JHSV Analysis Engine

2007-01-10

Gregory Alan Hildstrom
Hildstrom Engineering
Subcontract/D.O. DDL 04-D-4032-010
DDL Omni Engineering
Prime Contract N00178-04-D-4032, D.O. FD02
NSWCCD Code 653
hildstrom@hildstrom.com
gregory.hildstrom@navy.mil
Mobile: (210)413-6082
Office: (301)227-3967

NAVSEA
WARFARE CENTERS
Carderock Division

This page is intentionally blank.

Table of Contents

Illustration Index

Index of Tables

Introduction

This document aims to provide enough information for a programmer or engineer to read the JHSV SCL-format binary data and use the JHSV Analysis Engine (AE).

Distribution of recorded time histories to other entities is increasingly required during and after analysis. Distribution is particularly important for third party review and designer understanding of our recorded data. Data format standardization allows for data sharing and greatly simplifies overall system design. In this document, a complete specification of the SCL binary data format is discussed and C++ code examples are given. The binary data for numerous model tests and full-scale sea trials were recorded in SCL format with National Instruments LabView. The data acquisition programs were written by Jesus Rosario. The AE reads SCL binary data files only; other data file formats are not implemented, which greatly reduces code size. Future tests will also utilize this Code 653 standard data format. Below is a summary of the ship tests that have been recorded using the SCL binary format.

Model Tests:

- DD-21
- HSS
- DDX

Sea Trials:

- HSV-X1
- HSV2
- INLS
- JHSV

The AE calculates numerous statistics from recorded time histories and virtual channels immediately following SCL data file acquisition. This reduces the data to a more manageable size for satellite transmission and reduces the calculations necessary in system interface and display software. The AE calculates the following statistics:

- rainflow cycle counting
- fatigue damage accumulation
- time-domain statistics (MMM)
    - mean
    - max
    - min
    - variance
    - skewness
    - kurtosis
    - computed from
        - raw (combined) original time histories
        - lowpass Fast Fourier Transform (FFT) (Wiener) filtered time histories
        - highpass (raw-lowpass) filtered time histories

- power spectral density (PSD)
- response amplitude operator (RAO)
- Weibull 2-parameter amplitude statistics
  - amplitude mean
  - amplitude max
  - amplitude min
  - amplitude variance
  - amplitude skew
  - amplitude kurtosis
  - linear regression beta (shape parameter)
  - linear regression y-intercept
  - linear regression characteristic value (scale parameter)
  - linear regression correlation
  - moment method beta (shape parameter)
  - moment method y-intercept
  - moment method characteristic value (scale parameter)
  - amplitude types (one amplitude per lowpass wave cycle)(or one amplitude per raw wave cycle if cutoff frequency is 0)
    - raw max-mean
    - raw min-mean
    - raw peak to peak (p2p)
    - lowpass max-mean
    - lowpass min-mean
    - lowpass p2p
    - highpass max-mean
    - highpass min-mean
    - highpass p2p

System Diagram

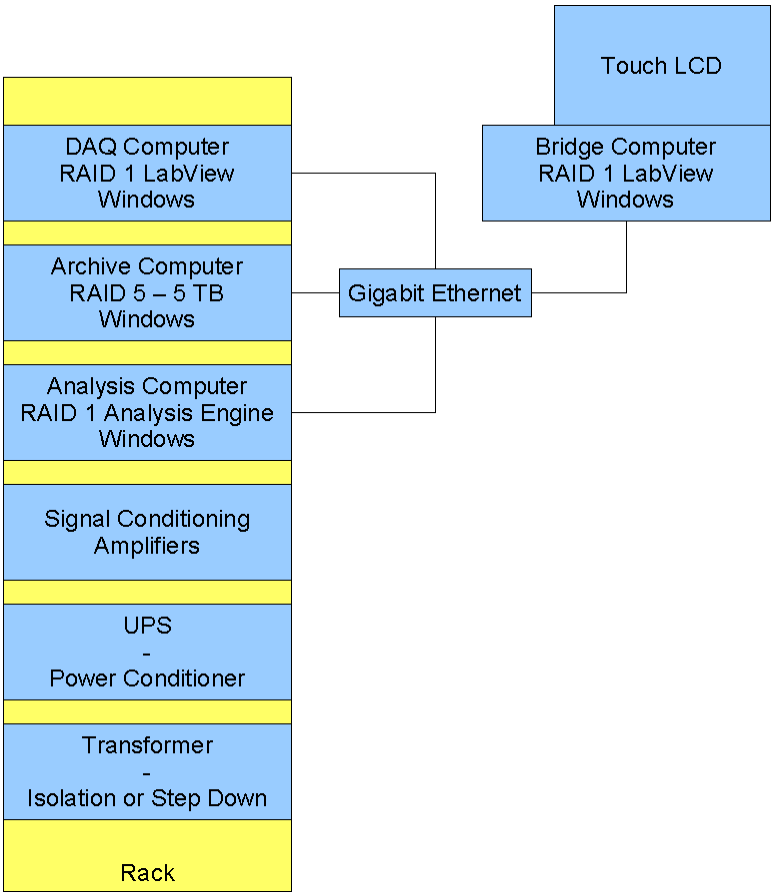

Illustration 1: JHSV Hull Monitoring System

Data Acquisition

The data acquisition computer saves SCL binary data files to the archive computer, via a mapped network drive, at a given time interval such as 30 minutes or 1 hour. After a binary data file has been written to the archive, the data acquisition software appends the name of the data file to a log file called daqlog.txt, which is located in the same folder as the data files. Each line in the log file contains only one data file name followed by a newline character.

The log file can be polled for changes by any computer attached to the network.

SCL Data Files

Binary Data Format

The SCL data files consist of a preceding header and interleaved data scans. The header is always (Number of channels + 1)*8 bytes. The number of channels is the first binary number stored in the data file, which enables a program to read the entire file without additional information. Table 1 shows an example byte layout of a 4-channel 2-scan binary SCL (example.scl) data file. The header is (Number of channels + 1)*8 bytes, (4+1)*8, or 40 bytes.

Section	Byte Range	Data Value	Data Type	Byte Ordering	Data Size (Bytes)
Header	0-3	Number of channels	Integer	high-byte-first	4
Header	4-7	Scan rate	Float	high-byte-first	4
Header	8-15	Calibration factor Channel 0	Double	high-byte-first	8
Header	16-23	Calibration factor Channel 1	Double	high-byte-first	8
Header	24-31	Calibration factor Channel 2	Double	high-byte-first	8
Header	32-39	Calibration factor Channel 3	Double	high-byte-first	8
Data	40-43	Channel 0 Scan 0	Float	high-byte-first	4
Data	44-47	Channel 1 Scan 0	Float	high-byte-first	4
Data	48-51	Channel 2 Scan 0	Float	high-byte-first	4
Data	52-55	Channel 3 Scan 0	Float	high-byte-first	4
Data	56-59	Channel 0 Scan 1	Float	high-byte-first	4
Data	60-63	Channel 1 Scan 1	Float	high-byte-first	4
Data	64-67	Channel 2 Scan 1	Float	high-byte-first	4
Data	68-71	Channel 3 Scan 1	Float	high-byte-first	4

Table 1: Example 4-channel 2-scan SCL Byte Layout

SCL data files are stored using high-byte-first (HBF) (big-endian) byte-ordering for compatibility with National Instruments hardware and software defaults. HBF is the native byte order on Motorola, Freescale, and IBM PowerPC processors, which are common in Apple Macintosh, IBM RS6000, and embedded computers.

Common i386 or x86 processor based computer systems use low-byte-first (LBF) (little-endian) byte-ordering as the fundamental system byte-order. LBF is the native byte order on Intel, AMD, and Via processors, which are common on Microsoft Windows based computers. WaveMetrics IGOR Pro can read this HBF “General Binary” data format if the entire header is skipped and the number of channels is known or determined. In order to correctly read HBF binary numbers on a LBF architecture, the bytes must be swapped in memory before use. The same is true of saving HBF binary numbers on a LBF architecture; bytes must be swapped in memory before being written to disk.

Example C++ Code

Below are three functions for switching 4-byte integers, 4-byte floats, and 8-byte doubles from HBF to LBF or LBF to HBF in C++.

```
int swapbytes(int input){
        int output = 0;

        ((char*)&output)[0] = ((char*)&input)[3];
        ((char*)&output)[3] = ((char*)&input)[0];

        ((char*)&output)[1] = ((char*)&input)[2];
        ((char*)&output)[2] = ((char*)&input)[1];

        return output;
}//end swapbytes


float swapbytes(float input){
        float output = 0;

        ((char*)&output)[0] = ((char*)&input)[3];
        ((char*)&output)[3] = ((char*)&input)[0];

        ((char*)&output)[1] = ((char*)&input)[2];
        ((char*)&output)[2] = ((char*)&input)[1];

        return output;
}//end swapbytes


double swapbytes(double input){
        double output = 0;

        ((char*)&output)[0] = ((char*)&input)[7];
        ((char*)&output)[7] = ((char*)&input)[0];

        ((char*)&output)[1] = ((char*)&input)[6];
        ((char*)&output)[6] = ((char*)&input)[1];

        ((char*)&output)[2] = ((char*)&input)[5];
        ((char*)&output)[5] = ((char*)&input)[2];

        ((char*)&output)[3] = ((char*)&input)[4];
        ((char*)&output)[4] = ((char*)&input)[3];

        return output;
}//end swapbytes
```

The following C++ main function, for LBF architectures, reads the entire SCL data header and interleaved data and then prints the output to the standard output as comma-delimited text. This is just an example, many large binary data files are too big to make ASCII conversion useful. It would be much more efficient to load the desired channel(s) into memory and perform analysis there. After the entire header has been read or skipped, the ifstream read pointer will point to the first byte of the first channel of the first scan. Code can then read in the interleaved data. Notice the byte swap function calls that are necessary to convert the HBF data for use on the LBF architecture. This simple program reads from the standard input and writes to the standard output. Usage: "readSCL < Fast_Run_0.scl > output.csv".

```
#include<iostream>
#include<fstream>
using namespace std;

int main(void){
	int numchan = 0;
	float scanrate = 0;
	double cal = 0;
	float datapt = 0;

	//double-check size of int and float on this system (should be 4)
	cout << "size of int," << sizeof(int) << endl;
	cout << "size of float," << sizeof(float) << endl;
	//double-check size of double on this system (should be 8)
	cout << "size of double," << sizeof(double) << endl;

	//read number of channels
	cin.read((char*)&numchan,4);
	numchan = swapbytes(numchan);
	cout << "number of channels," << numchan << endl;

	//read scanrate (model/test scale)
	cin.read((char*)&scanrate,4);
	scanrate = swapbytes(scanrate);
	cout << "scan rate," << scanrate << endl;

	//read all cal factors
	for(int i=0; i<numchan; i++){
		cin.read((char*)&cal,8);
		cal = swapbytes(cal);
		cout << "chan," << i << ",cal," << cal << endl;
	}//end for

	//ready to read interleaved data at this point using numchan until end of file
	while(!cin.eof()){
		for(int i=0; i<numchan && !cin.eof(); i++){
			cin.read((char*)&datapt,4);
			datapt = swapbytes(datapt);
			cout << datapt << ",";
		}//end for
		cout << endl;
	}//end while

	return 0;
}//end main
```

Analysis Engine Installation

The following steps are needed to install and run the JHSV Analysis Engine; until an installer program is created.

1) Download the Microsoft .NET framework 2.0.
2) Install the .NET framework 2.0, which provides most of the system DLL files needed. If this does not provide enough DLL files on a new system and runtime errors are encountered, also install Microsoft Visual C++ 2005 Express Edition, which is used for development.
3) Download the JHSV Analysis Engine.zip file.
4) Unzip the zip file, JHSVAnalysisEngine.zip, in a permanent location (usually C:\) to reveal the folder C:\JHSVAnalysisEngine.
5) Add the folder containing ae.exe to the system path variable. This allows the system to be able to open all of the Analysis Engine applications from anywhere regardless of where they are installed. The following steps are clearly shown in illustrations 2, 3, and 4.
    1) Click Start, then right-click on My Computer, then click on Properties.
    2) Click on the Advanced tab, then click on the Environment Variables button.
    3) Navigate to Path under System Variables, then double-click on Path to edit this variable.
    4) Add the folder path to the end of the path list; entries are separated by semicolons. Go to the end of the list, add a semicolon, then add the path to the folder like: C:\JHSVAnalysisEngine\bin.
6) Optional: Add the folder containing AcroRd32.exe to the system path. This allows AEGUI to open Adobe Reader regardless of where it is installed. The folder is usually something like: C:\Program Files\Adobe\Acrobat 7.0\Reader.
7) Optional: Add the folder containing Igor.exe to the system path. This allows AEGUI and loadsclfilewithigor to open WaveMetrics IGOR Pro regardless of where it is installed. The folder is usually something like: C:\Program Files\WaveMetrics\Igor Pro Folder.
8) Optional: Merge the three registry files: scldatafile1.reg, scldatafile2.reg, scldatafile3.reg, into the Windows system registry. Highlight all three registry files in C:\JHSVAnalysisEngine\bin, then right-click one and left-click Merge. This will add many of the AE tools to the Windows context menu. This sets double-clicking on .scl files to open them in IGOR and double-clicking on .csv files to open them in Excel. Right-clicking of *single* .scl files allows ae, scl2ascii, and sclscanrate to be executed on the selected data file. Right-clicking of *single* .csv files allows ascii2scl or slae to be executed on the selected data file. Operations involving multiple data files, like executing aerunner or scl2ascii on 1000 SCL data files, should be performed using drag and drop functionality on aerunner.exe or scl2ascii.exe, which handles the data files sequentially. The context menu options only handle *single* files gracefully; Windows will attempt to execute multiple selected files simultaneously instead of sequentially, which can overload a system quickly. Illustrations 5 and 6 show merging registry files and the resulting context menu options.

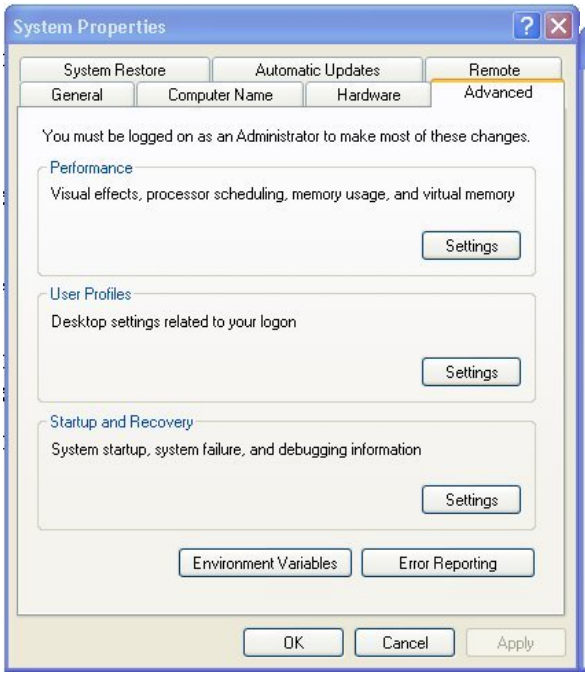

Illustration 2: System Properties Advanced Tab

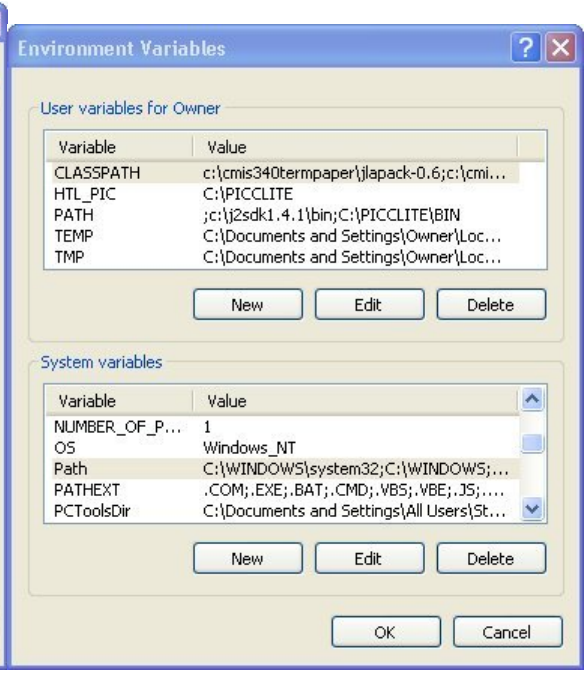

Illustration 3: Environment Variables Window

Illustration 4: Path Environment Variable Editing

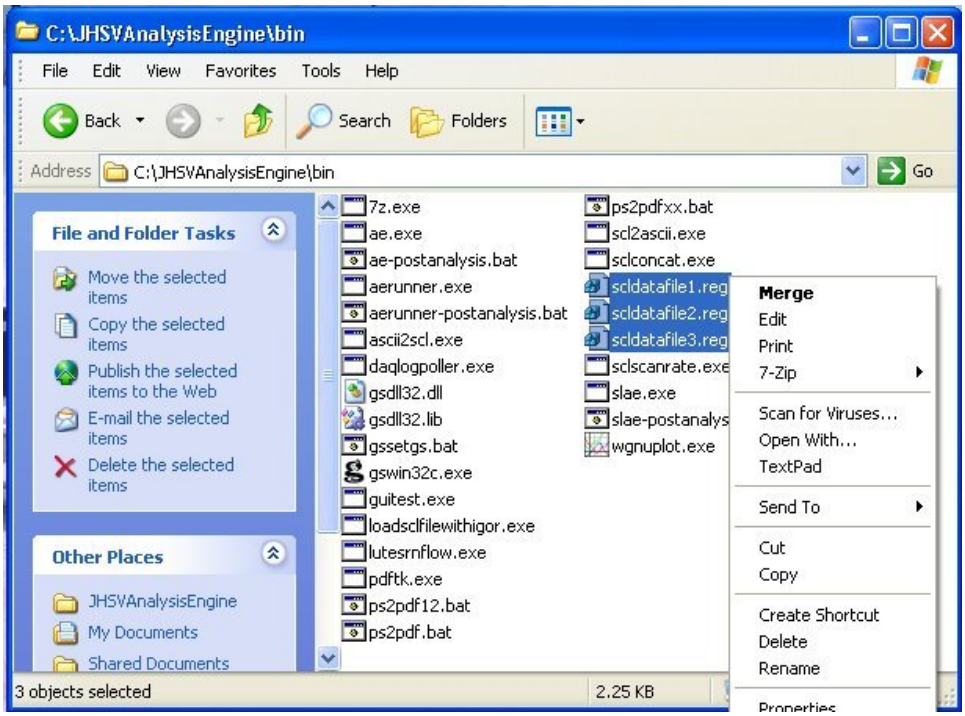

Illustration 5: Merging Registry Files

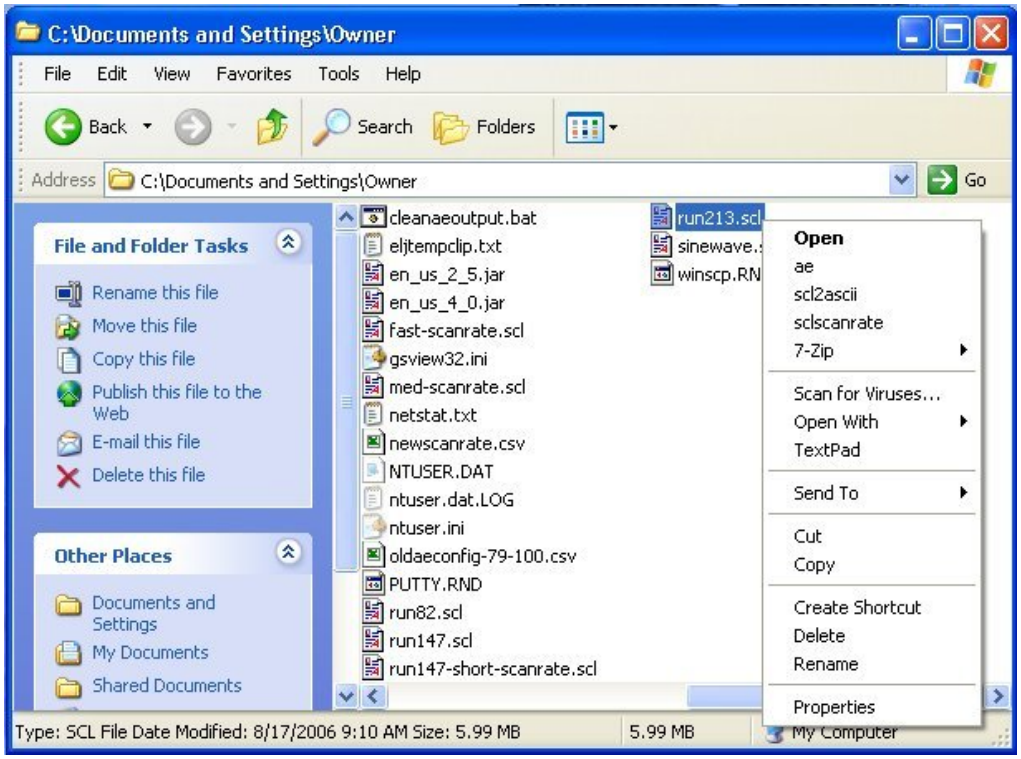

Illustration 6: SCL File Context Menu

Analysis Engine Application

Overview

AE is written in ANSI standard C++ and requires no external libraries. This minimizes code size, maximizes portability, and maximizes speed. The build environment for development and testing is a computer with an Intel P4 processor, Microsoft Windows XP SP2, and Microsoft Visual C++ 2005 Express Edition. AE is a single processor command line program that accepts the data file name as its only argument. Example: "c:\datafolder>ae datafilename.scl". Drag and drop also works in Windows.

The input data file is a standard SCL binary data file as described in previous sections of this document. The optional configuration file is aeconfig-numchan-scanrate.csv. The output files are:

- aeconfig-numchan-scanrate.csv: optional ASCII comma-delimited configuration file
- damagesum-numchan-scanrate.double: accumulated damage in LBF binary format
- damagesum-numchan-scanrate.csv: an ASCII copy of the binary damage file
- datafilename-fch.scl: an SCL file containing rainflow-detected full cycle histograms
- datafilename-psd.scl: an SCL file containing PSDs of the data file time histories
- datafilename-rao.scl: an SCL file containing RAOs of the data file time histories
- statslog.csv: an ASCII comma-delimited log of fatigue, MMM, PSD, and Weibull statistics. Detailed descriptions of the columns in this file are in appendix A.

Rainflow Cycle Counting

Rainflow cycle counting is used to detect and count closed stress/strain hysteresis loops, which are then used to determine cumulative fatigue damage. This analysis is performed on channels selected in the optional configuration file. Rainflow Algorithm I, developed by Downing and Socie, was chosen for simplicity. (Downing, 1982, p. 32) The more complex Rainflow Algorithm II, also developed by Downing and Socie, is utilized by Sandia National Laboratories' LIFE2 code in the analysis of wind turbine components. Algorithm II was chosen for LIFE2 because of memory constraints at the time of development. (Schluter, 1991, p. 7, 10, 29) The algorithm used in AE is also called "Simplified Rainflow Counting for Repeating Histories". (ASTM, 2005, p. 5)

Another useful discussion on rainflow analysis can be found on Wikipedia's website. (Rainflow, n.d.)

The full cycles are also counted in an SCL histogram file to save storage space over saving every cycle. The output file is named the same as the input data file with -fch appended before the file extension. Example: inputfile.scl would yield the histogram output file inputfile-fch.scl, which can be loaded directly into IGOR or LabView. The maximum cycle detected for each channel is stored in the SCL header as the calibration factor for that channel, which allows the x-axis of each histogram to be properly scaled when loaded by another program. The number of histogram bins is defined in the optional configuration file. These histograms are not used in the internal AE fatigue damage calculations; they are made available for external use. Cycles are used individually and are not grouped together when they are applied to the S-N curves.

Fatigue Damage Accumulation

Cumulative damage is calculated by applying Miner's rule to an S-N curve of the gaged material and the individual rainflow-detected cycles. Binning is not used in the damage calculation. A discussion of material fatigue and Miner's rule can be found on Wikipedia's website. (Fatigue, n.d.)

Miner's rule is typically defined as $\sum_{(i=1)}^{k}\left[\frac{n_i}{N_i}\right]=C$, where k is the number of different stress levels, n_i is the number of cycles found at a given stress level $n(S_i)$, N_i is the number of cycles to failure at the same given stress level $N(S_i)$, and C is the cumulative damage. The problem with this formula is that it is typically used on cycle histograms that have significantly different magnitude cycles binned together, which is often undesirable. The formula used here assumes that every cycle is completely unique and independent and that the S-N curve is continuous; no grouping or binning. The modified Miner's rule used here is $\sum_{(i=1)}^{k}\left[\frac{1}{N_i}\right]=C$, where k is the number of detected cycles and n_i is always 1 for each unique independent cycle.

In an imaginary time-history, a 260 MPa magnitude cycle is detected during rainflow cycle counting along with many others. That channel's example SN curve is presented in illustration 7. The magnitude of this single cycle is associated with 100 cycles to failure. Another cycle with a magnitude of 215 MPa is associated with 150 cycles to failure.

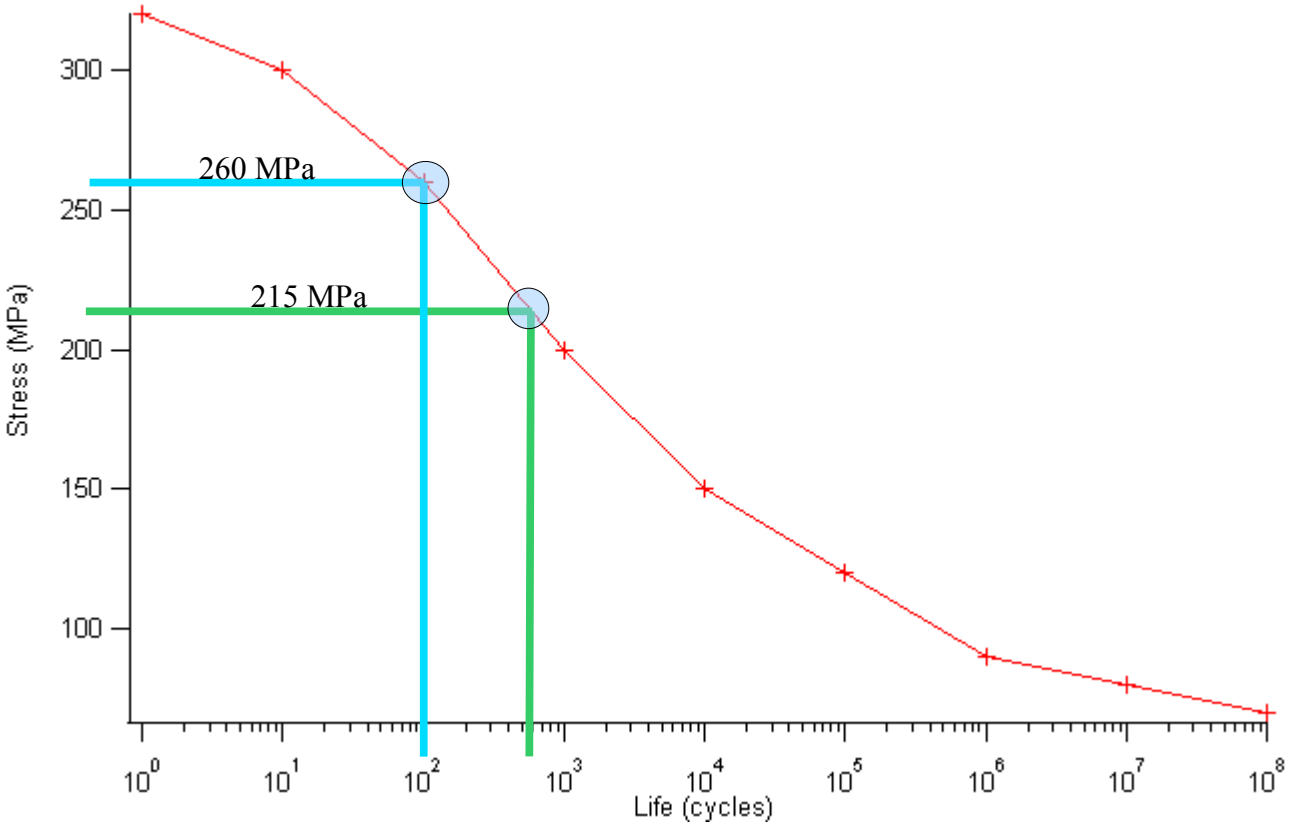

Illustration 7: Example SN Curve and Single Cycle Damages

The damage for the single 260 MPa cycle is 1/100 or 0.01. The damage for the single 215 MPa cycle is 1/150 or 0.0066666. The cumulative damage, C, for just these two cycles is the sum of the individual damages; $0.01 + 0.0066666 = 0.0166666$.

The S-N curves are defined in the optional configuration file as a set of discrete points on the S-N curve (1e0 through 1e8). Linear interpolation is used to determine the number of cycles to failure for stress levels that fall between points on the S-N curve.

The cumulative fatigue damage, C, from each input data file is computed and reported along with all of the time domain statistics in statslog.csv on a run by run and channel by channel basis. The cumulative aggregate fatigue damage, also C, from all of the analyzed data files is stored in two files: damagesum-numchan-scanrate.double and damagesum-numchan-scanrate.csv. The damagesum-numchan-scanrate.double file saves the cumulative C for each channel in a binary double-precision (64-bit) number, which preserves calculation precision from one data file analysis to the next. Some precision is lost in the default conversion to ASCII, but for display purposes the ASCII version is more than sufficient, so damagesum-numchan-scanrate.csv contains an ASCII copy of the contents of damagesum-numchan-scanrate.double.

Time Domain Statistics (MMM)

Most of the time domain statistics (MMM), come straight out of Numerical Recipes in C, section 14.1. (Press, 1999, p. 610) This analysis is performed on channels selected in the optional configuration file.

The mean is computed by $\bar{x} = \frac{1}{N}\sum_{j=1}^{N} x_j$.

The maximum is the maximum recorded value.

The minimum is the minimum recorded value.

The variance is computed by $Var(x_1 \ldots x_N) = \frac{1}{(N-1)}\sum_{j=1}^{N}(x_j - \bar{x})^2 = \sigma^2$.

The skewness is computed by $Skew(x_1 \ldots x_N) = \frac{1}{N}\sum_{j=1}^{N}\left[\frac{(x_j - \bar{x})}{\sigma}\right]^3$.

The kurtosis is computed by $Kurt(x_1 \ldots x_N) = \frac{1}{N}\sum_{j=1}^{N}\left[\frac{(x_j - \bar{x})}{\sigma}\right]^4$; subtracting 3 is not necessary.

Weibull Amplitude Statistics

Weibull amplitudes are parsed using one amplitude per wave encounter, which are determined from mean crossings in the lowpass time history. If the filter cutoff frequency is zero, mean crossings in the raw time history are used. The max and min amplitudes have the mean subtracted. Amplitude parsing will stop at the last scan or when the number of amplitudes reaches the number of scans / 10. This analysis is performed on channels selected in the optional configuration file. The amplitude types were discussed in the introduction.

2-parameter Weibull distributions are calculated using the linear regression and moment methods as described in "High Volume Data Analysis Suite", NSWCCD-65-TR–2005/17. (Hildstrom, 2006, p. 32)

A great discussion of the Weibull distribution can be found on Wikipedia's website. (Weibull, n.d.) The AE computed value of beta corresponds to the shape parameter k . The AE computed characteristic value corresponds to the scale parameter λ . The computed Weibull distribution parameters can be entered into the cumulative distribution function (CDF) to solve for expected and extreme values.

The 2-parameter Weibull CDF is $F(x;k,\lambda) = 1 - e^{-\left(\frac{x}{\lambda}\right)^k}$. (Weibull, n.d.) Solving for x in the CDF yields the equation $x = \lambda\left(-\ln(1 - F(x;k,\lambda))\right)^{\left(\frac{1}{k}\right)}$.

Fast Fourier Transform (FFT)

The FFT implementation used is also straight out of Numerical Recipes in C, section 12.2; function four1. (Press, 1999, p. 504) The FFT requires the complex input to be a power of 2, but zero padding or power of 2 sized segments can be used to overcome this limitation. This algorithm is of order $Nlog_2(N)$.

The FFT of a real function h(t) is $FFT(h(t)) = H(f) = \int_{-\infty}^{\infty} h(t)e^{(2\pi ift)}dt$; equation 12.0.1. (Press, 1999, p. 496)

Power Spectral Density (PSD)

The PSD is computed by $P_h(f) = 2|H(f)|^2$; equation 12.0.14. (Press, 1999, p. 498) A number of smaller segments, with 50% overlap, are used to create an average PSD of lower resolution to represent the entire time history. The mean of each segment is subtracted before analysis to prevent ultra-low frequencies from aliasing to DC. (Hildstrom, 2006, p. 26) The size of these FFT segments must be an integer power of 2, decided at compile or run time, and entered into the global variable fftsize or the optional configuration file aeconfig-numchan-scanrate.csv. This analysis is performed on channels selected in the optional configuration file.

The PSDs are output to an SCL file to save storage space and CPU cycles. The output file is named the same as the input data file with -psd appended before the file extension. Example: inputfile.scl would yield the PSD output file inputfile-psd.scl, which can be loaded directly into IGOR or LabView. The resulting PSD ranges from 0 Hz (DC) to the Nyquist frequency (scanrate/2 Hz).

Response Amplitude Operator (RAO)

The RAO is computed by $RAO_{(channel)}(f) = \sqrt{\frac{[PSD_{(channel)}(f)]}{[PSD_{(wave\,channel)}(f)]}}$. (Hildstrom, 2006, p. 29) The wave channel is selected at compile time or run time and entered into the global variable wavechannel or the optional configuration file aeconfig-numchan-scanrate.csv. The f=0 DC component of the RAO is set to 0 because the DC components of the PSDs are nearly 0 and the result is meaningless. This analysis is performed on channels selected in the optional configuration file.

The RAOs are output to an SCL file to save storage space and CPU cycles. The output file is named the same as the input data file with -rao appended before the file extension. Example: inputfile.scl would yield the RAO output file inputfile-rao.scl, which can be loaded directly into IGOR or LabView. The resulting RAOs range from 0 Hz (DC) to the Nyquist frequency (scanrate/2 Hz) in encounter frequency.

FFT, Optimal (Wiener), Lowpass Filtering

The time domain statistics designated lowpass use a lowpass filtered version of the original raw data. This lowpass filter is implemented with the FFT and filters entirely in the frequency domain. This method, also known as Optimal or Wiener filtering, comes straight out of Numerical Receipes in C, sections 13.3 and 13.5. (Press, 1999, p. 547 p. 558)

This filtering method uses the following basic steps:

1. Perform an FFT on the raw data, which converts the raw data in the time domain to spectrum data in the frequency domain.
2. Multiply the spectrum data by a filter function in the frequency domain. This passes, removes, or amplifies desired frequencies in the frequency domain.
3. Perform an inverse FFT on the spectrum data, which converts the spectrum data in the frequency domain back to filtered data in the time domain.

This method is implemented for lowpass filtering only; the cutoff frequency is defined at compile time or run time in the global variable cutofffrequency or in the optional configuration file aeconfig-numchan-scanrate.csv. If this variable is set to 0, no filtering is performed. The cutoff frequency is used to define a very simple filter function that is 1 (pass band) from 0 to the cutoff frequency and 0 (stop band) from the cutoff frequency to the Nyquist frequency. This sharp filter works well for non-impulsive data. Single-point drop-outs in the data (max or min impulse events) are lowpass filtered well if the cutoff frequency is kept less than $\frac{scanrate * stdev}{(max - mean)}$ and the $\frac{(max - mean)}{stdev}$ ratio is less than about 100. The lower the $\frac{(max - mean)}{stdev}$ ratio, the better the lowpass filter can eliminate single-sample impulses for a given cutoff frequency. The original raw data are padded up to the next power of 2 to satisfy the FFT implementation. The pad is removed after filtering to yield a filtered version of the raw data. This algorithm is of order $2N\log_2(N)$.

Difference Highpass Filtering

The time domain statistics designated highpass use a highpass filtered version of the original raw data. The FFT filtering is very accurate. For real signals, adding the FFT lowpass filtered time history and the FFT highpass filtered time history yields the original raw time history to near perfection. This code takes advantage of the FFT lowpass filter accuracy, so it computes the highpass filtered time history by subtracting the FFT lowpass filtered time history from the original raw time history, which saves many calculations and works very well. The algorithm can be represented by

$Data_{highpass}(t) = Data_{raw}(t) - Data_{lowpass}(t)$.

This algorithm is of order N. The FFT filter algorithm is of order $2N\log_2(N)$, so the two filter algorithm would be $2N\log_2(N) + 2N\log_2(N)$ for both lowpass and highpass FFT filtering. However, since the difference between raw and FFT lowpass is used to yield the highpass version, the two filter algorithm is of order $2N\log_2(N) + N$, which is much faster and saves $2N\log_2(N) - N$ computations.

Optional Configuration File aeconfig-numchan-scanrate.csv

The optional configuration file, aeconfig-numchan-scanrate.csv, is a standard comma-delimited ASCII csv file. It can be loaded with any text editor or Microsoft Excel. If AE cannot load aeconfig-numchan-scanrate.csv from the working directory, it creates one containing the default settings used during analysis. If the contents of the configuration file are modified, the new settings will be used next time AE is executed. This setup allows SCL files with different numbers of channels and scan rates to be analyzed together. The table 2 shows an example aeconfig-4-100.csv loaded into a spreadsheet program; cells highlighted in blue may be modified by the user.

fftsize	2048			
wavechannel	0			
cutofffrequency	0			
histogrambins	20			
modelscaleratio	1			
numchan	4			
scanrate	100			
channelnumber	0	1	2	3
name	ch0	ch1	ch2	ch3
scalefactor	1	1	1	1
offset	0	0	0	0
sn failure 1e0 cycle	320	320	320	320
sn failure 1e1 cycles	300	300	300	300
sn failure 1e2 cycles	260	260	260	260
sn failure 1e3 cycles	200	200	200	200
sn failure 1e4 cycles	150	150	150	150
sn failure 1e5 cycles	120	120	120	120
sn failure 1e6 cycles	90	90	90	90
sn failure 1e7 cycles	80	80	80	80
sn failure 1e8 cycles	70	70	70	70
performrainflowanalysis	1	1	1	1
performmmmanalysis	1	1	1	1
performpsdanalysis	1	1	1	1
performraoanalysis	1	1	1	1
performweibullanalysis	1	1	1	1
emptyscalevalue	-320	-320	-320	-320
yellowlinevalue	250	250	250	250
redlinevalue	300	300	300	300
fullscalevalue	400	400	400	400
piersidezerovalue	50	50	50	50
channelinformation	extra measurement info	extra measurement info	extra measurement info	extra measurement info
prescalefactorunits	Volts x Cal	Volts x Cal	Volts x Cal	Volts x Cal
postscalefactorunits	Other EUs	Other EUs	Other EUs	Other EUs
channellocation	frame 6 bulkhead 4	frame 6 bulkhead 4	frame 6 bulkhead 4	frame 6 bulkhead 4
virtualchanneltype	0	0	0	0
virtualchannelname	vc0	vc1	vc2	vc3
virtualchannelprescalefactor	1	1	1	1
virtualchannelpostscalefactor	1	1	1	1
vctype1rect45rosette0degCH	0	0	0	0
vctype1rect45rosette0degOffset	0	0	0	0
vctype1rect45rosette45degCH	1	1	1	1
vctype1rect45rosette45degOffset	0	0	0	0
vctype1rect45rosette90degCH	2	2	2	2
vctype1rect45rosette90degOffset	0	0	0	0
vctype1rect45rosetteOffsetIsMean	0	0	0	0
vctype2add2CH	0	0	0	0
vctype2add2CH	1	1	1	1
vctype3sub2CH	0	0	0	0
vctype3sub2CH	1	1	1	1
vctype4avg2CH	0	0	0	0
vctype4avg2CH	1	1	1	1
vctype5suboffsetCH	0	0	0	0
vctype5suboffsetValue	0	0	0	0
vctype6avg3CH	0	0	0	0
vctype6avg3CH	1	1	1	1
vctype6avg3CH	2	2	2	2

Table 2: aeconfig-4-100.csv default configuration file

Table 3 describes the global, real channel, and shared variables; the shared variables are used to analyze the real and virtual channels in that column and are highlighted in yellow. Virtual channel variables are described in the next section.

Row Label	Type	Description
fftsize	integer	the size of the FFT segments used during PSD/RAO analysis; power of 2
wavechannel	integer	the channel containing the wave height time history used during RAO analysis
cutofffrequency	real	the cutoff/crossover frequency used for lowpass/highpass filtering; 0 does not filter
histogrambins	integer	the number of histogram bins used for rainflow full-cycle histogram output
modelscaleratio	real	The scale ratio of the model like 30 for 1:30 or 1 for 1:1 full scale
numchan	integer	echoed from SCL input file
scanrate	integer	echoed from SCL input file
channelnumber	integer	real channel number of SCL data channel
name	text	real channel name of SCL data channel
scalefactor	real	scale factor applied to SCL data after loading
offset	real	offset subtracted from SCL data before scale factor after loading
sn failure 1e0 cycle	real	failure magnitude at 1e0 cycle
sn failure 1e1 cycles	real	failure magnitude at 1e1 cycles
sn failure 1e2 cycles	real	failure magnitude at 1e2 cycles
sn failure 1e3 cycles	real	failure magnitude at 1e3 cycles
sn failure 1e4 cycles	real	failure magnitude at 1e4 cycles
sn failure 1e5 cycles	real	failure magnitude at 1e5 cycles
sn failure 1e6 cycles	real	failure magnitude at 1e6 cycles
sn failure 1e7 cycles	real	failure magnitude at 1e7 cycles
sn failure 1e8 cycles	real	failure magnitude at 1e8 cycles
performrainflowanalysis	integer	1 is perform specified analysis, 0 is do not perform specified analysis
performmmmanalysis	integer	1 is perform specified analysis, 0 is do not perform specified analysis
performpsdanalysis	integer	1 is perform specified analysis, 0 is do not perform specified analysis
performraoanalysis	integer	1 is perform specified analysis, 0 is do not perform specified analysis
performweibullanalysis	integer	1 is perform specified analysis, 0 is do not perform specified analysis
emptyscalevalue	real	absolute minimum possible value (binary 0 or min from DAQ)
yellowlinevalue	real	yellow line warning value
redlinevalue	real	red line warning value
fullscalevalue	real	absolute maximum possible value (binary 11111 or max from DAQ)
piersidezerovalue	real	pier side zero value
channelinformation	text	extra channel information like gauge name or number
prescalefactorunits	text	units of measurement before scale factor is applied
postscalefactorunits	text	units of measurement after scale factor is applied
channellocation	text	channel location like frame and bulkhead numbers

Table 3: Global, Real Channel, and Shared Configuration Variables

Virtual Channels

Virtual channels are time histories derived from recorded time histories that were not directly recorded during data acquisition. Two of the most common uses for virtual channels are rosette analysis (Strain Gage Rosettes, 2005), for determining principle strain from three gauges, and combining two deck edge strains, for determining vertical or lateral bending.

Virtual channels are defined in the optional configuration file. The maximum number of virtual channels is equal to the number of real channels in the input data file. For example, if the SCL file contains 4 real data channels, a maximum of 4 virtual channels may be computed. The primary reason for this limit is to keep the single configuration file simple. AE loads the SCL file, loads the configuration file, determines if virtual channels are to be computed, computes the new time histories, appends them to the end of the data file in memory, and duplicates key configuration file values as if the original data file also contained the virtual channels. This treats the original real data channels and the virtual channels as if they were all in the original SCL file from the beginning. For example, if a 4-channel SCL file calls for 4 virtual channels in aeconfig-4-100.csv, all outputs will reference the new numchan of 8 (4 real + 4 virtual).

Each real channel's column in the configuration file may also define a new virtual channel. The S-N curve, analysis selection, red line values, and descriptions of the real channel column are duplicated and used for the corresponding virtual channel analysis. Table 4 describes the variables used for virtual channel analysis.

Row Label	Type	Description
virtualchanneltype	integer	0-6: type 0 does not create a new virtual channel using this column
virtualchannelname	text	name of the new virtual channel
virtualchannelprescalefactor	real	scale factor applied to channels before virtual channel combination
virtualchannelpostscalefactor	real	scale factor applied to virtual channel after combination; like strain to stress
vctype1rect45rosette0degCH	integer	far left gauge in a 0, 45, 90 rectangular rosette; real channel number
vctype1rect45rosette0degOffset	real	far left gauge in a 0, 45, 90 rectangular rosette; offset to subtract
vctype1rect45rosette45degCH	integer	middle gauge in a 0, 45, 90 rectangular rosette; real channel number
vctype1rect45rosette45degOffset	real	middle gauge in a 0, 45, 90 rectangular rosette; offset to subtract
vctype1rect45rosette90degCH	integer	far right gauge in a 0, 45, 90 rectangular rosette; real channel number
vctype1rect45rosette90degOffset	real	far right gauge in a 0, 45, 90 rectangular rosette; offset to subtract
vctype1rect45rosetteOffsetIsMean	integer	subtract mean of real channels instead of offsets
vctype2add2CH	integer	add two channels; first real channel number
vctype2add2CH	integer	add two channels; second real channel number
vctype3sub2CH	integer	subtract two channels; first real channel number
vctype3sub2CH	integer	subtract two channels; second real channel number
vctype4avg2CH	integer	average two channels; first real channel number
vctype4avg2CH	integer	average two channels; second real channel number
vctype5suboffsetCH	integer	subtract an offset from a channel; real channel number
vctype5suboffsetValue	real	subtract an offset from a channel; offset to subtract
vctype6avg3CH	integer	average three channels; first real channel number
vctype6avg3CH	integer	average three channels; second real channel number
vctype6avg3CH	integer	average three channels; third real channel number

Table 4: Virtual Channel Configuration Variables

Helper Applications

Most of these helper applications work from the command line, from the Windows context menu, and from the Windows Desktop using drag and drop. Drag and drop is when one or more selected files are dragged (click, hold, move) to an executable program, like aerunner.exe, and dropped (release). When using drag and drop or context menu with command line programs like AE, the default working directory is the XP User's home directory, usually: C:\Documents and Settings\Owner or C:\Documents and Settings\Administrator. This means that configuration files and output files will appear in that directory. When executing command line programs from the Command Window (Start->Run->cmd), the working directory is user selectable (cd c:\somedir) and output files will appear there.

aerunner

This program accepts one or more scl input files and executes AE on each of them. This is useful for analyzing large numbers of data files. Example: "c:\datafolder>aerunner datafilename.scl datafilename2.scl datafilename3.scl". Drag and drop is much faster and easier.

ascii2scl

This program accepts one or more comma or tab-delimited ASCII csv input files, with 3 lines of skipped header, and converts them to scl binary files, with a scan rate of 1, that can be read by IGOR or LabView. Example: "c:\datafolder>ascii2scl datafilename-psd.csv", which will create datafilename-psd.scl.

daqlogpoller

The AE could be executed directly by the data acquisition software, but that would consume valuable CPU cycles on the data acquisition computer that are necessary for flawless data acquisition. The system would operate much smoother and faster if the analysis were performed on a separate CPU and computer. This small helper application, called daqlogpoller, polls the data acquisition log file and the analysis log file at a given time interval like 1 minute or 5 minutes. The data acquisition log and analysis log files are both lists of file or path names where each line ends in a newline character '\n'. If any line in the data acquisition log (daqlog.txt) does not exist in the analysis log (analysislog.txt), the AE application is executed on the data file and the filename is appended to the analysis log. This command line program takes no arguments.

loadsclfilewithigor

This program accepts a single scl input file and loads it with WaveMetrics IGOR Pro. This program only works with drag and drop or double-click. Command line local file references will not work. The easiest way to configure this is to add the folder containing Igor.exe to the system path and tell Windows Explorer to open .scl files with loadsclfilewithigor.exe. The macro function loadGenericSCLFileSP4BHBF must be installed in IGOR, which is part of the nswccd65igor macro package. (NSWCCD, 2006) Then double-clicking on scl files will load them in IGOR's working folder; or the root folder if IGOR was not already open.

scl2ascii

This program accepts one or more scl input files and converts them to comma-delimited ASCII text files that can be read by Microsoft Excel, notepad, or IGOR. Example: "c:\datafolder>scl2ascii datafilename-psd.scl", which will create datafilename-psd.csv.

sclconcat

This program accepts one or more scl input files and concatenates them together into a single new larger scl file. No processing is performed at the splice points. This is useful for splicing together runs from the same condition. Example: "c:\datafolder>sclconcat datafilename.scl datafilename2.scl datafilename3.scl", which will create datafilename-concat.scl. An error message will be displayed if the scan rate or number of channels differs from the first file.

sclscanrate

This program accepts one or more scl input files and changes the scan rate of each to a new scan rate defined in the the global variable newscanrate or an optional configuration file newscanrate.csv. No other processing is performed. This is useful in the conversion from ASCII to scl because the scan rate will have the default value of 1. Example: "c:\datafolder>sclscanrate datafilename.scl datafilename2.scl datafilename3.scl", which will create datafilename-scanrate.scl, datafilename2-scanrate.scl, and datafilename3-scanrate.scl. The configuration file newscanrate.csv contains a single line of text that is the desired new scanrate.

slae

This program accepts a single statslog.csv input file and performs statistical analysis on the time history statistics. These statistics help determine the quality and validity of recorded measurements and are saved in the comma-delimited ASCII csv file statsstatslog.csv. Example: "c:\datafolder>slae statslog.csv", which will create statsstatslog.csv. Detailed descriptions of the columns in this file are in appendix B.

AEGUI Graphical Monitoring Application

The AEGUI program is a graphical user interface designed to monitor AE output. The interface runs on Microsoft Windows and is built with the FLTK GUI library. The AEGUI performs the following monitoring functions for three different data acquisition systems:

- Cumulative fatigue damage
- Current damage rate
- Basic variance, max, min, and damage statistics
- Warning levels with color changing indicators
- Trend generation across all analyzed data files using:
  - mean
  - max
  - min
  - range
  - variance
  - skewness
  - kurtosis
  - damage
  - psd peak frequency
  - warning flags
  - mean-pier side zero value
- Plot generation of selected channels using last recorded data file
- Gauge location drawing display
- Automatic PDF Report Generation

This program uses a minimum of three AE output files as inputs: aeconfig-numchan-scanrate.csv, damagesum-numchan-scanrate.csv, and statslog.csv. Ideally there is one configuration file and one damage file for each data acquisition system. These files are reloaded at a set interval of 1 minute, which insures that the display is updated within 1 minute of new AE analysis results. The numchan-scanrate combinations for each data acquisition system are hard-coded and set at compile time. The drawings for each measurement, which are optional, should follow the naming convention drawing-numchan-scanrate-channelnumber.pdf. An example drawing for channel 0 from a 79 channel 100 Hz SCL file should be named drawing-79-100-0.pdf. The automatic PDF report, SystemReport.pdf, can contain hundreds of plots and may take several minutes to generate. Screen shots of AEGUI running are shown in appendix C.

References

ASTM E 1049-85. (Reapproved 2005). Standard practices for cycle counting in fatigue analysis. ASTM International. Retrieved November 13, 2006 from http://www.astm.org

Downing, S. D., Socie, D. F. (1982). Simple rainflow counting algorithms. International Journal of Fatigue, Volume 4, Issue 1, January, 31-40. Retrieved July 1, 2006 from http://www.sciencedirect.com/

Fatigue (material). (n.d.). Retrieved July 1, 2006, from http://en.wikipedia.org/wiki/Fatigue_%28material%29

Hildstrom, G. (2006). High Volume Data Analysis Suite (HVDAS) 2006-04. Navy Technical Report NSWCCD-65-TR-2005/17. West Bethesda, MD: Naval Surface Warfare Center Carderock Division. Retrieved July 1, 2006 from http://www.hildstrom.com/publications/publicdocs/HVDAS-R0517-Hildstrom-2006-04.pdf

NSWCCD Code 65 IGOR Macros. (2006). Retrieved October 23, 2006, from http://www.hildstrom.com/projects/nswccd65igor/

Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (1999). Numerical Recipes in C: The Art of Scientific Computing Second Edition. Cambridge, UK: Cambridge University Press. Retrieved July 1, 2006 from http://www.nrbook.com/a/bookcpdf.html

Rainflow counting algorithm. (n.d.). Retrieved July 1, 2006, from http://en.wikipedia.org/wiki/Rainflow-counting_algorithm

Schluter, L. (1991). Programmer's Guide for LIFE2's Rainflow Counting Algorithm. Sandia Report SAND90-2260. Albuquerque, NM: Sandia National Laboratories. Retrieved July 1, 2006 from http://www.sandia.gov/wind/topical.htm

String Gage Rosettes. (2005). Vishay Micro-Measurements Tech Note TN-515. Retrieved November 27, 2006 from http://www.vishay.com/docs/11065/_vmr-tc0.pdf

Weibull distribution. (n.d.). Retrieved November 2, 2006, from http://en.wikipedia.org/wiki/Weibull_distribution

Appendix

A: AE Time-History Statistics Output File statslog.csv

The statslog.csv output file is a standard ASCII comma-delimited text file with one line of column headers followed by the data. The columns are described in the following tables.

Column Label	Type	Description
filename	text	input data file name
numchan	integer	number of channels from the input data file (real+virtual)
virtualnumchan	integer	number of virtual channels from the input data file
scanrate	real	scan rate of the input data file
analysisdate	text	date and time the file was analyzed
analysistime	integer	system time the file was analyzed
numscans	integer	number of scans in the input data file
seconds	real	number of seconds recorded in the input data file
version	integer	version of the AE code in YYYYMMDDHHMM format
fftsize	integer	fftsize used for PSD and RAO calculation
wavechannel	integer	channel used as the wave channel for RAO
cutofffrequency	real	cutoff frequency used in digital filtering
channel	integer	channel number corresponding to the current row's statistics
cal	real	cal factor used on the current channel
scalefactor	real	scale factor defined in optional configuration file
offset	real	offset defined in optional configuration file
channelname	text	channel name defined in optional configuration file
mean	real	mean of the unfiltered raw data channel
max	real	maximum of the unfiltered raw data channel
min	real	minimum of the unfiltered raw data channel
variance	real	variance of the unfiltered raw data channel
skewness	real	skewness of the unfiltered raw data channel
kurtosis	real	kurtosis of the unfiltered raw data channel
meanlp	real	mean of the lowpass filtered data channel
maxlp	real	maximum of the lowpass filtered data channel
minlp	real	minimum of the lowpass filtered data channel
variancelp	real	variance of the lowpass filtered data channel
skewnesslp	real	skewness of the lowpass filtered data channel
kurtosislp	real	kurtosis of the lowpass filtered data channel
meanhp	real	mean of the highpass filtered data channel
maxhp	real	maximum of the highpass filtered data channel
minhp	real	minimum of the highpass filtered data channel
variancehp	real	variance of the highpass filtered data channel
skewnesshp	real	skewness of the highpass filtered data channel
kurtosishp	real	kurtosis of the highpass filtered data channel
num extrema	integer	number of extrema detected for rainflow analysis
num cycles	integer	number of cycles counted by rainflow analysis
damage	real	damage accumulated, C, for the current file and channel
meancycle	real	mean of the rainflow-detected cycles
maxcycle	real	maximum of the rainflow-detected cycles
mincycle	real	minimum of the rainflow-detected cycles
variancecycles	real	variance of the rainflow-detected cycles
skewnesscycles	real	skewness of the rainflow-detected cycles
kurtosiscycles	real	kurtosis of the rainflow-detected cycles

Table 5: statslog.csv file info, mmm statistics, and rainflow statistics column descriptions

Column Label	Type	Description
rawmaxnumberofweibullamplitudes	integer	number of amplitudes detected for Weibull analysis
rawmaxamplitudemean	real	amplitude mean
rawmaxamplitudemax	real	amplitude max
rawmaxamplitudemin	real	amplitude min
rawmaxamplitudevariance	real	amplitude variance
rawmaxamplitudeskew	real	amplitude skew
rawmaxamplitudekurtosis	real	amplitude kurtosis
rawmaxlrbeta	real	Weibull linear regression beta
rawmaxlryint	real	Weibull linear regression y-intercept
rawmaxlrcharval	real	Weibull linear regression characteristic value
rawmaxlrcorrelation	real	Weibull linear regression correlation
rawmaxmmbeta	real	Weibull moment method beta
rawmaxmmyint	real	Weibull moment method y-intercept
rawmaxmmcharval	real	Weibull moment method characteristic value
rawminnumberofweibullamplitudes	integer	number of amplitudes detected for Weibull analysis
rawminamplitudemean	real	amplitude mean
rawminamplitudemax	real	amplitude max
rawminamplitudemin	real	amplitude min
rawminamplitudevariance	real	amplitude variance
rawminamplitudeskew	real	amplitude skew
rawminamplitudekurtosis	real	amplitude kurtosis
rawminlrbeta	real	Weibull linear regression beta
rawminlryint	real	Weibull linear regression y-intercept
rawminlrcharval	real	Weibull linear regression characteristic value
rawminlrcorrelation	real	Weibull linear regression correlation
rawminmmbeta	real	Weibull moment method beta
rawminmmyint	real	Weibull moment method y-intercept
rawminmmcharval	real	Weibull moment method characteristic value
rawp2pnumberofweibullamplitudes	integer	number of amplitudes detected for Weibull analysis
rawp2pamplitudemean	real	amplitude mean
rawp2pamplitudemax	real	amplitude max
rawp2pamplitudemin	real	amplitude min
rawp2pamplitudevariance	real	amplitude variance
rawp2pamplitudeskew	real	amplitude skew
rawp2pamplitudekurtosis	real	amplitude kurtosis
rawp2plrbeta	real	Weibull linear regression beta
rawp2plryint	real	Weibull linear regression y-intercept
rawp2plrcharval	real	Weibull linear regression characteristic value
rawp2plrcorrelation	real	Weibull linear regression correlation
rawp2pmmbeta	real	Weibull moment method beta
rawp2pmmyint	real	Weibull moment method y-intercept
rawp2pmmcharval	real	Weibull moment method characteristic value

Table 6: statslog.csv raw Weibull column descriptions

Column Label	Type	Description
lpmaxnumberofweibullamplitudes	integer	number of amplitudes detected for Weibull analysis
lpmaxamplitudemean	real	amplitude mean
lpmaxamplitudemax	real	amplitude max
lpmaxamplitudemin	real	amplitude min
lpmaxamplitudevariance	real	amplitude variance
lpmaxamplitudeskew	real	amplitude skew
lpmaxamplitudekurtosis	real	amplitude kurtosis
lpmaxlrbeta	real	Weibull linear regression beta
lpmaxlryint	real	Weibull linear regression y-intercept
lpmaxlrcharval	real	Weibull linear regression characteristic value
lpmaxlrcorrelation	real	Weibull linear regression correlation
lpmaxmmbeta	real	Weibull moment method beta
lpmaxmmyint	real	Weibull moment method y-intercept
lpmaxmmcharval	real	Weibull moment method characteristic value
lpminnumberofweibullamplitudes	integer	number of amplitudes detected for Weibull analysis
lpminamplitudemean	real	amplitude mean
lpminamplitudemax	real	amplitude max
lpminamplitudemin	real	amplitude min
lpminamplitudevariance	real	amplitude variance
lpminamplitudeskew	real	amplitude skew
lpminamplitudekurtosis	real	amplitude kurtosis
lpminlrbeta	real	Weibull linear regression beta
lpminlryint	real	Weibull linear regression y-intercept
lpminlrcharval	real	Weibull linear regression characteristic value
lpminlrcorrelation	real	Weibull linear regression correlation
lpminmmbeta	real	Weibull moment method beta
lpminmmyint	real	Weibull moment method y-intercept
lpminmmcharval	real	Weibull moment method characteristic value
lpp2pnumberofweibullamplitudes	integer	number of amplitudes detected for Weibull analysis
lpp2pamplitudemean	real	amplitude mean
lpp2pamplitudemax	real	amplitude max
lpp2pamplitudemin	real	amplitude min
lpp2pamplitudevariance	real	amplitude variance
lpp2pamplitudeskew	real	amplitude skew
lpp2pamplitudekurtosis	real	amplitude kurtosis
lpp2plrbeta	real	Weibull linear regression beta
lpp2plryint	real	Weibull linear regression y-intercept
lpp2plrcharval	real	Weibull linear regression characteristic value
lpp2plrcorrelation	real	Weibull linear regression correlation
lpp2pmmbeta	real	Weibull moment method beta
lpp2pmmyint	real	Weibull moment method y-intercept
lpp2pmmcharval	real	Weibull moment method characteristic value

Table 7: statslog.csv lowpass Weibull column descriptions

Column Label	Type	Description
hpmaxnumberofweibullamplitudes	integer	number of amplitudes detected for Weibull analysis
hpmaxamplitudemean	real	amplitude mean
hpmaxamplitudemax	real	amplitude max
hpmaxamplitudemin	real	amplitude min
hpmaxamplitudevariance	real	amplitude variance
hpmaxamplitudeskew	real	amplitude skew
hpmaxamplitudekurtosis	real	amplitude kurtosis
hpmaxlrbeta	real	Weibull linear regression beta
hpmaxlryint	real	Weibull linear regression y-intercept
hpmaxlrcharval	real	Weibull linear regression characteristic value
hpmaxlrcorrelation	real	Weibull linear regression correlation
hpmaxmmbeta	real	Weibull moment method beta
hpmaxmmyint	real	Weibull moment method y-intercept
hpmaxmmcharval	real	Weibull moment method characteristic value
hpminnumberofweibullamplitudes	integer	number of amplitudes detected for Weibull analysis
hpminamplitudemean	real	amplitude mean
hpminamplitudemax	real	amplitude max
hpminamplitudemin	real	amplitude min
hpminamplitudevariance	real	amplitude variance
hpminamplitudeskew	real	amplitude skew
hpminamplitudekurtosis	real	amplitude kurtosis
hpminlrbeta	real	Weibull linear regression beta
hpminlryint	real	Weibull linear regression y-intercept
hpminlrcharval	real	Weibull linear regression characteristic value
hpminlrcorrelation	real	Weibull linear regression correlation
hpminmmbeta	real	Weibull moment method beta
hpminmmyint	real	Weibull moment method y-intercept
hpminmmcharval	real	Weibull moment method characteristic value
hpp2pnumberofweibullamplitudes	integer	number of amplitudes detected for Weibull analysis
hpp2pamplitudemean	real	amplitude mean
hpp2pamplitudemax	real	amplitude max
hpp2pamplitudemin	real	amplitude min
hpp2pamplitudevariance	real	amplitude variance
hpp2pamplitudeskew	real	amplitude skew
hpp2pamplitudekurtosis	real	amplitude kurtosis
hpp2plrbeta	real	Weibull linear regression beta
hpp2plryint	real	Weibull linear regression y-intercept
hpp2plrcharval	real	Weibull linear regression characteristic value
hpp2plrcorrelation	real	Weibull linear regression correlation
hpp2pmmbeta	real	Weibull moment method beta
hpp2pmmyint	real	Weibull moment method y-intercept
hpp2pmmcharval	real	Weibull moment method characteristic value
psdpeakfrequency	real	The frequency of the peak in the psd

Table 8: statslog.csv highpass Weibull and PSD column descriptions

Column Label	Type	Description
performrainflowanalysis	integer	variable echoed from the configuration file
performmmmanalysis	integer	variable echoed from the configuration file
performpsdanalysis	integer	variable echoed from the configuration file
performraoanalysis	integer	variable echoed from the configuration file
performweibullanalysis	integer	variable echoed from the configuration file
emptyscalevalue	real	variable echoed from the configuration file
yellowlinevalue	real	variable echoed from the configuration file
redlinevalue	real	variable echoed from the configuration file
fullscalevalue	real	variable echoed from the configuration file
piersidezerovalue	real	variable echoed from the configuration file
channelinformation	text	variable echoed from the configuration file
prescalefactorunits	text	variable echoed from the configuration file
postscalefactorunits	text	variable echoed from the configuration file
channellocation	text	variable echoed from the configuration file
zerovarianceflag	integer	flag set if variance is zero, which indicates a dead channel
emptyscaleflag	integer	flag set if min <= empty scale value (binary 0) empty DAQ buffer
yellowlineflag	integer	flag set if yellow line value exceeded
redlineflag	integer	flag set if red line value exceeded
fullscaleflag	integer	flag set if max >= full scale value (binary 1111) clipping
hplpvarianceflag	integer	flag set if highpass variance exceeds lowpass variance
hplpmaxflag	integer	flag set if highpass max-mean exceeds lowpass max-mean
psd99pctfrequency	real	The frequency by which 99% of the psd energy occurs
modelscaleratio	real	The scale ratio of the model like 30 for 1:30 or 1 for 1:1 full scale
fullscalescanrate	real	The full scale scan rate = acquisition scanrate/sqrt(ratio)
fullscaleseconds	real	The full scale seconds = scans/fullscalescanrate

Table 9: statslog.csv configuration echo and flag descriptions

B: SLAE Time-History-Statistics Statistics Output File statsstatslog.csv

The statsstatslog.csv output file is a standard ASCII comma-delimited text file with one line of column headers followed by the data. The statistics computed from the statistics are summarized in the three tables below.

Column Label	Type	Description
intputfile	text	The input file statslog.csv used
numfiles	integer	The number of data files' statistics used in the calculation
scanrate	real	The scan rate of the original data
channel	integer	The channel number of the original data
channelname	text	The channel name of the original data
meanofmeans	real	The mean of the means of the original data
maxofmeans	real	The max of the means
maxmeanfilename	text	The filename associated with the max mean
minofmeans	real	The min of the means
minmeanfilename	text	The filename associated with the min mean
varianceofmeans	real	The variance of the means
skewofmeans	real	The skew of the means
kurtosisofmeans	real	The kurtosis of the means
lryintofmeans	real	The linear regression y-intercept of the means
lrbetaofmeans	real	The linear regression slope of the means
lrcorrelationofmeans	real	The linear regression correlation of the means
meanofmaxes	real	The mean of the maxes of the original data
maxofmaxes	real	The max of the maxes
maxmaxfilename	text	The filename associated with the max max
minofmaxes	real	The min of the maxes
minmaxfilename	text	The filename associated with the min max
varianceofmaxes	real	The variance of the maxes
skewofmaxes	real	The skew of the maxes
kurtosisofmaxes	real	The kurtosis of the maxes
lryintofmaxes	real	The linear regression y-intercept of the maxes
lrbetaofmaxes	real	The linear regression slope of the maxes
lrcorrelationofmaxes	real	The linear regression correlation of the maxes
meanofmins	real	The mean of the mins of the original data
maxofmins	real	The max of the mins
maxminfilename	text	The filename associated with the max min
minofmins	real	The min of the mins
minminfilename	text	The filename associated with the min min
varianceofmins	real	The variance of the mins
skewofmins	real	The skew of the mins
kurtosisofmins	real	The kurtosis of the mins
lryintofmins	real	The linear regression y-intercept of the mins
lrbetaofmins	real	The linear regression slope of the mins

Table 10: statsstatslog.csv column descriptions

Column Label	Type	Description
meanofvariances	real	The mean of the variances of the original data
maxofvariances	real	The max of the variances
maxvariancefilename	text	The filename associated with the max variance
minofvariances	real	The min of the variances
minvariancefilename	text	The filename associated with the min variance
varianceofvariances	real	The variance of the variances
skewofvariances	real	The skew of the variances
kurtosisofvariances	real	The kurtosis of the variances
lryintofvariances	real	The linear regression y-intercept of the variances
lrbetaofvariances	real	The linear regression slope of the variances
lrcorrelationofvariances	real	The linear regression correlation of the variances
meanofskews	real	The mean of the skews of the original data
maxofskews	real	The max of the skews
maxskewfilename	text	The filename associated with the max skew
minofskews	real	The min of the skews
minskewfilename	text	The filename associated with the min skew
varianceofskews	real	The variance of the skews
skewofskews	real	The skew of the skews
kurtosisofskews	real	The kurtosis of the skews
lryintofskews	real	The linear regression y-intercept of the skews
lrbetaofskews	real	The linear regression slope of the skews
lrcorrelationofskews	real	The linear regression correlation of the skews
meanofkurtosi	real	The mean of the kurtosi of the original data
maxofkurtosi	real	The max of the kurtosi
maxkurtosisfilename	text	The filename associated with the max kurtosis
minofkurtosi	real	The min of the kurtosi
minkurtosisfilename	text	The filename associated with the min kurtosis
varianceofkurtosi	real	The variance of the kurtosi
skewofkurtosi	real	The skew of the kurtosi
kurtosisofkurtosi	real	The kurtosis of the kurtosi
lryintofkurtosi	real	The linear regression y-intercept of the kurtosi
lrbetaofkurtosi	real	The linear regression slope of the kurtosi
lrcorrelationofkurtosi	real	The linear regression correlation of the kurtosi
meanofcals	real	The mean of the cals of the original data
maxofcals	real	The max of the cals
maxcalfilename	text	The filename associated with the max cal
minofcals	real	The min of the cals
mincalfilename	text	The filename associated with the min cal
varianceofcals	real	The variance of the cals
skewofcals	real	The skew of the cals
kurtosisofcals	real	The kurtosis of the cals
lryintofcals	real	The linear regression y-intercept of the cals
lrbetaofcals	real	The linear regression slope of the cals
lrcorrelationofcals	real	The linear regression correlation of the cals

Table 11: statsstatslog.csv column descriptions 2

Column Label	Type	Description
meanofpsdpeakfreqs	real	The mean of the psd peak frequencies of the original data
maxofpsdpeakfreqs	real	The max of the psd peak frequencies
maxpsdpeakfreqfilename	text	The filename associated with the max psd peak frequency
minofpsdpeakfreqs	real	The min of the psd peak frequencies
minpsdpeakfreqfilename	text	The filename associated with the min psd peak frequency
varianceofpsdpeakfreqs	real	The variance of the psd peak frequencies
skewofpsdpeakfreqs	real	The skew of the psd peak frequencies
kurtosisofpsdpeakfreqs	real	The kurtosis of the psd peak frequencies
lryintofpsdpeakfreqs	real	The linear regression y-intercept of the psd peak frequencies
lrbetaofpsdpeakfreqs	real	The linear regression slope of the psd peak frequencies
lrcorrelationofpsdpeakfreqs	real	The linear regression correlation of the psd peak frequencies
meanofscalefactors	real	The mean of the scalefactors of the original data
maxofscalefactors	real	The max of the scalefactors
maxscalefactorfilename	text	The filename associated with the max scalefactor
minofscalefactors	real	The min of the scalefactors
minscalefactorfilename	text	The filename associated with the min scalefactor
varianceofscalefactors	real	The variance of the scalefactors
skewofscalefactors	real	The skew of the scalefactors
kurtosisofscalefactors	real	The kurtosis of the scalefactors
lryintofscalefactors	real	The linear regression y-intercept of the scalefactors
lrbetaofscalefactors	real	The linear regression slope of the scalefactors
lrcorrelationofscalefactors	real	The linear regression correlation of the scalefactors
meanofoffsets	real	The mean of the offsets of the original data
maxofoffsets	real	The max of the offsets
maxoffsetfilename	text	The filename associated with the max offset
minofoffsets	real	The min of the offsets
minoffsetfilename	text	The filename associated with the min offset
varianceofoffsets	real	The variance of the offsets
skewofoffsets	real	The skew of the offsets
kurtosisofoffsets	real	The kurtosis of the offsets
lryintofoffsets	real	The linear regression y-intercept of the offsets
lrbetaofoffsets	real	The linear regression slope of the offsets
lrcorrelationofoffsets	real	The linear regression correlation of the offsets

Table 12: statsstatslog.csv column descriptions 3

Column Label	Type	Description
meanofdamage	real	The mean of the damage of the original data
maxofdamage	real	The max of the damage
maxdamagefilename	real	The filename associated with the max damage
minofdamage	real	The min of the damage
mindamagefilename	real	The filename associated with the min damage
varianceofdamage	real	The variance of the damage
skewofdamage	real	The skew of the damage
kurtosisofdamage	real	The kurtosis of the damage
lryintofdamage	real	The linear regression y-intercept of the damage
lrbetaofdamage	real	The linear regression slope of the damage
lrcorrelationofdamage	real	The linear regression correlation of the damage
sumofflags	integer	The total number of flags for the current channel
lastzerovarianceflagfile	text	The last file that the named flag was set
lastemptyscaleflagfile	text	The last file that the named flag was set
lastyellowlineflagfile	text	The last file that the named flag was set
lastredlineflagfile	text	The last file that the named flag was set
lastfullscaleflagfile	text	The last file that the named flag was set
lasthplpvarianceflagfile	text	The last file that the named flag was set
lasthplpmaxflagfile	text	The last file that the named flag was set
meanofpsd99pctfreqs	real	The mean of the psd 99pct frequencies of the original data
maxofpsd99pctfreqs	real	The max of the psd 99pct frequencies
maxpsd99pctfreqfilename	text	The filename associated with the max psd 99pct frequency
minofpsd99pctfreqs	real	The min of the psd 99pct frequencies
minpsd99pctfreqfilename	text	The filename associated with the min psd 99pct frequency
varianceofpsd99pctfreqs	real	The variance of the psd 99pct frequencies
skewofpsd99pctfreqs	real	The skew of the psd 99pct frequencies
kurtosisofpsd99pctfreqs	real	The kurtosis of the psd 99pct frequencies
lryintofpsd99pctfreqs	real	The linear regression y-intercept of the psd 99pct frequencies
lrbetaofpsd99pctfreqs	real	The linear regression slope of the psd 99pct frequencies
lrcorrelationofpsd99pctfreqs	real	The linear regression correlation of the psd 99pct frequencies

Table 13: statsstatslog.csv damage and flag descriptions

C: AEGUI Screen Shots

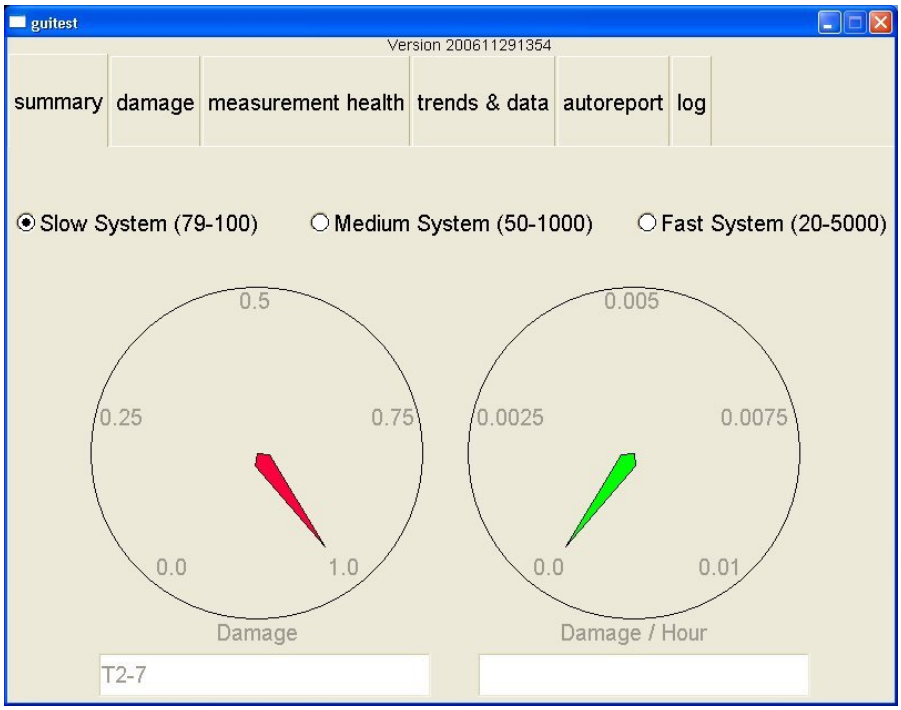

Illustration 8: AEGUI Screen Shot: summary tab

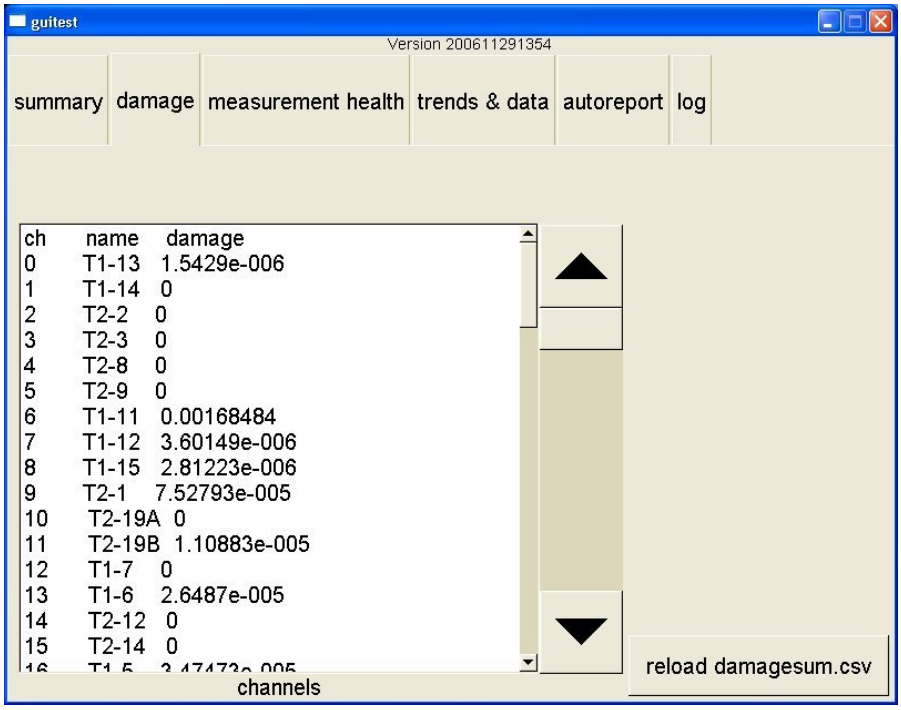

Illustration 9: AEGUI Screen Shot: damage tab

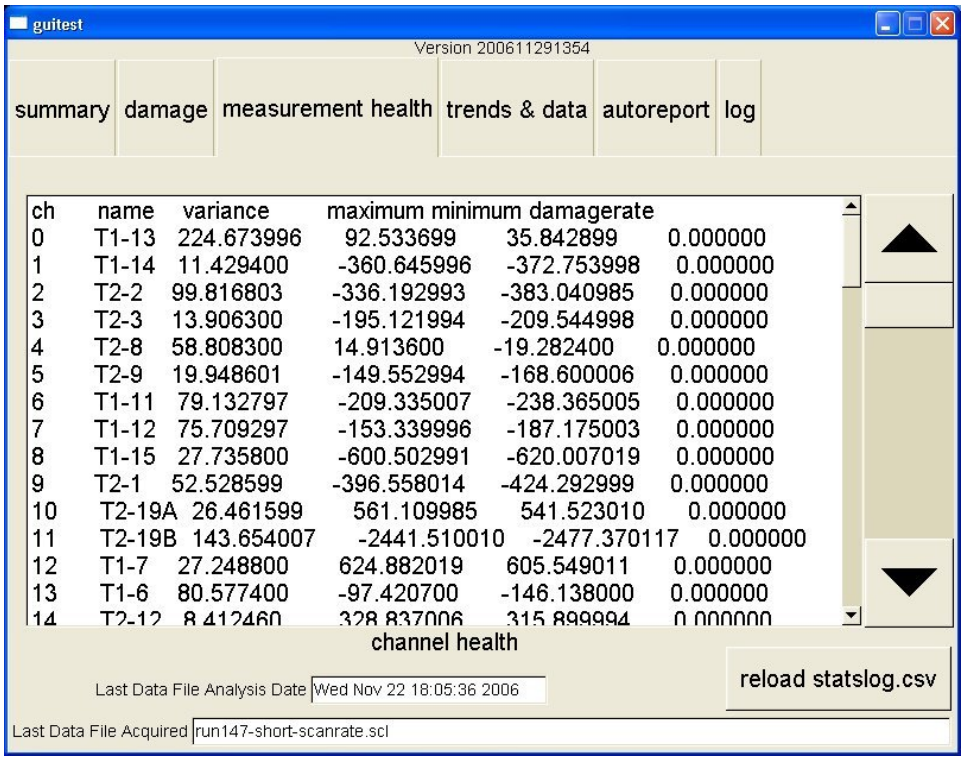

Illustration 10: AEGUI Screen Shot: measurement health tab

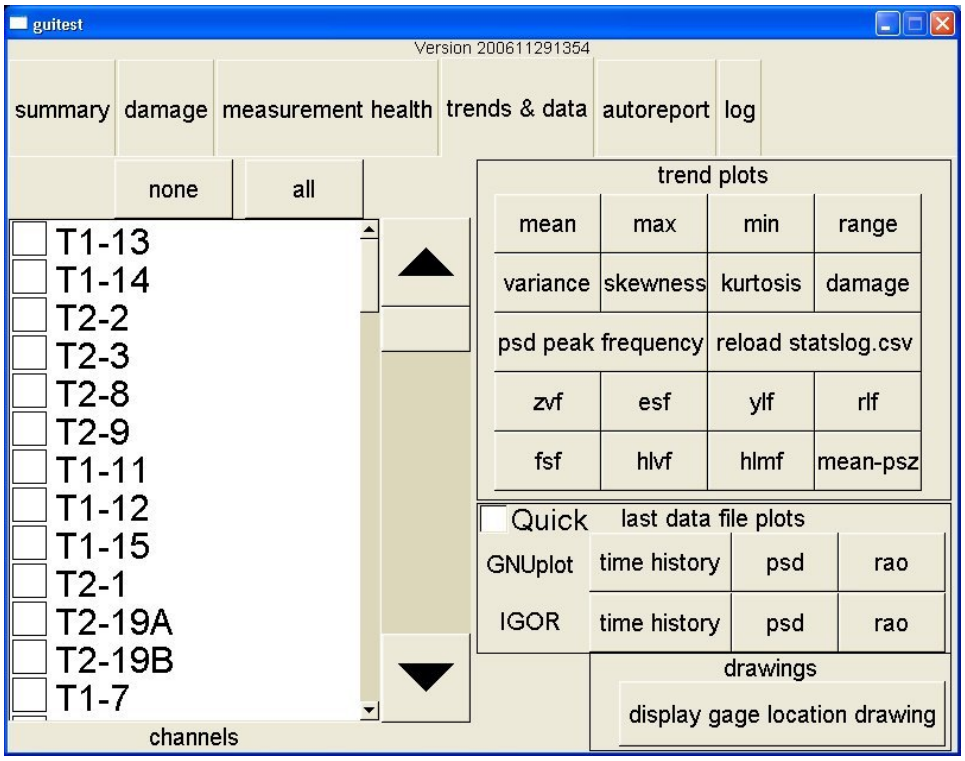

Illustration 11: AEGUI Screen Shot: trends & data tab

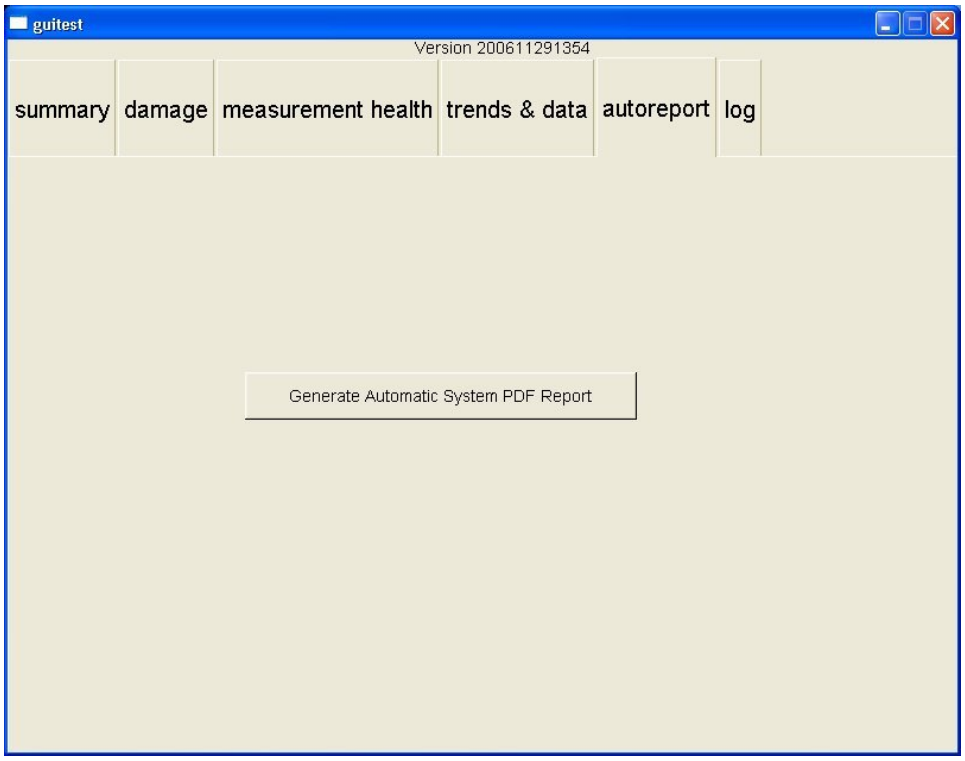

Illustration 12: AEGUI Screen Shot: autoreport tab

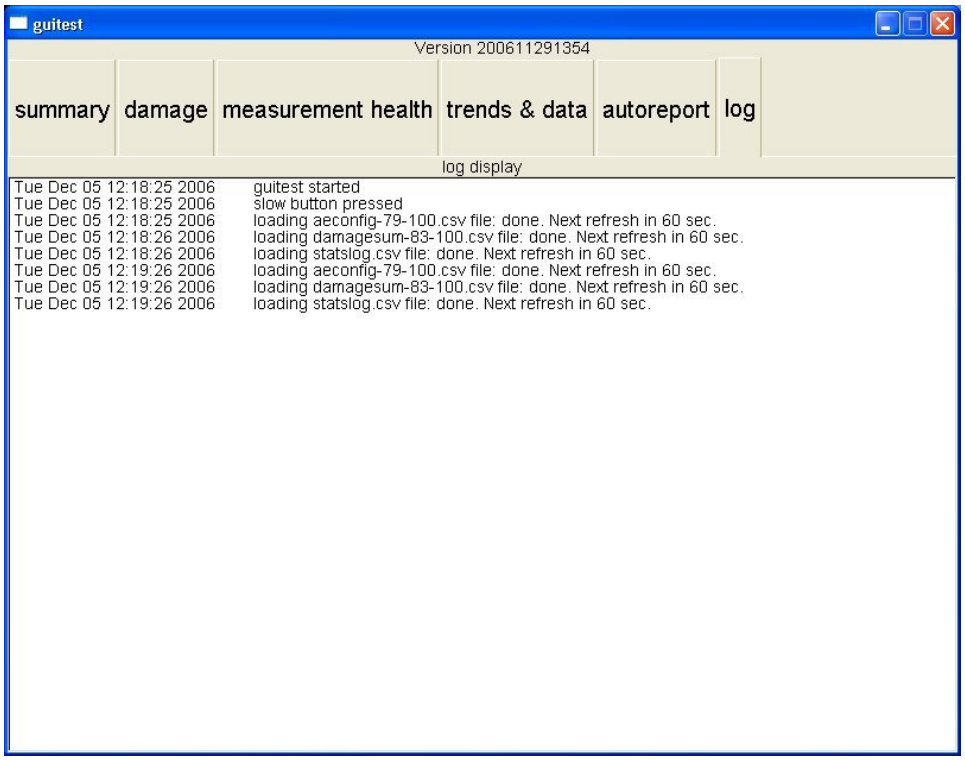

Illustration 13: AEGUI Screen Shot: log tab

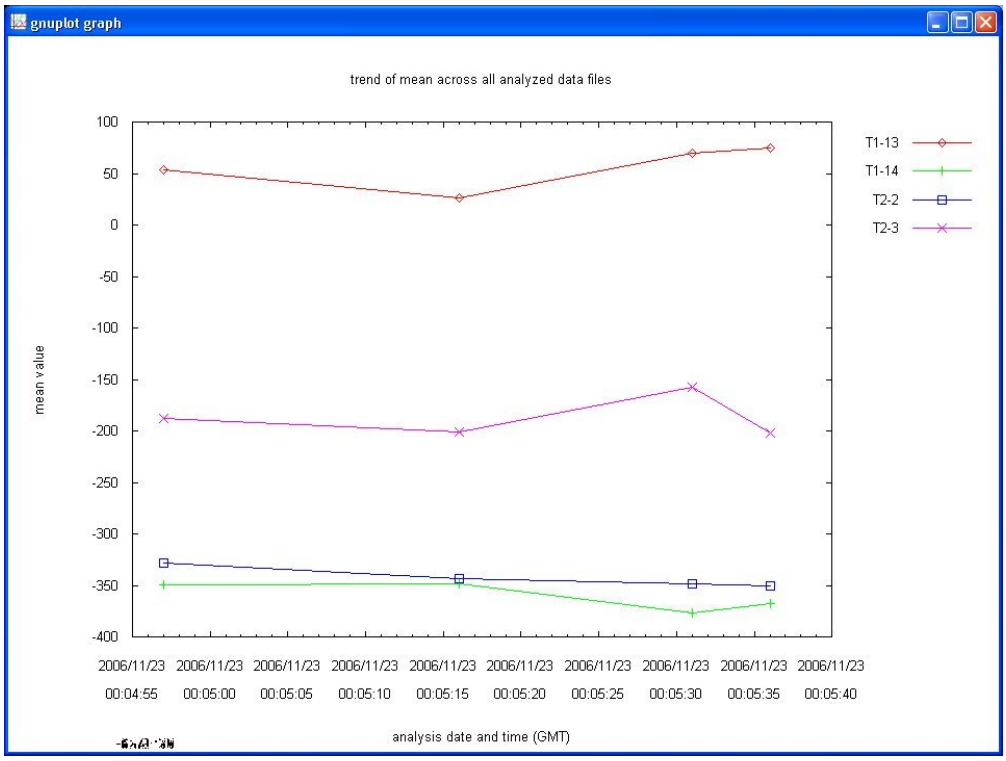

Illustration 14: AEGUI Screen Shot: GNUplot interactive window