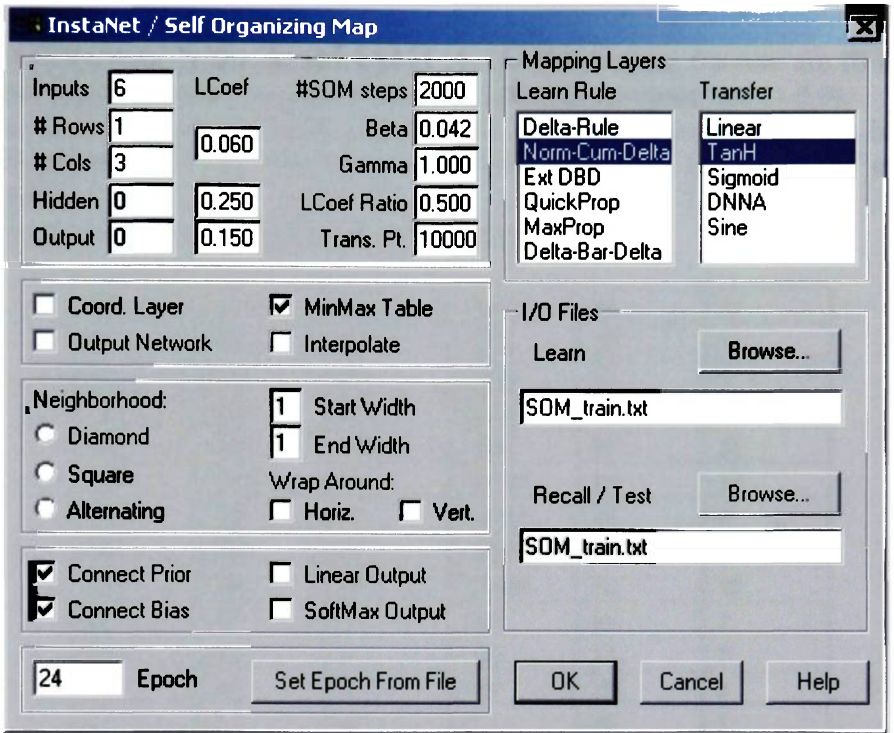
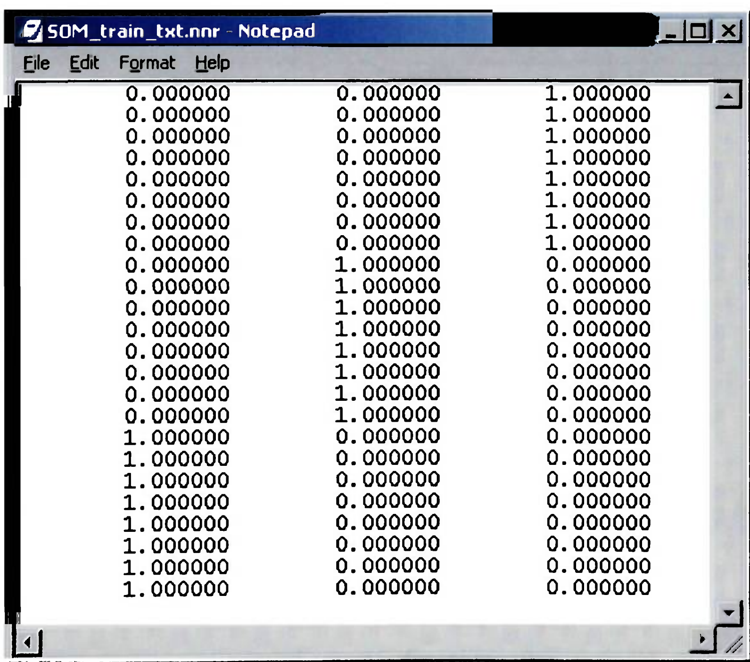


Figure B3.8 – Sample Screen Plot

3. After the initial training, the same training file will be used to test the network to determine the categorical numbers that are assigned by the network.



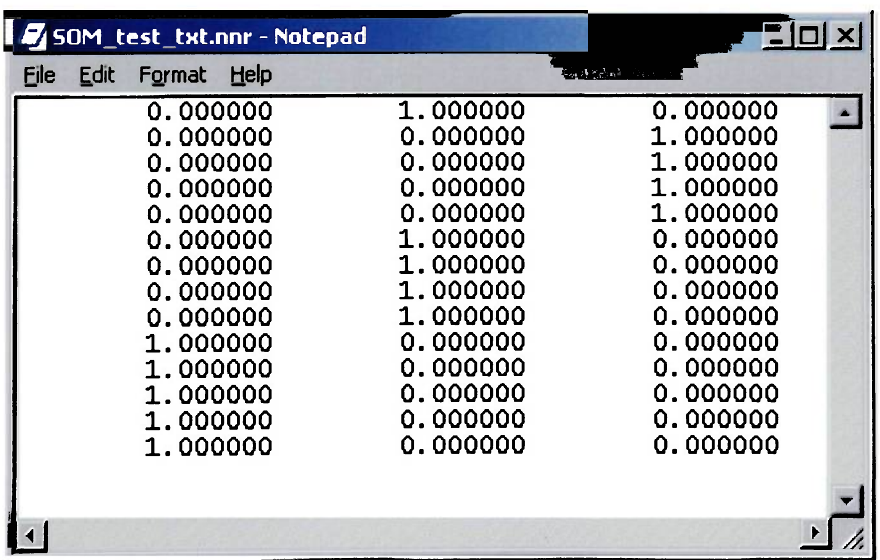
0.000000	0.000000	1.000000
0.000000	0.000000	1.000000
0.000000	0.000000	1.000000
0.000000	0.000000	1.000000
0.000000	0.000000	1.000000
0.000000	0.000000	1.000000
0.000000	0.000000	1.000000
0.000000	0.000000	1.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000

Figure B3.9 – Sample Text File

4. The output file shows the categories for the plane strain fracture are (0 0 1), mixed mode fracture are (0 1 0) and the plastic deformation signals are (1 0 0).
5. Now, a test file may be used on the trained network to show the network is capable of categorizing the data into three predefined categories based on the training file created.
   The test file will contain the following information:

228	1341	97	20727	71	27
41	223	44	1751	72	13
6	202	89	2918	75	1
118	210	94	2284	71	22
324	300	176	2886	77	32
228	1341	97	20727	71	27
42	799	37	12651	47	3
126	909	43	16026	44	8
221	624	30	11446	43	10
59	29	1	502	41	3
56	12	0	390	37	5
37	16	0	191	39	4
1	1	0	1	30	1
161	1	0	162	31	1

After the testing, the output of the testing file should like this:


SOM_test_txt.nnr - Notepad
File Edit Format Help

0.000000	1.000000	0.000000
0.000000	0.000000	1.000000
0.000000	0.000000	1.000000
0.000000	0.000000	1.000000
0.000000	0.000000	1.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000
1.000000	0.000000	0.000000

Figure B3.10

The first data is considered as mixed mode fracture, the second through fifth data are consider as plane S Introduction to Kohonen Self-Organizing Neural Network by Hand strain fracture and so on.

Conclusions

- A SOM is a powerful technique that can be used to categorize data that does not have an obvious pattern.

- The network must be tested with the same training data set to determine the categorization scheme used by the network.

- The training data needs to have the same amount of data for each category.

Appendix C

Example of Back Propagation Neural Network by Hand Calculation

Objective

Determine new weights for back propagation network with two inputs (X1, X2), two middle layer processing elements (PEs) (Y1, Y2), and a single output (Z1). Use a learning coefficient (LC) of .25 and a sigmoid activation function.

Network Schematic

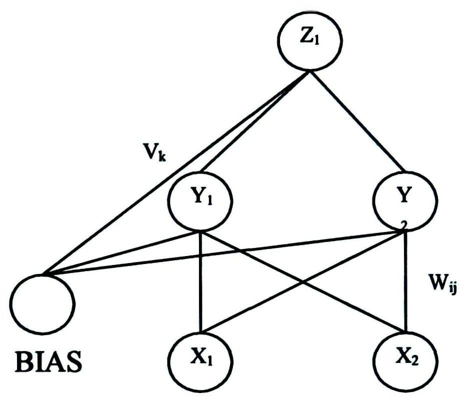


Figure C1 – Network Schematic

Initial Data

$$W_{ij} = \left| \begin{matrix} 0.7 & -0.4 & \vert 0.4 \\ -0.2 & 0.3 & \vert 0.6 \end{matrix} \right| \qquad V_k = | \, 0.5 \quad 0.1 \, | \; -0.3 |$$

Process

First compute middle layer output:

$$y_j = \Sigma\, W_{ij} * x_I$$
$$Y_{(out)} = f(y_j) = 1/(1+e^{-y_j})$$

Next compute the output and associated error:

$$Z_1 = V_{11}Y_1 + V_{12}Y_2 + V_{1B}$$

$\delta_k = \delta Z_1 = (T_k - Z_k)f'(Z_1) = (T_1 - Z_1) * f(z_1) * (1\text{-}f(z_1))$

Update middle to output layer weights:

$\Delta W_{jk} = LC * \delta_k * Y_j$

Compute middle layer error:

$\delta_j = \delta_k * V_{jk} * f'(Y_j)$

Update input to middle layer weights:

$\Delta W_{ij} = LC * \delta_i * X_j$

Repeat process iteratively until required error value is reached (reference hand calculations).

Summary of Results

Table 1 lists the iterative weight values for the given initial data set. Iterations were stopped upon reaching 10% error.

Iteration:	**0**			**Iteration:**	**10**		
Wij:	0 7	-0 4	0 4	**Wij:**	0 7	-0 3632	0 4368
	-0 2	0 3	0 6		-0 2	0 3100	0 6100
Vk:	0 5	0 1	-0 3	**Vk:**	0 646	0 288	-0 036
Iteration:	**1**			**Iteration:**	**25**		
Wij:	0 7	-0 3962	0 4038	**Wij:**	0 7	-0 3182	0 4818
	-0 2	0 3006	0 6006		-0 2	0 3293	0 6293
Vk:	0 517	0 121	-0 27	**Vk:**	0 791	0 472	0 220
Iteration:	**2**			**Iteration:**	**50**		
Wij:	0 7	-0 3924	0 4076	**Wij:**	0 7	-0 2677	0 5323
	-0 2	0 3014	0 6014		-0 2	0 3559	0 6559
Vk:	0 531	0 142	-0 24	**Vk:**	0 929	0 648	0 461
Iteration:	**3**			**Iteration:**	**75**		
Wij:	0 7	-0 3889	0 4111	**Wij:**	0 7	-0 2342	0 5658
	-0 2	0 3022	0 6022		-0 2	0 3752	0 6752
Vk:	0 546	0 162	-0 212	**Vk:**	1 013	0 753	0 604
Iteration:	**4**			**Iteration:**	**100**		
Wij:	0 7	-0 3851	0 4149	**Wij:**	0 7	-0 2094	0 5906
	-0 2	0 3031	0 6031		-0 2	0 3900	0 6900
Vk:	0 561	0 182	-0 184	**Vk:**	1 072	0 826	0 702
Iteration:	**5**			**Iteration:**	**133**	(solved)	
Wij:	0 7	-0 3813	0 4187	**Wij:**	0 7	-0 1845	0 6155
	-0 2	0 3041	0 6041		-0 2	0 4052	0 7052
Vk:	0 576	0 202	-0 156	**Vk:**	1 128	0 897	0 797

Table C1 – Summary of Iterative Solutions (1st Data Set)

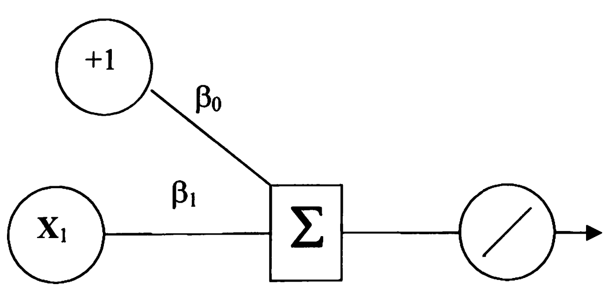
Table C2 lists the iterative weight values for a second data set: $X_1 = 3.0$, $X_2 = 4.0$ & $Z_1 = 1.0$.

Iteration:	**0**			**Iteration:**	**10**		
Wij:	0 7	-0 4	0 4	**Wij:**	0 7493	-0 3343	0 4164
	-0 2	0 3	0 6		-0 1746	0 3339	0 6085
Vk:	0 5	0 1	-0 3	**Vk:**	0 699	0 257	-0 068
Iteration:	**1**			**Iteration:**	**25**		
Wij:	0 7055	-0 3927	0 4018	**Wij:**	0 7828	-0 2897	0 4276
	-0 1981	0 3026	0 6006		-0 149	0 3681	0 6170
Vk:	0 524	0 1181	-0 272	**Vk:**	0 830	0 366	0 079
Iteration:	**2**			**Iteration:**	**50**		
Wij:	0 7109	-0 3854	0 4036	**Wij:**	0 8325	-0 2233	0 4442
	-0 1959	0 3055	0 6014		-0 1021	0 4306	0 6326
Vk:	0 547	0 1357	-0 245	**Vk:**	1 035	0 548	0 304
Iteration:	**3**			**Iteration:**	**75**		
Wij:	0 7163	-0 3783	0 4054	**Wij:**	0 8527	-0 1964	0 4509
	-0 1936	0 3086	0 6021		-0 0809	0 4588	0 6397
Vk:	0 569	0 1528	-0 219	**Vk:**	1 126	0 632	0 402
Iteration:	**4**			**Iteration:**	**94**	(solved)	
Wij:	0 7215	-0 3714	0 4072	**Wij:**	0 8631	-0 1825	0 4544
	-0 191	0 3119	0 603		-0 0696	0 4738	0 6435
Vk:	0 590	0 1693	-0 194	**Vk:**	1 175	0 677	0 454
Iteration:	**5**						
Wij:	0 7265	-0 3646	0 4088				
	-0 1884	0 3154	0 6039				
Vk:	0 610	0 1852	-0 171				

Table C2 – Summary of Iterative Solutions (2nd Data Set)

Examples of Linear Regression Analysis and Back Propagation Neural Network Software

Linear Regression Analysis (LRA) has been used as a method to help model linear problems. Traditionally, it was done by finding the slope (m) of a linear function (y =mx + c) that best fits the data. The linear function can then be used to predict the outcome of new data values found within the defined limits of the line. An alternative to the above method may be performed by training a neural network. A Back Propagation Neural Network (BPNN) is developed and then trained with test data values. The network learns the relationship between the input and output values by iteratively comparing the error between the predicted and actual outcome and then adjusting the system weights. The BPNN will then be able to predict the outcome of the new data. The structure of the BPNN used for LRA is the following:



Linear Transfer Function

$$Y = \beta_0 + \beta_1 X_1$$

Figure C1.1

This tutorial consists of step-by-step documentation supplemented by screen shots for the software tool utilized and the selected options demonstrated. The conclusion contains a comparison of the results between using LRA from Microsoft (MS) Excel and a BPNN on the data and recommendations for better training of the BPNN.

Objective of tutorial

The objectives of this tutorial are to:

- Demonstrate the effectiveness of the BBPN in achieving the same goal of using LRA.
- To familiarize the user with the software tools.
    - MS Excel
    - NeuralWorks Professional II/PLUS (NWP2+)
- To develop specific performance skills.
    - Ability to perform LRA using Excel
    - Construct a BBPN using NWP2+
    - Train the constructed BBPN
    - Supply the BBPN with test data

Collected Data

Table C1.2.1 contains a set of data that was collected, and then plotted in Figure C1.2.2 using Excel (Steps of data plotting are not discussed in this tutorial).

X	Y
27.20	11.88
27.25	11.02
28.38	12.13
30.29	11.23
33.10	12.54
34.18	10.31
35.07	10.98
38.93	9.82
43.65	8.87
45.35	8.24
45.38	11.60
48.56	9.35
56.93	9.72
57.79	8.81
58.01	8.63
58.07	10.22
61.17	9.49
70.09	6.68
79.13	7.99
79.21	7.95
70.34	8.38
71.03	7.91
74.64	8.78
74.72	6.21
77.43	8.44

Table C1.2.1 – Sample Data Set

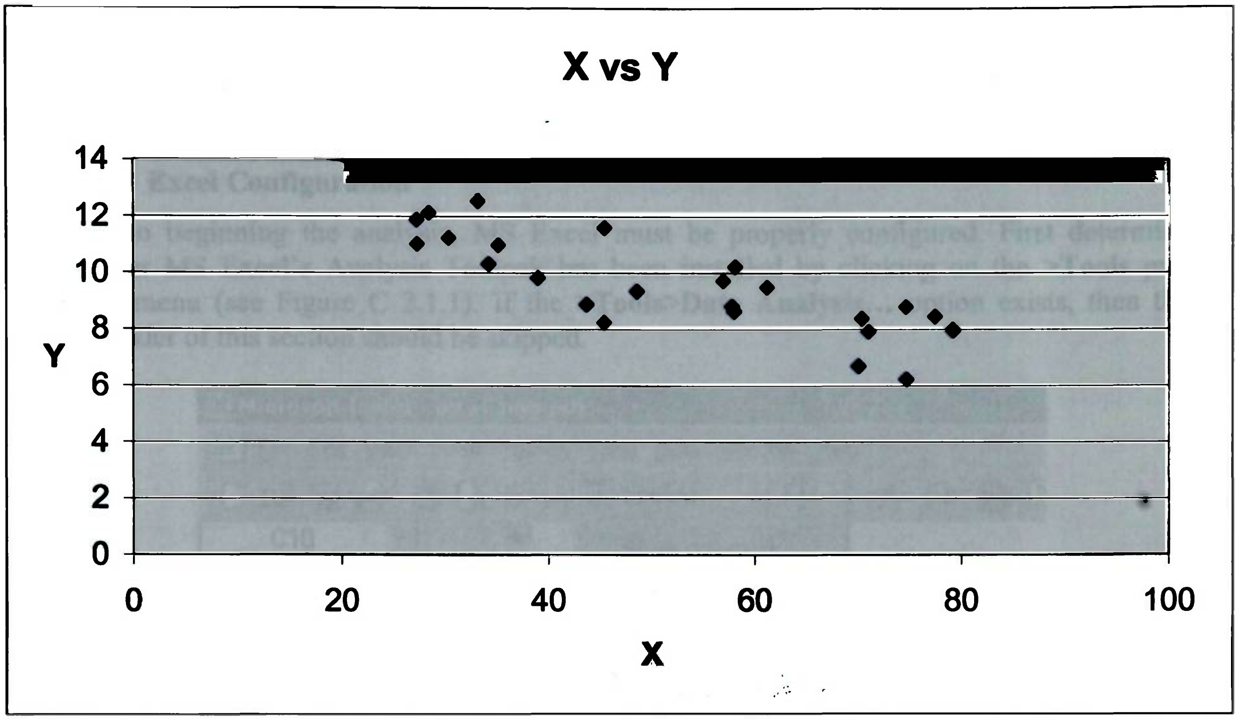


Figure C1.2.2 – Scatter Plot of Sample Data

As Figure C1.2.2 illustrates, the data demonstrates linear properties. It is therefore feasible to model this data with a linear equation. The next step is to find the straight line that best fits the data. That procedure is demonstrated using MS Excel and NWP2+ with BBPN.

Linear Regression Analysis with Excel

In this section, MS Excel is used as a tool to help find a best-fit straight line to fit the data in Table C1.2.1. The data should be formatted as previously shown into two-separated columns.

Excel Configuration

Prior to beginning the analysis, MS Excel must be properly configured. First determine whether MS Excel's Analysis Toolpak has been installed by clicking on the >**Tools** pull down menu (see Figure C 2.1.1). If the **>Tools>Data Analysis...** option exists, then the remainder of this section should be skipped.

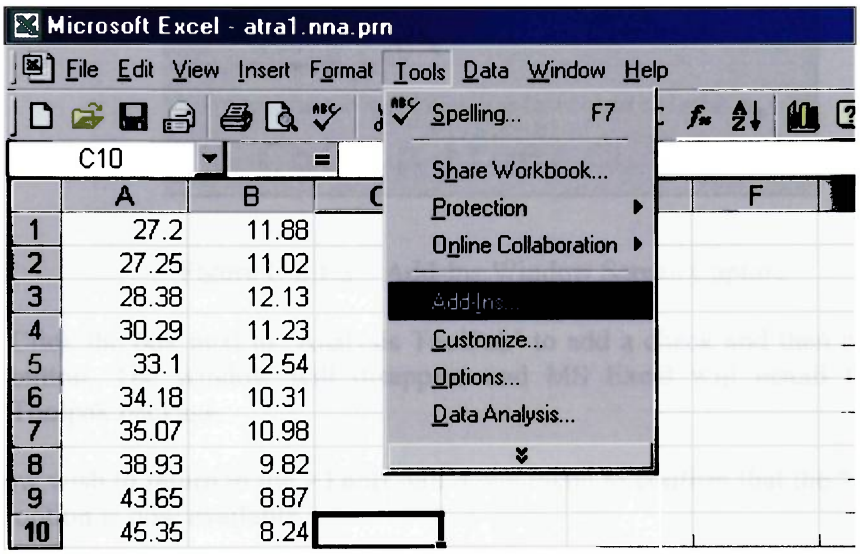


Figure C 2.1.1 – >Tools>Data Analysis... Screen Capture

If the **>Tools>Data Analysis...** option is not available, follow the next steps to install the Data Analysis Toolpak:

1. Select >**Tools>Add-Ins...** item under the >**Tools** pull down menu. The following window will be displayed:

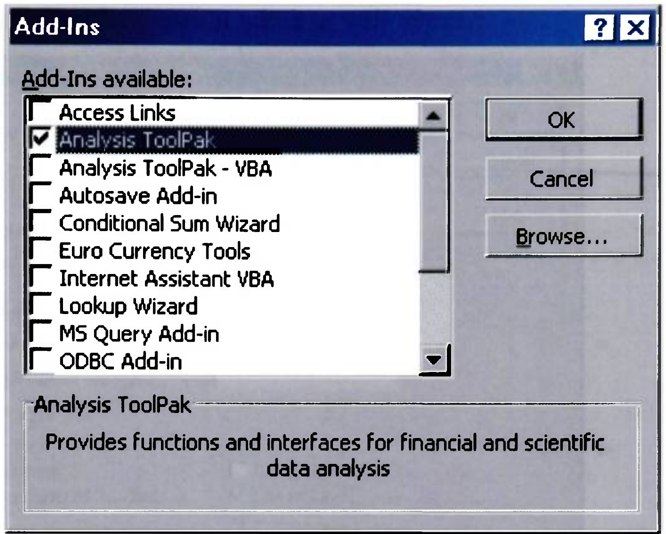


Figure C 2.1.2 – Add-Ins Window Screen Capture

2. Click the box next to 'Analysis ToolPak' to add a check and then click the **OK** button. The window will disappear and MS Excel will install the Analysis Toolpak package.

You may now wish to return to the **>Tools** pull down menu to confirm that the **>Tools>Data Analysis...** option is now available.

Data Analysis – LRA

Prior to this point, the data in Table C1.2.1 should have been formatted into two separate columns.

The following steps describe how to perform LRA using Excel:

1. Select **>Tools>Data Analysis...** under the **Tools** pull down menu which will bring up the following window:

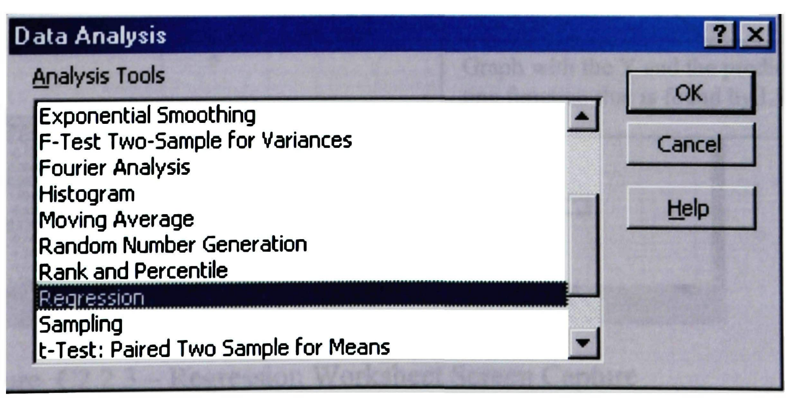


Figure C 2.2.1 – Data Analysis Window Screen Capture

2. Scroll down the list and select 'Regression' as shown in Figure C2.2.1. Click the OK button and then Regression window will appear:

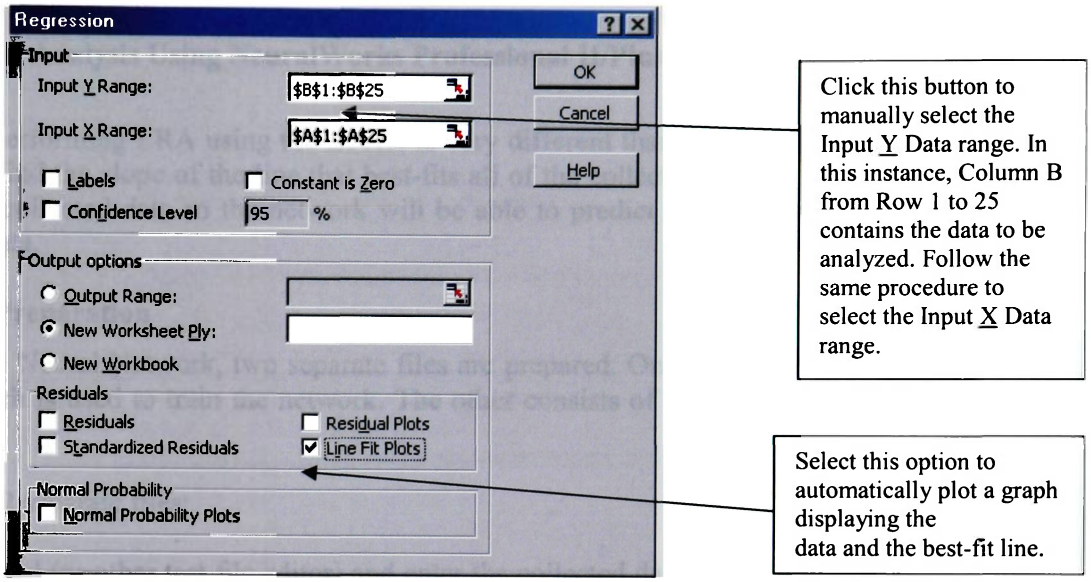


Figure C2.2.2 – Regression Window Screen Capture

3. Select the input range for X and Y as shown in Figure C2.2.2 and select the Residuals option for **Line Fit Plots**. Click on **OK**.

4. A new worksheet will then be generated:

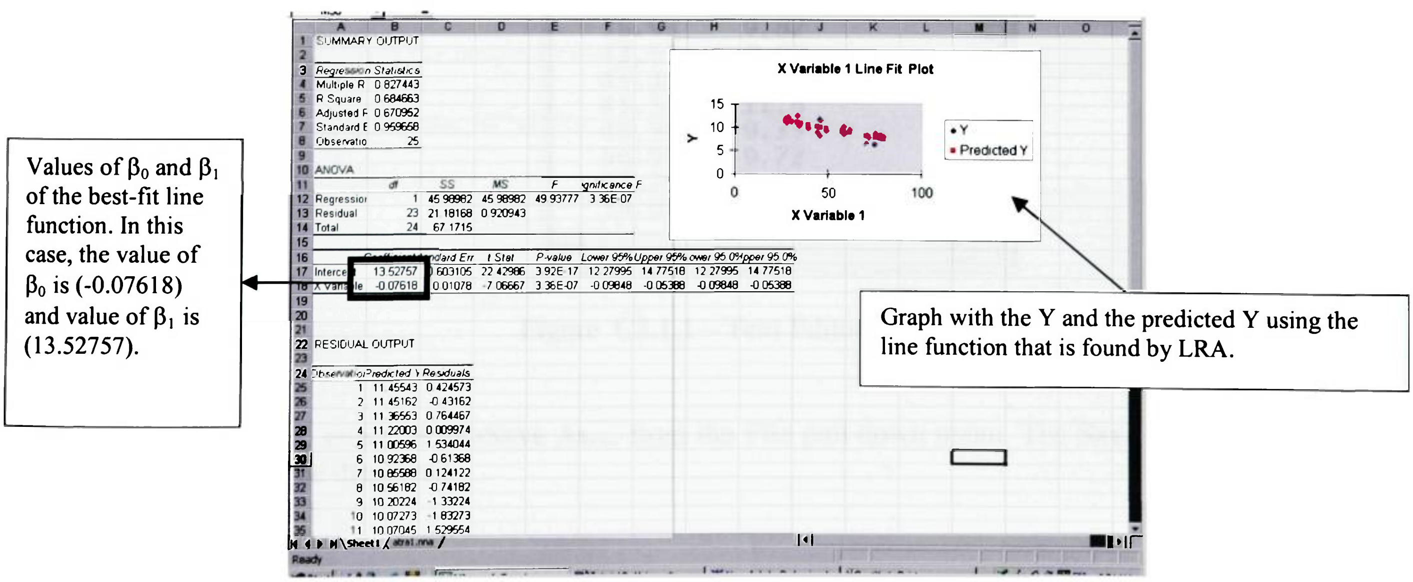


Figure C2.2.3 – Regression Worksheet Screen Capture

The equation for the best-fit straight line can then be determined to be:

$\mathbf{Y} = -0.07618 + 13.52757\mathbf{X}_1$

Linear Regression Analysis Using NeuralWorks Professional II/Plus

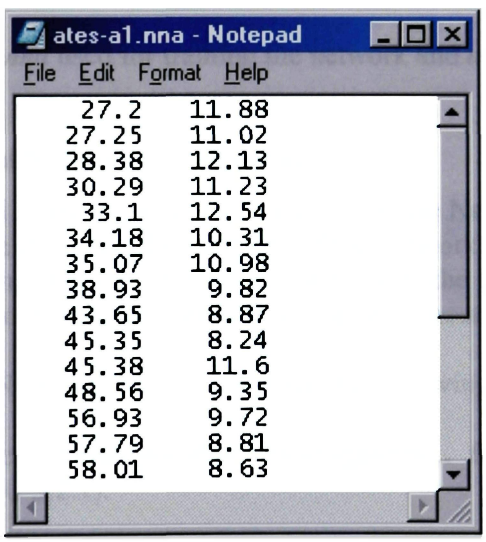
The approach of performing LRA using the BPNN is very different than using MS Excel. It does not actually find the slope of the line that best-fits all of the collected data. It trains the network with the collected data so the network will be able to predict the outcome of new sets of collected data.

Network Preparation

In order to use the Neural Network, two separate files are prepared. One file consists of the training data, which is used to train the network. The other consists of the test data used to test the network.

Steps for creating 2 separate files:

1. Start Notepad (or other test file editor) and enter the collected data in two columns as shown:



27.2	11.88
27.25	11.02
28.38	12.13
30.29	11.23
33.1	12.54
34.18	10.31
35.07	10.98
38.93	9.82
43.65	8.87
45.35	8.24
45.38	11.6
48.56	9.35
56.93	9.72
57.79	8.81
58.01	8.63

Figure C3.1.1 – Text Editor Screen Capture

2. Then, select **>File>Save As...** from the **File** pull down menu. The **Save As** window will be displayed:

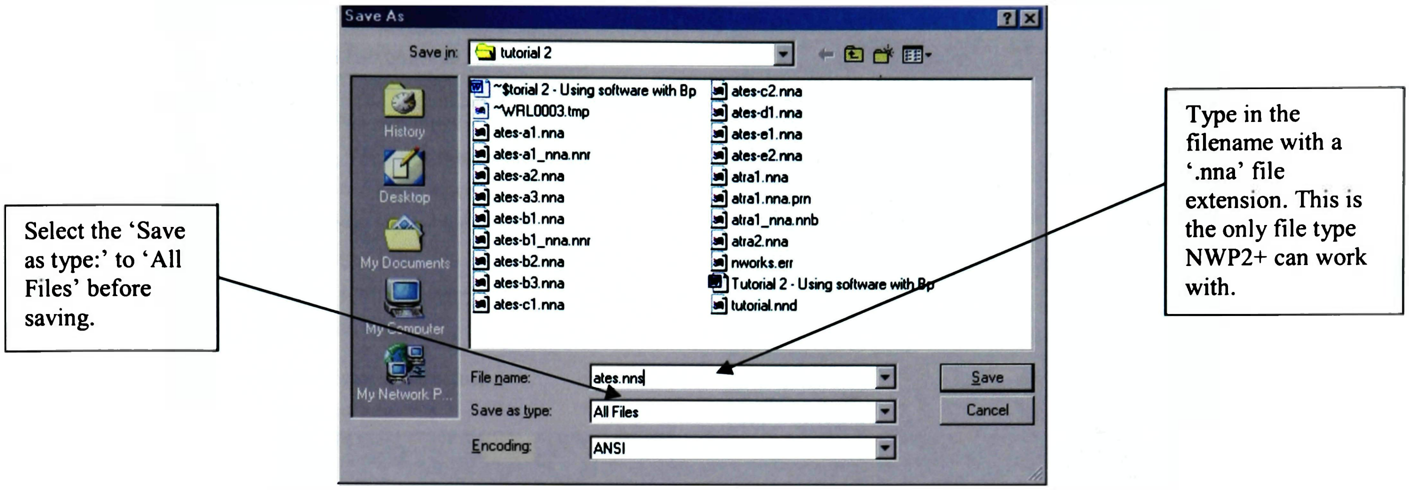


Figure C3.1.2 – Save As Window Screen Capture

3. Now, save the file *ates-a1.nna* and *ates.nna* in the same folder. (Any folder is acceptable so long as it may be located later.)

Two files are then created: *ates.nna* used for training the network and *ates-a1.nna* used to test the network.

Back Propagation Neural Network Construction

Before you construct the Neural Network, a defined structure of the Neural Network needs to be defined and decided upon. In this case, the structure of the network will be the same as in Figure 1.1. The reason for using this structure is that we know the collected data has the linear properties and the output of this neural network will give a linear equation.

The steps to construct the BPNN for the collected data are the following:

1. Startup NWP2+ and select **>InstaNet>Back Propagation...** from the **InstaNet** pull down menu (see Figure C 3.2.1).

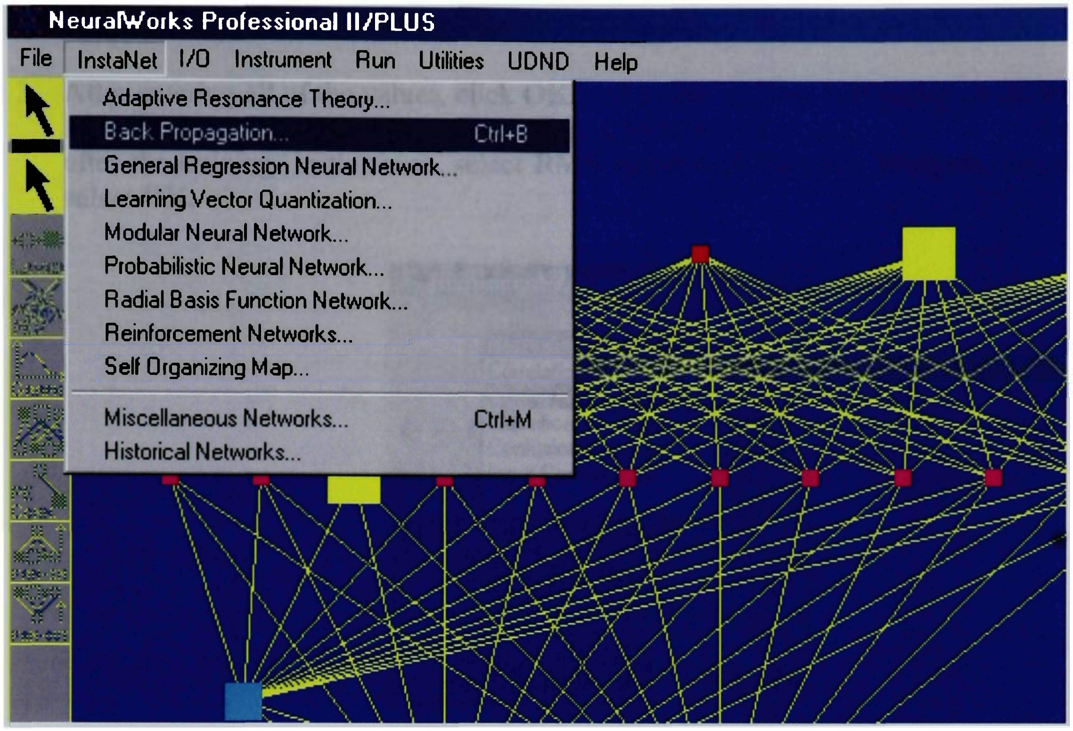


Figure C 3.2.1 – Creating BPNN

2. After selecting **Back Propagation...**, a new window will pop up as shown in Figure C 3.2.2. This window defines the structure of the network. Set the variables accordingly using the values in Figure C 3.2.2.

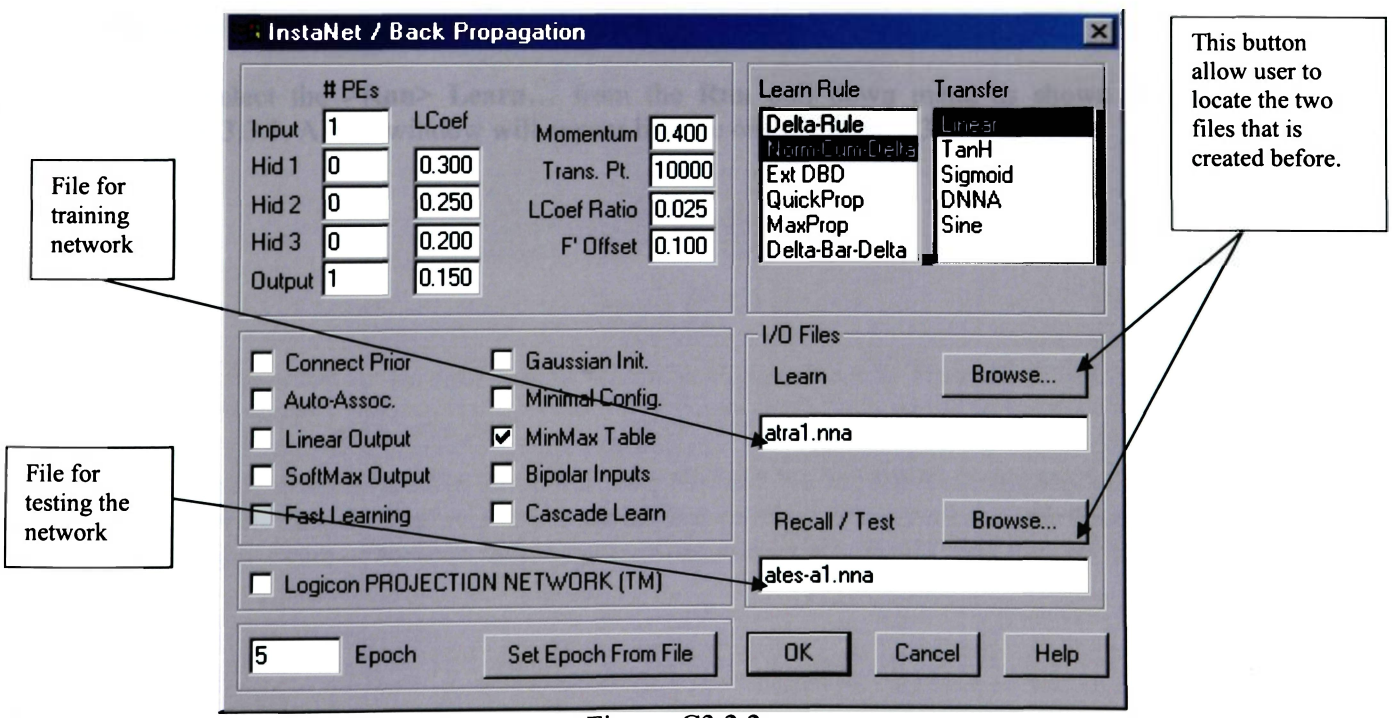


Figure C3.2.2

3. After entering all of the values, click **OK**, and a new window will popup as shown in Figure C3.2.3. This window allows the user to select what results will be displayed after the training. In this case, select **RMS Error** and **Network Weights**, and then select **OK.**

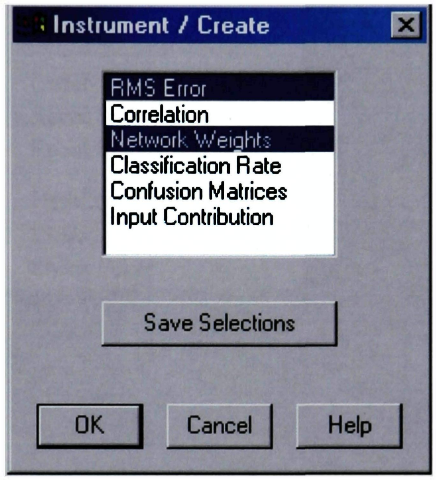


Figure C3.2.3

Back Propagation Neural Network Training

With the network now constructed, you are seconds away to train the network.

The steps to train the network are as follows:

1. Select the **>Run> Learn...** from the **Run** pull down menu as shown in Figure C3.3.1. A new window will appear like the one in Figure C3.3.2.

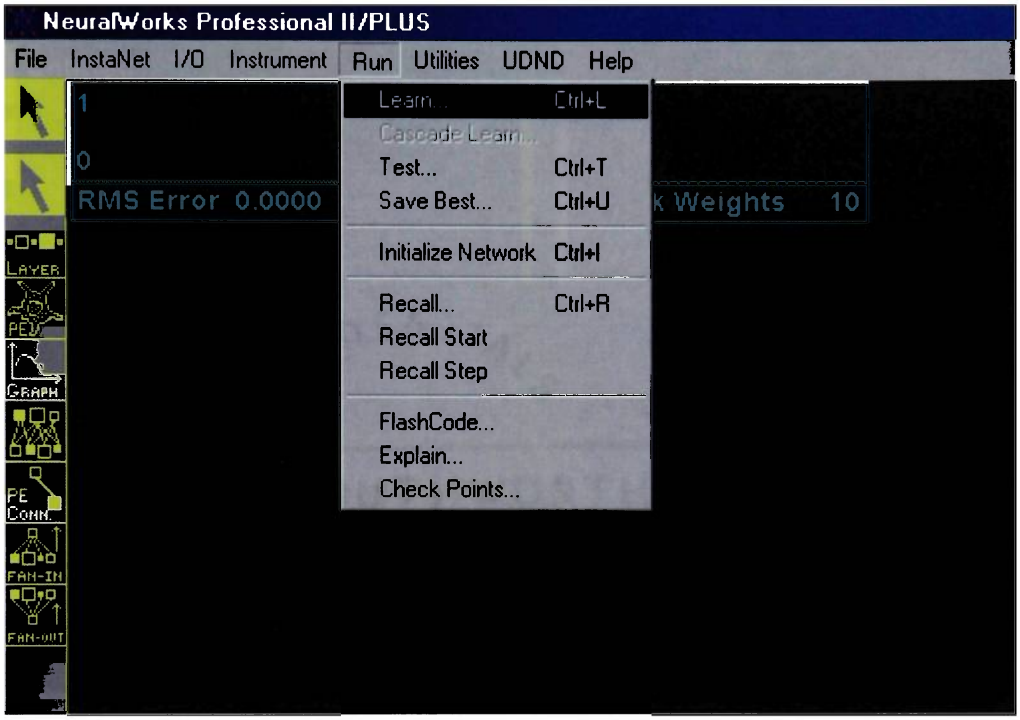


Figure C3.3.1

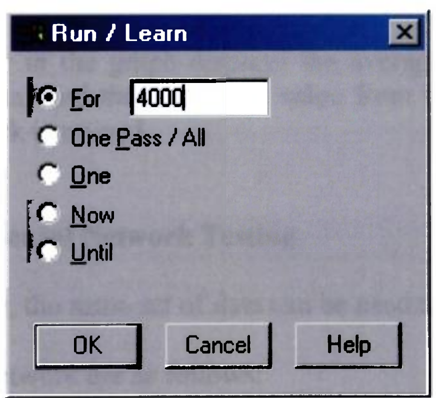


Figure C3.3.2

2. Enter 4000 in the field box as shown in Figure C 3.3.2. Then, click **OK** to start the training.

While the network is training, the screen may flicker a bit, especially on the top left hand side of the screen. It will display a wave signal line moving. Wait until the wave signal stops, signifying the end of training. After the training, the screen should look similar to Figure C 3.3.3.

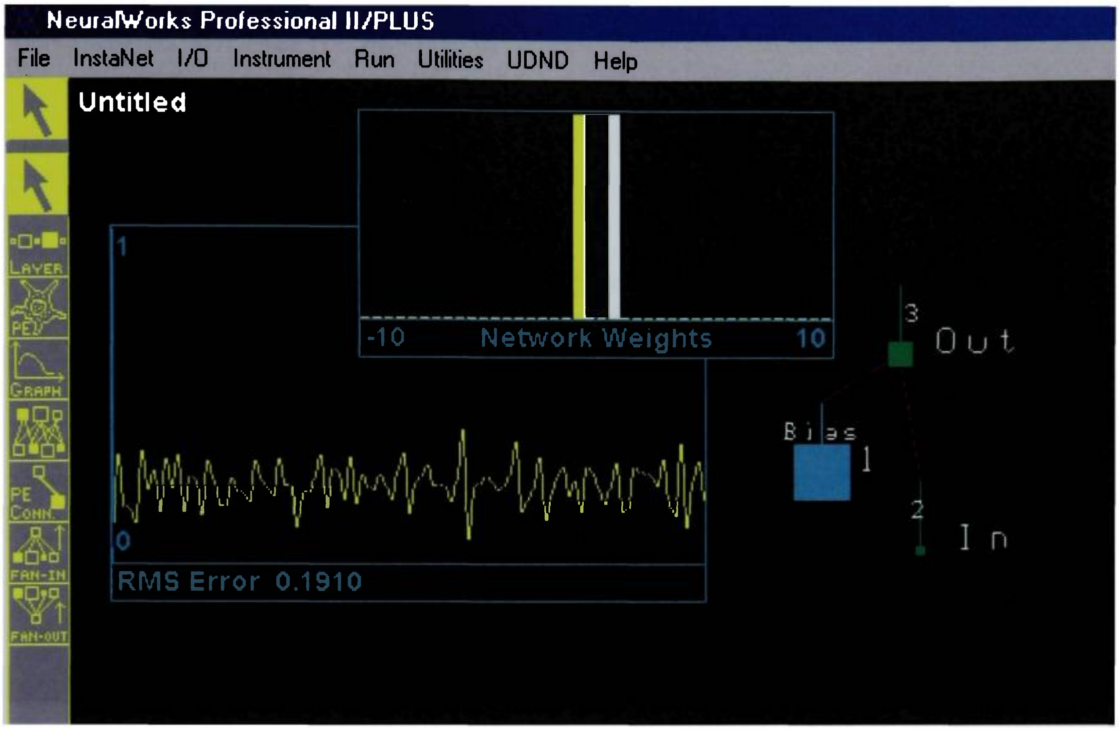


Figure C 3.3.3

The value of the RMS Error in the graph displays the average difference of the expected value (from the collected data) and the computed value from the network. The smaller the number, the closer the network is trained.

Back Propagation Neural Network Testing

With the trained BPNN ready, the same set of data can be used to test the network again.

The steps to test the neural network are as follows:

1. Select **>Run>Test** from the **Run** pull down menu (see Figure C3.3.1). A new window will appear as shown in Figure C3.3.2. This time select **One Pass/All** instead of **For**.
2. Click **OK**, and the network will begin testing. Wait until the signal wave stops moving indicating the testing is complete. After the testing, the screen should look similar to Figure C3.4.1.

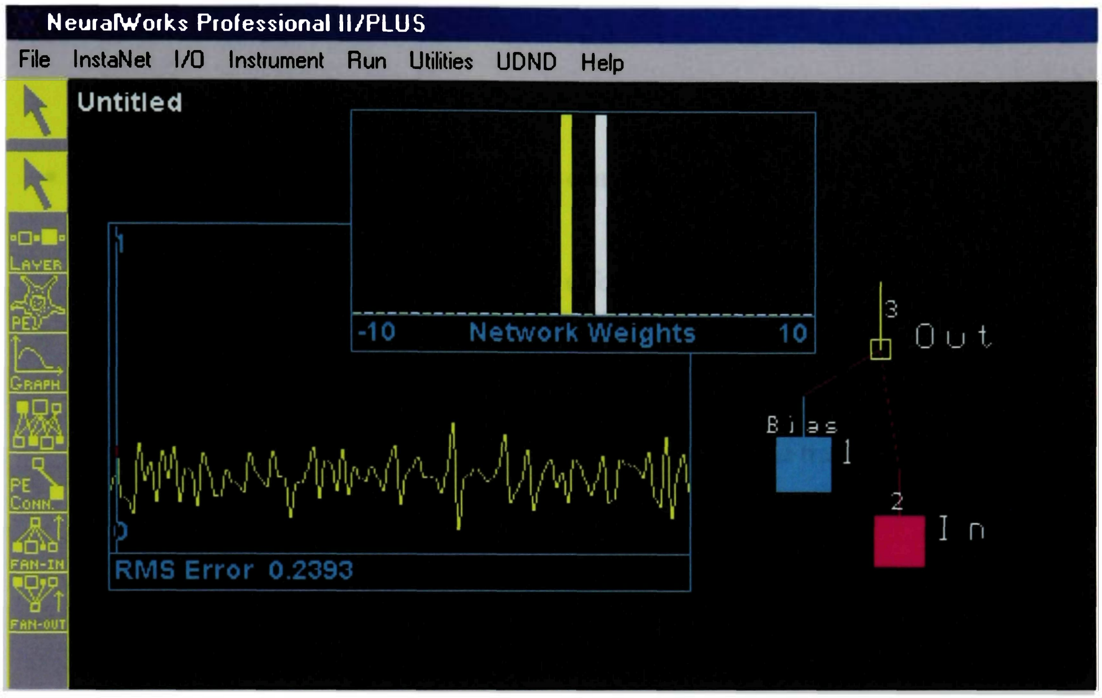


Figure C3.4.1

After the execution of the testing on the network, an output result file will be created. In this case, *ates-a1.nna* is used for the testing, so the output file will be *atra-a1_nna.nnr*. Open *atra-a1_nna.nn* using the Notepad application or other text editor. It will look like the one shown in Figure C3.4.2.

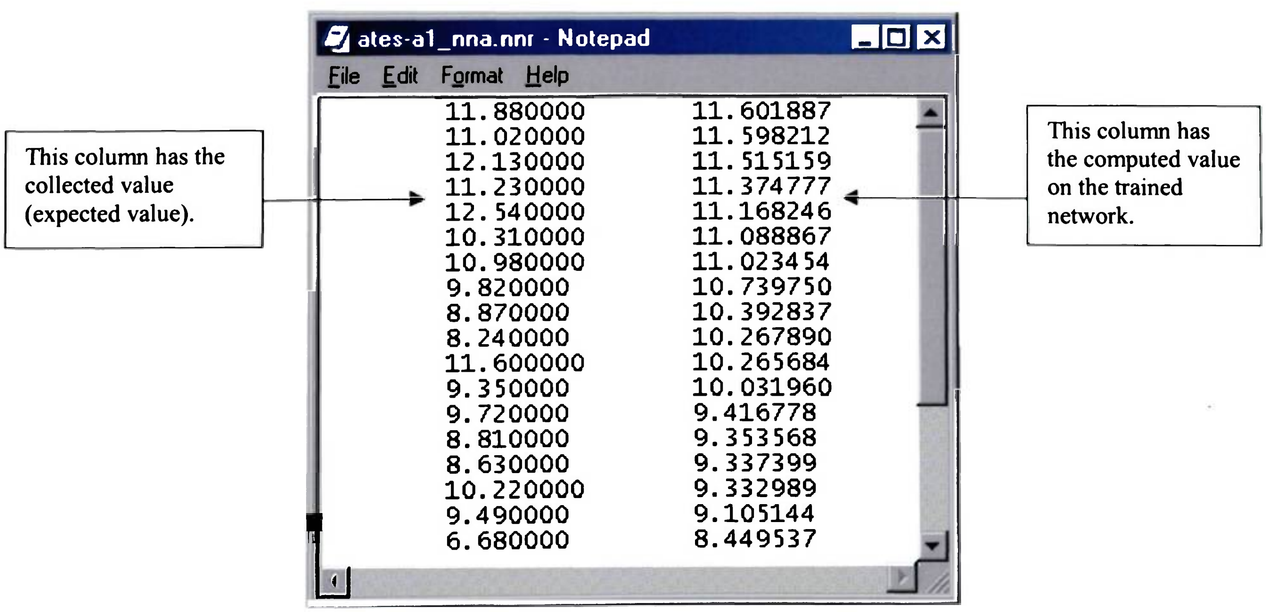


Figure C3.4.2

Conclusions

- Back Propagation Neural Networks can accurately simulate Linear Regression Analysis as performed by MS Excel.

- However, the degree of accuracy and effectiveness of using BPNN to achieve this goal depends on the quality of the training. It was shown in the tutorial that results varied with training parameters.

- Since BPNN is highly dependent on training, the training process may be augmented by using random input training sequences and lowering the transition point.