

12-2006

## Neural Network Burst Pressure Prediction in Composite Overwrapped Pressure Vessels from Acoustic Emission Data

Seth-Andrew T. Dion

*Embry-Riddle Aeronautical University - Daytona Beach*

Follow this and additional works at: <https://commons.erau.edu/db-theses>



Part of the [Aerospace Engineering Commons](#)

---

### Scholarly Commons Citation

Dion, Seth-Andrew T., "Neural Network Burst Pressure Prediction in Composite Overwrapped Pressure Vessels from Acoustic Emission Data" (2006). *Theses - Daytona Beach*. 43.

<https://commons.erau.edu/db-theses/43>

This thesis is brought to you for free and open access by Embry-Riddle Aeronautical University – Daytona Beach at ERAU Scholarly Commons. It has been accepted for inclusion in the Theses - Daytona Beach collection by an authorized administrator of ERAU Scholarly Commons. For more information, please contact [commons@erau.edu](mailto:commons@erau.edu).

**NEURAL NETWORK BURST PRESSURE PREDICTION IN  
COMPOSITE OVERWRAPPED PRESSURE VESSELS FROM  
ACOUSTIC EMISSION DATA**

Seth-Andrew T. Dion

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  
December 2006

UMI Number: EP32102

### INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI<sup>®</sup>

---

UMI Microform EP32102  
Copyright 2011 by ProQuest LLC  
All rights reserved. This microform edition is protected against  
unauthorized copying under Title 17, United States Code.

---

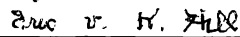
ProQuest LLC  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106-1346

**NEURAL NETWORK BURST PRESSURE PREDICTION IN  
COMPOSITE OVERWRAPPED PRESSURE VESSELS FROM  
ACOUSTIC EMISSION DATA**

Seth-Andrew T. Dion

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

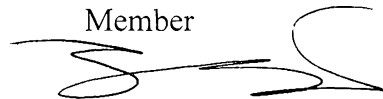
THESIS COMMITTEE



Dr. Eric v. K. Hill  
Chairman



Dr. Eric R. Perrell  
Member



Dr. Yi Zhao  
Member



Dr. Habib Eslami  
Department Chair, Aerospace Engineering

1/17/07  
Date



Dr. Christina M. Frederick-Recascino  
Associate Provost

1/17/07  
Date

“X-RAYS: Their moral is this – that a right way of looking at things  
will see through almost anything.”

Samuel Butler

## ACKNOWLEDGEMENTS

I would like to thank everyone that has helped me along my academic journey. I especially need to thank my mother, Judith, who has provided endless inspiration and encouragement in all of my endeavors, my father, Richard, who has been my advocate and advisor, and my sister, Alyssa, for being a source of levity and maintaining my sanity. Thank you also to my fellow students Justin Karl, my abductor, and Nicholas Spivey who both started this project with me and spent innumerable nights in the lab processing and analyzing data. Christopher Cassano, thank you for providing a fresh view on much of this project and dedicating so much time to its progress.

In the Aerospace Engineering Department, I must thank Mary Balistrone for being the best ally and surrogate mother that a graduate student could want. I owe Dr. Eric v. K. Hill an unpayable debt of gratitude for being my mentor and forcing me to grow immeasurably as a student as well as expertly playing the role as my thesis chair, guiding me throughout the process. Drs. Yi Zhao and Perrell, thank you for dedicating the time to participate on my committee. Finally, thank you Dr. Habib Eslami for so many classes over the years, the confidence in me as a student, and ultimately the confidence showed in me as a teacher.

There are so many other people that have impacted my life, sometimes only through the briefest of interactions. To these people I owe my final thanks. It is through the summation of these innumerable, ordinary, and occasionally extraordinary, events that my life has been shaped.

## ABSTRACT

Author: Seth-Andrew T. Dion  
Title: Neural Network Burst Pressure Prediction in Composite Overwrapped Pressure Vessels from Acoustic Emission Data  
Institution: Embry-Riddle Aeronautical University  
Degree: Master of Science in Aerospace Engineering  
Year: 2006

Composites have grown in importance in the aerospace industry where high specific strength is a priority. Weight reduction in space vehicles is critical because of the exorbitant cost associated with placing objects into space. Major weight savings have been obtained by switching from all metal pressure vessels to composite overwrapped pressure vessels (COPVs). Due to the nature of composites, current nondestructive analysis procedures for COPVs are not adequate for assessing structural integrity. As such, new methods must be developed. Presented herein is one such method.

A method for burst pressure prediction using parametric filtering of acoustic emission (AE) data along with the specification of a categorical variable defining damage type has yielded accurate results for COPVs. The process, while accurate -- 5.85 % worst case prediction error -- required that the inflicted damage type of the bottle be known in order to make accurate predictions.

The newly developed method relied heavily upon filtering of the parametric data recorded by an acoustic emission detection system. This edited data set was then used to make burst pressure predictions using a three layer backpropagation neural network given the AE amplitude distributions as input.

## TABLE OF CONTENTS

Acknowledgements.....	i
Abstract.....	ii
Table of Contents.....	iii
List of Tables.....	iv
List of Figures.....	iv
List of Symbols and Abbreviations.....	v
1.0 Introduction.....	1
1.1 Composite Overwrapped Pressure Vessels.....	1
1.2 Project History.....	5
1.3 Acoustic Emission Theory.....	9
1.4 Failure Mechanisms in Composites.....	11
1.5 Classical Wave Theory.....	13
2.0 Parametric Filtering.....	15
2.1 Introduction.....	15
2.2 Initial File Handling.....	16
2.3 Filtering Process.....	17
3.0 Statistical Modeling of Data.....	21
3.1 Introduction.....	21
3.2 Models.....	25
4.0 Results.....	27
5.0 Conclusions.....	29
6.0 References.....	31
7.0 Appendix.....	32
7.1 Amplitude Distributions.....	32
7.2 Bounded Johnson Distributions.....	38
7.3 Perl Codes.....	44
7.3.1 Comma Delimiting Converter.....	44
7.3.2 Parameter Separator.....	46
7.4 MATLAB Codes.....	48
7.4.1 Frequency_Analyzer.m.....	48
7.4.2 Johnson_dsitribution.m.....	53
7.4.3 Wave Speed.....	56
7.4.4 TWPV_burst_pressure_prediction.m.....	58



## LIST OF TABLES

Table 1: COPV Bottle Data .....	7
Table 2: Calculated wave speeds in the COPV aluminum liner .....	14
Table 3: BNN Architectural Summary .....	27
Table 4: Neural Network Results.....	28

## LIST OF FIGURES

Figure 1: Carbon/Epoxy COPV during manufacture.....	2
Figure 2: Fiber breaks as a function of ultimate load .....	4
Figure 3: Representative COPV used in burst pressure testing .....	5
Figure 4: AE transducer locations.....	6
Figure 5: Illustration of data variability between two similar bottles.....	8
Figure 6: Acoustic emission waveform and quantification parameters.....	10
Figure 7: AE Amplitude Distribution SN014 300 Hits.....	12
Figure 8: Converted data file .....	16
Figure 9: Energy vs. Duration - unfiltered.....	18
Figure 10: Energy vs. Duration - post zero-energy filtration.....	18
Figure 11: Energy vs. Duration - post 100 ms duration filter.....	20
Figure 12: Energy vs. Duration – post 25 $\mu$ s rise time filter.....	20
Figure 13: Johnson distribution approximation curves.....	23
Figure 14: Example of Bounded Johnson distribution approximation curve .....	24
Figure 15: Bounded Johnson distribution SN020.....	26
Figure 16: SN002 Amplitude Distributions.....	33
Figure 17: SN003 Amplitude Distributions.....	33
Figure 18: SN009 Amplitude Distributions.....	34
Figure 19: SN010 Amplitude Distributions.....	34
Figure 20: SN013 Amplitude Distributions.....	35
Figure 21: SN014 Amplitude Distributions.....	35
Figure 22: SN018 Amplitude Distributions.....	36
Figure 23: SN020 Amplitude Distributions.....	36
Figure 24: SN025 Amplitude Distributions.....	37
Figure 25: SN026 Amplitude Distributions.....	37

## LIST OF SYMBOLS AND ABBREVIATIONS

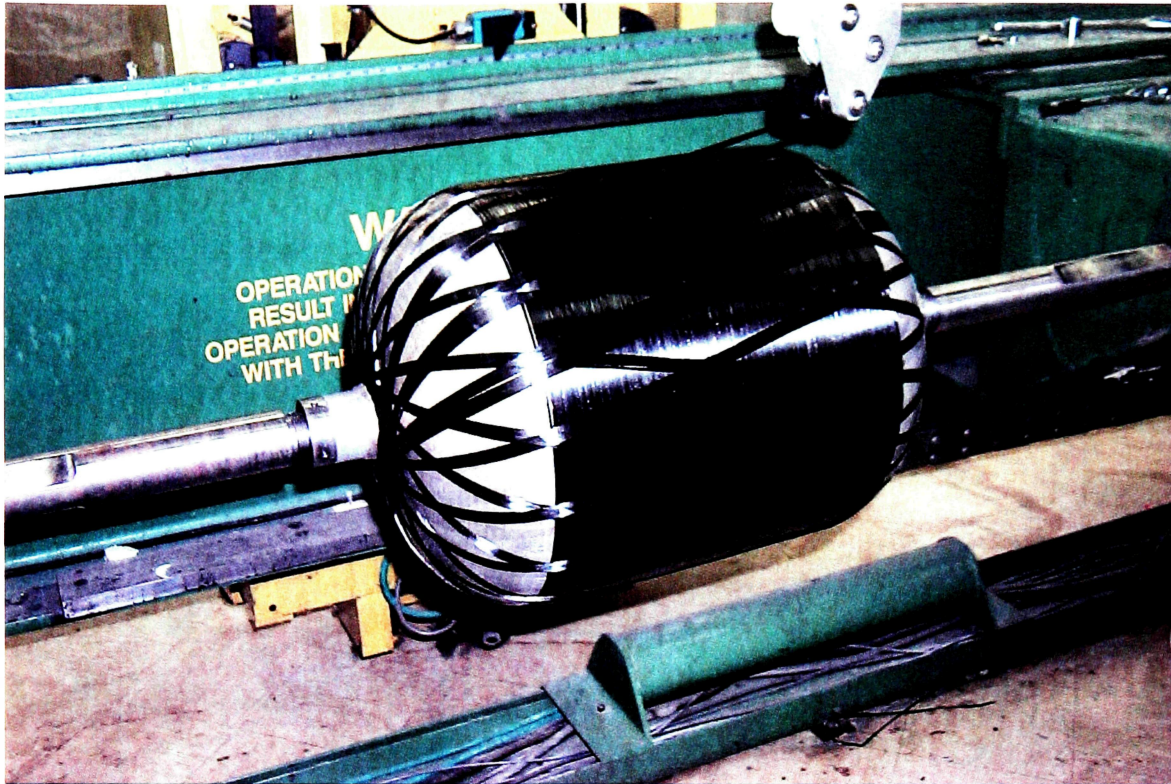
AE	Acoustic emission
ANN	Artificial neural network
BPNN	Backpropagation neural network
BVID	Barely visible impact damage
C	Counts is the number of voltage versus time waveform signal excursions above threshold
COPV	Composite overwrapped pressure vessel
D	Duration of signal waveform above threshold [ $\mu$ s]
MEOP	Maximum expected operating pressure
PFM	Parametric filtering method
$\beta_1$	Square of relative skewness
$\beta_2$	Relative kurtosis

## 1.0 INTRODUCTION

### 1.1 Composite Overwrapped Pressure Vessels

Advanced composite materials play an important role in the aerospace industry where high strength and low weight are given priority over cost or ease of manufacture. This is especially evident in space applications. At the cost of tens of thousands of dollars per pound to place an object into orbit, the importance of minimum weight components and structures becomes apparent.

Pressure vessels find widespread use in space vehicles as propellant and pressurant tanks. For example, the Space Shuttle Orbiter uses 24 spherical COPVs ranging from 19 to 40 inches in diameter in four different subsystems (Sutter, 1). Pressure vessels have been made from metals like steel, aluminum, or titanium, but the potential weight savings associated with using composites has led to an increase in popularity of composite overwrapped pressure vessels (COPVs) as shown in Figure 1. As the name suggests, COPVs are pressure vessels that have composite layers wrapped over an ultra-thin, typically metallic, inner liner. The inner liner serves as the container for the fluid under pressure and also as the forming mandrel, while the composite overwraps provide structural strength. The weight savings of COPVs over metallic pressure vessels warrant their use; however, there are several drawbacks of using COPVs in place of their fully metallic counterparts, the most significant drawback being the high sensitivity to impact damage.



**Figure 1: Carbon/Epoxy COPV during manufacture**

There exists a great potential for COPVs to receive impact damage during their operational lifetimes. Damage ranging from a dropped wrench to a dropped bottle has the potential to significantly decrease the strength of the bottle. In some cases, impact damage that is severe enough to profoundly decrease the strength of the bottle is not readily visible to the naked eye. This is termed barely visible impact damage (BVID).

The possibility of BVID in combination with the wide variability in composite end-product quality and mechanical properties, make proof testing of COPVs even more important than for metallic pressure vessels. Proof testing of metallic pressure vessels generally consists of a controlled pressurization up to approximately 150% of the maximum expected operating pressure (MEOP), depending on the standard being used,

followed by controlled depressurization. During the proof test the pressure vessel should remain within the elastic region. As such, when depressurized, its strength is not substantially affected. This is not the case with composite pressure vessels. As a COPV is pressurized up to and above the MEOP the matrix material begins to crack, composite layers may delaminate and fibers may break, resulting in a bottle that could be substantively different, with respect to physical properties, than the bottle that began the proof test. This problem is compounded every time the vessel is pressurized because of the effects of cumulative damage. As a result, COPVs are sometimes tested in the range of 70-80% of the ultimate load. However, this pressurization can still lead to fiber breakage (NDT Handbook, 172).

Of the failure mechanisms previously mentioned, fiber breaks have the most profound effect on the strength of the pressure vessel. It can be seen in Figure 2 that as loading increases toward, and then above, the proof pressure, the resulting number of fiber breaks increases at an exponential rate (Hill, 745). This presents a dilemma: proof testing of COPVs is imperative for their safe operation, yet proof testing can detrimentally affect COPV strength. It is apparent from this discussion that a need exists for a new method of testing COPVs that does not require such high proof pressures.

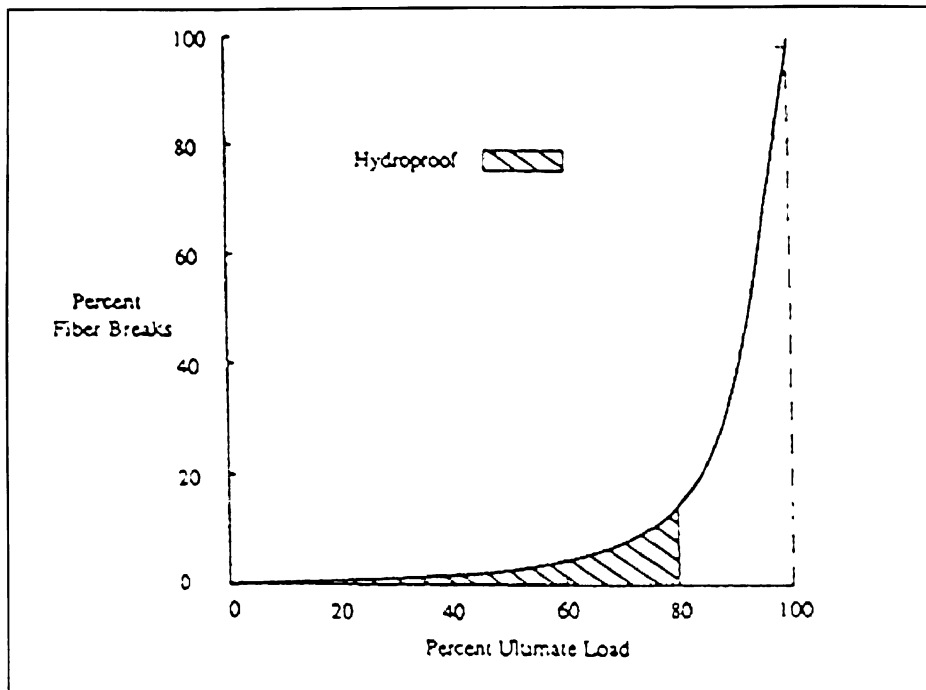


Figure 2: Fiber breaks as a function of ultimate load

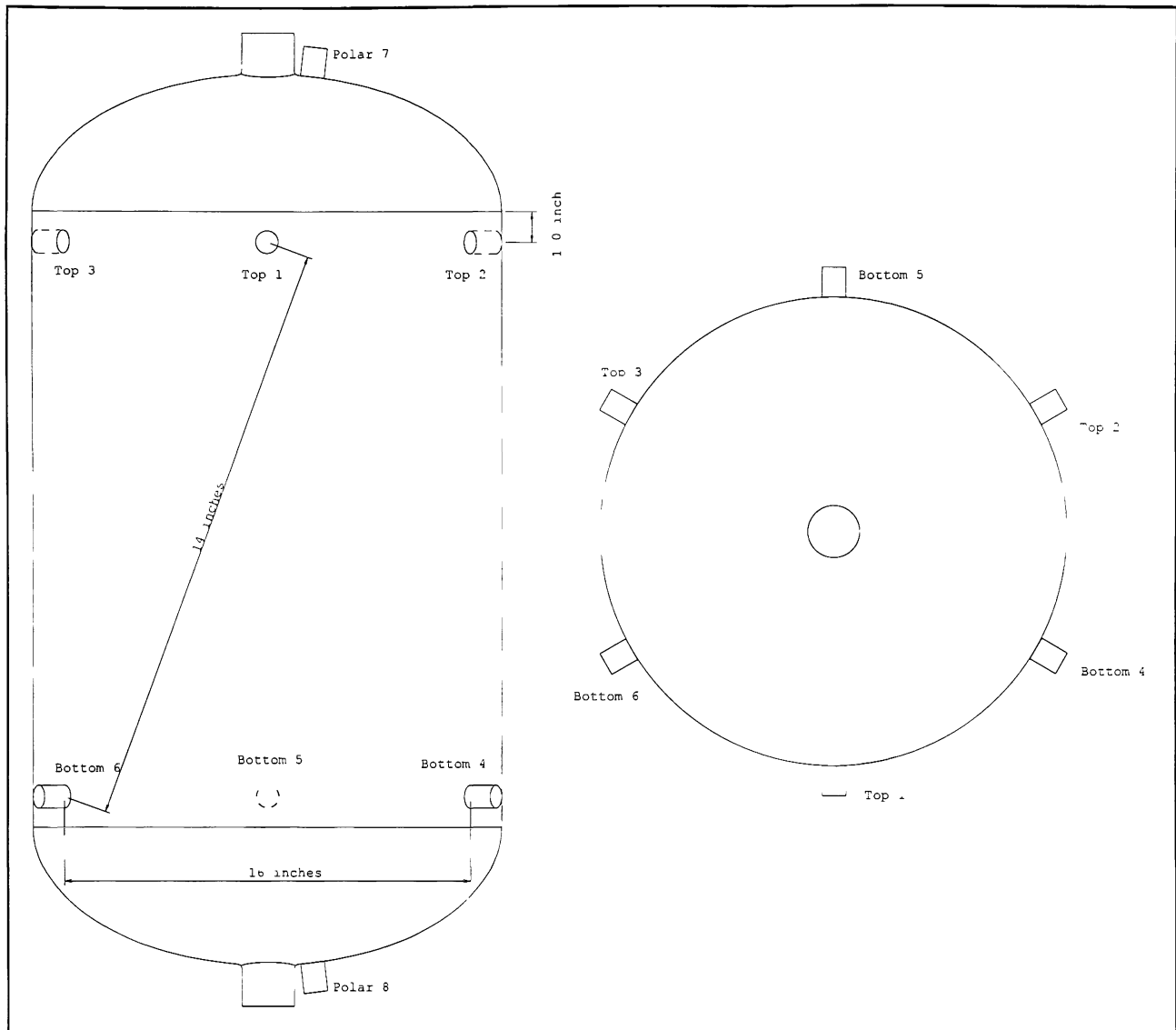
## 1.2 Project History

A number of COPVs were manufactured by the NASA Marshall Space Flight Center for the purpose of performing destructive burst pressure testing. Each bottle was 15 inches in diameter and made from filament wound graphite/epoxy. A representative bottle used in the testing can be seen in Figure 3.



**Figure 3: Representative COPV used in burst pressure testing**

The bottles were fitted with instrumentation in order to measure various parameters including pressure and acoustic emission (AE) signals from flaw growth activity. Eight AE transducers were attached to each bottle, with the seventh and eight transducers near the polar bosses and the remaining six arrayed in two sets of three around the mid-section. A schematic illustration of transducer locations is given in Figure 4.



**Figure 4: AE transducer locations**

It should be noted that not all of the bottles had the *eighth* transducer mounted.

A summary of the manufacturing and testing data can be found in Table 1 for each of the ten bottles. Note that all bottles experienced differing pressurization ramp schemes, different types and amounts of damage, different curing methods, and different pressurant temperatures (cryogenic and room temperature).



Table 1: COPV Bottle Data

SN	Damage	Cure (Oven)	Cryo	Pressurization Scheme
SN002	Impact	Static	Yes	5 ramps w/ unloads, "plateau" has creep
SN003	Impact	Rotisserie	Yes	3 ramps w/ unloads, short-uphill "holds"
SN009	None	Static	Yes	5 abrupt ramps
SN010	None	Static	Yes	6 ramps
SN013	None	Rotisserie	No	4 smooth ramps
SN014	None	Rotisserie	Yes	5 ramps
SN018	Cut Hoop Fibers (5 tows, midhoop 1 <sup>st</sup> outer hoop)	Rotisserie	No	1 slow ramp
SN020	Impact	Rotisserie	Yes	4 ramps, 4 <sup>th</sup> long hold
SN025	Cut Hoop Fibers (5 tows, midhoop, 1 <sup>st</sup> outer hoop)	Rotisserie	Yes	5 ramps w/ unloads, short-uphill "holds"
SN026	Cut Hoop Fibers (5 tows, 3" from dome, inner hoop)	Rotisserie	No	Unavailable

The lack of a systematic treatment of the bottles, as well as the notoriously noisy nature of AE data from composites, negated successful attempts to make accurate burst pressure predictions. Only with the development of parametric analysis including categorical variables for noise rejection were accurate predictions achieved. This is partly due to the large variability of the intrinsic patterns in the AE data from one bottle to the next, but more a function of noise suppression.

An example of the difference in AE data from the damage progression of two similar bottles (one red, the other blue) can be seen in Figure 5. Furthermore, virgin stressing of composites is oftentimes typified by dramatic acoustic activity. This noisy behavior is a direct result of the many failure modes in composite structures coupled with the brittle mechanical properties of both the fibers and matrix materials.

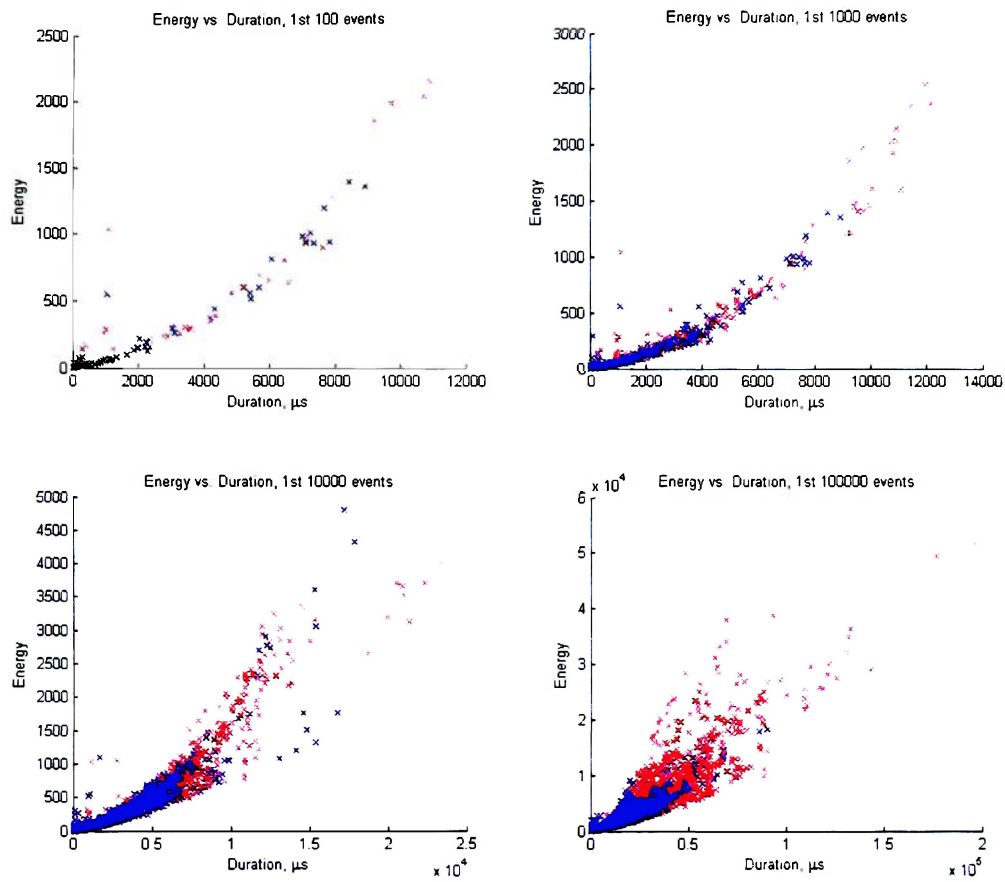


Figure 5: Illustration of data variability between two similar bottles

### 1.3 Acoustic Emission Theory

Acoustic emission is a nondestructive testing method that involves instrumenting a specimen with piezoelectric transducers and recording parametric representations of the waveform data from flaw growth activity in order to perform some type of analysis. Analysis of the data will oftentimes allow for the determination of failure mechanisms that are active in the specimen. Acoustic emission is caused by the rapid release of energy within the system due to dissipative mechanisms in the solid.

During the testing of a specimen, any waveforms that have amplitudes greater than a preset threshold value are quantified by the acoustic emission system. Each signal picked up by the transducer is called a "hit", and each hit can be defined by a combination of five parameters. These quantification parameters include rise time, energy, amplitude, duration, and counts, as seen in Figure 6. Rise time is the time it takes for the signal from the initial excursion above the threshold to reach its peak amplitude. Amplitude is a measure of the signal strength and is commonly measured in decibels [dB]. The duration of the AE signal is the time [ $\mu\text{s}$ ] from the initial to the final excursions above the threshold. Counts is a measure of the number excursions above the threshold that the rectified signal makes during its duration, and energy is the area under the rectified signal envelope.

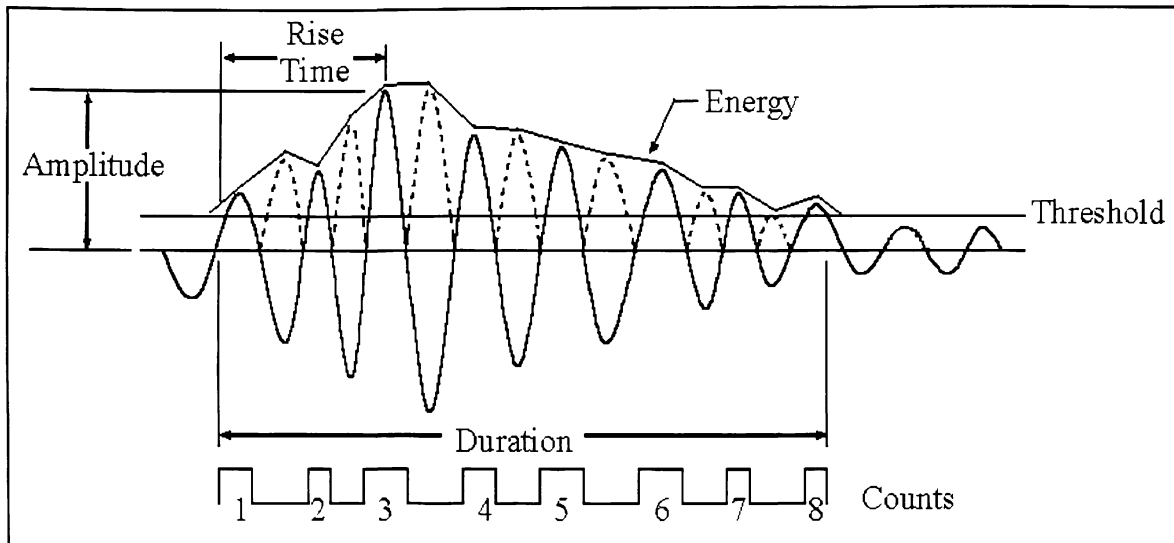


Figure 6: Acoustic emission waveform and quantification parameters

The acquisition of acoustic emission data can be highly sensitive to the recording hardware's setup parameters. These parameters are not of any particular consequence to this paper; however, it should be noted that as a result of these parameters, it is possible for multiple hit data (MHD) to occur (Karl, 14). When AE signals begin to occur faster than the recording hardware can save them, signals may be partially lost or superimposed onto other signals. It therefore becomes difficult to identify and remove the two overlapping signals, or MHD, without some form of parametric filtering. The techniques used herein will be presented in a subsequent section.

#### **1.4 Failure Mechanisms in Composites**

Composite material systems are typified by brittle fibers embedded in a brittle matrix. Since each component of the material system has unique mechanical properties, failure can occur in either system independently or in combination. For failure to occur, damage must accumulate. Damage can be defined as an irreversible microscopic change in the solid that leads to a lower energy state. The different modes of damage include fiber breaks, transverse matrix cracking (cracks in the matrix perpendicular to the fiber direction), fiber-matrix debonding, and delaminations.

However, AE does discriminate if the damage is occurring on the micro or macro scale. It will record the transient elastic waves regardless of the source so long as the wave has sufficient energy and is within the frequency range of the transducers. As such, there are other significant sources of AE in the system during testing. In composite laminates it has been shown (Talreja, 13) that there is a progression of damage from matrix cracking, interfacial (fiber-matrix) debonding, delamination, fiber breakage, and ultimately total failure of the laminate. In the real-world testing of COPVs it would also be expected to have noise from inter-laminar rubbing during cyclic pressurization, debonding of the composite from the metallic liner, electromagnetic interference, miscellaneous background noise, etcetera. Therefore, it is the goal of the analyst to filter out as much of the noise that is not directly related to the failure of the specimen as possible. Even with careful consideration to data acquisition, further data processing is more often than not needed.

In composite structures, the amplitude frequencies generated during loading can be grouped and classified into failure mechanisms. These mechanisms are sometimes seen as “humps” in the AE (differential) amplitude distributions. Figure 7 shows the amplitude distribution for bottle SN014, an undamaged graphite/epoxy COPV that was pressurized with a cryogenic fluid. Although it is not immediately evident, there are distinct humps in the AE amplitude distribution that can only be separated via a classification process. The failure mechanism humps are not always evident due to the multidimensional nature of the data plus larger numbers of signals will result in data overlap. The five failure mechanisms seen here were confirmed by analyzing the data with a self-organizing map (SOM) neural network (Karl, 8). The SOM classified three primary mechanisms and fourth and fifth less prevalent mechanisms. This result confirmed what was known about composite failure modes and what was visually observed in the amplitude frequency distributions.

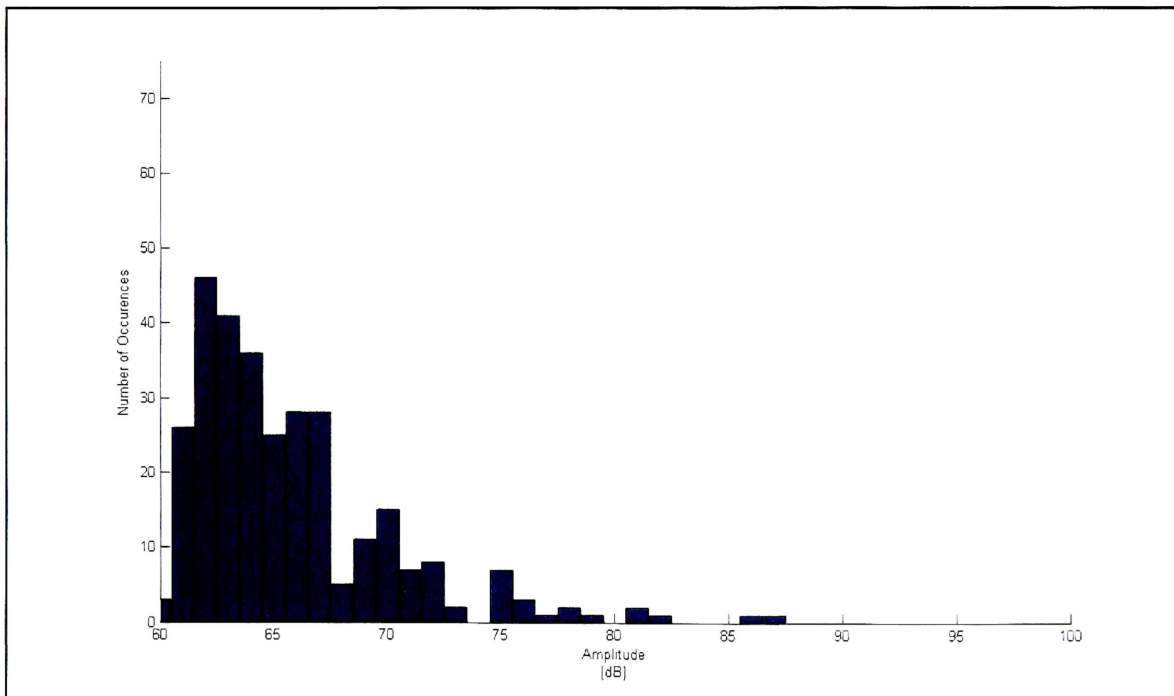


Figure 7: AE Amplitude Distribution - SN014 - 300 Hits

## 1.5 Classical Wave Theory

Acoustic emission relies on the transient elastic waves propagating in the solid to reach the data acquisition system. As the wave travels through the medium the signal attenuates. The magnitude of attenuation is a property of the geometry of the solid in which the elastic wave is propagating and the mechanical properties of the solid. One typical goal of AE monitoring is the location of the damage source mechanism. As such, the velocity of the waves must be calculated and multiple transducers must be used for flaw growth triangulation.

The wave velocities can be calculated by first finding Lamé's constants:

$$\lambda = \frac{(E\nu)}{(1+\nu)(1-2\nu)}$$

$$\mu = \frac{E}{2(1+\nu)}$$

Then the wave speeds for the different wave types can then be calculated using the following four equations.

$$c_l = \sqrt{\frac{\lambda + 2\mu}{\rho}} \quad \text{Longitudinal Wave Speed}$$

$$c_t = \sqrt{\frac{\mu}{\rho}} \quad \text{Transverse Wave Speed}$$

$$c_R = \frac{0.87 + 1.12\nu}{1 + \nu} c_l \quad \text{Rayleigh Wave Speed}$$

$$c_p = c_t \sqrt{\frac{2}{1 - \nu}} \quad \text{Lamb Wave Speed}$$

A simple program was written to calculate the wave speeds and the results for the aluminum liner are listed in Table 2.

**Table 2: Calculated wave speeds in the COPV aluminum liner**

Wave Type	Wave Speeds [m/s]
Longitudinal	6149
Transverse	3097
Rayleigh	2887
Lamb	5351

These wave types are generalizations made in an attempt to accurately describe the wave motion in the medium. However, these wave types are based on mathematical models that fit imposed assumptions. While the wave motion in the solid absolutely follows an equation, that equation is not bounded to fit the imposed model (Pollock, 6). Therefore, the motion in the real-world structure can be extremely complicated to solve.

With COPVs there are interactions between the metallic liners and the composite shells that provide the strength. The interactions are magnified because the mechanical properties of the two layers are significantly different. Furthermore, in a real pressure vessel wave coupling occurs with the contained fluid providing another means for the waves to propagate. The inherent difficulty of modeling a real structure becomes immediately apparent. The modeling of the waves in these structures is far beyond the scope of this research.

Since a COPV is essentially a thin shell, the Rayleigh and the Lamb waves will be of particular interest here. These two wave types carry the bulk of the information about the AE signal and have been successfully used in the location of damage (Scott, 160-161).



## **2.0 PARAMETRIC FILTERING**

### **2.1 Introduction**

The long-term goal of recording and analyzing the AE from the COPVs is to develop a real-time health monitoring system. Before such a system can be established, the mechanisms leading to structural failure must be well understood. Through strict filtering of the data, it is desirable to reduce the bulk of the information to an extremely concise data set consisting only of information that can be directly used by an artificial neural network (ANN), or other prediction toolsets, to make burst pressure predictions. Once an ANN can be relied upon to make accurate predictions, the next step to developing an active monitoring system can be taken. During the filtering process, data sets were made smaller and more manageable, erroneous data due to ambient noise were eliminated, and failure mechanisms in the specimens were identified to allow for further analysis in future burst pressure predictions.

## 2.2 Initial File Handling

NASA's AE data sets for the pressurization of the bottles existed in very large files, originally in DTA format. Prior to filtering, the AE data had to be converted to text format. The raw DTA files were converted into TXT files with a MISTRAS utility program, ATASC.exe. The converted text files format is given in Figure 8 below.

```

"(FILE d \aedata\sn002c02 dta)"

"(TEST START DATE & TIME)"
"(Tue Mar 4 20 02 53 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 00 00 00 3072210      0 01      1      2      2      1      13      65
0 00 00 00 8134537      0 01      4      15     4      4      49      65
0 00 00 00 8187160      0 01      3      28     18     25     222     73
0 00 00 00 9896647      0 02      7      40     96     315     901     88
0 00 00 00 9897120      0 02      6     213     96     207     923     78

```

**Figure 8: Converted data file**

The resulting text files typically comprised over 300,000 lines, largely comprised of whitespace (nonnumerical spaces), as shown in Figure 8. A script written in Perl (a computer program adept at handling large file I/Os) was created that condensed the files by removing all whitespace, effectively making the files smaller and easier to manipulate.

### 2.3 Filtering Process

The first filtering step was to remove all zero-energy hits. Zero-energy hits are signals whose waveform has just enough energy to trip the data acquisition system before attenuating back below the threshold. Consequently, zero-energy hits contain no useful parametric information for analysis. Filtering the zero-energy hits removed the most data of all the filtering steps described herein. A representative energy versus duration plot for data after zero-energy event filtration can be seen in Figure 10. Note that while Figure 9 and Figure 10 appear to be identical, the data that were removed fell in the data dense area at the left-hand end of the plot, and as such, the changes are not readily apparent.

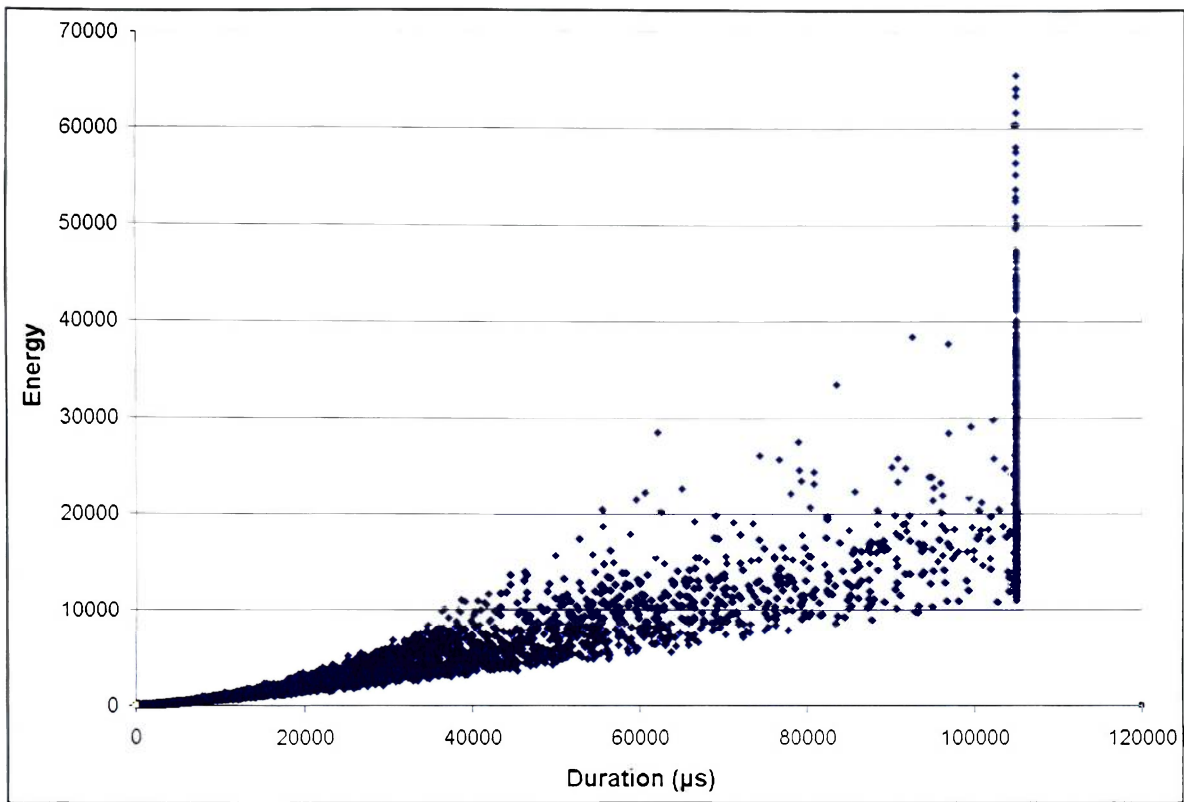


Figure 9: Energy vs. Duration - unfiltered

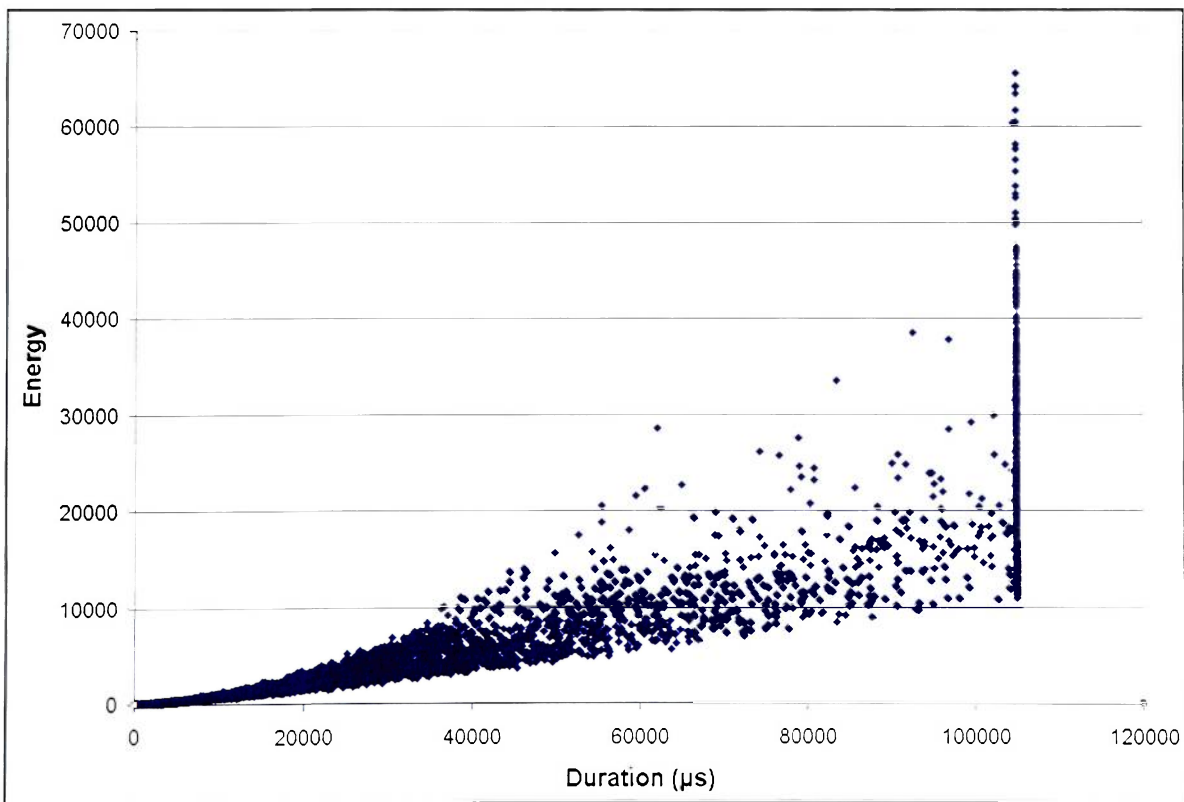


Figure 10: Energy vs. Duration - post zero-energy filtration

After filtering zero-energy hits, the next process was to filter events that had very long durations, i.e., greater than 100,000  $\mu\text{s}$  (100 ms). The logic behind this was twofold: first, any duration greater than this would most likely be either ambient noise or multiple hit data (MHD). Second, many data points were exceeding the maximum allowable duration for the data acquisition system as seen in Figure 9 and Figure 10 as the straight vertical line to the far right of the graph. The resulting data after filtering by duration can be seen in Figure 11.

Removing hits with rise times greater than 25  $\mu\text{s}$  was the last step in this parametric editing procedure. Events with rise times shorter than 25  $\mu\text{s}$  are most likely flaw growth and are used for prediction; conversely, events with rise times greater are probably mechanical in nature (Moore, 42) and therefore need to be eliminated. Compared to filtering zero-energy and long-duration hits, the number of hits filtered by the rise time stipulation was relatively few. However, this final step proved to be significant in the creation of data sets that were easier to analyze. The resulting energy versus duration plot can be seen in Figure 12.

The above three step process was termed the parametric filtering method (PFM). After the application of the PFM, the data sets were significantly reduced in size. The final step was the preparation of amplitude distributions for prediction. Here the first 2000 hits were selected from the filtered data. While 2000 is a somewhat arbitrary number, it provided enough data to be statistically significant and yet was small enough to have had all the hits occur at low loads – less than 25 % of ultimate (Hill, 745).

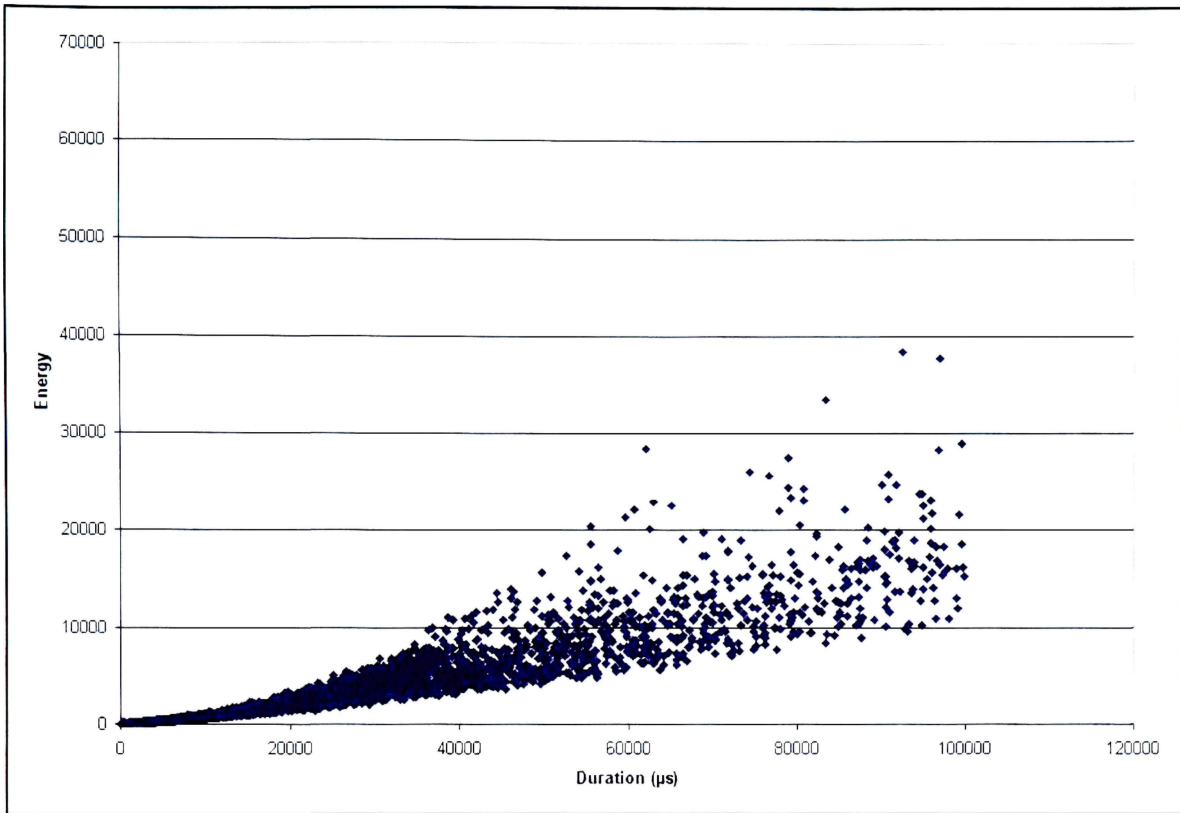


Figure 11: Energy vs. Duration - post 100 ms duration filter

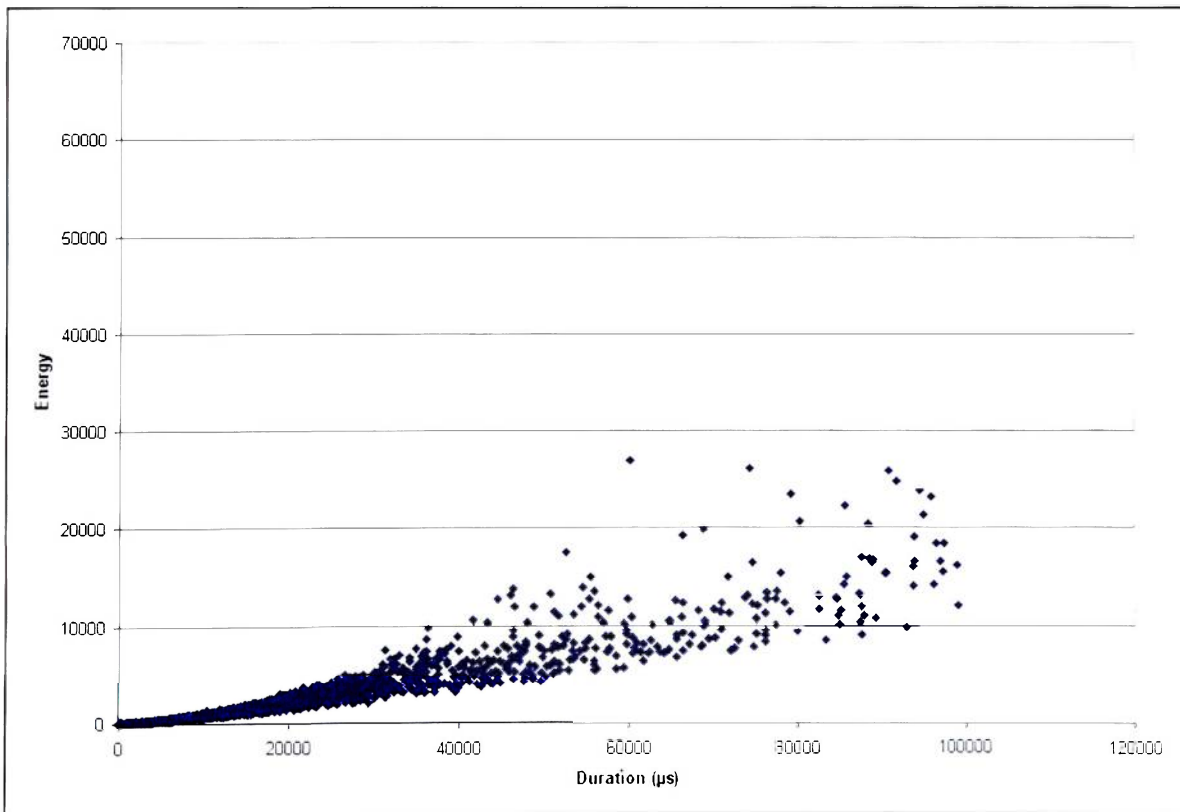


Figure 12: Energy vs. Duration – post 25 μs rise time filter

### 3.0 STATISTICAL MODELING OF DATA

#### 3.1 Introduction

Before the advent, and subsequent modernization, of digital computing systems, it was common to model empirical engineering data through statistical analysis. While there are numerous distribution models in use today (normal, lognormal, and extreme value functions), an often overlooked model is the Johnson distribution. The Johnson distribution is an extremely versatile data fitting model based on the transformation of the standard normal variate (Hahn and Shapiro, ).

The first step in the modeling process is to make estimations for the central moments, as is often the case with statistic-based techniques. The moments are calculated using the following equations:

$$m_2 = \hat{\sigma}^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

$$m_3 = \frac{\sum_{i=1}^n (x_i - \bar{x})^3}{n}$$

$$m_4 = \frac{\sum_{i=1}^n (x_i - \bar{x})^4}{n}$$

Next the relative skewness,  $\beta_1$ , and the relative kurtosis,  $\beta_2$ , are calculated for the data sets.

$$\sqrt{\beta_1} = \frac{m_3}{(m_2)^{3/2}}$$

$$\beta_2 = \frac{m_4}{(m_2)^2}$$

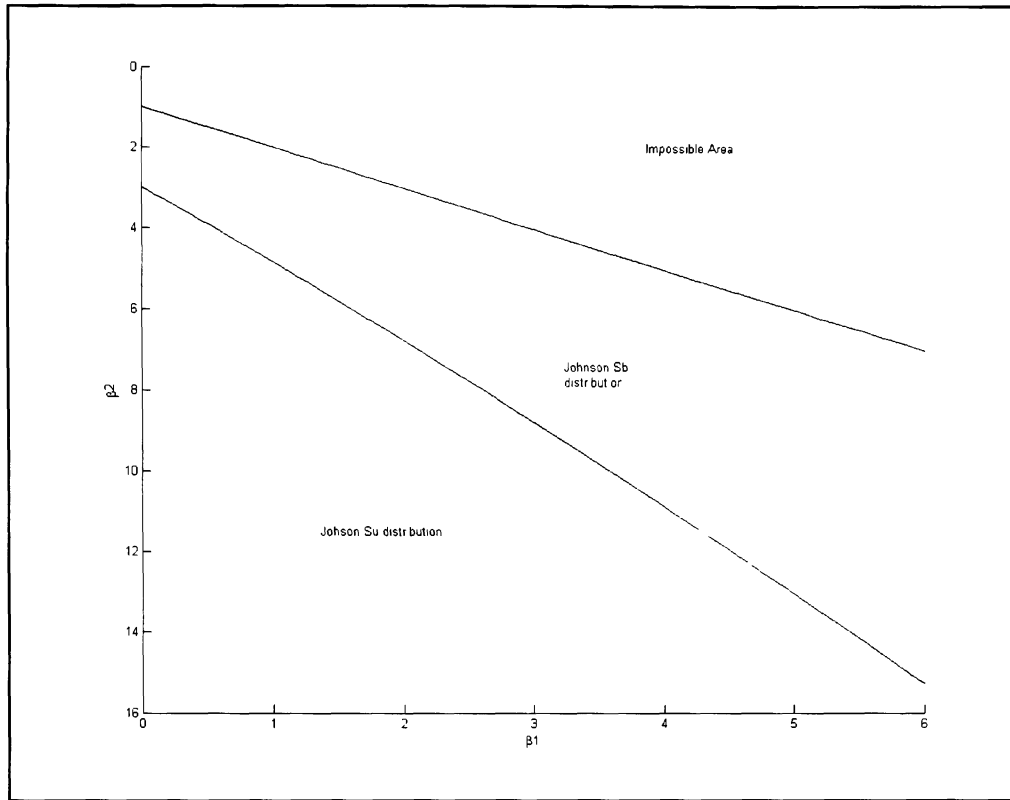
Once the  $\beta_1$  and  $\beta_2$  values are known, they can be used in conjunction with Johnson distribution approximation curves given by the following formulas.

$$\beta_1 = (\omega - 1)(\omega + 2)^2 \quad \text{where } \omega \text{ is a range of values to plot the Johnson approximation curves}$$

$$\beta_2 = \omega^4 + 2\omega^3 + 3\omega^2 - 3$$

The result of the distribution curves is seen in Figure 13. The region that the data point falls into determines the best statistical model.





**Figure 13: Johnson distribution approximation curves**

The result of this determination is that the bounded Johnson distribution is the best statistical model for all the data sets. Thus, the formula for the probability distribution is given by

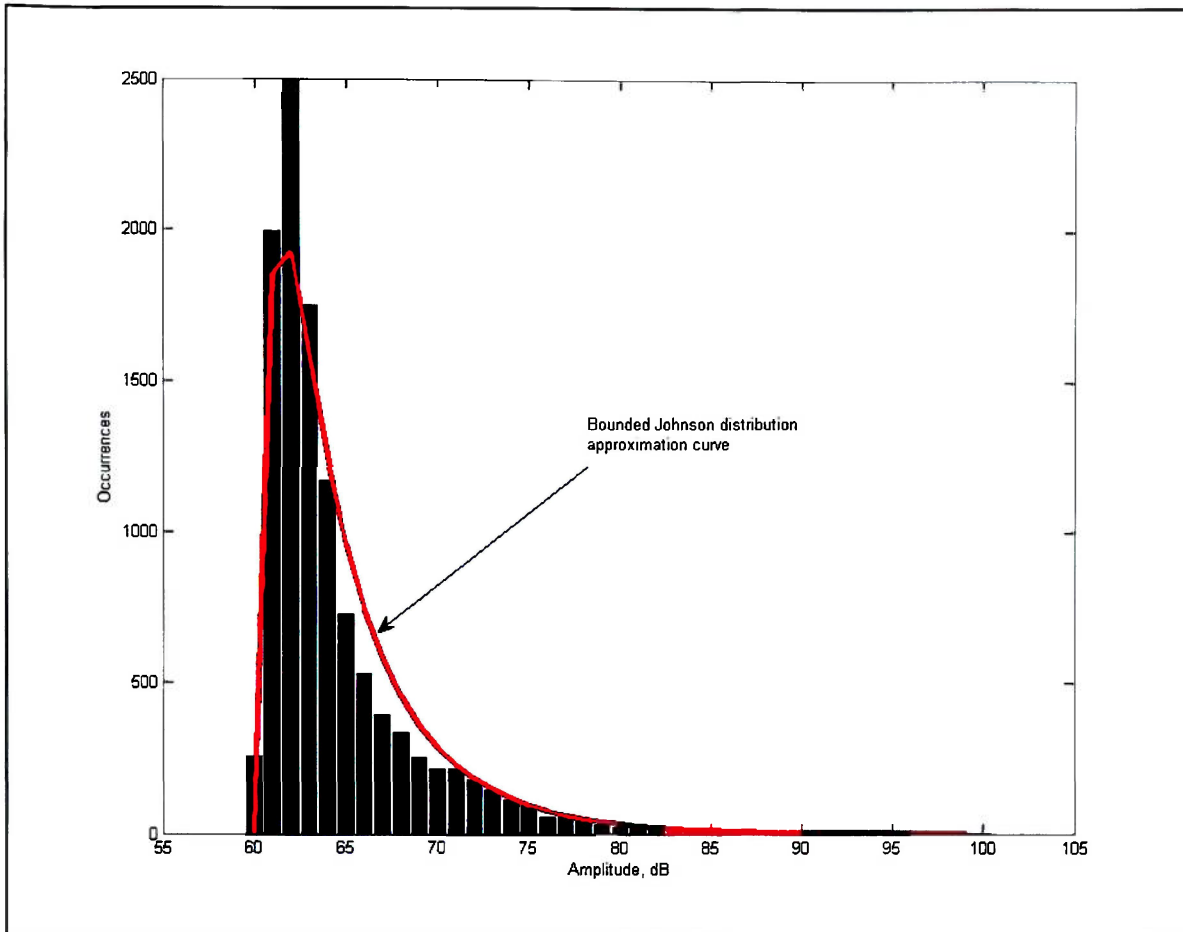
$$f_x(x) = \frac{\eta}{\sqrt{2\pi}} \frac{\lambda}{(x-\varepsilon)(\lambda-x+\varepsilon)} e^{-\frac{1}{2} \left[ \gamma + \eta \ln \left( \frac{x-\varepsilon}{\lambda-x+\varepsilon} \right) \right]^2},$$

where the shape parameters  $\eta$  and  $\gamma$  are approximated as

$$\hat{\eta} = \frac{z_{1-\alpha'} - z_\alpha}{\ln \left[ \frac{(x_{1-\alpha'} - \varepsilon)(\varepsilon + \lambda - x_\alpha)}{(x_\alpha - \varepsilon)(\varepsilon + \lambda - x_{1-\alpha'})} \right]}$$

$$\hat{\gamma} = z_{1-\alpha'} - \hat{\eta} \ln \left( \frac{x_{1-\alpha'} - \varepsilon}{\varepsilon + \lambda - x_{1-\alpha'}} \right).$$

An example of the resulting probability distribution is shown in Figure 14.



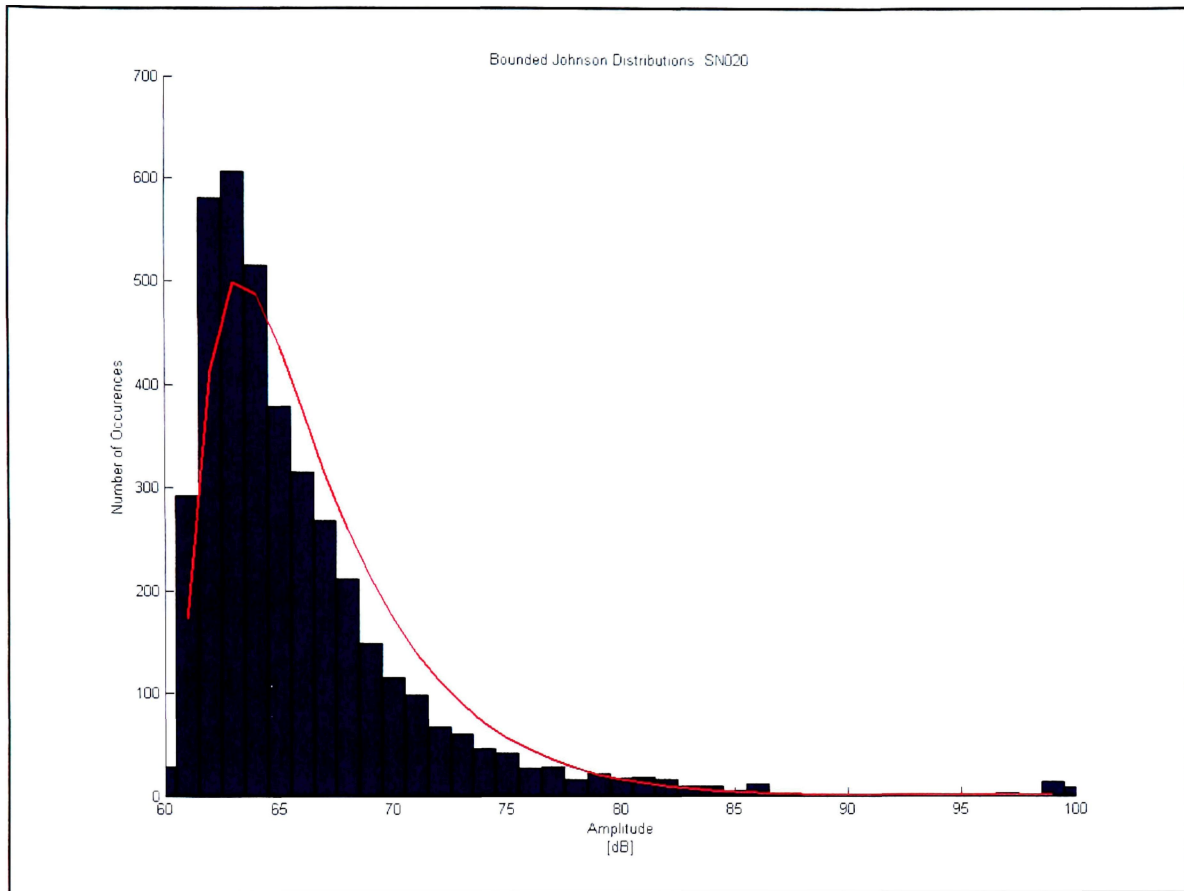
**Figure 14: Example of Bounded Johnson distribution approximation curve**

A probability density distribution curve can thus be generated for any amplitude distribution.

### 3.2 Models

Models for each of the COPVs amplitude distributions were generated from the parametrically filtered data. All of the bounded Johnson distribution calculations can be found in section 7.2.

There exists, however, a major limitation in the curve fitting. The curves were fit to the entire resulting amplitude distribution data set. Since it is known that there exist at least five failure mechanisms, and all of the failure mechanisms are included in each amplitude distribution, a highly accurate fit cannot be expected. This is readily seen in Figure 15 where the 80 and 100 dB data offset the curve at the lower amplitudes. The first failure mechanism (which is likely matrix cracking due the large number of low amplitude hits) is located on the left of the plot. However, the Johnson distribution curve can be seen to be skewed (or pulled over as it were) to the right. That is due to the large moment arm that the sparse, high-amplitude data to the far right possess.



**Figure 15: Bounded Johnson distribution - SN020**

In order to improve the accuracy of the modeling, the failure mechanisms must first be separated and then each individual mechanism modeled with its own distribution curve.

## 4.0 RESULTS

The parameterized data from the COPVs were recorded into DTA file format by a Physical Acoustics MISTRAS<sup>®</sup> AE acquisition system. Acoustic emission hits were recorded along with the transducer channel, pressure, and the AE parameters counts, rise time, energy, duration, and amplitude. Filtering was accomplished by using the aforementioned parametric frequency method. The results of the PFM calculations yielded frequencies from the sub 10 kHz to 1 MHz range. It is important to understand that the acoustic emission signal frequency spectrums are very sensitive to the resonance and transmission characteristics of the AE transducer and the emitting object's geometry and acoustic properties (Miller, 35).

Once the amplitude distributions were completed, it was possible to begin data preparation for input into a backpropagation neural network (BPNN). The architectural design of the three layer network is beyond the scope of this paper; however, the details of the network architecture and settings are summarized in Table 3.

**Table 3: BNN Architectural Summary**

Inputs	41
Hidden Layer Neurons	11
Output	1
Momentum	0.400
F' Offset	0.100
Learning Coefficient	0.300
Transfer Function	Hyperbolic Tangent
Learning Rule	Norm-Cum-Delta
Convergence Criterion	0.070

The results from the BPNN's output are listed in Table 4. The predictions were not within the desired range of  $\pm 5$  percent; however, they were within an acceptable range.

The worst case prediction was for SN010 with a prediction error of +5.85%.

**Table 4: Neural Network Results**

S/N	Condition	Cure Type	Temperature	Use	[psig]	Prediction	% Error
002	Impacted	Oven	Cryogenic	Train	1880	1853.69	1.40
003	Impacted	Rotisserie	Cryogenic	Train	2004	2069.75	-3.28
009	Good	Oven	Cryogenic	Train	2544	2630.34	-3.39
013	Good	Rotisserie	Ambient	Train	2874	2866.85	0.24
014	Good	Rotisserie	Cryogenic	Train	2390	2402.04	-0.50
018	Lacerated	Rotisserie	Ambient	Train	2864	2945.45	-2.84
025	Lacerated	Rotisserie	Cryogenic	Train	2393	2399.95	-0.29
010	Good	Oven	Cryogenic	Test	2460	2316.07	5.85
020	Impacted	Rotisserie	Cryogenic	Test	1967	1955.75	0.57
026	Lacerated	Rotisserie	Ambient	Test	2675	2572.71	3.82

## 5.0 CONCLUSIONS

The parametric filtering method yielded burst pressure predictions with a high level of accuracy from the AE parametric data. Increasing the sample size substantially should yield more consistent and accurate test results while being less sensitive to small disturbances. Predictions based off of the Johnson distributions were inconsistent and outside the bounds of tolerable limits. The erroneous predictions are likely due to the modeling of multiple failure mechanisms with only one curve. Therefore, predicting via the PFM in conjunction with an ANN is preferable.

In well defined successful experiments, it is essential to define a control group that sets a foundation on which to compare all other variables in a singular manner in order to determine each variable's contribution to the applied parameters. Without such a group, the purpose of changing variables to determine their respective effects borders on futility. There is also a need for systematic control over changing variables of interest. In this case, there clearly was no control group, nor was there a systematic approach to changing variables. Therefore, the effects on burst pressure by variables such as curing, cryogenics, impact, and laceration could not be sufficiently determined. Many bottles were varied in more than one manner and devoid of a foundation for comparison; little could be drawn from the effect of each variable. However, the purpose of the original testing was not to gather any specific or useful information for this research, but rather to test other equipment. Despite that, accurate predictions were still attained. In the future, and for verification, it would be highly beneficial to maintain a proper control group.

In the future, additional attention should be given to separating the individual failure mechanisms and making a comparison between the prediction methodologies', mentioned herein, accuracies. A long-term goal should be to determine the failure mechanism that allows for the most accurate, repeatable predictions with the least amount of post-acquisition manipulation with the intent of creating a real-time, health monitoring system.



## 6.0 REFERENCES

1. Hahn, Gerald and Shapiro, Samuel. Statistical Models in Engineering. New York: Wiley Classics Library Edition, 1994.
2. Hill, E. v. K., Walker II, J. L., Rowell, G. H., "Burst Pressure Prediction In Graphite/Epoxy Pressure Vessels Using Neural Networks And Acoustic Emission Amplitude Data." Materials Evaluation (1996, June): 744-754.
3. Karl, Justin. "Filtering of Acoustic Emission Data Through Principal Frequency Component Extraction." Embry-Riddle Aeronautical University, 2006
4. Miller, R. K., Hill, E. v. K., Moore, P. O., eds. "Acoustic Emission Testing" vol. 6, Nondestructive Testing Handbook. 3<sup>rd</sup> ed. Columbus OH: American Society of Nondestructive Testing, 2005.
5. Pollock, A. A., "Classical Wave Theory in Practical AE Testing." Physical Acoustics Corporation: Princeton, New Jersey, 1988.
6. Scott, Ian G. Basic Acoustic Emission. New York: Gordon and Breach Science Publishers, 1991.
7. Sutter, Material Issues in Space Shuttle Composite Overwrapped Pressure Vessels. NASA: 2006.
8. Talreja, R. "Modeling of Damage Development in Composites Using Internal Variable Concepts." Damage Mechanics in Composites AD-Vol 12 (1987): 11-16.

## **7.0 APPENDIX**

### **7.1 Amplitude Distributions**

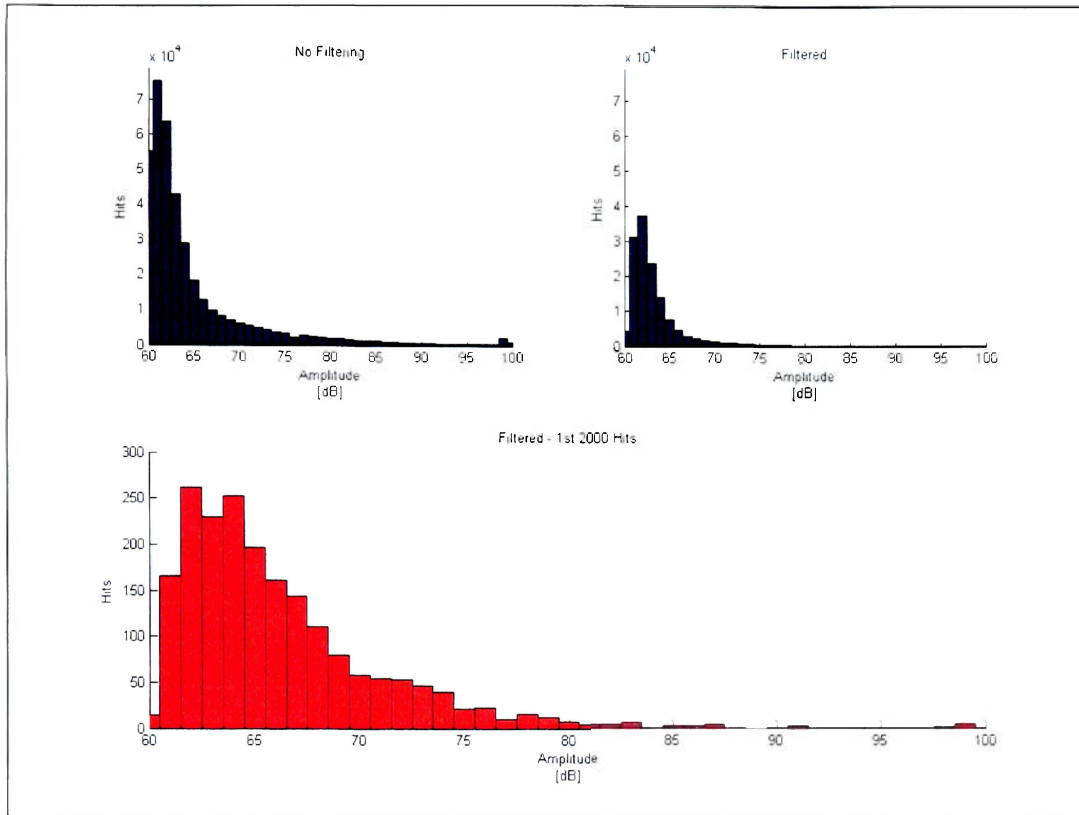


Figure 16: SN002 Amplitude Distributions

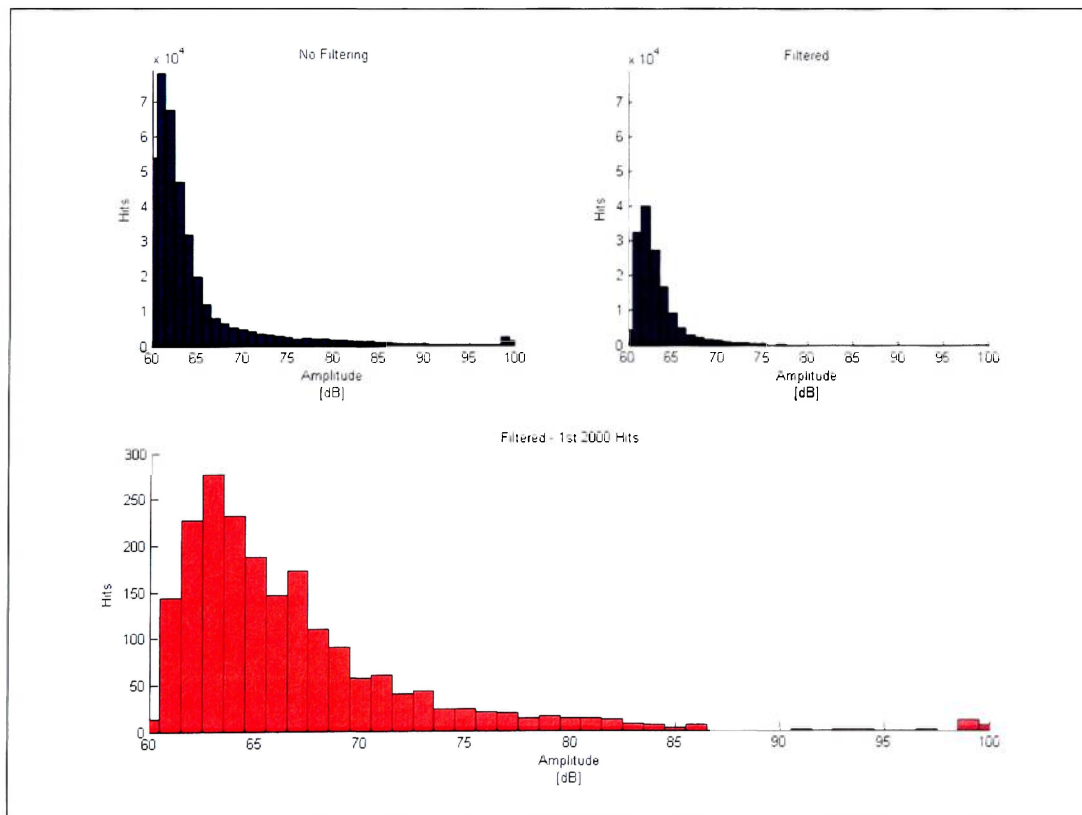


Figure 17: SN003 Amplitude Distributions

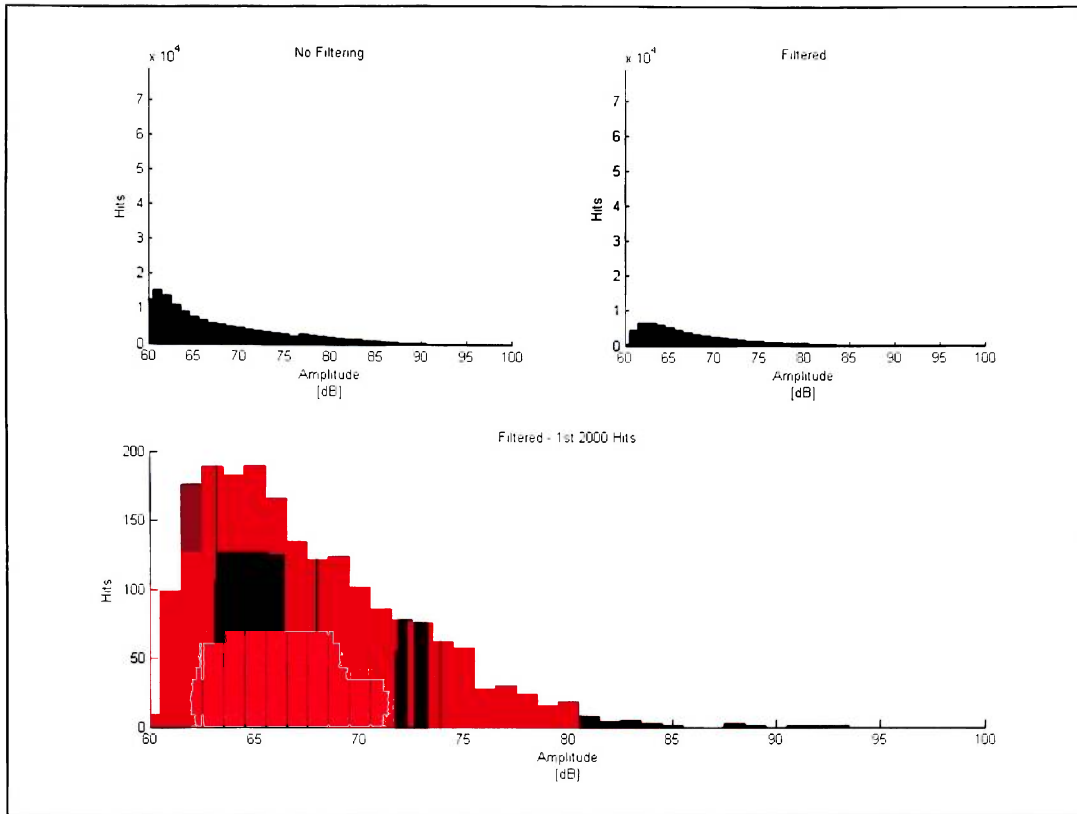


Figure 18: SN009 Amplitude Distributions

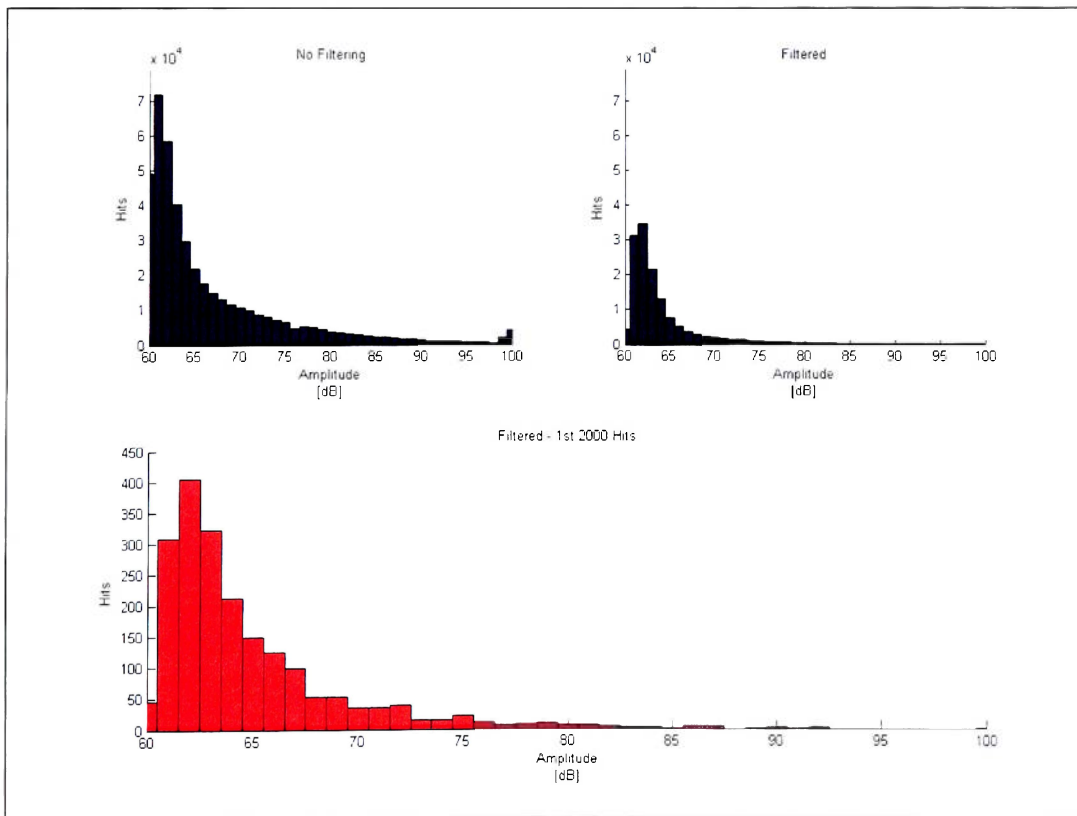


Figure 19: SN010 Amplitude Distributions

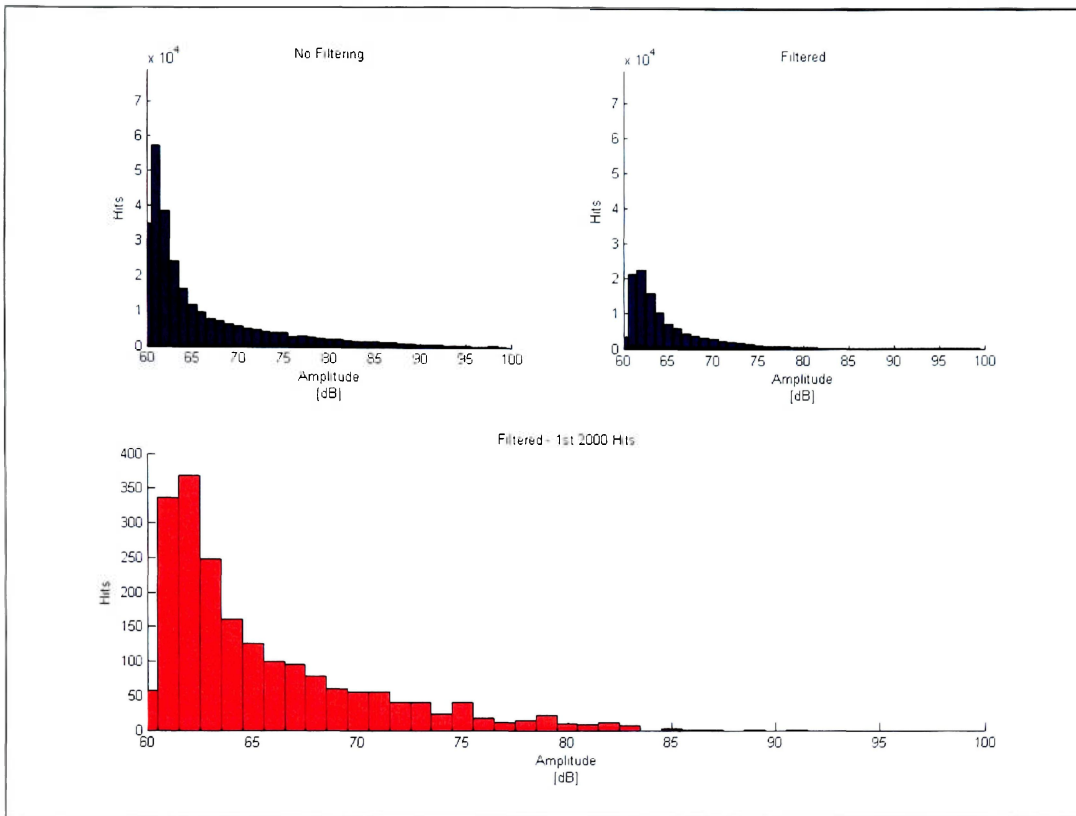


Figure 20: SN013 Amplitude Distributions

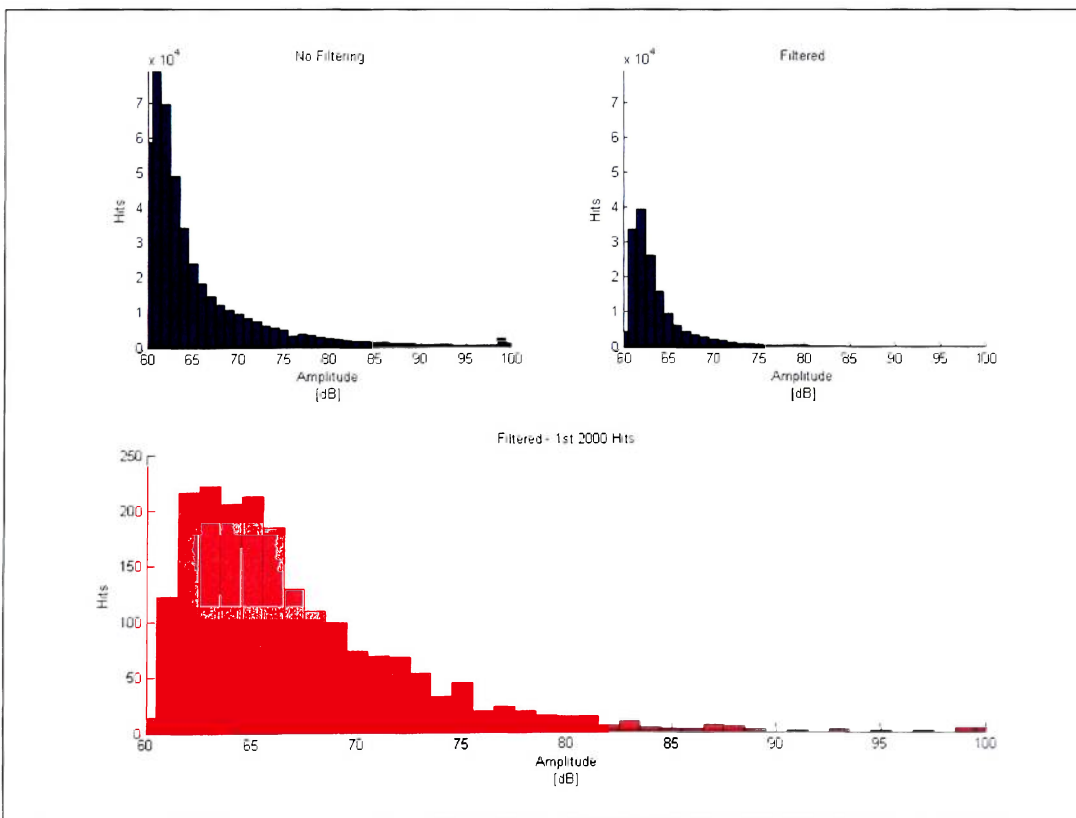


Figure 21: SN014 Amplitude Distributions

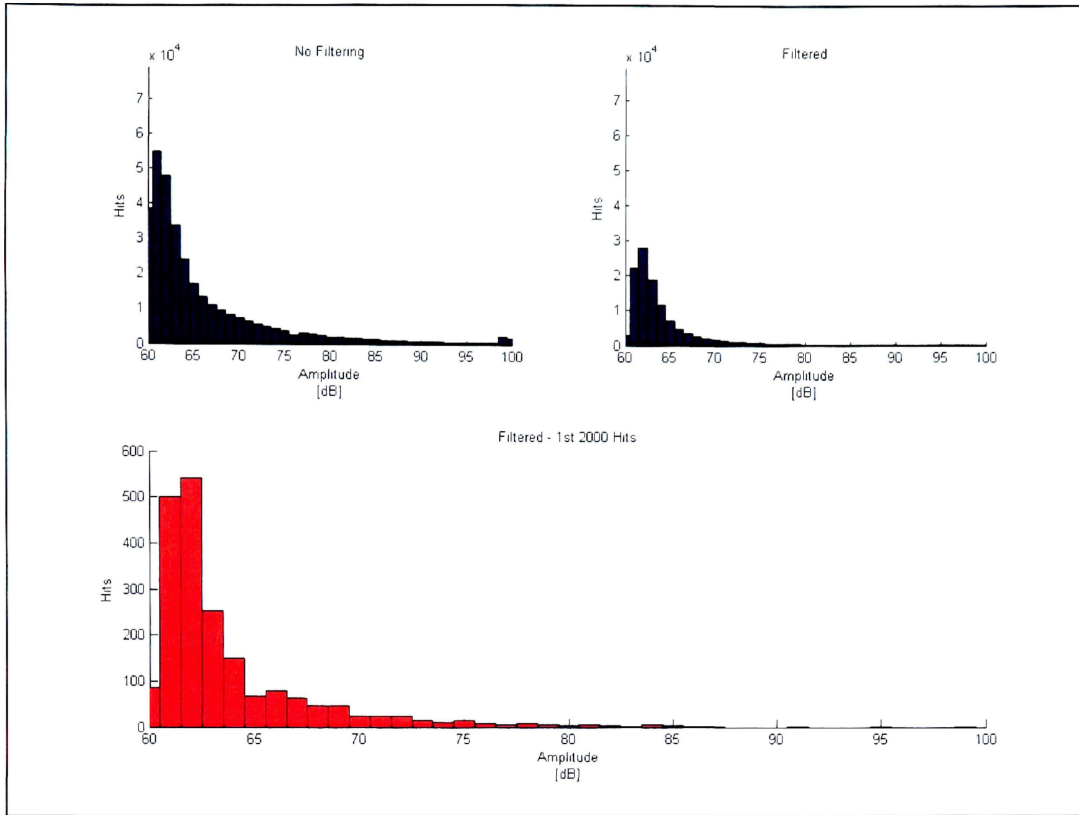


Figure 22: SN018 Amplitude Distributions

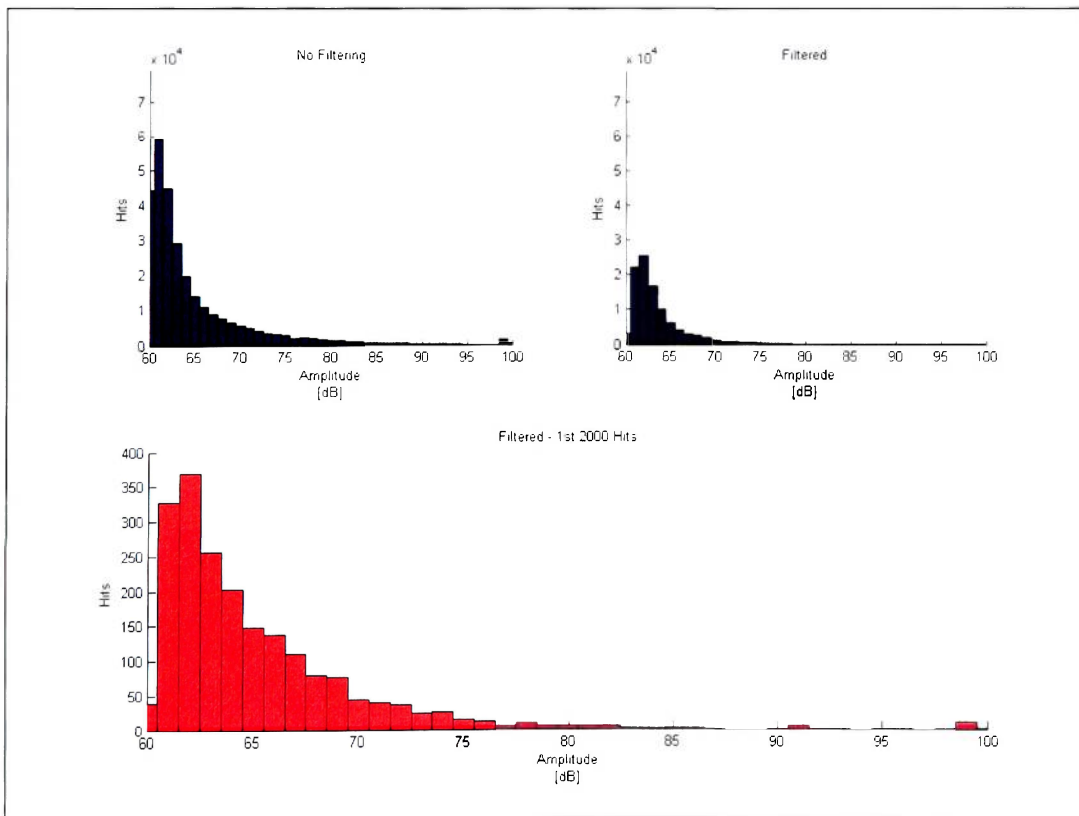


Figure 23: SN020 Amplitude Distributions

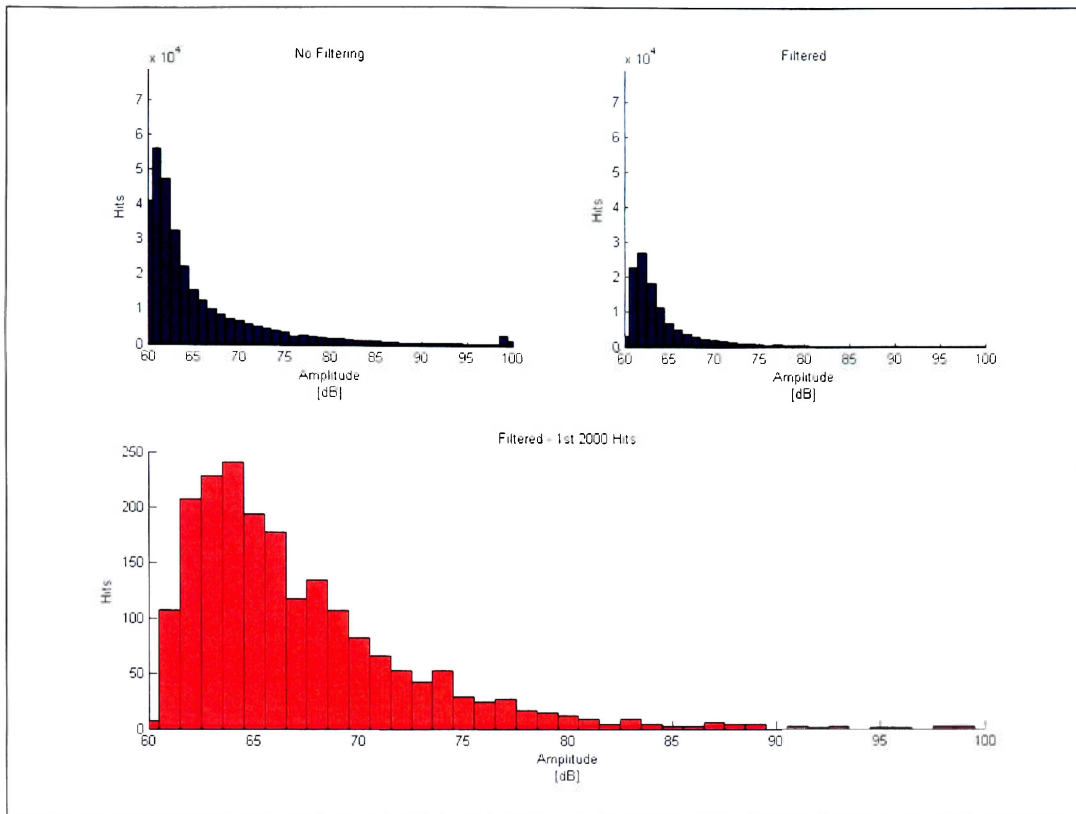


Figure 24: SN025 Amplitude Distributions

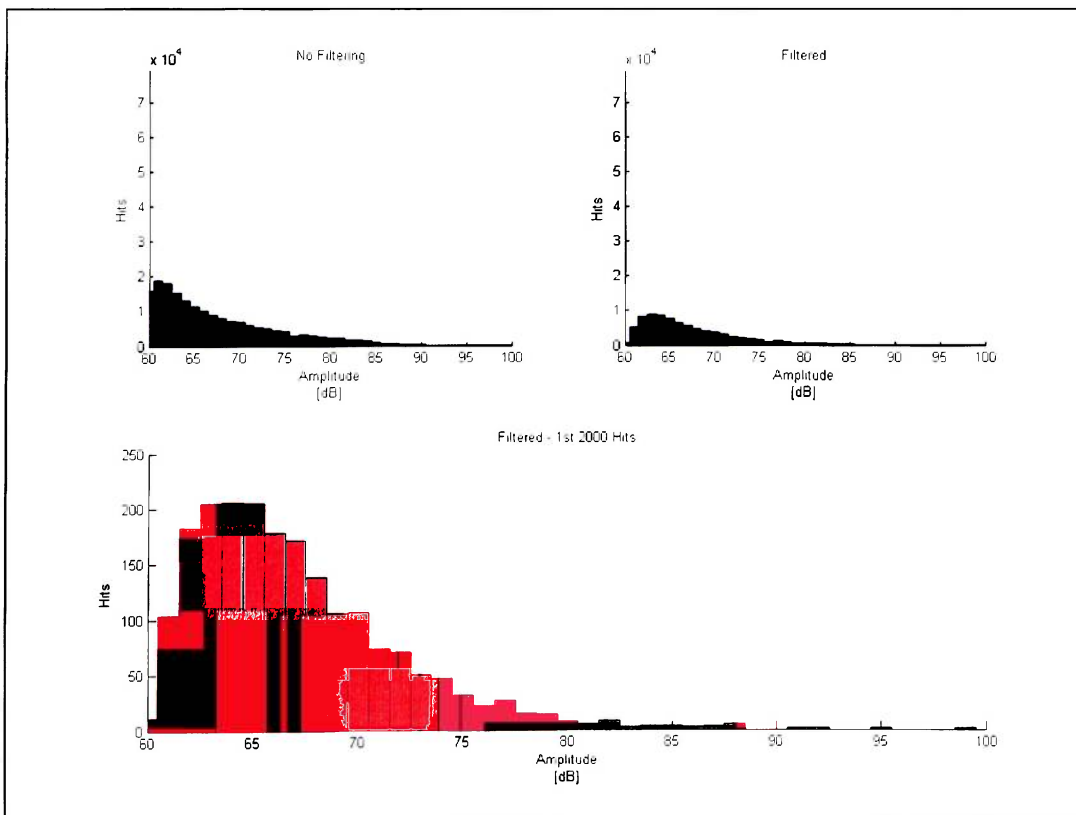
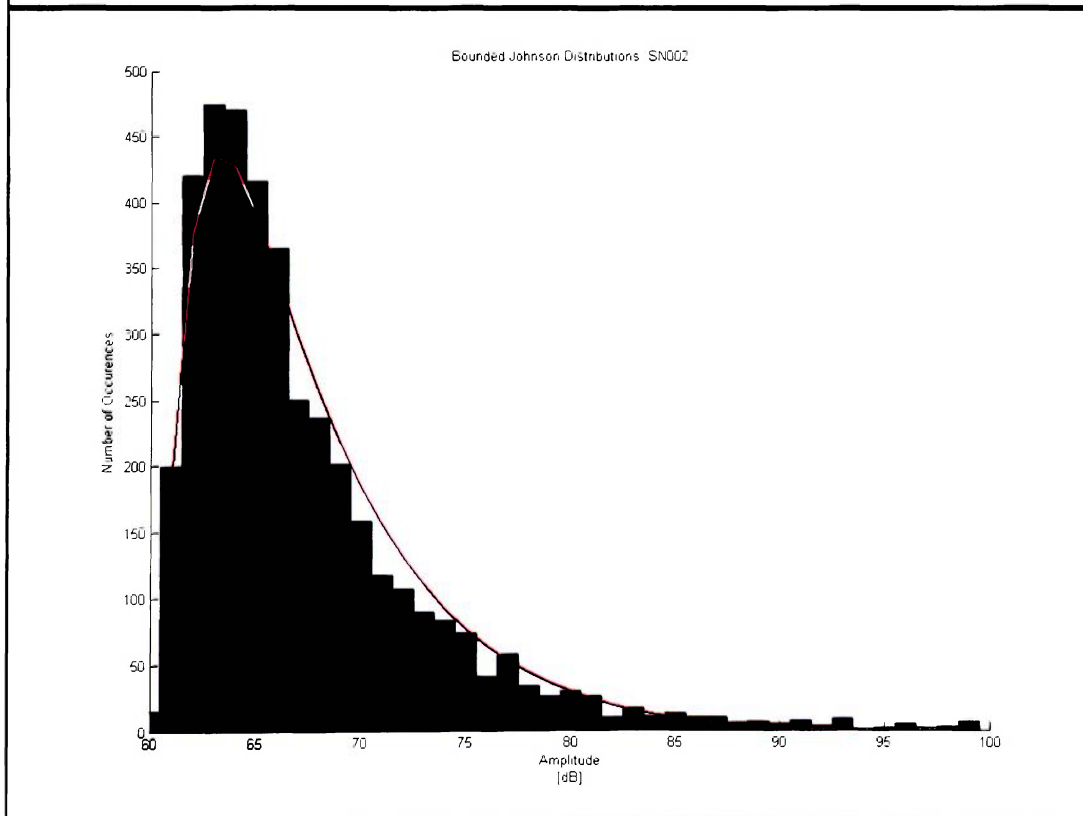
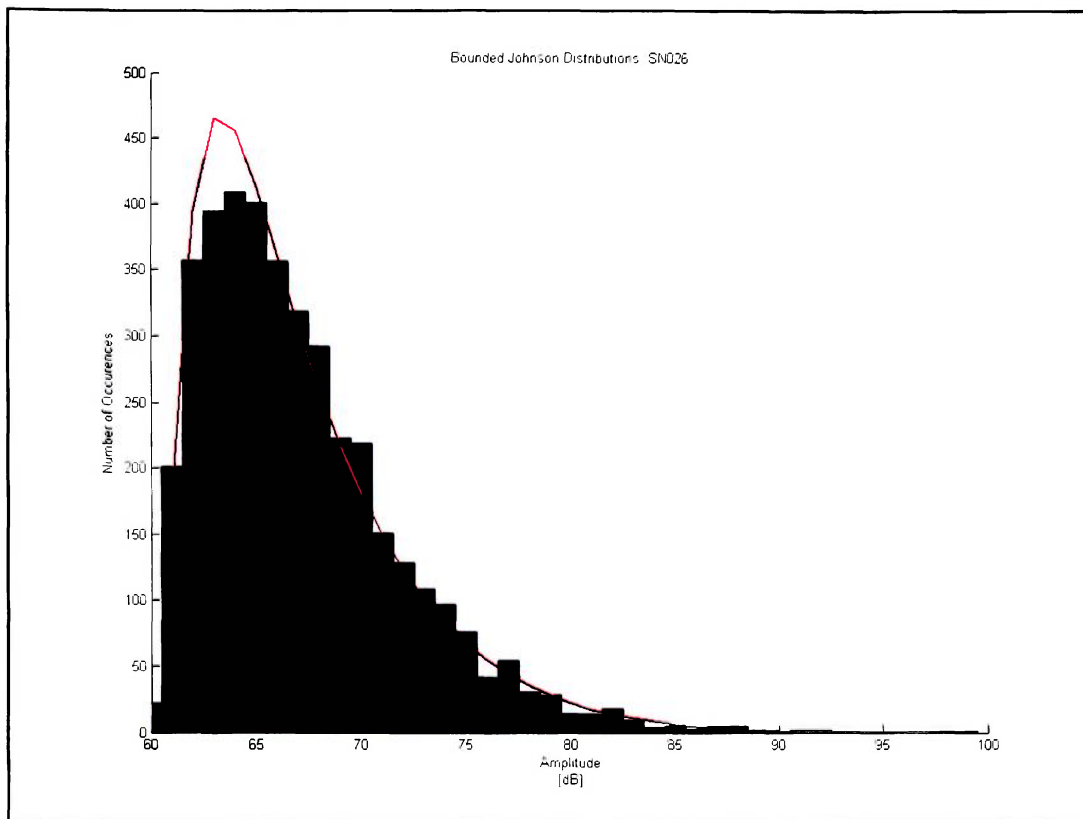
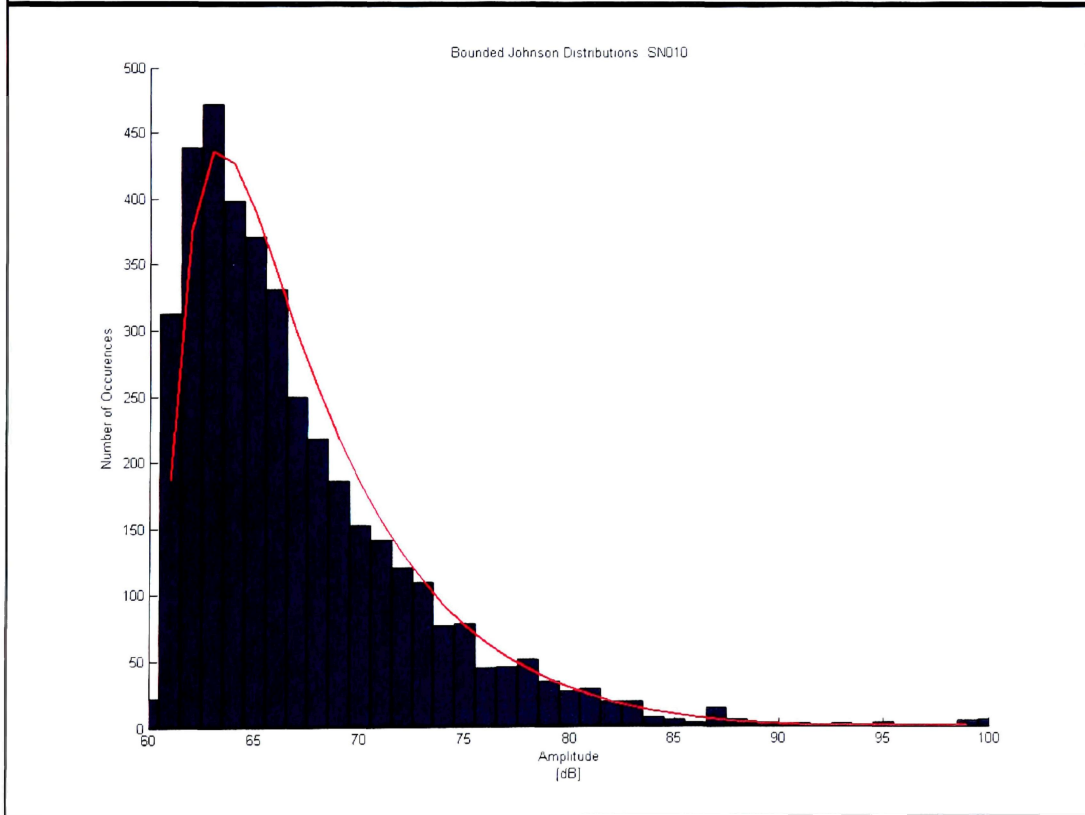
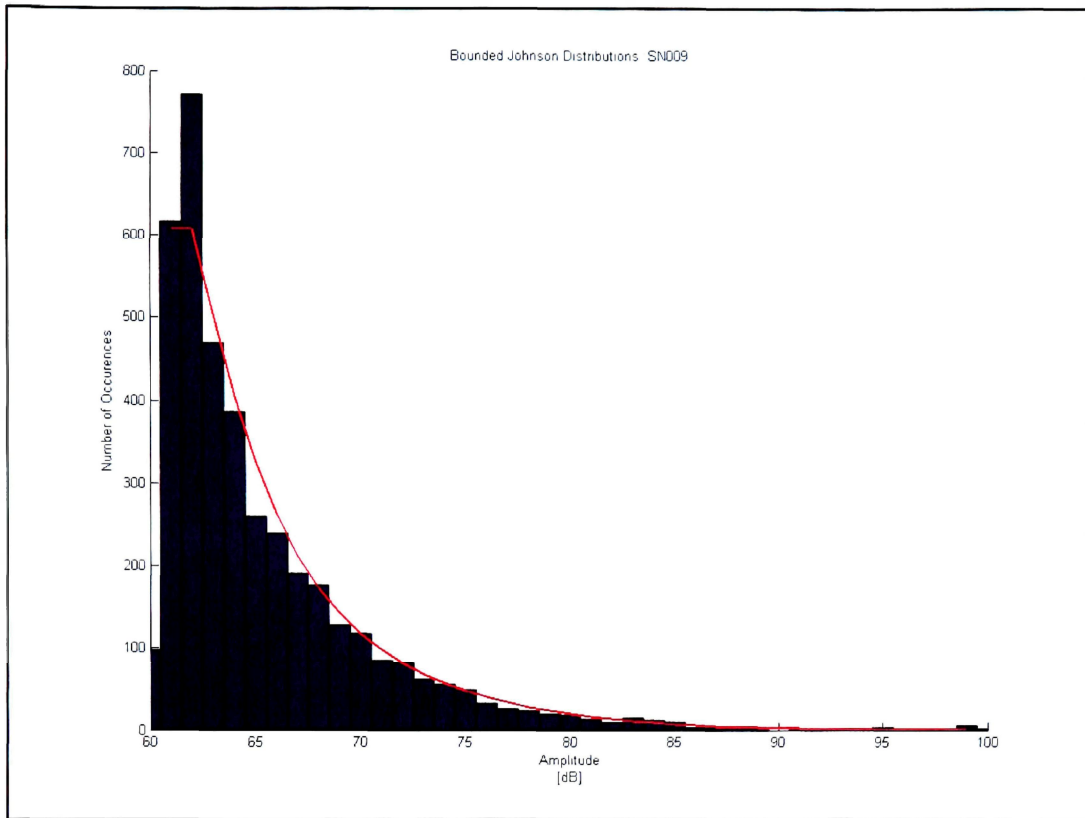


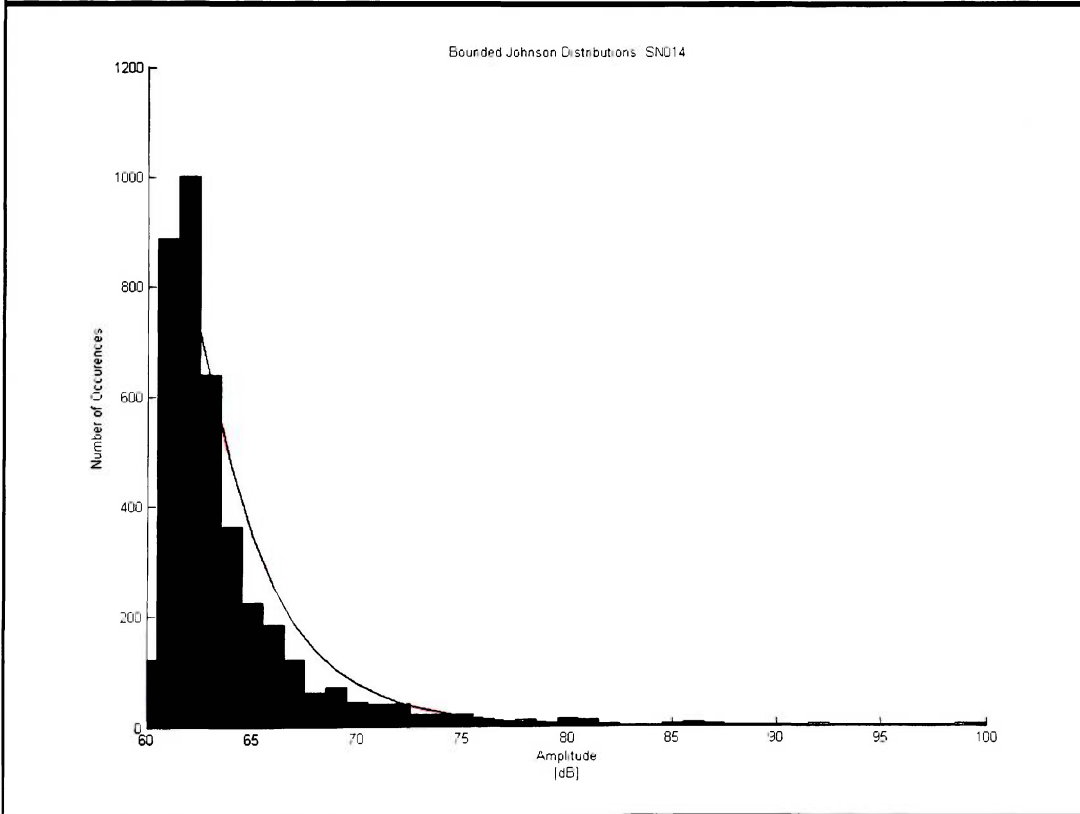
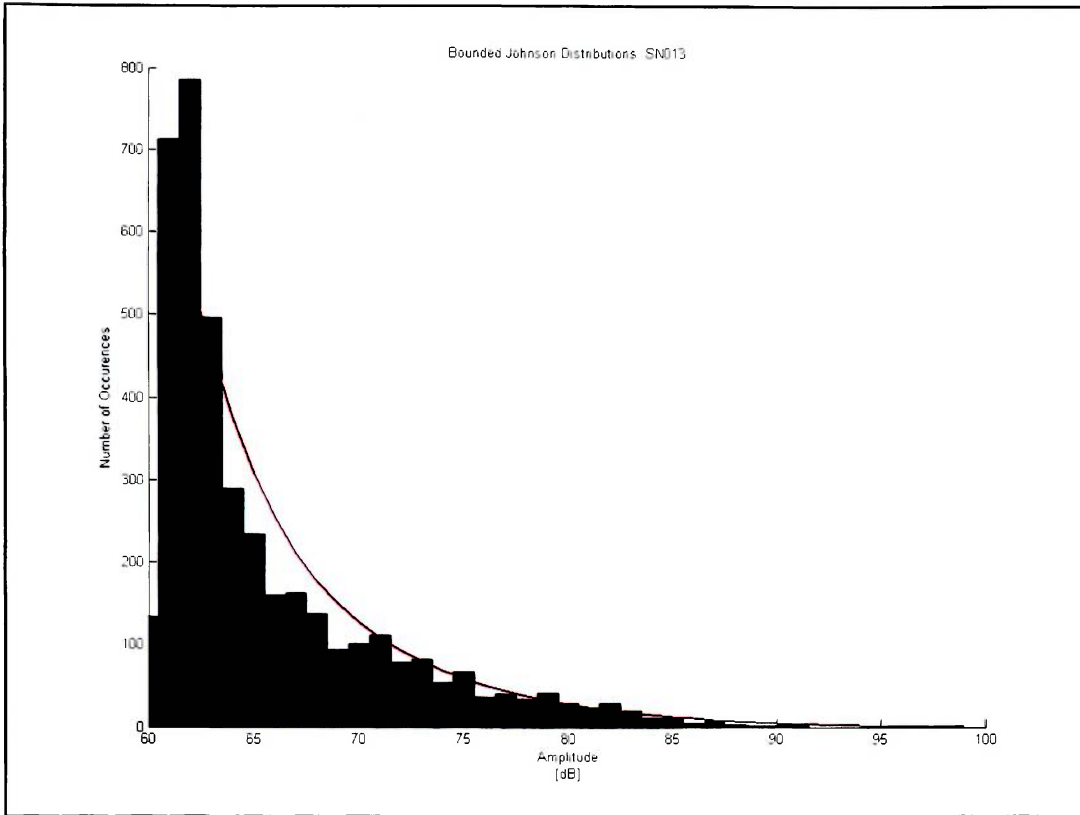
Figure 25: SN026 Amplitude Distributions

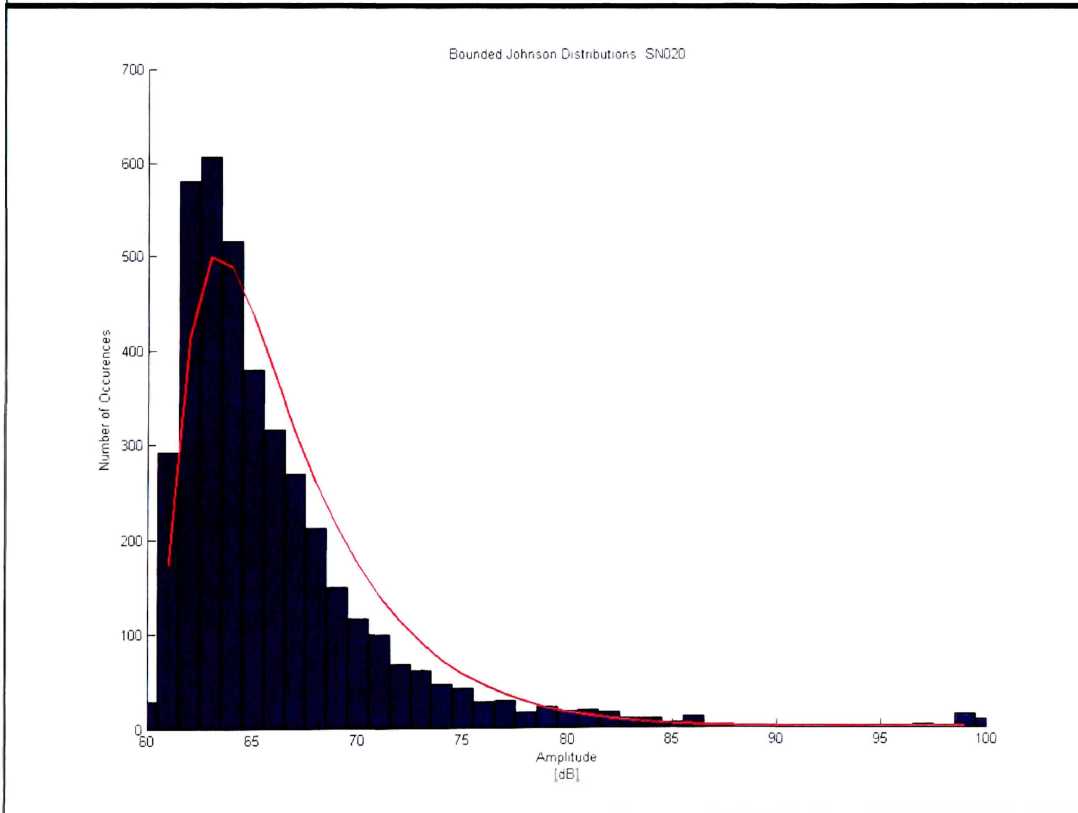
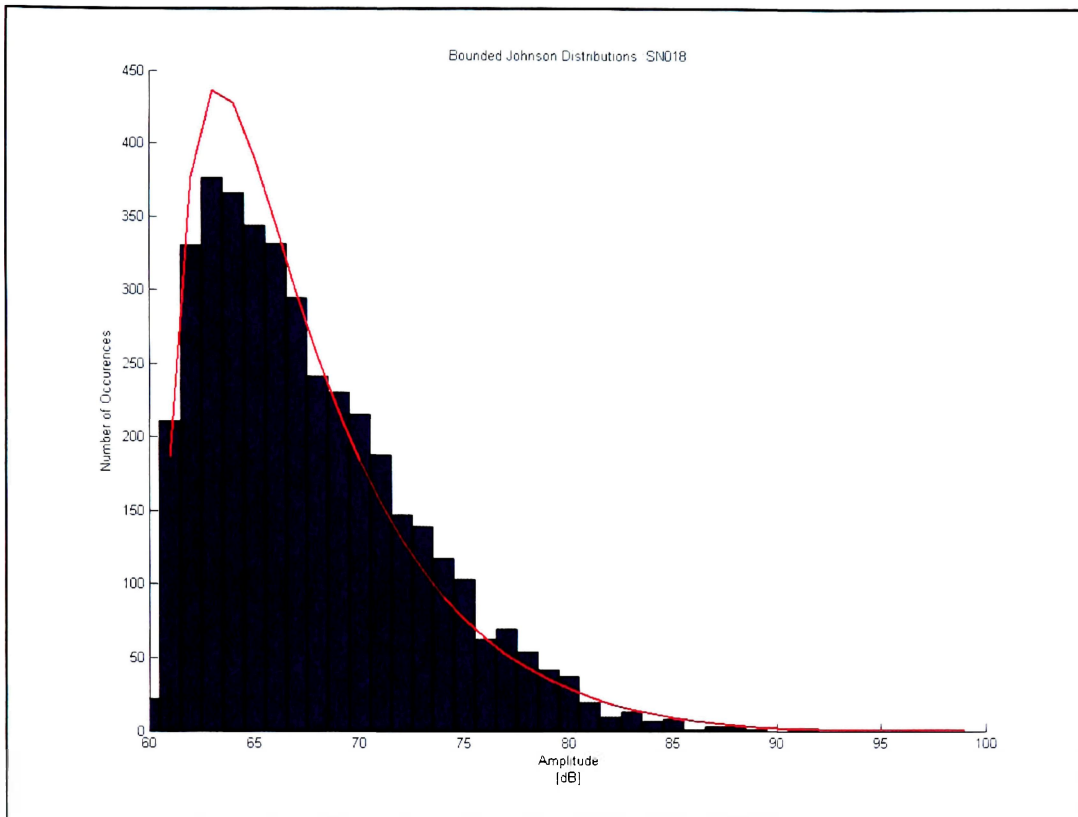
## 7.2 Bounded Johnson Distributions

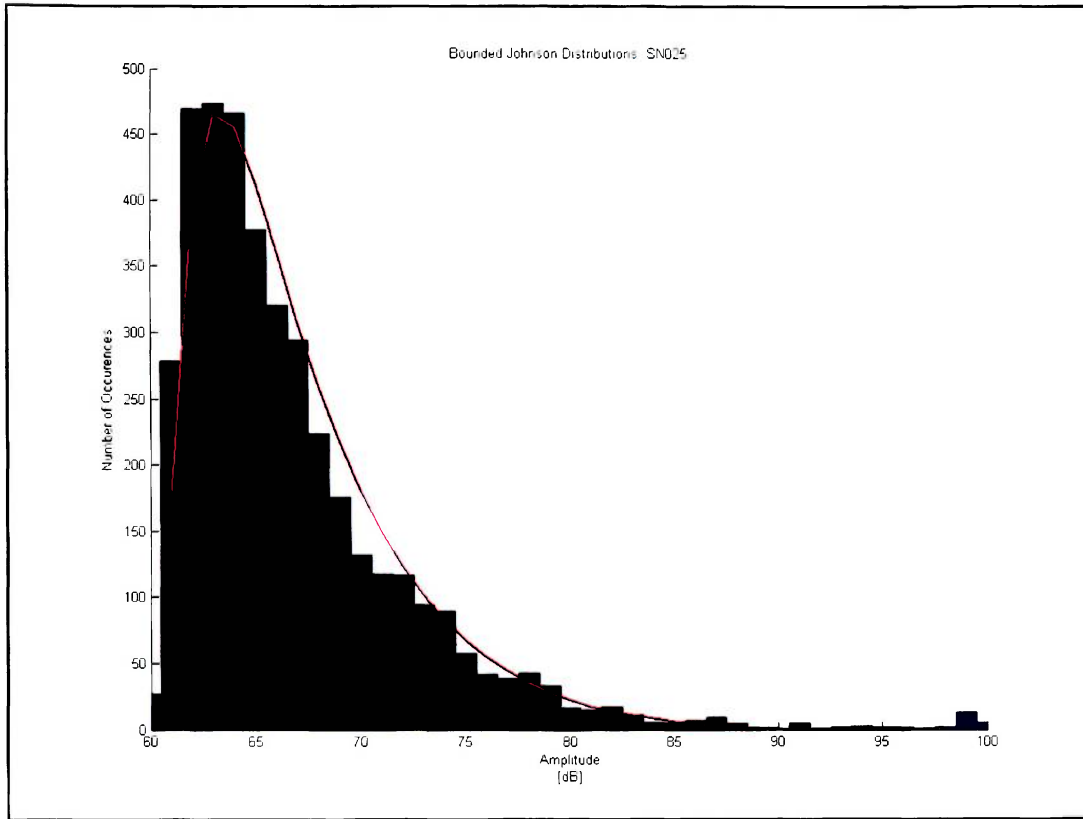












## 7.3 Perl Codes

### 7.3.1 Comma Delimiting Converter

```

# This program is intended to take the AE data that was converted from
# RAW.DTA into RAW.TXT and remove all but the selected transducer's
# data.
# The new file is named CHANNEL#_OriginalFileName.txt in a subdirectory
# April Fools, 2006
# Viperstd, a.k.a. CaptainInsane-O || GSSpectre

# Declare subroutines
sub ltrim($);
sub csv($);

system "cls"; # Clears the DOS screen
print "This is a test of the Perl enviroment.\n";

for ($Channel = 1;$Channel<8;$Channel++)
{

print "\nThe selected channel is: $Channel\n";

# Check if directory exists and if it doesn't, makes it
if (-d "Channel_$Channel")
{
    print "\nERROR ERROR ERROR ERROR ERROR ERROR\n";
    print "Directory Already Exists!!!";
    print "\nERROR ERROR ERROR ERROR ERROR ERROR\n\n\n";
    die
}
else
{
    print "Directory Channel_$Channel was successfully created.\n";
    mkdir("Channel_$Channel");
}

@filelist = glob("*.txt"); # list of all the .txt files in the dir
print "\nThe following files will be used:\n";
print @filelist;

$length=@filelist+0; # Converts the array of file names into a length

$i=0;

while ($i<=$length-1)
{
    print "\n-----Processing $filelist[$i]-----\n";
    $file = @filelist[$i];
    # Opens the original for reading
    open(ORIGINAL, $file) || die "\nCANNOT OPEN FILE!\n$!";
    $file_new = @filelist[$i];

```

```

# Opens/writes the new file
open(NEW, ">>Channel_{$Channel}/$file_new") || die "\nCANNOT OPEN FILE!\n$!";
$num_lines = 0;
# Reads through the original file line by line
while (<ORIGINAL>)
{
    $line = $_;
    # Checks if the data is for the correct transducer && that the
    # file is less than 65536 lines, if yes: written to new file
    if (($line =~ /\d\d{2}\s{9}$Channel\s/) && ($num_lines<65535)) # #.# followed by 9 spaces
    {
        $line = ltrim($line);
        $line = csv($line);
        $line = "\n";
        print NEW $line;
        $num_lines++;
    }
}

# Closes out the files when read/writes complete
close(NEW);
close(ORIGINAL);

    $i++;
}
}

# This subroutine removes the leading spaces and the first column
sub ltrim($)
{
    my $string = shift;
    $string =~ s/^\s+//;
    $string = substr($string,3);

    return $string;
}

# This subroutine converts from space-delim to csv
sub csv($)
{
    my $string = shift;
    $string =~ s/\s+/,/g;
    chop ($string);
    return $string;
}

```

### 7.3.2 Parameter Separator

*# This program is intended to take the csv, text files and convert them # into single column arrays named bottle\_attribute.txt*

*# November 2006*

*# Viperstd, a.k.a. CaptainInsane-O || GSSpectre*

system "cls"; *# Clears the DOS screen*

print "This is a test of the Perl enviroment.\n";

*@filelist = glob("\*.txt"); # Get a list of all the .txt files in the directory*

*\$length=@filelist+0; # Converts the array of file names into a length for looping*

\$i=0;

**while** (\$i<=\$length-1)

```
{
  print "\n-----Processing $filelist[$i]-----\n";
  $file = @filelist[$i];
```

*# Opens the original file for reading*

**open**(ORIGINAL, \$file) || **die** "\nCANNOT OPEN FILE!\n\$!";

*@raw\_data = <ORIGINAL>;*

*# remove the .txt from the file name*

*\$file\_new = "@filelist[\$i]";*

*chop \$file\_new;*

*chop \$file\_new;*

*chop \$file\_new;*

*chop \$file\_new;*

*# create filenames for coloumn arrays*

*\$file\_time = \$file\_new "\_time.txt";*

*\$file\_para = \$file\_new "\_para.txt";*

*\$file\_ch = \$file\_new "\_ch.txt";*

*\$file\_rt = \$file\_new "\_rt.txt";*

*\$file\_counts = \$file\_new "\_counts.txt";*

*\$file\_energy = \$file\_new "\_energy.txt";*

*\$file\_duration = \$file\_new "\_duration.txt";*

*\$file\_amp = \$file\_new "\_amp.txt";*

*# Opens/writes the new files*

**open**(TIME, ">>\$file\_time") || **die** "\nCANNOT OPEN FILE!\n\$!";

**open**(PARA, ">>\$file\_para") || **die** "\nCANNOT OPEN FILE!\n\$!";

**open**(CH, ">>\$file\_ch") || **die** "\nCANNOT OPEN FILE!\n\$!";

**open**(RT, ">>\$file\_rt") || **die** "\nCANNOT OPEN FILE!\n\$!";

**open**(COUNTS, ">>\$file\_counts") || **die** "\nCANNOT OPEN FILE!\n\$!";

**open**(ENERGY, ">>\$file\_energy") || **die** "\nCANNOT OPEN FILE!\n\$!";

**open**(DURATION, ">>\$file\_duration") || **die** "\nCANNOT OPEN FILE!\n\$!";

**open**(AMP, ">>\$file\_amp") || **die** "\nCANNOT OPEN FILE!\n\$!";

*# read in the file's content as an array:*

**foreach** \$row (@raw\_data)

```
{
```



```

chop($row);    #remove the trailing newline
# read in each value into the array & assign it to the proper file
($timing,$para,$ch,$rt,$counts,$energy,$duration,$amplitude)=split(/\./,$row);
printf TIME "$timing\n";
printf PARA "$para\n";
printf CH "$ch\n";
printf RT "$rt\n";
printf COUNTS "$counts\n";
printf ENERGY "$energy\n";
printf DURATION "$duration\n";
printf AMP "$amplitude\n";
}

#Close the open files when read/writes complete
close(TIME);
close(PARA);
close(CH);
close(RT);
close(COUNTS);
close(ENERGY);
close(DURATION);
close(AMP);
close(ORIGINAL);

#increment the counter (go to the next bottle's file)
$i++;
}

```

## 7.4 MATLAB Codes

### 7.4.1 *Frequency\_Analyzer.m*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program is designed to read in the contents of the
% files containing the column vectors of the parameterized data
% & then create the family of plots required to visually rep. it.
% It also generates the PFM distributions for the BPNN
% THIS VERSION USES MINIMAL PARAMETRIC FILTERING
% Author: Seth-Andrew T. Dion
% Date: November 2006
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;
clc;
close all;

lowerbound=input('Please enter the minimum frequency too allow
(min=0) [kHz]: ');
upperbound=input('Please enter the maximum frequency too allow
(max=1000)[kHz]: ');
primary_fr=input('Please enter the # of primary frequencies to
investigate : ');

tic

%
filelist=['SN002';'SN003';'SN005';'SN009';'SN010';'SN013';'SN014';'S
N018';'SN020';'SN025';'SN026']
burst_p = [1880;2004;2760;2544;2460;2874;2390;2864;1967;2393;2675];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Scan through all the files 1 at a time and determine Primary Fs %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

filelist=['SN002';'SN003';'SN005';'SN009';'SN010';'SN013';'SN014';'S
N018';'SN020';'SN025';'SN026']

for i=1:length(filelist)

fn_1=strcat(filelist(i,1),filelist(i,2),filelist(i,3),filelist(i,4),
filelist(i,5));
fn_2='_amp.txt';
fn_3='_ch.txt';
fn_4='_counts.txt';
fn_5='_duration.txt';
fn_6='_energy.txt';
fn_7='_para.txt';
fn_8='_rt.txt';
fn_9='_time.txt';
% strcat 'adds' the strings together to make compound file names
fnamp=strcat(fn_1,fn_2);
fnch=strcat(fn_1,fn_3);
fncounts=strcat(fn_1,fn_4);
fnduration=strcat(fn_1,fn_5);

```

```

fnenergy=strcat(fn_1,fn_6);
fnpara=strcat(fn_1,fn_7);
fnrt=strcat(fn_1,fn_8);
fntime=strcat(fn_1,fn_9);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      scan in all of the column vectors into arrays      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

amp=load(fnamp);
ch=load(fnch);
counts=load(fncounts);
duration=load(fnduration);
energy=load(fnenergy);
para=load(fnpara);
rt=load(fnrt);
time=load(fntime);

freq = floor(counts./(duration.*1e-6)/1000); %calculated based
on c&d

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculates the average frequencies                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
min_freq = min(freq);
max_freq = max(freq);

% remove frequency values that are not in the desired range
for j=length(freq):-1:1
    if ((freq(j,:)<lowerbound) | (freq(j,*)>upperbound) && j>0)
        freq(j,:)=[];
        amp(j,:)=[];
        ch(j,:)=[];
        counts(j,:)=[];
        duration(j,:)=[];
        energy(j,:)=[];
        para(j,:)=[];
        rt(j,:)=[];
        time(j,:)=[];
    end
end
% remove zero energy events
if (energy(j,:)==0)
    freq(j,:)=[];
    amp(j,:)=[];
    ch(j,:)=[];
    counts(j,:)=[];
    duration(j,:)=[];
    energy(j,:)=[];
    para(j,:)=[];
    rt(j,:)=[];
    time(j,:)=[];
end
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       write the filtered files for post processing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    freq_gl{i}=freq;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Handles the frequency groupings for avg. value determination %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% create the array of frequencies to group together
c=1;
for l=lowerbound:1:upperbound-1;
    freq_bins(c,1) = l;
    c=c+1;
end

freq_events = zeros(length(freq_bins),1);

for k=1:length(freq_bins)
    for c=1:length(freq)
        if (freq(c,1)==freq_bins(k,1))
            freq_events(k,1)=freq_events(k,1)+1;
        end
    end
end

freq_events_new(:,i)=freq_events(:,1);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculates the n-primary frequencies, where n is user input
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
freq_avg=mean(freq_events_new,2);
xy(:,1)=freq_bins;
xy(:,2)=freq_avg;
xy=sortrows(xy,2);
pf=xy(length(xy)-primary_fr:length(xy),:);
pf=sortrows(pf,1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% writes the frequency distribution files for the backpropagation
% neural network processing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:length(pf)
    pfm_bot2(i,1)=length(find(freq_gl{1}(:)==pf(i,1)));
    pfm_bot3(i,1)=length(find(freq_gl{2}(:)==pf(i,1)));
    pfm_bot5(i,1)=length(find(freq_gl{3}(:)==pf(i,1)));
    pfm_bot9(i,1)=length(find(freq_gl{4}(:)==pf(i,1)));
    pfm_bot10(i,1)=length(find(freq_gl{5}(:)==pf(i,1)));
    pfm_bot13(i,1)=length(find(freq_gl{6}(:)==pf(i,1)));
    pfm_bot14(i,1)=length(find(freq_gl{7}(:)==pf(i,1)));
end

```

```

    pfm_bot18(i,1)=length(find(freq_gl{8}(:)==pf(i,1)));
    pfm_bot20(i,1)=length(find(freq_gl{9}(:)==pf(i,1)));
    pfm_bot25(i,1)=length(find(freq_gl{10}(:)==pf(i,1)));
    pfm_bot26(i,1)=length(find(freq_gl{11}(:)==pf(i,1)));
end

dlmwrite('SN002_PFFM.txt', transpose(pfm_bot2), 'precision', '%.2f')
dlmwrite('SN002_PFFM.txt', burst_p(1,1), '-append', 'precision',
'%.2f')
dlmwrite('SN003_PFFM.txt', transpose(pfm_bot3), 'precision', '%.2f')
dlmwrite('SN003_PFFM.txt', burst_p(2,1), '-append', 'precision',
'%.2f')
dlmwrite('SN005_PFFM.txt', transpose(pfm_bot5), 'precision', '%.2f')
dlmwrite('SN005_PFFM.txt', burst_p(3,1), '-append', 'precision',
'%.2f')
dlmwrite('SN009_PFFM.txt', transpose(pfm_bot9), 'precision', '%.2f')
dlmwrite('SN009_PFFM.txt', burst_p(4,1), '-append', 'precision',
'%.2f')
dlmwrite('SN010_PFFM.txt', transpose(pfm_bot10), 'precision',
'%.2f')
dlmwrite('SN010_PFFM.txt', burst_p(5,1), '-append', 'precision',
'%.2f')
dlmwrite('SN013_PFFM.txt', transpose(pfm_bot13), 'precision',
'%.2f')
dlmwrite('SN013_PFFM.txt', burst_p(6,1), '-append', 'precision',
'%.2f')
dlmwrite('SN014_PFFM.txt', transpose(pfm_bot14), 'precision',
'%.2f')
dlmwrite('SN014_PFFM.txt', burst_p(7,1), '-append', 'precision',
'%.2f')
dlmwrite('SN018_PFFM.txt', transpose(pfm_bot18), 'precision',
'%.2f')
dlmwrite('SN018_PFFM.txt', burst_p(8,1), '-append', 'precision',
'%.2f')
dlmwrite('SN020_PFFM.txt', transpose(pfm_bot20), 'precision',
'%.2f')
dlmwrite('SN020_PFFM.txt', burst_p(9,1), '-append', 'precision',
'%.2f')
dlmwrite('SN025_PFFM.txt', transpose(pfm_bot25), 'precision',
'%.2f')
dlmwrite('SN025_PFFM.txt', burst_p(10,1), '-append', 'precision',
'%.2f')
dlmwrite('SN026_PFFM.txt', transpose(pfm_bot26), 'precision',
'%.2f')
dlmwrite('SN026_PFFM.txt', burst_p(11,1), '-append', 'precision',
'%.2f')
code_time=toc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plots
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hold all

scatter(pf(:,1),pf(:,2))
plot(freq_bins,freq_avg)
scatter(pf(:,1),pfm_bot2(:,1))

```

```
scatter(pf(:,1),pfm_bot3(:,1))  
scatter(pf(:,1),pfm_bot5(:,1))  
scatter(pf(:,1),pfm_bot9(:,1))  
scatter(pf(:,1),pfm_bot10(:,1))  
scatter(pf(:,1),pfm_bot13(:,1))  
scatter(pf(:,1),pfm_bot14(:,1))  
scatter(pf(:,1),pfm_bot18(:,1))  
scatter(pf(:,1),pfm_bot20(:,1))  
scatter(pf(:,1),pfm_bot25(:,1))  
scatter(pf(:,1),pfm_bot26(:,1))
```

## 7.4.2 Johnson\_dsitribution.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Calculate the Johnson Distribution of a bottle filtered to 91kHz%
% Author: Seth-Andrew T. Dion %
% Date: November 2006 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;
clc;
close all;

%Channel 1, 91kHz filtering, bottle SN002
%
A=[207;1451;1766;1273;933;767;617;498;414;354;279;247;233;191;158;
131;85;116;74;78;56;52;40;35;27;20;26;23;19;14;10;10;7;10;4;7;6;2;
8;6;15;];
A=[259;1994;2498;1750;1169;727;529;394;337;254;218;218;179;147;114
;91;55;66;44;31;23;27;25;15;16;13;9;7;6;6;2;4;4;2;4;0;1;4;2;1;1];
B=[60;61;62;63;64;65;66;67;68;69;70;71;72;73;74;75;76;77;78;79;80;
81;82;83;84;85;86;87;88;89;90;91;92;93;94;95;96;97;98;99;100];

n=max(size(A));
A_sum = sum(A);
%calculate the estimation of central moments from data
m2 = 1/n*sum(A.^2)-1/n^2*(sum(A))^2;
m3 = sum(A.^3)/n-3*sum(A.^2)/n*sum(A)/n+2*(sum(A)/n)^3;
m4 = sum(A.^4)/n -
4*sum(A)/n*sum(A.^3)/n+6*(sum(A)/n)^2*sum(A.^2)/n-3*(sum(A)/n)^4;

beta1 = (m3/m2^(3/2))^2; %relative skewness
beta2 = m4/m2^2; %relative kurtosis (peakedness)

% based on these values of beta, use Fig 6-2 pg 200 Hahnn &
Shapiro
% to determine Johnson shape -->Sb (bounded Johnson) region
% for Sb, epsilon=0 (lower bound is 0 data pts, lambda unknown

eps = 59.99; % this is the lower bound of the data.
lambda = 40.01; %the upper limit of the data set
x_5 = median(A);

% Calculate x_a based on the 9% data
i=1;
value = A(i);
while value<=sum(A)*.09
    value=value+A(i+1);
    i=i+1;
end
x_a = B(i);
% Calculate x_1-a based on the 91% data
i=1;
value = A(i);
while value<=sum(A)*.91
    value=value+A(i+1);
    i=i+1;

```

```

end
x_l_ap = B(i);

% values read off of the normal-distribution tables
z_l_ap = 1.34;
z_a = -1.34;

ada_hat_p1 = (z_l_ap - z_a);
ada_hat_p2_n = (x_l_ap-eps)*(eps+lambda-x_a);
ada_hat_p2_d = (x_a-eps)*(eps+lambda-x_l_ap);
ada_hat = ada_hat_p1/(log(ada_hat_p2_n/ada_hat_p2_d));

gamma_hat = z_l_ap-ada_hat*log((x_l_ap-eps)/(eps+lambda-x_l_ap));
% Eq 6.27

for i=1:1:n

    f1(i,1)=ada_hat/((2*pi)^.5);
    f2(i,1)=lambda/((B(i)-eps)*(lambda-B(i)+eps));
    f3(i,1)=exp(-.5*(gamma_hat+ada_hat*log((B(i)-eps)/(lambda-
B(i)+eps))))^2);
end

f=f1.*f2.*f3;
% f=max(A)/max(f).*f;
f=sum(A).*f
% P = sum(A)*1*f;

% Plotting
bar(B,A)
hold on
plot(B,f,'r',...
'LineWidth',3)
hold off

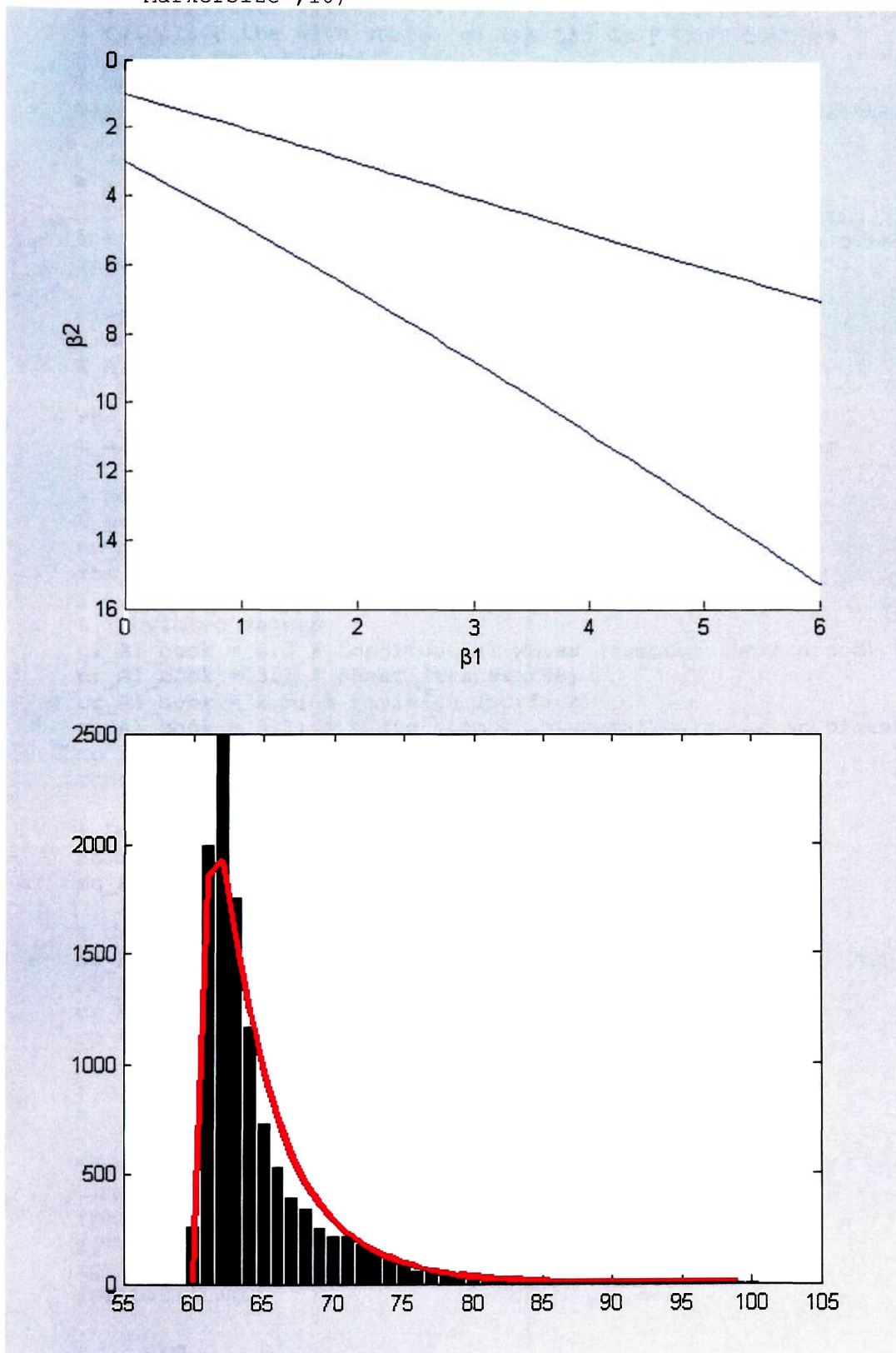
%plots the johnson distribution curves
omega = 1:.01:1.5;
b1 = (omega-1).*(omega+2).^2;
b2 = omega.^4+2.*omega.^3+3.*omega.^2-3;

% Create figure
figure2 = figure;
% Create axes
axes1 = axes('YDir','reverse','Parent',figure2);
% Create plot
hold on
plot(b1,b2)
plot(b1,b1+1)
ylim(axes1,[0 16]);
xlim(axes1,[0 6]);
xlabel(axes1,'\beta1');
ylabel(axes1,'\beta2');
hold(axes1,'all');

```



```
plot(beta1,beta2,':bs',...  
      'MarkerSize',10)
```



### 7.4.3 Wave Speed

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate the wave speeds of the 15" Gr/E COPV Bottles
% Author: Seth-Andrew T. Dion
% Date:   November 2006
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% NOTE: in an infinite medium two wave types propagate
% Dilational - Longitudinal, volume changes, angles remain fixed
% Equivoliminal - transverse (t) or shear (2D), volume fixed,
angles change

clear;clc;
% Material Properties of 6061-T6 Al
E_Al = 68.9e9; %Pa
nu_Al = 0.33;
rho_Al = 2700; %kg/m^3
t_Al = .12*(1/(25.4*1000)) % conversion to m from inches

% Material Properties of Hexcel Gr/Epoxy AS4
E_ge = 231e9; % Pa
nu_ge = 0.33; %
rho_ge = 1780; %kg/m^3

% Tabulated values
cl_Al_book = 6.3 % Longitudinal waves (compressional/sound)
ct_Al_book = 3.1 % shear (transverse)
cr_Al_book = 2.9; % rayleigh (surface)
cp_Al_book = 5.1; % plate (Lamb...dispersive, occur in plates due
to finite t)
rhocl_Al_book = 17; %acoustic impedance 10^6 kg/(m^2*s)

% Lamé's constants, Al
lambda_Al = E_Al*nu_Al/((1+nu_Al)*(1-2*nu_Al));
mu_Al = E_Al/(2*(1+nu_Al));

% Calculated wave speeds
cl_Al = sqrt((lambda_Al+2*mu_Al)/rho_Al); % Longitudinal WS, m/s
ct_Al = sqrt(mu_Al/rho_Al); % Transverse WS, m/s
cr_Al = ((0.87+1.12*nu_Al)/(1+nu_Al))*ct_Al; % Rayleigh WS, m/s
cp_Al = ct_Al*sqrt(2/(1-nu_Al)); % Lamb WS, m/s

% group velocity curves
% v = cp+Al - lambda*

fprintf('The following wave speeds were calculated for an Aluminum
Liner:\n')
fprintf('Longitudinal:      %g m/s\n',cl_Al)
fprintf('Transverse:         %g m/s\n',ct_Al)
fprintf('Rayleigh:              %g m/s\n',cr_Al)
fprintf('Lamb:                  %g m/s\n',cp_Al)

```

*The following wave speeds were calculated for the Aluminum Liner:*

Longitudinal: 6148.93 m/s  
Transverse: 3097.33 m/s  
Rayleigh: 2886.8 m/s  
Lamb: 5351.36 m/s

*Published with MATLAB® 7.0*

#### 7.4.4 TWPV\_burst\_pressure\_prediction.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate the theoretical burst pressure of a COPV
% Author: Seth-Andrew T. Dion
% Date: November 2006
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
clc
close all
format short g

% Bottle Properties
n_hoop = 5; % # of hoop plies
n_polar = 2; % # of polar plies
d = 15; % diameter, inches

% Material Properties
t_hoop = .007; % ply thinckness, inches
t_polar = t_hoop;
x_t = 310; % ksi
x_c = -240; % ksi
y_t = 8.7; % ksi
y_c = -8.7; % ksi
s12 = 18.5; % ksi

% thin walled pressure vessels
syms sig2
sig1 = 2*sig2;
t12 = 0;
f = 0; % default value

% Calculate Fs
F1 = 1/x_t + 1/x_c;
F2 = 1/y_t + 1/y_c;
F11 = -1/(x_t*x_c);
F22 = -1/(y_t*y_c);
F66 = 1/(s12)^2;
F12 = f*sqrt(F11*F22);

% Failure Envelope
% matrix splitting occurs at these values
A=solve(F1*sig1+F2*sig2 +
F11*sig1^2+F22*sig2^2+F66*t12^2+2*F12*sig1*sig2-1, sig2);
sig2 = double(A)
sig1 = 2*sig2

% Calculate pressure
% sig1 = pr/t
P = 2*sig1*(n_hoop*t_hoop+n_polar*t_polar)/(d-
2*(n_hoop*t_hoop+n_polar*t_polar))*1000 %psi

% Calculate ultimate failure pressure (hoop plies)
clear sig1 sig2
sig2 = 0;
syms sig1

```

```
A=solve(F1*sig1+F2*sig2 +  
F11*sig1^2+F22*sig2^2+F66*t12^2+2*F12*sig1*sig2-1, sig1);  
sig1 = double(A)  
P = max(2*sig1*(n_hoop*t_hoop+n_polar*t_polar)/(d-  
2*(n_hoop*t_hoop+n_polar*t_polar))*1000) %psi  
  
sig2 =  
  
    8.7536  
   -8.6117  
  
sig1 =  
  
    17.507  
   -17.223  
  
P =  
  
    115.13  
   -113.27  
  
sig1 =  
  
    310  
   -240  
  
P =  
  
    2038.7
```