Development and Implementation of Metrics for Identifying Military Impulse Noise

Project WP-1585

University of Pittsburgh

Jeffrey S. Vipperman
Mathew A. Rhudy
Brian A. Bucci

2-September 2010

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
10-06-2010	Final	Mar 2005-Jun 2010

4. TITLE AND SUBTITLE

Development and Implementation of Metrics to Identify Impulse Noise

5a. CONTRACT NUMBER

W912HQ-07-C-0004

5b. GRANT NUMBER

SERDP SI-1585

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Vipperman, Jeffrey S.
Rhudy, Mathew B.
Bucci, Brian A.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Pittsburgh
3520 Fifth Avenue
Pittsburgh, PA 15213-3320

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Strategic Evironmental Research and Development Program
901 North Stuart Street, Suite 303
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)

SERDP

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Ultimately will be Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project sought to more accurately classify military noise. A high-fidelity library of measured noise waveforms was created. Four signal metrics (kurtosis, crest factor, spectral slope, and weighted square error) were identified as features that can be used for noise classification using an artificial neural network (ANN). Two prototype systems were fabricated and deployed. The University of Pittsburgh (UPitt) algorithm was combined with the Bearing Amplitude and Measurement Array System (BAMAS) (Applied Physical Sciences, Inc.) in a PC/104 based hardware platform. Two prototypes were deployed at Marines Corps Base Camp Lejeune. While the overall accuracy of the UPitt classifier alone was somewhat lower (89%) than desired, nonblast noise was rejected at over a 99% rate (which meets the objective to reduce false positives). When the UPitt and BAMAS algorithms were combined, blast noise was identified at over 98% accuracy, while aircraft, wind, and vehicle noise were rejected at over 98-99% accuracy. The device is at Technology Readiness Level 7, ready for demonstration and validation, before being commercialized.

15. SUBJECT TERMS

Community Noise, Impulse Noise, Noise Classifier, Noise Measurement, Noise Monitoring, Annoyance, Artificial Neural Network, Bayesian Classifier, Signal Detection, Noise measurements

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jeffrey Vipperman
					19b. TELEPHONE NUMBER *(Include area code)* 412-624-1643

Reset

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Table of Contents

List of Acronyms

AC	Alternating current
ADC	Analog to digital converter
ANN	Artificial neural network
API	Applications programming interface
APS	Applied Physical Sciences, Inc.
BAMAS	Noise bearing and amplitude measurement and analysis system
BEAR	Boom Event Analyzer Recorder
BLAM	Blast analysis and monitoring algorithm
CF	Crest factor
CPU	Central processing unit
CST	Central Standard Time
DAQ	Data aquisition
DFR	Decimated, filtered, and re-sampled
DOA	Direction of arrival
DoD	Department of Defense
DSP	Digital signal processor
ERDC-CERL	Engineer Research and Development Center/Construction
FLOP	Floating point operation
FFT	Fast Fourier transform
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FtC	Fort Carson, CO
GIS	Geographic Information System
GMM	Gaussian mixture model
Hz	Hertz
JASA	Journal of the Acoustical Society of America
kHz	kilohertz
L_{DN}	Day-night sound level
L_{pk}	Peak sound level
LSLC	Least-square linear classifier
MCBCL	Marine Corps Base Camp Lejeune
MFLOP	Million floating point operations
MHz	Megahertz
MLP	Multilayer perceptron ANN
NI	National Instruments, Inc.
PC/104	Embedded computer standard governed by the PC/104 consortium
PDF	Probability density function
PSD	Power spectral density function
RF	Radio Frequency
RMS	Root-mean-square

RMTK	Range managers toolkit
ROC	Receiver Operating Characteristic
SBC	Single board computer
SEED	SERDP Exploratory development
SERDP	Strategic Environmental Research and Development Program
SLM	Sound level meter
SNR	Signal to Noise Ratio
SOM	Self-organizing map ANN
SON	Statement of Need
SVM	Support vector machine ANN
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
tANN	Temporal ANN
UPitt	University of Pittsburgh
USA-CHPPM	US Army Center for Health Promotion and Preventive Medicine
WSE	Weighted square error

List of Figures

List of Tables

Key Words

Community Noise, Impulse Noise, Noise Classifier, Noise Measurement, Noise Monitoring, Annoyance, Artificial Neural Network, Bayesian Classifier, Signal Detection, Noise measurements

Acknowledgements[1]

This research was supported wholly by the U.S. Department of Defense, through the Strategic Environmental Research and Development Program (SERDP). The authors gratefully acknowledge this financial support, as well as the support of many individuals that have helped make this research possible, including:

- Dr. John Hall, Sustainable Infrastructure Program Manager SERDP/ESTCP
- Dr. Robert Holst, Former Program Manager, SERDP/ESTCP
- Mr. Ed Nykaza (current COR), Army ERDC-CERL
- Dr. Larry Pater (past COR), Army ERDC-CERL
- Dr. William A. Russell, Sc.D., former Program Manager Army Operational Noise, USACHPPM
- Dave Whiteford, USACHPPM
- Gary Oles, former Range Control officer, MCBCL
- Paul Peterson, MCBCL
- David Salzman, Deputy Director of Range Control, MCBCL
- Michael Wannamaker, Range Control Operations Chief, MCBCL
- Col. William Yearwood, FTIG, PA
- John Fronko, FTIG, PA
- LtCol Jamison, Range Control, FTIG
- James Benford, Range Control, Fort Collins, CO
- Scott Clark, CEM Energy, Noise and P2 Program Coordinator Fort Collins, CO
- Carlos Rivero-deAguilar, Chief, Environmental Compliance, Restoration and Prevention Division (ECRPD) Fort Collins, CO
- Carin Shoemaker, Noise Manager, Fort Riley, KS
- Mr. Robert Kennedy, Noise Manager, Fort Hood, TX
- Sergio Sergi, Fort AP Hill, VA
- Mel Soult, OIC, Demolition Site 71-A, Fort AP Hill, VA
- Jeffrey Allanach, APS
- Bruce Abraham, Senior Scientist, APS
- Justin Borodinsky, APS
- Ellis Leeder, Fort Benning, GA
- The SERDP Program support staff of Hydrogeologic, Inc. (Kristen Lau, Johnathon Thigpen, Carrie Wood, Susan Walsh)

[1] Our apologies for to anyone that was inadvertently left off

1. Abstract

This project was a SERDP exploratory development (SEED) continuation that sought to more accurately classify military noise. During the SEED project, a library of recorded noise sources was created. In addition, four signal metrics (kurtosis, crest factor, spectral slope, and weighted square error) were identified as features that can be used for noise classification using an artificial neural network (ANN). The specific aims of the current project were to 1) expand the measurement library, 2) refine the SEED classifiers, 3) establish algorithm hardware requirements, 4) perform a real-time laboratory implementation, and 5) demonstrate two prototypes. In all, 11,600 waveforms were recorded. Several ANN structures were investigated, including multi-layer perceptron (MLP), self-organizing map (SOM), support vector machine, and an image recognition network. A Bayesian classifier was also created that used Gaussian mixture models to fit the multivariate data. The MLP, SOM, and Bayesian classifiers were found to perform to nearly 100% accuracy during development. The University of Pittsburgh (UPitt) algorithm was combined with the Bearing Amplitude and Measurement Array System (BAMAS) (Applied Physical Sciences, Inc.) in a PC/104 based hardware platform. Two prototypes were deployed at Marines Corps Base Camp Lejeune. Both algorithms performed well individually. While the overall accuracy of the UPitt classifier alone was somewhat lower (89%) than desired, nonblast noise was rejected at over a 99% rate (which meets the objective to reduce false positives). When the UPitt and BAMAS algorithms were combined, blast noise was identified at over 98% accuracy, while aircraft, wind, and vehicle noise were rejected at over 98-99% accuracy. The device is at Technology Readiness Level 7, ready for demonstration and validation, before being commercialized.

2. Objectives

The over-arching goal of this project was to support military readiness by providing better management of noise encroachment issues for military bases. It was also desired to improve accuracy of noise monitoring while requiring less human interpretation. An ancillary benefit was to provide a means to verify noise prediction models that are used for impact studies.

This 2.9-year project represented a continuation of a Strategic Environmental Research and Development Project (SERDP) exploratory development (SEED) effort [SERDP (SI-1436)] to improve community noise monitoring around military bases. The scientific objective was to continue development of an improved, autonomous military impulse noise detector. Toward this goal, five primary technical objectives for the project were identified (see Table 1). The work was divided into two phases with a go/no-go point in between. The project commenced on May 1, 2007 with prototypes deployed in Oct-Dec 2009 and final data analyses completed in June 2010.

Table 1. Project Objectives

Phase I	1.	Expand noise measurement library
	2.	Refine noise classifier developed in SEED proposal
	3.	Establish Hardware Requirements for Classifier Algorithms (with APS)
Phase II	4.	Create real-time laboratory demo (with APS)
	5.	Develop and demonstrate a prototype system (with APS)

The project investigated the use of a mix of artificial neural network (ANN) and Bayesian statistical methods to create a robust noise classifier. Collaboration on a "sister" SERDP project SI-1427 (Noise Bearing and Amplitude Measurement and Analysis System or BAMAS) was executed with Applied Physical Sciences, Inc. (APS) [Allanach (2009)]. By combining testing efforts, better value was provided to the military. Algorithms were developed that work independently of BAMAS microphone array and in concert with it. As expected, the best performance was achieved from the combined systems. The classifier was "generalized" by collecting large amounts of algorithm training data under various conditions and locations from military installations across the country. A methodical approach was presented to take the technology from simulation to laboratory demonstration and then to final product. The technology has been demonstrated in a prototype developed by APS, with the intention of transitioning to a product.

The initial key step in this project was collecting a library of high-amplitude operational noise around military bases, including military impulse noise. Before the initial SEED project, no such library was found to be in existence. Without an appropriate library of noise recordings, it would not have been possible to develop and test new classification algorithms. The library contains actual recorded waveform data, sampled at a sufficient frequency to observe all important features. It contains a wide variety of military impulse noise sources, with measurements

conducted at either key locations of noise-level concern, or appropriate distance to simulate these situations. Measurements were also conducted for a wide variety of environmental conditions. The library also includes noise sources from the respective locations that are believed to be responsible for current false positive detection and any other high-amplitude noise sources of interest that would also occur in these locations. For example, aircraft noise (e.g. close air mission support) has been found to be significant. The structure of the library allows the user to easily navigate the database and select any related data which are of interest.

With a suitable waveform library in place, it was possible to investigate the dataset for suitable means of discerning military impulse noise from other high level noise sources. These methods started with the development or evaluation of signal metrics that were based on statistical, temporal, spectral *features*. These waveform features then served as a dataset to be used to train, validate, and test *classification* algorithms that were investigated as a means to identify military impulse noise. Two specific types of classifiers were investigated: artificial neural networks and Bayesian. The performance of the original multilayer perceptron ANNs developed during the SEED effort was evaluated on any newly collected data to determine if the classifiers were generalized. When performance was found to be subpar, retraining or restructuring of the ANNs was used. Further refinements are also possible by introducing new features to the ANN. A Bayesian classifier based directly on the original computed metrics was also developed.

Once the metrics were identified and an accurate classification algorithm was found, the goals of the project turned toward prototype development and field demonstration. Two prototypes were constructed and deployed in the field. Performance of BAMAS, the University of Pittsburgh (UPitt) Classifier, and a combined system was examined, including some comparison with existing noise monitors.

3. Background

Automatic classifiers go by many names, including artificial intelligence, biologically inspired computing, expert systems, and machine learning [Shapiro (1987)]. Besides classification, these systems are also capable of detection, estimation, signal extraction, control, and analysis. In particular, speech and pattern recognition have received a lot of attention in recent years [Anil (2000), Chan (1982), Horvath (1982), Moser (1986), Pearl (1986), Ripley (1996)]. The algorithms utilized by these systems are also varied. Whatever the goal, the algorithms have the same purpose: to gather a set of *features* and draw generalizations from them, also known as *classification*. Some popular choices include regression analysis, maximum entropy and maximum likelihood methods, principal component analyses, hidden Markov models, fuzzy systems, artificial neural networks (ANNs), and Bayesian statistics. Wavelet transforms are also useful for extracting fine signal features that could be used by other types of classifiers [Ganesan (2004), Guo (1997)]. They can also be more efficient than fast Fourier transforms (FFTs) at frequency decomposition [Chen (1982)]. Thus, many techniques are useful across a wide spectrum of uses [Shapiro (1987), Scharf (1991)].

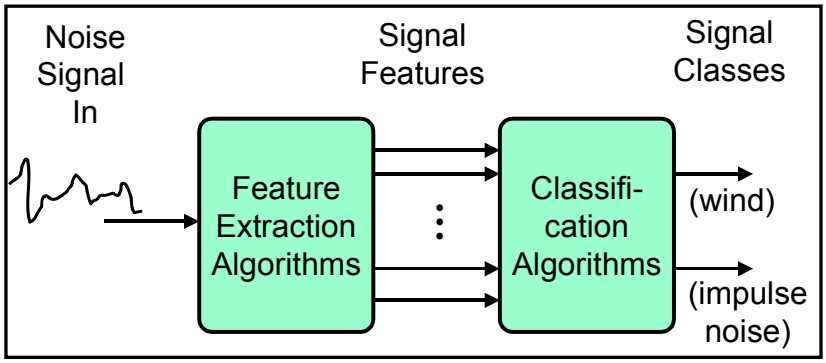

Figure 1. Schematic Diagram of Noise Classification System.

Noise classifiers have existed for a number of years and have been used for a variety of uses including damage detection [Chan (1982), Horvath (1982)], speech recognition [Moser (1986)], and noise source identification [Moukas (1982), Greene (1997)], including for detection of aircraft and helicopter noise [Cabell (1992)]. Note that prior reports of noise source identification primarily deal with continuous noise sources. All classification systems typically depend upon a classifying function, $f(\underline{x})$ which maps a set of features, $\underline{x}$, from feature space, $\mathcal{F}$, into class space vector $c \in \mathcal{C}$ (See Fig. 1) as $f(\underline{x})$: $\underline{x} \rightarrow \mathcal{C}$ (see Figure 1). Features are simply the chosen set of distinguishing characteristics of the noise sources and typically consist of a mixture of time domain and frequency domain metrics. Classes are the different categories of noise sources that are to be discerned.

Neural networks in particular have proved to be a very powerful device for information processing [Anil (2000), Kosko (1992), Ripley (1996)]. They are capable of determining complex, nonlinear input/output relationships from data. No mathematical model or statistical

methods are required to implement them. Adaptive algorithms can be incorporated, giving them learning capabilities, which can compensate for changes in a system. They have been successfully used in identifying continuous noise sources [Bucci (2006a), Bucci (2006b), Cabell (1992), Vipperman (2005a), Vipperman (2005b), Vipperman (2006)]. Accuracies nearing 100% were reported in a SERDP SEED Effort [Vipperman (2005b), Vipperman (2006)]. These algorithms are based upon gradient-descent optimization techniques, such as the back-propagation algorithms [Kosko (1992), Widrow (1985)]. ANN-based noise classifiers have even demonstrated the ability to account for variables such as spherical spreading, atmospheric absorption, and ground impedance, which can have a strong impact on acoustic propagation.

At the same time, probabilistic learning algorithms [Scharf (1991)] should be studied that might provide better robustness, particularly across different military installations. In particular, Bayesian methods have proven to be relatively simple, yet effective in many applications of parameter estimation and spectral analysis [Bretthorst (1988) and (1990), Jaynes (1985) & (2003)].

Most commercially available noise-monitoring systems are nothing more than a threshold activated sound level meter (SLM). They typically log either A- or C-weighted peak level (L_{pk}) or day-night sound level (L_{DN}) and sometimes sound exposure level (SEL). They often have remote communication capabilities and some have the ability to record a "snippet" of the offending sound. More advanced sensors employ detection algorithms. For example, the NoiseWatch® system by McQ Associates, Inc. (McQ) uses the blast analysis and monitoring (BLAM) algorithm, which is based upon sonic boom detection technology [McQ]. While much more accurate than previous systems deployed by the Department of Defense (DoD), it still reportedly suffers from an unacceptably high false positive rate [Luz (2004)].

Wind noise has proved to be the biggest problem for high-intensity military impulse noise monitors. Since energy from both wind and large ordnance is concentrated at very low-frequency, simple filtering schemes are ineffective [Benson (1996)]. Benson developed a 2-microphone method to improve wind-noise rejection. Unlike blast noise, the measured wind noise is *mostly* uncorrelated at each microphone. The method was experimentally demonstrated, although there is no evidence that it was ever permanently implemented anywhere. Applied Physical Sciences, Inc. (APS) has developed a similar, though more advanced monitoring concept sponsored by SERDP [SERDP (SI-1427), Allanach (2009)]. The Noise Bearing and Amplitude Measurement and Analysis System (BAMAS) uses 4 microphones to determine the Direction of Arrival (DOA) of the sound (see Figure 2). Similar to Benson, correlation methods are used to enhance the wind noise rejection of the system. Multiple BAMAS sensors can be used to triangulate the noise source based on their DOA outputs and determine the distance to the source.

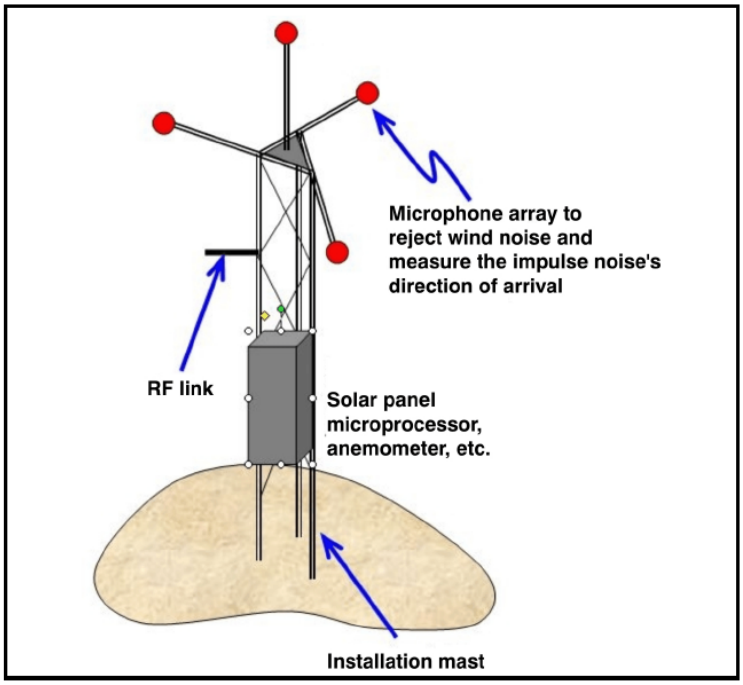

Figure 2. Schematic of Bearing Amplitude and Measurement System (BAMAS) [SERDP-1427].

The proposed effort is complementary to the BAMAS approach. While BAMAS is a spatial signal processing method, the current approach uses a temporal and spectral processing for a single measurement channel. *BAMAS provides acoustic detection while the UPitt algorithm provides signal classification.* Such a system could be instrumental in executing long-term range management as well as providing instant feedback to assess whether conditions are favorable for military training exercises.

In the SEED effort, a noise library was developed to be used for developing noise classifier algorithms [Vipperman (2006)]. It contained recordings of various ordnance, wind, and aircraft. An ANN classifier was developed that exhibited essentially 100% accuracy for the given data. Four signal metrics were identified as effective inputs to the ANN, including two conventional metrics (kurtosis and crest factor) and two new frequency domain metrics (spectral slope and weighted square error) [Vipperman (2006)]. This continuation effort expanded upon the SEED effort, including integration of the new algorithms into a hardware prototype.

4. Materials and Methods

4.1. Data Collection

The noise measurements library developed in the SEED effort has been greatly expanded in the current effort. It was important to measure many kinds of ordnance under varying conditions (e.g. weather, topography, distance between source and receiver, vegetation, etc.). Doing so helped to ensure that the noise classifiers are sufficiently generalized. UPitt has collaborated with APS by gathering BAMAS array data during field collection trips. To facilitate these measurements, National Instruments (NI) hardware was purchased that permits four channels of simultaneous data acquisition.

In FY09, two additional collection trips were performed at Fort Benning, GA. This location has the McQ NoiseWatch™ (i.e. BLAM) monitors. Microphone array measurements were taken using a UPitt constructed microphone array with similar geometry to the APS array. A picture of the setup is given in Figure 3. The data collected from Fort Benning include wind and 25mm Bradley fire waveforms. Four channels of array data were collected for each waveform. The data were analyzed using both single channel classification methods and using the BAMAS. A list of all measurement sites and the type of noise contained there is given in Table 2.

Table 2. List of Measurement Sites Visited.

Location	Number of Visits	Types of Noise
MCBCL, NC	3	Tank, artillery, vehicle, mortars, wind noise, dummy bombs, demolition charges, grenades, Bradley fire, aircraft, wind
Fort Indiantown Gap, PA	2	Mortars, Howitzer, dummy bombs, canon straif, vehicle strikers
Fort Benning, GA	2	Tanks, demolition, Bradley fire, mortars, vehicle, aircraft, wind
Fort Hood, TX	1	Tanks, Palladin, Bradley, aircraft, mortars, artillery, vehicle, wind
Fort AP Hill, VA	1	Demolition, grenades, mines, mortars, artillery, aircraft, wind
Fort Riley, KS	1	Tanks, Bradley, demolition, aircraft, strong winds
Fort Carson, CO	1	Tanks, Bradley, demolition, mortars, aircraft, winds

Figure 3. Picture of array measurements at Fort Benning

4.2. Algorithm Development

The UPitt algorithm can work alone or in conjunction with the APS array and algorithms (by using the average output of all microphones (which provides additional wind noise rejection). Details for the APS classifier can be found in the final project report [Allanach, 2009]. The UPitt algorithm is described next.

4.2.1. Signal Metrics

In the SEED project (SI-1436), four scalar metrics that are effective in classifying military noise were identified or developed. These metrics are kurtosis, Crest factor (CF), spectral slope (*m*), and weighted-square error (*WSE*). The former two are metrics that have been widely used in acoustics and vibration analysis to characterize waveforms. The latter two metrics were developed in the initial effort based on observation made from the power spectral density (PSD) functions calculated for the waveforms collected. To review, Crest factor is

$$\mathrm{CF} = \frac{|p_{pk}|}{p_{rms}}, \tag{1}$$

where $|p_{pk}|$ is the maximum absolute instantaneous pressure amplitude, and p_{rms} is the root-mean-square (RMS) of the sound pressure over some period of time. Crest factor tends to be a good indicator of impulsiveness in a signal [Norton (2003)]. The statistical metric kurtosis is also a good indicator of impulsivenes. Kurtosis is the fourth central statistical moment and is given by

$$\mathrm{Kurtosis} = \frac{E\left[(p-\mu)^4\right]}{\sigma^4} = \frac{1}{\sigma^4 T}\int_0^T (x-\mu)^4\, dt\,, \tag{2}$$

where μ is the mean pressure (typically zero), σ is the variance of the signal, and T is the time frame over which kurtosis is computed [Norton (2003)]. Spectral Slope (*m*), is computed by creating a least-squares fit to a line,

$$\hat{y} = mx + b\,, \tag{3}$$

where $\hat{y} = \log_{10}(\mathrm{PSD})$ is the base-10 logarithm of the power spectral density and $x = \log_{10}(f)$ is the base-10 logarithm of frequency. The fit is conducted over the frequencies between 0.6 and 100 Hertz (Hz). Although the impulse and aircraft noise sources are poorly characterized by a linear trend, the slope is still useful for differentiating from other types of noise. There are 164 frequency bins from 0.6-100Hz for the given spectral resolution (4,096 point FFT) and sampling frequency of 5 kilohertz (kHz), and the WSE is computed from the first 163 of these bins. To complement metric *m*, the "goodness" of the linear fit is assessed with the WSE, which is computed as

$$WSE = \sum_{i=1}^{163}\left[y_i - \hat{y}_i\right]^2\left[f_{i+1} - f_i\right], \tag{4}$$

where y_i is based upon the $\log_{10}(\mathrm{PSD}_i)$ of the i^{th} frequency bin, $\hat{y}_i$ is the estimate of y_i from the linear curve fit, and f_i is the base-10 logarithm of the i^{th} frequency. Squaring the quantity $\left[y_i - \hat{y}_i\right]$ allows WSE to remain positive and also reflects the total magnitude of the error. The term $\left[f_{i+1} - f_i\right]$ serves to add greater weight to the error at the lower frequency bins. This is done because the best features for identifying military impulse noise from non-impulse noise occur at the lower reaches of the bandwidth of interest. Also, higher frequency bins within the bandwidth

of interest are more susceptible to contamination from other sources of environmental noise. In order to scale the metrics with respect to signal energy, the logarithmic PSD terms, y_i, are normalized, between [0,1], which correspond to the minimum and maximum values of $\log_{10}$(PSD), respectively. Thus, y_i is computed as

$$y_i = \frac{\log_{10}(PSD_i) - \min[\log_{10}(PSD)]}{\max[\log_{10}(PSD)] - \min[\log_{10}(PSD)]}. \quad (5)$$

It is important to note that the values of $\hat{y}_i$ are not normalized. They are simply the curve fit of y_i. Given the success of these metrics in the SEED project it was believed that continued investigation of these metrics would be prudent. For a most complete explanation of the function of these metrics description of other metrics that do not perform as well [Bucci (2007)].

4.2.2. Metric-Based ANN Classifier

As in the SEED effort, the multi-layer perceptron (MLP) ANN [Haykin (1999)] was used to classify the noise based upon the four signal metrics described in section 4.1. The ANN was trained using the Levenberg-Marquardt algorithm [Chong (2001]], with inputs of the four scalar metrics, with three hidden layers of four nodes each and two output nodes [Bucci, 2007, Bucci and Vipperman, (2006) and (2007b)]. Additionally, since the completion of the SEED effort, two additional ANN structures using the four scalar metrics have been evaluated. The first is a self-organizing map (SOM) ANN [Kohonen (1990] that uses the four scalar metrics as inputs and a 4×4 lattice of neurons as output space [Bucci (2007), Bucci and Vipperman (2007b)]. In addition, a support vector machine (SVM) ANN [Vapnik (1992)] using the four scalar metrics and 28 support vectors was investigated [Bucci (2007), Bucci and Vipperman (2007b), Vipperman 2006]. Based on studies conducted in 2007, the MLP ANN (simply referred to as "ANN" from this point forward) has been adopted for all future studies. Additionally, a study of using least-squares linear classifier (LSLC) was conducted. It was found that LSLC did not perform as well as their nonlinear counterparts (ANN).

4.2.3. Temporal ANN (tANN)

A novel ANN that is based on temporal processing was also developed. It offers the advantage of greater simplicity. Purely temporal approaches were originally avoided, given that current algorithms (that suffer from inaccuracies) look for a specific levels and waveform shapes in their classification scheme. A temporal only classification algorithm does offer the distinct advantage of avoiding the need to compute the FFT of a waveform, which would save programming effort and computational time. While computational demands are reduced, the lack of frequency domain information can challenge the algorithm to discern wind from impulse noise. It was believed that since waveforms of similar sources can appear very different for a wide variety of reasons (multi-path propagation effects, source/receiver distance, weather conditions, topography, vegetation, multiple sources, etc.), that a temporal approach to classification may not be the optimal choice.

It was observed that the same sources could look quite different under varying conditions. However, with experience, it became evident that experienced observer could often differentiate the source of a waveform with near perfect accuracy. Figure 4 shows some examples of waveforms of military impulse noise that illustrate how the waveforms may appear somewhat different even though they originate from similar noise sources. Figure 5 and Figure 6 show typical examples of wind noise and aircraft noise, respectively. Once again it is observed that all of the waveforms appear very different when compared to waveforms of the same class of noise. However, there are more profound differences between the respective classes of waveforms.

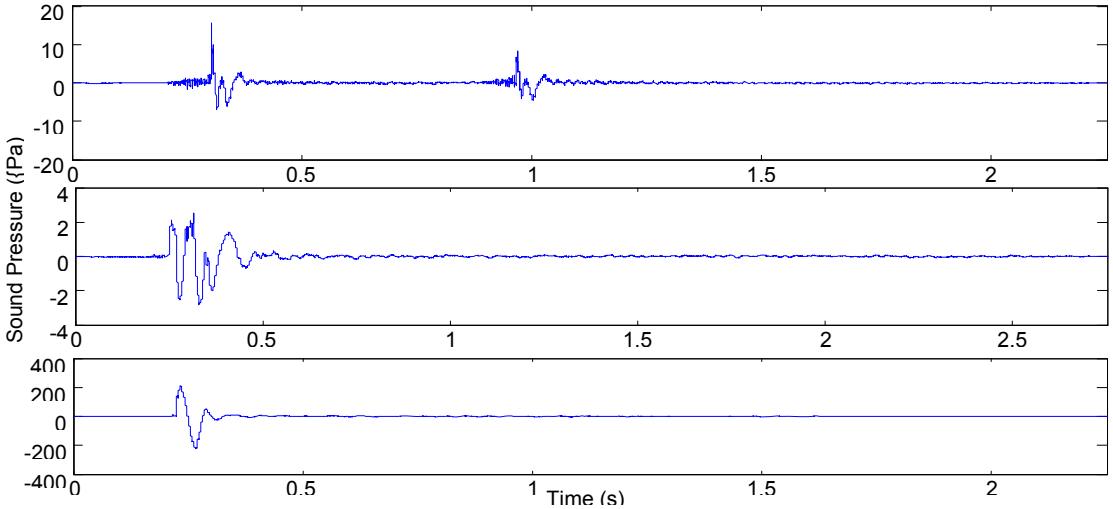

Figure 4. Examples of waveforms of military impulse noise illustrating how the waveforms may appear somewhat different even though they originate from similar noise sources. (top) 2 81mm detonations at 2km, (middle) 155mm high explosive round detonation at 6 km, (bottom) 3 Bangalore torpedoes at 2.5 km.

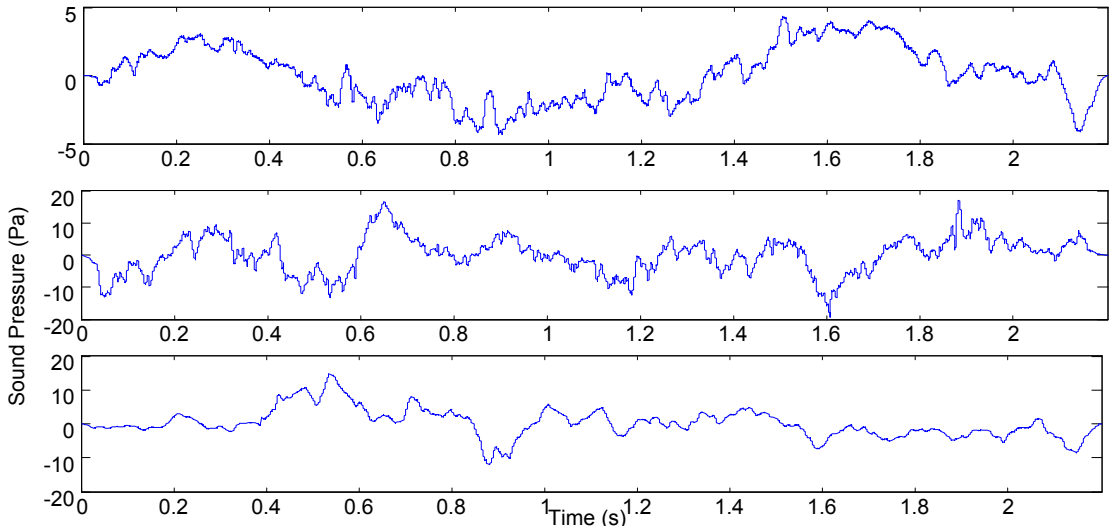

Figure 5. Three example recordings of wind noise.

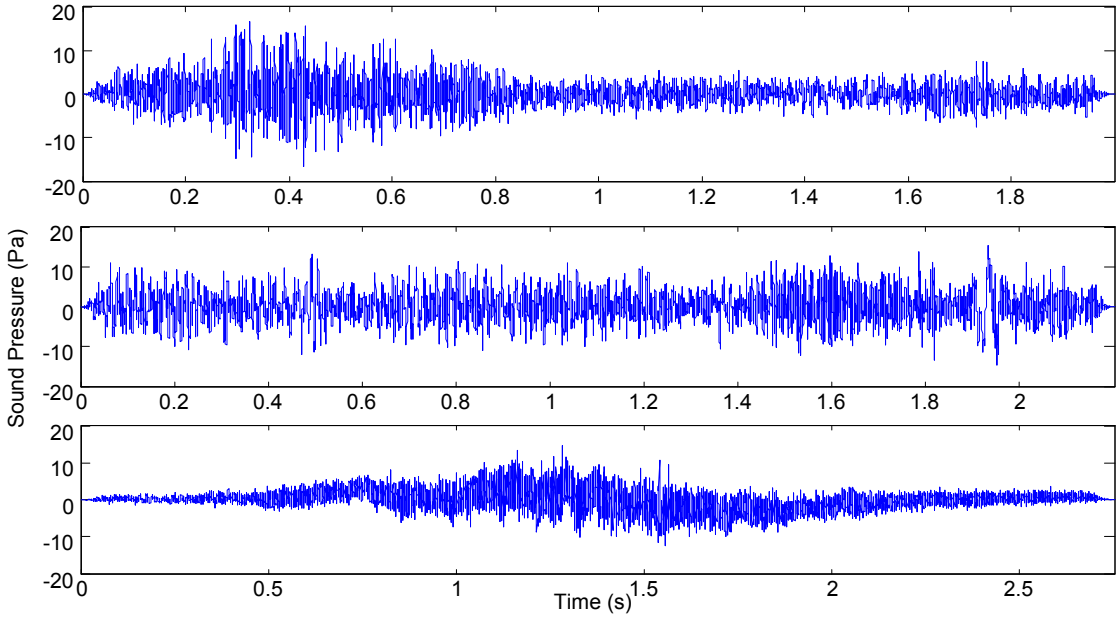

Figure 6. Examples of typical aircraft noise waveforms. (top) F-16 flyover, (middle) F-16 flyover, (bottom) A-10 flyover.

Borrowing from a proven approach in image processing, each of the waveforms is viewed as an image [Burr (1986), Jagota (1990), Abe (2001), Haykin (1999)]. This technique could be compared to the automatic address reading algorithm used by the United State Postal Service. A given character, for example "a", read from an envelope address may appear significantly different when penned by different senders. However, barring extremely poor penmanship, it is still intelligible to a human reader because the human reader is capable of drawing a generalization based on the many past observations of that character. Thus, it was hypothesized that, since ANNs have strength in drawing generalizations, given enough samples of alphabetic characters, an ANN would be able to identify the printed characters of an envelope address. The hypothesis was shown to be valid in 2007.

As this method applies to this problem, there are two "characters" to be identified, military impulse noise and non-impulse noise, (three if aircraft noise is decided to be considered as a separate class of noise). As with the character recognition problem, the image is first normalized and a grid of bins is placed over the image. This concept is illustrated in Figure 7. The feature set, or inputs to the ANN, are simply the number of sample points that reside within each of the respective bins. Additionally, the waveforms were decimated to 1 kHz to simulate using actual data collected by the current field implemented hardware. The ANN structure consists of a MLP with 100 inputs, two hidden layers of 100 neurons each, and one output (1 for military impulse noise and -1 for non-impulse noise).

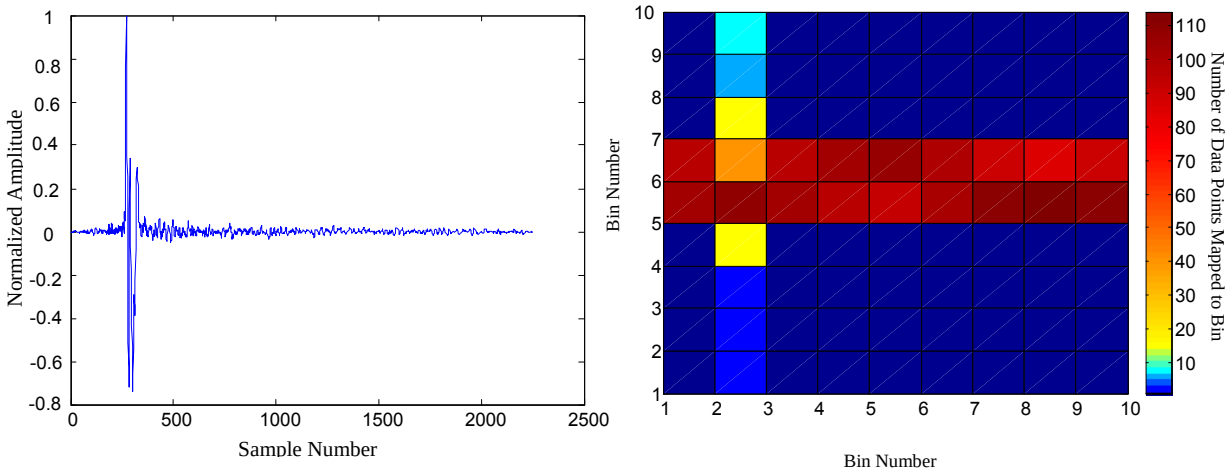

Figure 7. Computation of feature set for 81mm mortar recording

A secondary study on the efficacy of various activation functions was also performed in 2007. Since any algorithm resulting from these efforts is desired to eventually be implemented on a digital signal processing (DSP) board, computational efficiency is desired. The performance of the widely used sigmoidal activation function neurons were compared to the saturating linear activation function neurons. Figure 8 illustrates examples of common activation functions that may be used in the neurons of a network. The topmost plot shows the threshold function. While it is mathematically possible to approximate any function with the threshold function, it is not always practical to use and isn't conducive to the state of the art training algorithms. Examples of sigmoidal activations are given in the bottom two plots of the figure (the logsig and hyperbolic tangent functions). The chief advantage of this type of neuron is constant differentiability, which aids the computation of derivatives required in the training update equations. However, operation of these neurons in an ANN implementation would require multiple Taylor series expansions to estimate the value of these functions. Depending on the computational resources available to the monitoring stations, the total number of computations required to classify a waveform may become of concern. The advantage of the saturating linear type of neuron is that, since the activation function of a neuron is a piecewise linear function, Taylor series expansions would not be necessary. If the saturating linear activation function neurons are able to provide adequate performance, then this type of neuron would be the natural choice due to the computational savings.

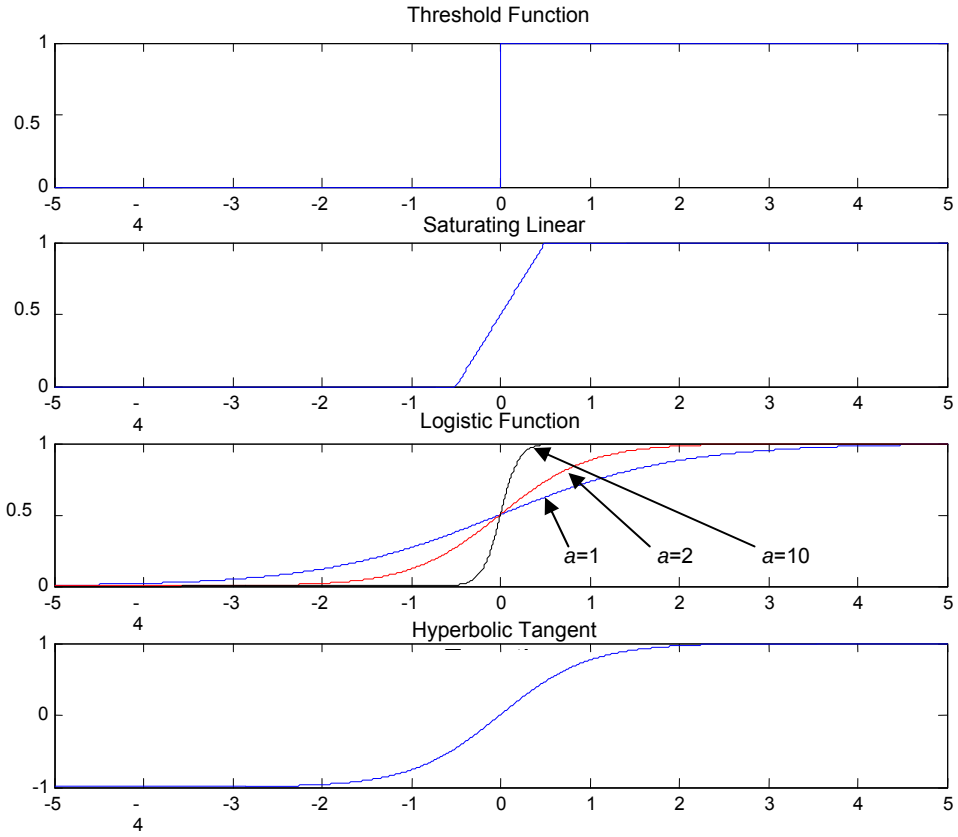

Figure 8. Plot of typical neuron activation functions

4.2.4. Metric-Based Bayesian Classifier

Given the great recent interest in exploring the strengths of Bayesian methods, a Bayesian classifier utilizing the four scalar metrics was developed [Bucci and Vipperman (2007c)]. Bayes' rule is stated as

$$P(C_i \mid \underline{x}) = \frac{p(\underline{x} \mid C_i)P(C_i)}{p(\underline{x})}, \tag{6}$$

where $\underline{x}$ is the vector of observations and C_i is the i^{th} class of data from n classes. In equation (6) $P(C_i|\underline{x})$ is the probability that a given data point is from class C_i given the observation vector $\underline{x}$, $p(\underline{x}|C_i)$ is the conditional probability density of class i, $P(C_i)$ is the prior probability of data class C_i being observed, and $p(\underline{x})$ is the probability density of observation vector $\underline{x}$, which is defined in terms of the conditional probability densities as

$$p(\underline{x}) = \sum_{i=1}^{n} p(\underline{x} \mid C_i)P(C_i). \tag{7}$$

Ideally, the conditional probability, $p(\underline{x} \mid C_i)$, would be Gaussian. In real life, the distributions of the signal metrics are not Gaussian and can be multi-modal. As a result, a mixture model is required. The actual distribution can be modeled as a multivariate Gaussian distribution or multi-

Gaussian. Typically, a linear combination of multi-Gaussians or Gaussian mixture model (GMM) is created, which can be expressed as [Quatieri (2002)]:

$$p(\underline{x} \mid C_i) = \sum_{k=1}^{r} w_k \frac{1}{(2\pi)^{m/2} \left(\det \underline{\underline{Q}}_{pq}\right)^{1/2}} \exp\left(-\frac{1}{2} \left(\underline{x} - \underline{\mu}_p\right)^T \underline{\underline{Q}}_{pq}^{-1} \left(\underline{x} - \underline{\mu}_p\right)\right). \qquad (8)$$

In equation (8), w_k is a weighting factor, which will be discussed shortly, $\underline{x}$ is the feature vector describing an input data, $\underline{\mu_p}$ is the mean vector of the data used to train the Bayesian classifier, Q_{pq} is the covariance matrix of the training data, and m is the length of $\underline{x}$ (4 in this case). The part of equation (8) to the right of the w_k term defines an m-dimensional Gaussian distribution. It is possible that this part of the equation could be any distribution function. However, Gaussian type distributions are usually the most convenient for approximating higher dimensional data. In some cases, a single multi-Gaussian may be adequate to approximate a given distribution of data however; a GMM is required to adequately represent the distribution. Thus, the w_k term represents the weights associated with each respective distribution in the GMM. To assure that the GMM represents a true probability density function (PDF), the sum of all w_k is equal to 1. The parameters w_k, μ_p, and Q_{pq} were optimized via the Expectation Maximization Algorithm [Stark et al (2002); Chong and Zak (2001)] to produce the GMM with the best fit to the data space given a set number of multi-Gaussian distributions. At first, intuition may lead one to want to include as many multi-Gaussian fits into the GMM as possible. Contrary to this belief, this may yield a classifier that is not optimal. This failure occurs because the resulting GMM will have been fit too closely to the data space of the training data and lost its ability to generalize about input data points that are close but not identical to data points used for training. This result forgoes a primary strength of Bayesian methods. The optimal number of Gaussians to include can be determined heuristically.

Figure 9 shows a diagram of information flow within a Bayesian classifier. The parameter Λ_i is known as the likelihood ratio for data class i and is based conditional probabilities of the respective classes of data. The formal definition of Λ_i is

$$\Lambda_i(\underline{x}) = \frac{p(\underline{x} \mid C_i)}{\sum_{\substack{s=1 \\ s \neq i}}^{n} p(\underline{x} \mid C_s)}. \qquad (9)$$

ξ_i is referred to as the threshold of test and has the corresponding definition of

$$\xi_i = \frac{\sum_{\substack{s=1 \\ s \neq i}}^{n} P(C_s)}{P(C_i)}, \qquad (10)$$

which is based on the prior probability that a given class of data is to be observed.

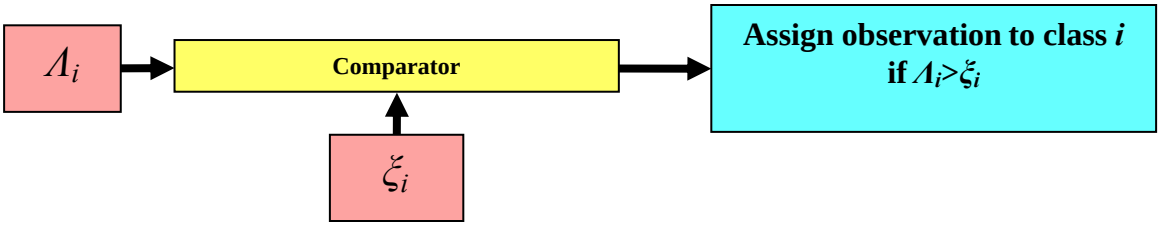

Figure 9. Bayesian Classifier

In the case of this particular problem, it is rather difficult to assign a true probability to military impulse noise being observed at any given time. Many complex factors would need to be taken into account, such as, training schedules and weather conditions, both of which could change the probabilities on the basis of a very small time scale. Added to which is the uncertainty of the exact state of each of these conditions.

Fortunately, it is observed that the likelihood ratios, computed for the different classes of data (military impulse noise and non-impulse noise) differ significantly. Thus, the choice of prior probability of each class of noise will have an effect equivalent to adjusting a parameter to make the classification algorithm more or less sensitive to the detection of military impulse noise. This concept will be further expounded upon in the results and accomplishments section with real data.

4.2.5. Assessing Generality of Classifiers

An assessment of the ability of the classifiers to generalize decisions must be made. Data are obtained from many sources and measurement conditions. Earlier in the study, each time data were obtained from a new measurement location, they were first tested on the classifier *before* retraining it. After the initial testing accuracy was obtained, then the network was retrained using the new (and old) data. This exercise helped indicate whether a particular location is significantly different from the data for which it was trained. A more formal method of assessing the generality was adopted in FY08. It involved training the classifiers using data from one specific location and then testing with the data from all other locations. In all, six military installations have been visited (some of them twice). Whenever a misclassification occurred, the waveform was closely examined to try to determine its cause. In most cases, the reason for the misclassification was obvious.

4.2.6. Determining the Performance of Combined vs. Separate Classifiers

The performance of the ANN, Bayesian, and temporal ANN were compared. Direct comparison of the two can only be made in general terms, since retraining the ANN can produce a slightly different result. Any waveform that was incorrectly classified was examined closely to learn more about why it may have been misclassified. In FY09, additional comparisons were made between the UPitt metric-based classifiers, the APS BAMAS detector, UPitt + BAMAS.

4.3. Establish Hardware Requirements

The required data fidelity and computational needs were established for the classifier algorithms. In turn, the hardware requirements were established. There are two parts to the monitoring system: the "remote" sensor and the "local" centralized database and analysis software. At the sensor end, the hardware components include the analog "front end" (microphone, preamplifier, and low-pass filter), analog to digital converter (ADC), DSP, communications, and memory. The algorithms dictated the requirements on bandwidth, dynamic range, ADC resolution, memory, and computational burden. A fixed-point DSP chip was chosen (cheaper, but has less accurate calculations or can be much slower if floating point emulation is used). The analog front end, in conjunction with the sampling rate and anti-aliasing low-pass filter, controlled the necessary bandwidth of the hardware. The dynamic range was typically set by the microphone and preamplifier, which is divided into a finite number of discrete values (e.g. 2^{16}) by the ADC. Algorithm complexity also determined the memory storage and computational requirements for the DSP, which also controlled many other functions of the noise monitor.

Originally, it was planned to work with a manufacturer to upgrade an existing noise monitoring product with the UPitt classification algorithms and APS BAMAS array. However, since writing the proposal, the manufacturer decided to pursue other projects and withdrew participation in either project. As a result, APS developed their own hardware platform to implement BAMAS and the UPitt algorithms. APS specified a hardware platform (with input from UPitt) that will met the needs of both projects, including provisions to log and display the additional information provided by BAMAS and UPitt algorithms, as well as the raw waveforms.

4.3.1. Streamline algorithms

The computational efficiency of the algorithms were optimized before deploying them in a field prototype. The algorithms were originally based upon the Enhanced Sound Level Meter code developed at the University of Pittsburgh. This code is implemented in MATLAB with a mix of time- and frequency-domain computations. Many of the subroutines are built into MATLAB and had to be recoded, in preparation for porting them to C language. In addition, little regard was initially given to computational or memory storage efficiency. Originally, efficiency was a critical step, in order to preclude extensive hardware upgrades to the existing noise monitoring product. Since APS developed a more powerful hardware platform, this step was somewhat less critical.

4.3.2. Convert to Fixed point

The target hardware will employ fixed point math. Consequently, the algorithms had to be converted from floating point to integer math. Maintaining adequate numerical accuracy can be tricky for fixed-point operations. Further conversion was necessary to realize the algorithms in C/C++ language. The original subcontractor, was going to help with this step, but UPitt assumed responsibility for this task while searching for another subcontractor. The APS hardware platform has built in floating point emulation, which facilitated this step. The algorithms were re-coded using fixed point operations wherever possible, and the remaining operations were done in emulated floating point. While this emulation requires some additional computation time, the resulting system runs within the designated time requirements.

4.3.3. Determine necessary data fidelity

The original intent of this step was to determine if the UPitt algorithms would work on the initial subcontractor's hardware. To determine this, the high-fidelity data that were recorded by UPitt were decimated and resampled to match the less-stringent hardware specifications of the existing system. At first, these lower-fidelity data were in turn classified by the system that was trained using the high-fidelity data. If performance was found to suffer, the classifier was to be retrained with the lower-fidelity data and reevaluated. This process was repeated until the minimum required data quality was established (successively degraded the data quality until the algorithms started losing accuracy).

The data-altering process occurred in steps. Firstly, the data were digitally low-pass filtered (4^{th} order Butterworth) and decimated to drop the sampling rate to 1 kHz and the bandwidth to 400 Hz, respectively. Secondly, a threshold of 100, 105, 110, or 115 dB was introduced to decrease effective dynamic range. This step set the effective trigger (measurement) threshold. Finally, the data were re-discretized across the new dynamic range (12-bit resolution). These steps are represented in Figure 10. Since the sampling rate is decreased, the resolution of the FFT can also be decreased to save computational time, if necessary.

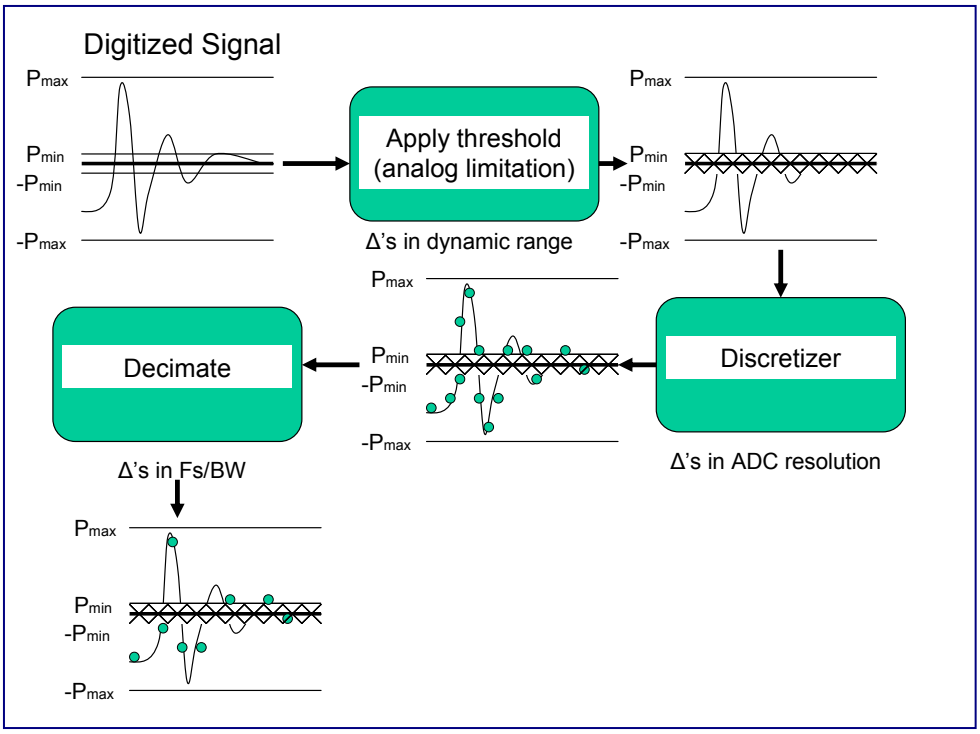

Figure 10. Schematic showing reduction of data fidelity.

The low-frequency cut off for the required bandwidth was also determined. Low frequency information was found to be particularly important information for discerning wind from impulse noise. The limit was established by high-pass filtering data (2^{nd}-order digital Butterworth filter) with successively higher corner frequencies until classification performance was found to suffer. In particular, some of the data from Fort Riley were used, given the problems with wind false positives that occur at this location.

4.3.4. Establish Hardware Requirements

The limits on data fidelity discussed above in section 4.3.3 directly impact the hardware requirements (microphone and preamp bandwidth and gain and ADC resolution). Another consideration is the computational requirements of the algorithms. These were established by simply counting the number of mathematical operations required to compute the metrics and implement the classification algorithms (including any overhead for floating point emulation). Establishing these requirements became somewhat moot, given that a new, scalable hardware platform was specified by APS. Should additional computational power have been required, an additional PC/104 single board computer (SBC) could have easily been added to the stack. Although the algorithms (and hardware) are amenable to distributed computations, the single SBC was sufficient for the current computational requirements of the overall system.

4.4. Create Real-time Implementation

Changes to this task were made due to a change in subcontractor to APS. Many of these steps revolved around integrating the UPitt algorithms, along with the BAMAS algorithms, into the existing hardware platform of the first subcontractor. Since APS and UPitt specified a new hardware platform, a different integration plan was required. In addition, UPitt assumed some of the development work that was originally slated for the initial contractor.

4.4.1. Port algorithms to C/C++

The codes that were developed in MATLAB were first ported to C/C++ language. Since calculations in MATLAB are vectorized, functions were re-written using loops to achieve vector and matrix operations. In turn, these streamlined algorithms were converted to C/C++ code under windows. Next, the codes were ported to Linux to run on the PC/104 hardware. At first, the microphone and preamp were bypassed and the algorithms will be tested on recorded waveforms. Each step of the calculation (signal metrics, classifier output, intermediate calculations, etc.) was checked against the simulation results. Once developed, UPitt supplied their debugged codes to APS for the final integration into the prototype units that were bench- and field-tested.

4.4.2. Debug signal metric computations

The signal metric computations were programmed into the prototype units. These calculations in the final real time system were checked against previous offline calculations. The results were comparable, so the prototype system signal metric computations were found to be properly debugged.

4.5. Develop field prototype hardware

Again, the original intent was to make upgrades to existing McQ NoiseWatch™ product. With the change of subcontractor, UPitt and APS collaborated to specify the requirements of the system. Two remote sampling stations were created that included the following features:

- Rechargeable battery powered operation with solar charging
- Radio frequency (RF) 900 megahertz (MHz Ethernet bridge) link to transmit data back to a base station
- Very low power single board computer based on an embedded computing standard; A powerful or scalable architecture is desired
- Microphone and preamp to provide sufficient bandwidth for the algorithms
- A minimum of four channels of 16-bit analog input with sufficient sampling capability to provide desired bandwidth

5. Results and Accomplishments

5.1. Data Collection

At present, the high-fidelity library now contains a total of 2,670 useful waveforms, including 848 impulse, 1,402 wind, and 420 aircraft or vehicle noise measurements. An additional 86 Bradley fire waveforms and 146 wind waveforms were collected from Fort Benning. Table 3 shows a summary of the waveforms collected. Additional data was collected from the prototype systems, as discussed in Section 5.6.

Table 3. Summary of waveforms recorded in 2009.

	Impulse	**Wind**	**Vehicle/Aircraft**	**Total**
Fort Benning	86	146	0	232
Total To Date	**848**	**1,402**	**420**	**2670**

All of the data from Fort Benning were collected using a microphone array. Each waveform file contains four channels of raw data corresponding to the four different microphones in the array. UPitt conducted various analyses on the multiple channel data utilizing the cross-correlations between the channels. For two given signals, $x(t)$ and $y(t)$, the cross-correlation is given by,

$$R_{xy}(\tau) = \frac{1}{T}\int_0^T x(t)\,y(t+\tau)\,dt \tag{11}$$

Another quantity of interest is the cross-correlation coefficient function, given by,

$$\rho_{xy}(\tau) = \frac{R_{xy}(\tau)}{\sqrt{R_{xx}(0)R_{yy}(0)}} \tag{12}$$

Using the cross-correlations between the different microphone channels, the time-delay between the channels was calculated. From the geometry of the array, a maximum possible time delay was determined for a propagating signal. Using this information, uncorrelated waveforms, such as wind, can be rejected if the time-delay between the channels exceeds the maximum possible time delay. Also, using the time delays between the channels, the angle of incidence of an acoustic event can also be determined. If a single event is recorded by two or more BAMAS units, the location of the event can be determined by triangulation. This additional information can be useful in eliminating false events, *e.g.*, if the event occurred from a location off-base. Another development from the added degrees of freedom of the microphone array is the ability to reduce uncorrelated noise in the signals by synthesizing the four channels into one. A method was developed to perform this synthesis by taking only the correlated portion of the signals and combining them into one overall correlated signal. UPitt executed some preliminary analysis of these correlation techniques, which were the basis of the previous work done by APS [Allanach

(2009)]. The final implementation of the correlation techniques was integrated by APS into the BAMAS.

5.2. Algorithm Development

The algorithm development consisted of implementing the metric calculations and the classifier code and then integrating them with the BAMAS code. Since APS developed the hardware with both algorithms in place, better integration was possible. For example, the UPitt algorithm used a single high fidelity measurement microphone, while BAMAS used the cheaper, 4-channel array. In addition, the UPitt classifier was set to operate on the raw or detected output (average of four array channels). Bearing and now possibly range information could also have been be integrated into the UPitt algorithms, if desired.

5.2.1. Signal Metrics

Some minor changes have been made to the previously developed signal metrics. The two frequency metrics had to be adjusted slightly due to the change in sampling frequency and FFT resolution. Previously, a sampling frequency of 10 kilohertz and a FFT resolution of 4096 points were used. Due to the hardware requirements of the system, the sampling frequency was reduced to 5 kHz, and a 8192 point FFT is implemented. These two differences effectively changed the bin size for the FFT. Previously, each frequency bin was approximately 2.5 Hz. With the changes in sampling frequency and FFT resolution, the frequency bins were now spaced 0.6 Hz apart. The calculation of the frequency metrics *m* and WSE use information in the frequencies up to 100 Hz, not including 0 Hz. Previously there were 42 frequency bins of interest, but now there are 164 frequency bins of interest. The spectral slope, *m*, was calculated from the first 164 bins, and the weighted square error (WSE) was calculated from the first 163 of these bins. The same formulas are used as before, except with the change in frequency bins. The details of these new calculations can be found in section 4.2.1.

5.2.2. Metric-Based ANN Classifier

The metric-based ANN classifier continues to work well during development. The classifier detection threshold could generally be adjusted to greatly reduce, if not completely remove the false positives (FPs). Figure 11 shows the rate of FPs (red curve) and false negatives (FNs) (black curve) versus the detection threshold when considering all data. The threshold has a value between 0 and 1. Note that thresholds below 0.05 and above 0.95 are not considered, since these are near the extremes (singularities) that would classify all signals one way or another. In general, choosing lower thresholds (abscissa) is observed to provide fewer FNs, while higher thresholds provide a lower number of FPs. The total percent of misclassified signals is the sum of the two curves. The data in the figure is the aggregate accuracy for all data with no measurement threshold imposed. As will be demonstrated shortly, as the measurement threshold is increased, accuracy improves. In general, the inaccuracy (ordinate) of the given classifier is about 2% when a threshold (abscissa) ranging between 0.1-0.9 is chosen.

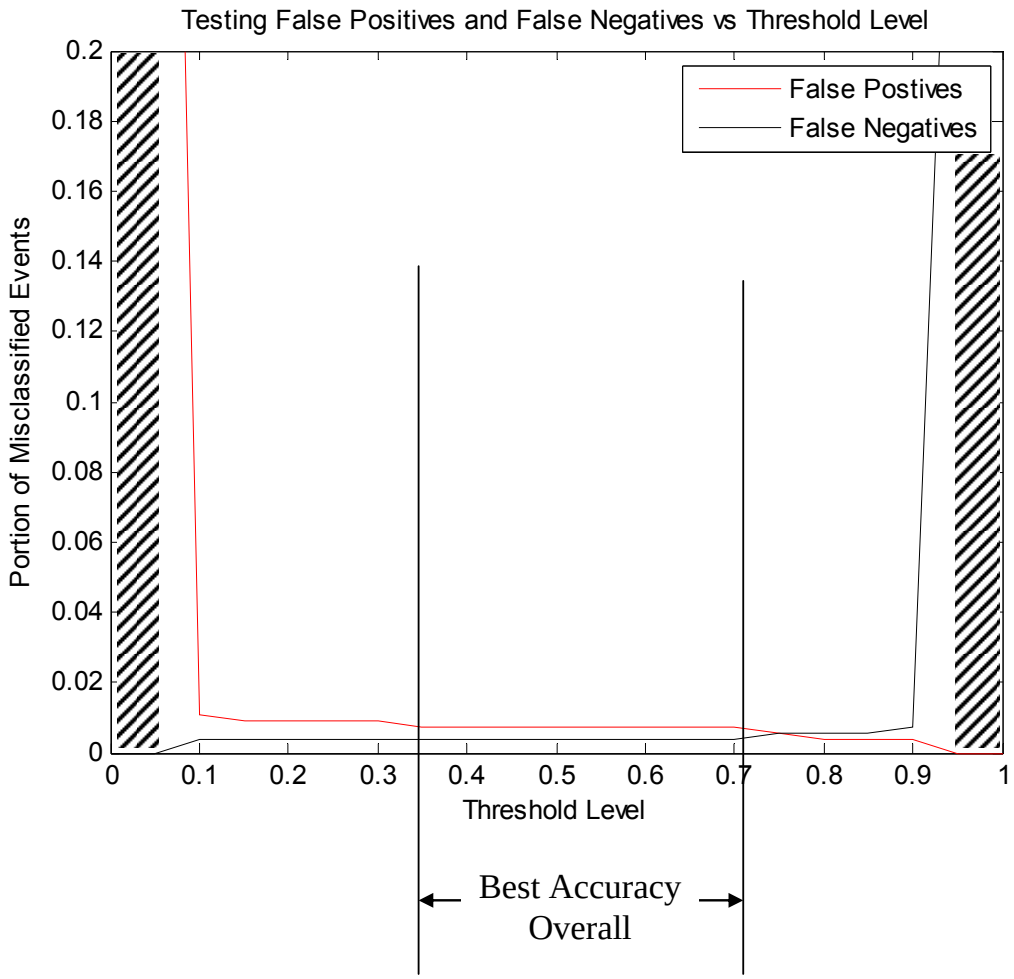

Figure 11. Example accuracy of the metric-based (ANN or Bayesian) classifier versus detection threshold.

5.2.3. Temporal ANN (tANN)

Results similar to Figure 11, except for the tANN classifier, are given in Figure 12. Accuracies are observed to generally be lower than those for the metric based classifiers. For detection thresholds between 0.46-0.68, the best overall accuracy was achieved (about 3% of waveforms are misclassified). The temporal ANN provides surprisingly good performance, given its relative simplicity. However, the lack of frequency domain information put it at a distinct disadvantage. For example, the tANN had a much harder time rejecting wind noise FPs. This result is not unlike the BLAM algorithm that also considers the shape of the waveform. An example of a misclassified tANN waveform will be presented in the next section will illustrate this concept. The tANN also had a harder time identifying blasts that had significant propagation path effects (echoes, multi-path propagation, long propagation distances, multiple events, etc.).

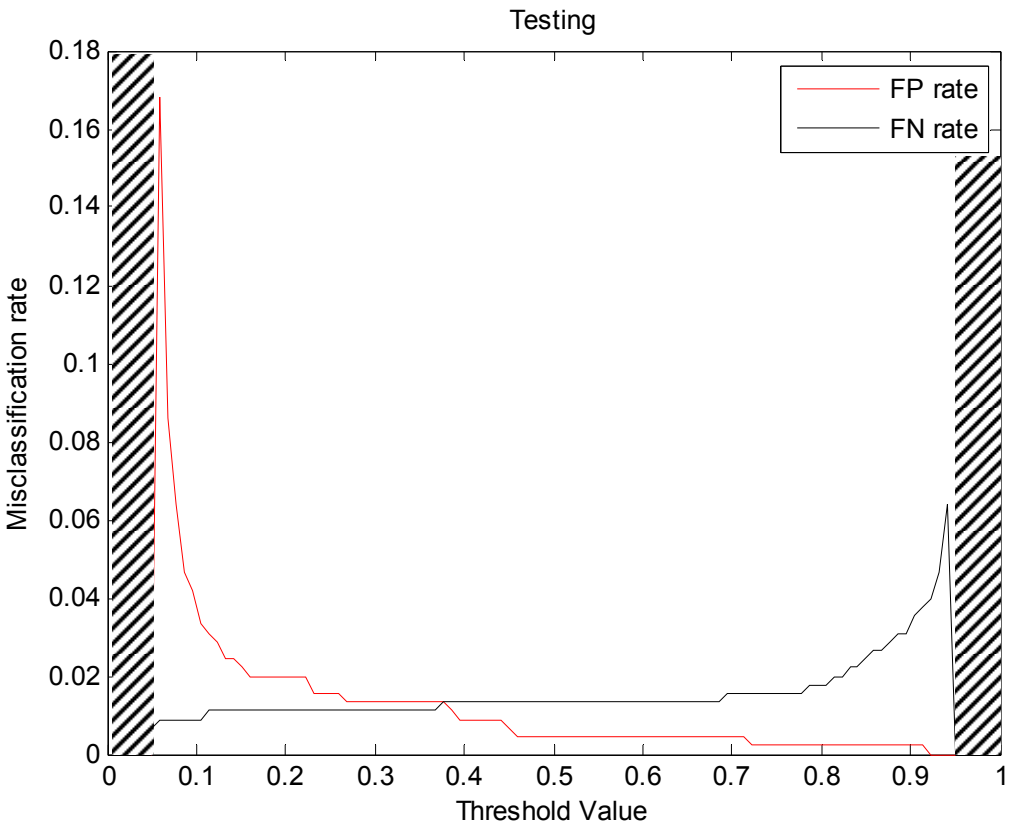

Figure 12. Accuracy of the tANN classifier versus detection threshold.

5.2.4. Metric-Based Bayesian Classifier

The performance of Bayesian classifiers was found to be nearly identical to the ANN in terms of overall accuracies and specific misclassified signals. Thus, combining these two classifiers would provide little, if any, gain in accuracy. In the end, this result is not surprising, since they use the same feature space (signal metrics). They differ in the way that the draw their respective decision boundaries. Either classifier can generally be adjusted to reduce, if not completely remove, all FPs. All algorithms had a hard time identifying Bradley fire (see Figure 13) in the presence of significant winds. In talking with the environmental noise managers, Bradley fire rarely generates complaints from neighbors. Indeed, most Bradley measurements were below 100 dB and were personally found to be far less startling than larger ordnance.

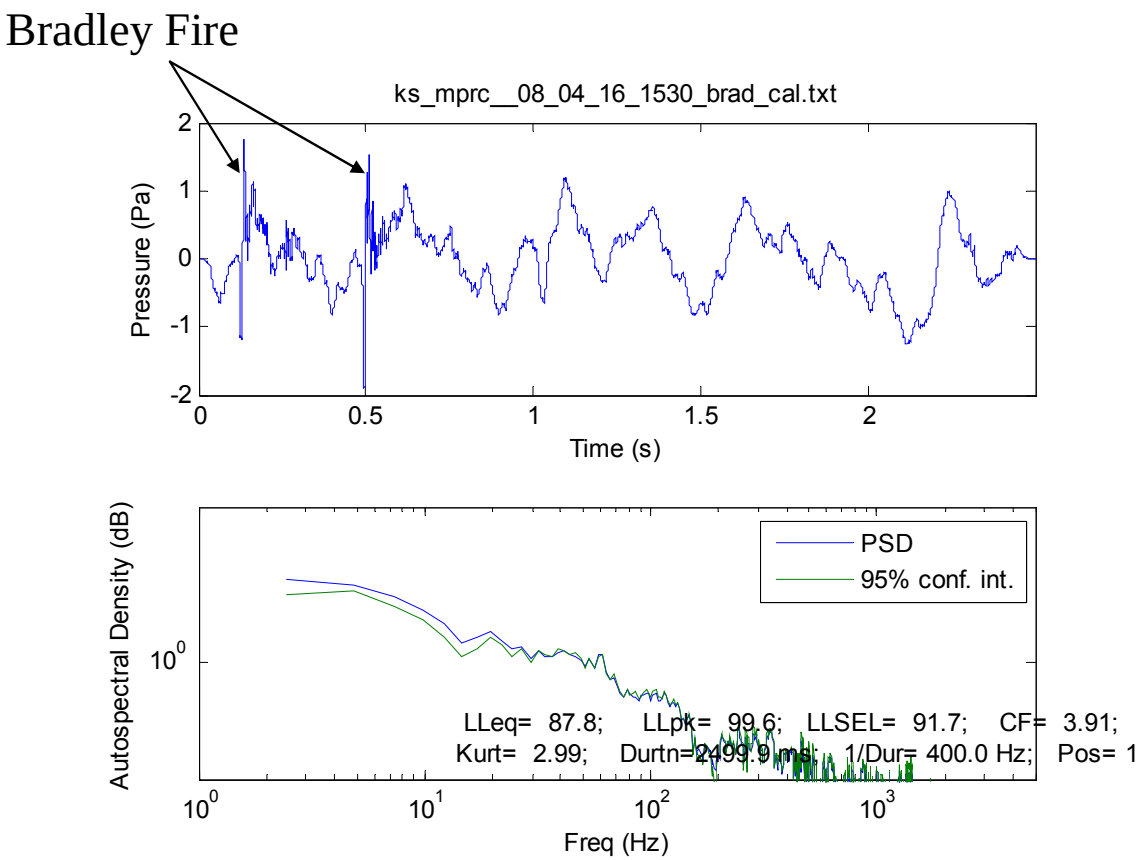

Figure 13. Misclassified waveform consisting of Bradley fire (25mm) in wind

5.2.5. Assessing Generality of classifiers

The ultimate goal was to develop a noise classifier that is sufficiently generalized such that it can be trained with the measurement library and then operated at any base with little error. To date, seven different measurement locations have been visited (Marine Corps Base Camp Lejeune (MCBCL), Fort Indiantown Gap (FTIG), and Fort Benning were visited twice or more). While not every combination has been exhausted, appreciable variations in ordnance, vegetation, topology, topography, and weather have been recorded.

Each measurement trip provided an opportunity to assess the classifier generality. This was done by testing the accuracy of the classifier on the new data before retraining with it. Each time, the classifier was found to work very well. A more stringent test of the classifier was devised by training it on all data except that for which it is tested. ANN training involves three different accuracies (training, validation, and testing) that correspond to the three partitions of the data [Vipperman (2006)]. Training accuracy is used to monitor the progression of the training (for ANN), while validation accuracy indicates if the ANN is becoming over-generalized. The most important accuracy is testing accuracy, since it indicates the accuracy expected if the trained classifier is put into service. In many cases, FP can be completely eliminated, but sometimes may involve a slight decrease in overall testing accuracy.

Table 4 summarizes the maximum achievable ANN accuracy for classifiers that were trained with data exclusively from the indicated testing location. Bayesian classifier performance is nearly identical to that of the ANN, and is not presented. Figure 14 shows a typical result for ANN that was trained on all data except MCBCL and then tested on MCBCL data. From the figure, if the threshold is chosen to be above about 0.22, then the FP rate will be minimized. In contrast, the lowest FN rate occurs at 0.05. The best accuracy (to foremost minimize FPs and then FNs) is to choose the threshold between 0.22-0.57, as indicated in the figure. The figure represents one possible result; higher accuracies were often observed with repeated retraining (starting from a different random initial set of weights).

Table 4. Training and testing ANN (metric-based) classifier accuracy when classifier were trained using all data except from the Testing Location

Testing Location	**Testing Max Accuracy (%)**	**Testing Max Accuracy (No FP) (%)**	**Training Max Accuracy (%)**	**Validation Max Accuracy (%)**
Camp Lejeune	99.3	99.3	100	99.0
Ft. Carson	99.8	99.3	99.6	98.4
Ft. Riley	99.2	N/A	99.9	99.6
Ft. Indiantown Gap	99.0	N/A	100	99.4
Ft. Hood	100	100	99.9	99.1
Ft. A. P. Hill	100	100	100	99.0
Ft. Benning	100	100	99.5	99.2

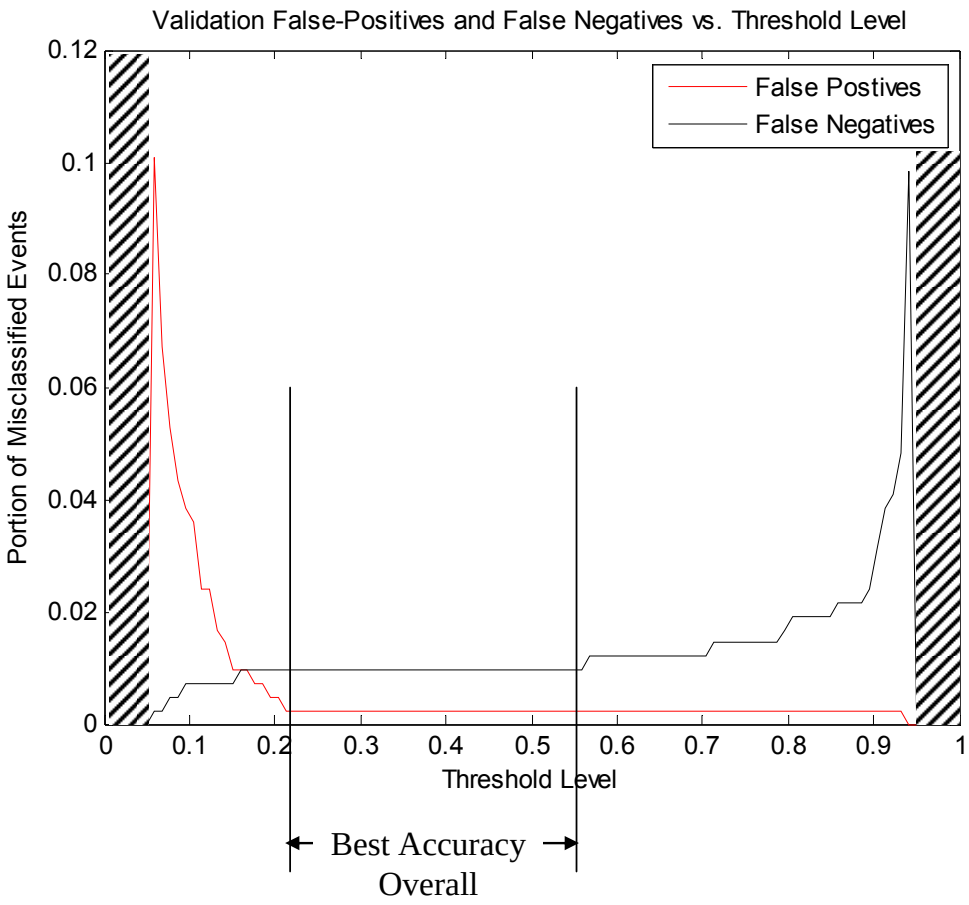

Figure 14. Accuracy of ANN classifier that was trained on all data except MCBCL, but tested on MCBCL data.

5.2.6. Combined vs. Separate Classifiers

Since similar performance was observed between Bayesian and ANN classifier and they use the same feature space, combining them would provide little advantage. In general, the metric-based classifiers were found to be superior to the tANN classifier. In particular, the frequency domain metrics seem to enhance the ability to correctly classify signals in mixed (event+noise) environments as well as when multiple events occurred simultaneously.

5.3. Establish Hardware Requirements

APS and UPitt worked together develop the hardware specifications. A hardware platform has been developed by APS to meet the needs of the projects.

5.3.1. Streamline algorithms

Although this was listed as a separate subtask, in reality, this process is integrated throughout the main task. One example of streamlining involved removing some instruction loops and replacing them with hard-coded instructions. Although this resulted in much longer programs, the computational efficiency was greatly improved by removing the loop overhead.

5.3.2. Convert to Fixed point

Fixed point codes were successfully implemented in MATLAB. Fixed-point kurtosis and crest factor calculations were found to have less than 0.01% error as compared to their floating point counterparts. These small errors were found to have no impact on the accuracy of the fixed point ANN classifier. Figure 15 shows the classification results from both the fixed and floating point ANN and they are nearly identical. Any "errors" denoted in the figure were common to both ANNs. The computational time of the fixed-point code was measured to be 51 times longer in MATLAB (classification of 1,615 waveforms took 347 versus 6.7 seconds). The increased computational time was a result of additional overhead of floating point emulation.

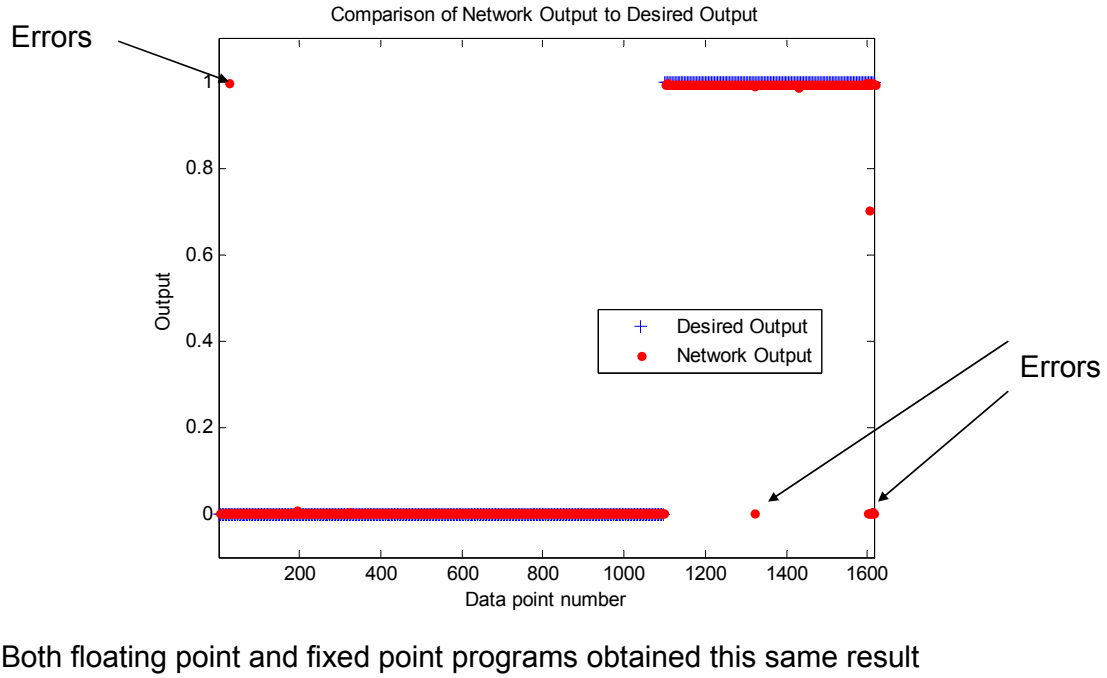

Figure 15. Comparison of floating- and fixed-point calculations for the ANN classifier in MATLAB.

It was also desirable to develop a fixed-point implementation of the ANN in C/C++. To do this, a set of structures and subroutines were written to implement the fixed point code, the output of which was checked against that of the floating point calculations. These were somewhat challenging, given that fixed point math only consists of addition and multiplication. Further complicating the conversion to fixed point was that the ANN activation functions (e.g. tansig and logsig) are nonlinear. They were implemented using a lookup table in conjunction with a routine

that performs linear interpolation between the table values. Another problem that surfaced was the implementation of a fixed point FFT. This prevented UPitt from developing a fully functional fixed-point code. Although the fixed point code was not fully implemented, it was ultimately superseded by the built in floating point emulator in the PC/104 hardware platform selected by APS. Some portions of the final code were implemented in fixed point, such as the FFT calculation. However, other portions of the code had to be coded using floating point calculations due to overflow and scaling issues. The final code runs well within the required execution time (BAMAS and UPitt classifier both run in less than one second).

5.3.3. Determine necessary data fidelity

In Figure 16-Figure 20, the rates of FPs and FNs versus the measurement threshold for the ANN classifier are given. The measurement threshold or level should not be confused with the "threshold level" used for classification. Each of the five figures has a different measurement threshold imposed (none, 100 dB, 105 dB, 110 dB, and 115 dB, respectively). Figure 16 shows the case for no imposed measurement threshold (varies from 80-95 dB depending upon the recorded data). FPs are observed from the figure to be at the lowest occurrence (~0.5%) when the threshold level (abscissa) is above 0.8. FNs are lowest (~0.5%) for a threshold below 0.7. The range of best overall accuracy (the sum of FP and FN rates of ~1.5%) occurs for threshold levels between 0.35-0.75. Choosing a threshold level in this range will provide the best overall accuracy for this network. The range of acceptable thresholds (to define the decision boundary) is fairly wide (0.75-0.35 = 0.4) in this case. Figure 17 shows very similar results when a 100 dB measurement threshold is imposed. For this case, the FP and FN rates are generally lower than the previous case, yet neither can be completely eliminated. The best overall accuracy (99.2%) is achieved when the threshold level is chosen to be ~0.1-0.38. Figure 18 shows the 105dB threshold case. The best overall accuracy occurs for a threshold level between 0.1-0.25. Interestingly, the FN rate seems higher in general than the previous case, which impacts the overall accuracy. This result is likely due to the fact that a more selective set of data was used for training and testing. However, FPs can be eliminated for a threshold above 0.6. In Figure 19 (110 dB measurement threshold), a new pattern emerges. The classifier now has the ability to provide 100% accuracy (for a threshold level between 0.15-0.29). Finally, if a measurement threshold of 115 dB (Figure 20) is chosen, the decision boundary with perfect detection widens to a range of 0.1-0.9.

It is not surprising that increasing the measurement threshold improves overall accuracy and provides the opportunity to reduce or eliminate FPs. Impulse noise is typically much higher level than wind noise. The exception is when impulse noise is very distant or of relatively small size. In either case, the Lpk level is much lower, making it of less concern.

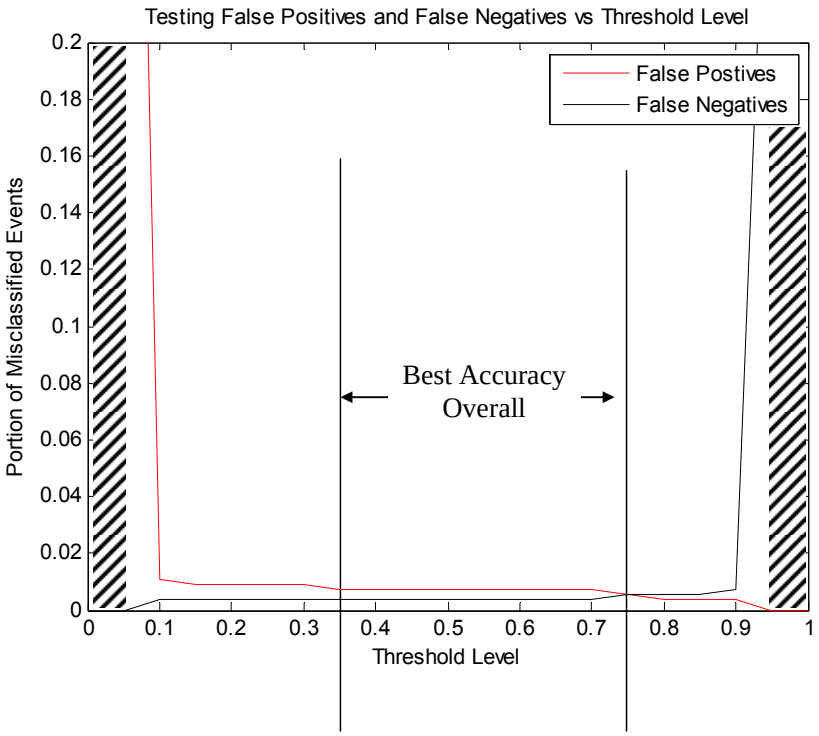

Figure 16. Accuracy of ANN classifier with no specified threshold.

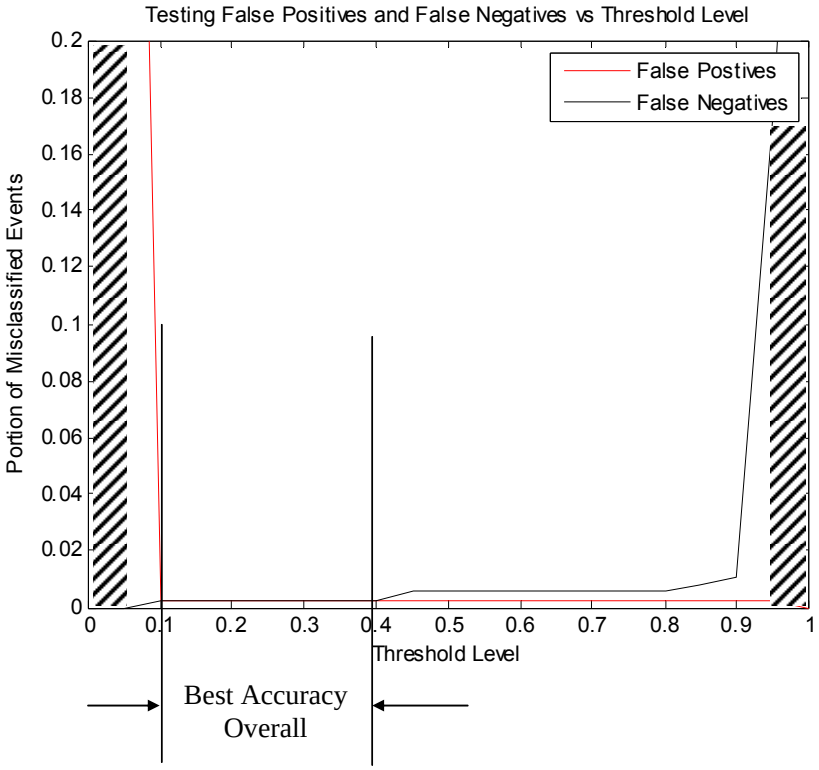

Figure 17. Accuracy of ANN classifier with 100 dB threshold.

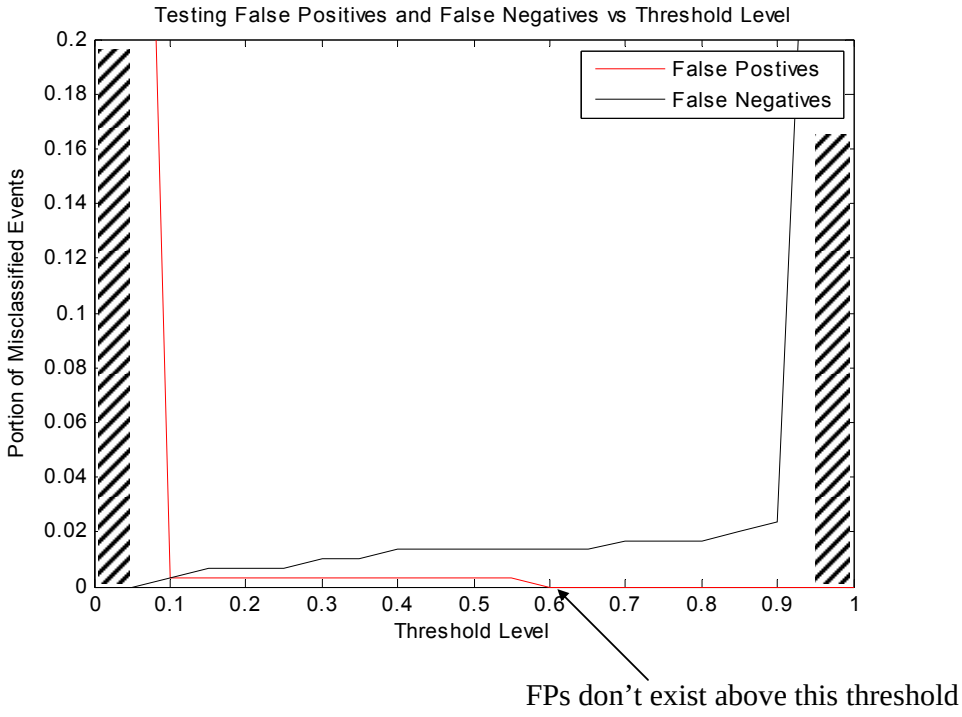

Figure 18. Accuracy of ANN classifier with 105 dB threshold.

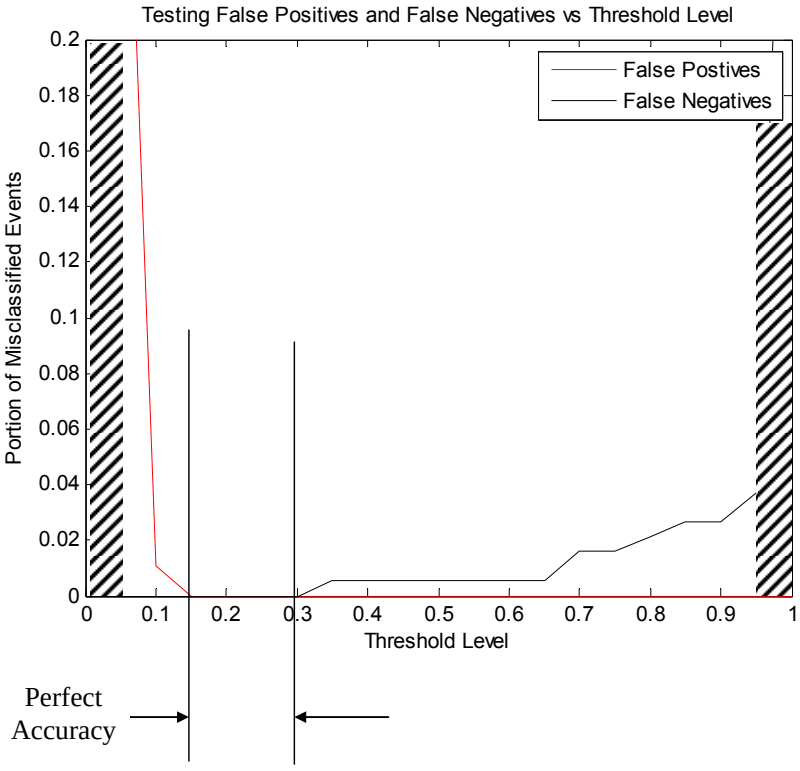

Figure 19. Accuracy of ANN Classifier with 110 dB threshold.

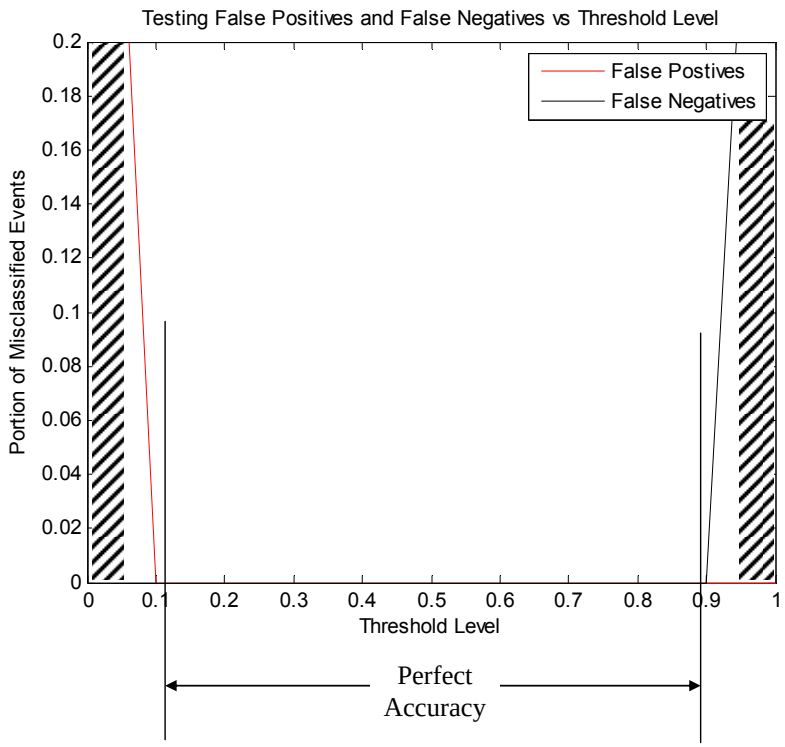

Figure 20. Accuracy of ANN classifier with 115 db threshold.

A similar study was also conducted on data that had been decimated, filtered, and re-sampled to match the hardware specifications of the McQ NoiseWatch. Results of this study can be found in Appendix B.

Results of the low-frequency cut-off study demonstrated that the good low-frequency response of the microphones is essential. It was first found that the UPitt classifiers did not work well on data sampled using the APS array microphones. The filtering study served to establish the actual lower bound on bandwidth at 0.5 Hz. From Figure 21, FNs, FPs, and total errors (FNs + FPs) are plotted versus the corner frequency of the high-pass filter. FPs are seen to start at zero and then start to increase just below 0.5 Hz. FNs are observed to start out nonzero and reach a minimum at 0.5 Hz, after which they increase along with the FPs. The minimum total error occurs at 0.5 Hz. It is questionable whether FNs really increase below 0.5 Hz, since the data acquisition was alternating current (AC)-coupled (has a low-frequency cutoff of a few tenths of a Hz).

Figure 22 illustrates the importance of retaining the low-frequency information for discerning wind noise from impulse noise. The upper trace shows a raw (blue) and high-pass filtered wind waveform (red) that has characteristics much like impulse noise. Little difference in the shape of the waveform is observed, other than the filtered waveform is attenuated slightly. The spectra of both waveforms are given in the lower plot of Figure 22. As observed, the high-pass filtering rolls off the lower spectrum of the wind making it look very similar to that for impulse noise.

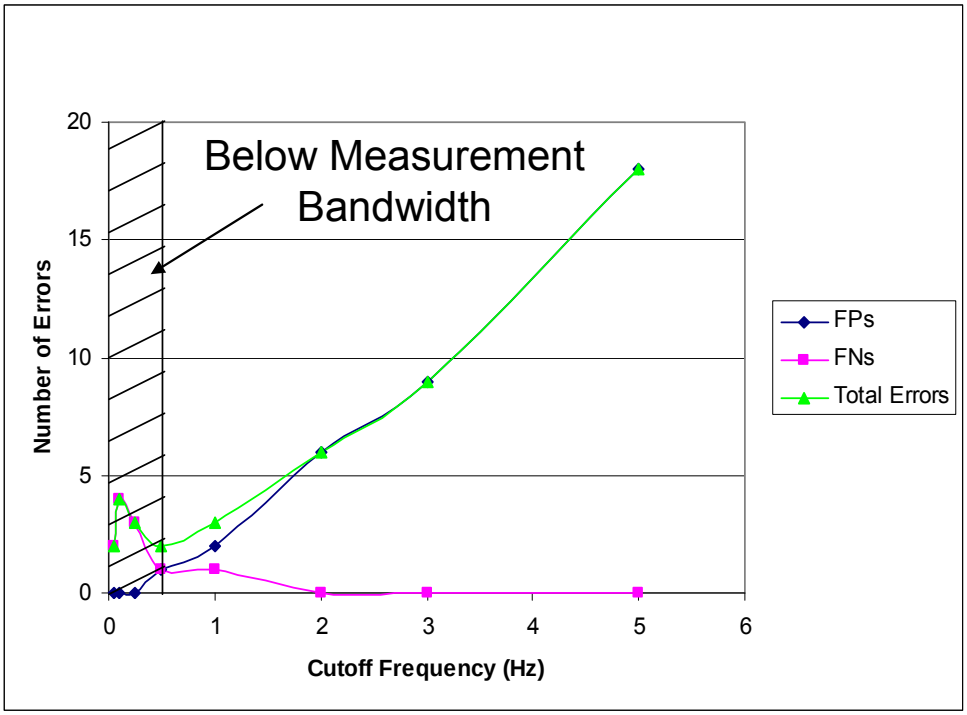

Figure 21. Classification errors as the high-pass cutoff frequency is increased.

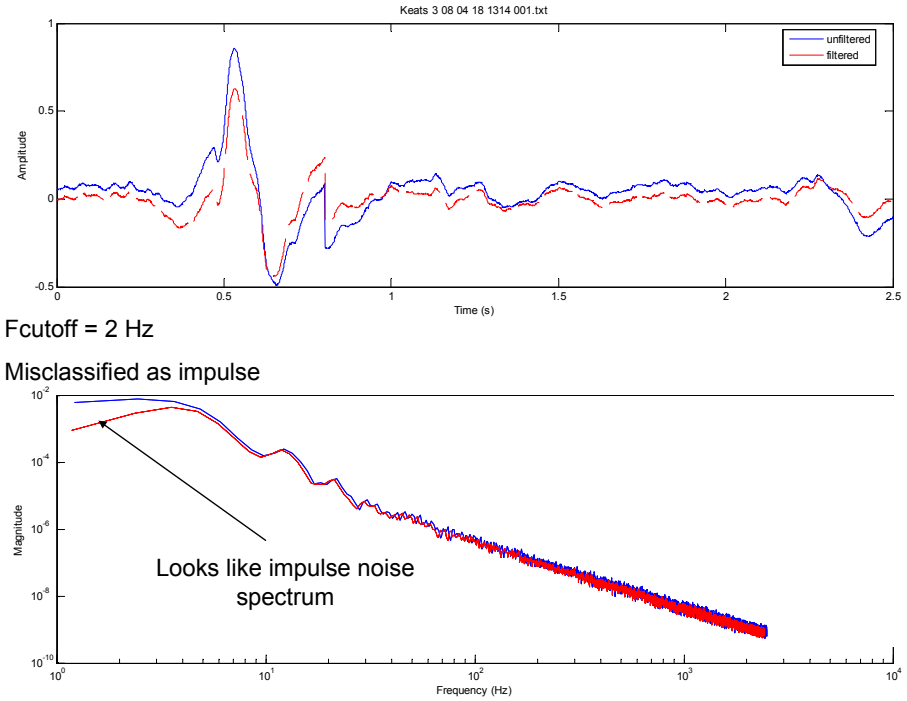

Figure 22. High-pass filtered (red) and unfiltered (blue) wind noise (top-waveform, bottom-spectrum).

5.3.4. Establish hardware requirements

The original concern that the existing hardware would not have enough computational power was no longer relevant, given that a new hardware platform was specified by APS. Further, the APS system is easily expandable by adding additional SBCs to increase computational power. Nonetheless, the computational requirements have been established for the algorithms. Table 5 gives a breakdown of the number of floating point operations (FLOPs) required to implement the metric calculations and ANN classifier for a n=5,000 point waveform. The total was 146,004 and thus is conservatively estimated to be less than 0.3 megaFLOPs (MFLOPs). A similar process was done for the Bayesian classifier, and the result was approximately 1 MFLOP. Due to implementing the algorithms on a fixed point chip, the floating point emulation will add additional overhead to the algorithms. The final code was found to execute properly and within a sufficient time window on the PC/104 platform.

Table 5. Number of computational operations in classifier.

Type of Operation	**Number of Operations**	**Number of Operations for n = 5,000**
Addition/Subtraction	$9 \cdot n + 739$	45,739
Multiplication/Division	$14 \cdot n + 540$	70,540
Exponential	270	270
Hyperbolic Tangent	2,560	2,560
Base 10 Logarithm	8,200	8,200
Square Root	200	200
Fast Fourier Transform (FFT)	$n \cdot \log(n)$	18,495
Total:		**146,004**

5.3.5. Determine Utility of Saving Statistical and other info.

Since teaming with APS, much more control over the software development was afforded. The signal metrics and waveform itself were saved and transmitted to the home base. There is also a provision to download the entire recorded waveform. Whether or not this data will be useful to the end user depends upon their interest and training levels. Retraining of the classifiers could be possible, but is somewhat discouraged, since the classifier could easily be made worse if done improperly.

5.4. Create Real-time Implementation

5.4.1. Port algorithms to C/C++

The MATLAB implementation of the algorithms was ported to C/C++. A successful C code was written to implement the ANN, as well as the calculation of the four metrics. Figure 23 shows a plot of the error between the outputs of the C-code and the original MATLAB code. These errors are relatively small (since the ANN output is between 0 and 1, errors on the order of 10^{-5} are insignificant).

Although there has yet to be a compelling reason to use the Bayesian classifier over the ANN, it was still implemented in C/C++ language. Unlike the ANN, this classifier was primarily implemented as a series of matrix operations.

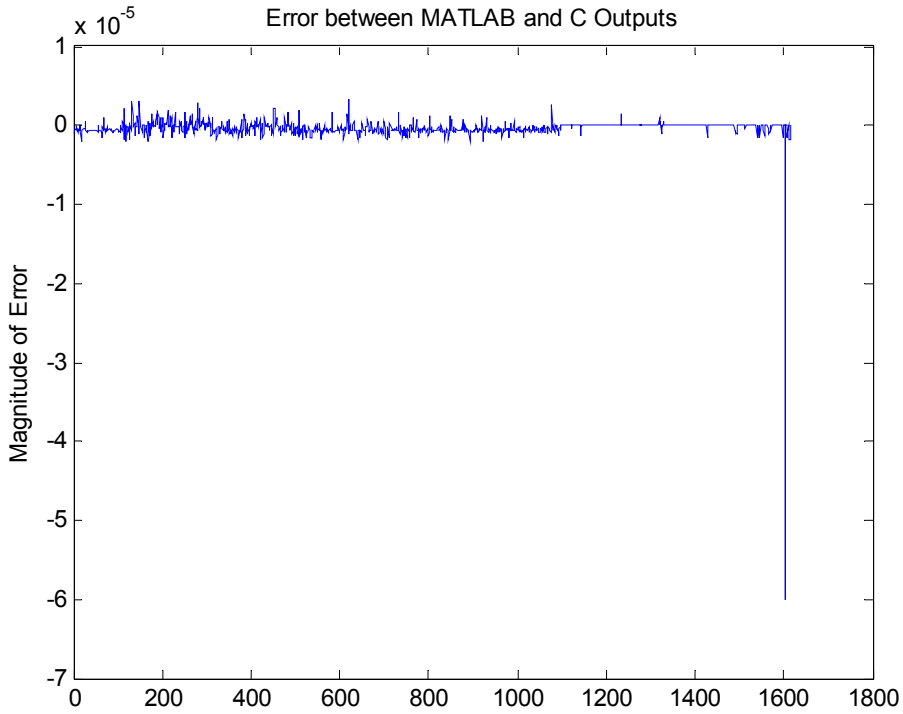

Figure 23. Error between classifiers implemented in C and MATLAB.

5.5. Develop Field Prototype Hardware

Figure 24 is a picture of the PC/104 stack which runs the BAMAS and UPitt real-time codes. The electronics are composed of four PC/104 modules, PC/104 ISA extension card, and rechargeable Li-Ion battery. The SBC developed by Arcom uses an Intel XScale PXA255 which

is a 400MHz ARM Ltd compliant central processing unit (CPU). This SBC was chosen because it dissipates only 2W (during typical operation) which allowed us to test the system for an entire day. Additionally, the CPU speed is sufficient enough to run the BAMAS and UPitt codes in real-time. Power, in the form of ±5V and ±12V, is regulated by a 60 Watt power supply (stacked beneath the CPU board). This particular power supply accepts SMBus (Smart Management Bus) compatible batteries which allows for battery-controlled charging. The data acquisition board located at the rear of the housing is capable of sampling 16 channels, with 16 bit resolution, at 200kHz.

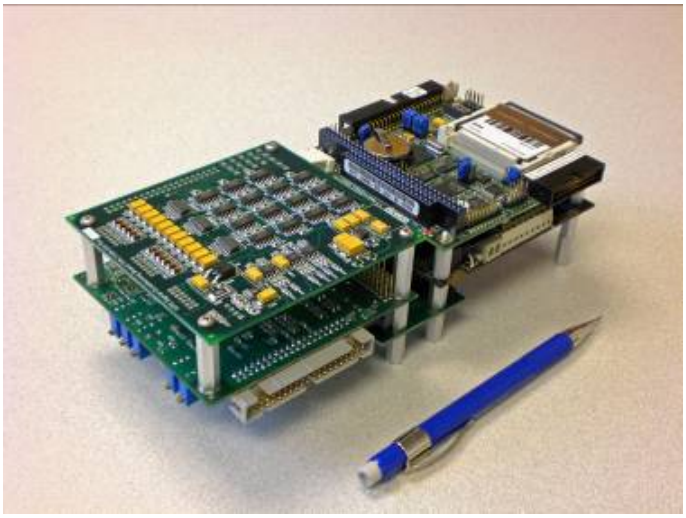

Figure 24. PC/104 Stack with low power 400MHz CPU, 200kHz data acquisition (DAQ), power supply, and custom signal conditioning board.

A block diagram of the BAMAS system is shown in Figure 25. Each remote system has the ability to store local detection information and then communicate that information to a central base station where it will be fused (i.e. triangulation) and logged to disk. This log file could be configured so that it may be imported, for example, into Google Earth or Google Maps using their open source applications programming interface (API) and can also be imported into Access or Excel for statistical analysis. Each waypoint (both detected events and false alarms) cataloged in Google Earth (see Figure 26) will display detailed detection statistics such as source level, time and date, and waypoint covariance (this may appear as an ellipse around the waypoint).

Individual BAMAS units communicate wirelessly with the base station computer using PC/104 compliant 900 megahertz Serial Port or Ethernet bridges. These can transmit/receive long distances approaching 20 miles using Yagi-directional antennas allowing the system to monitor large areas.

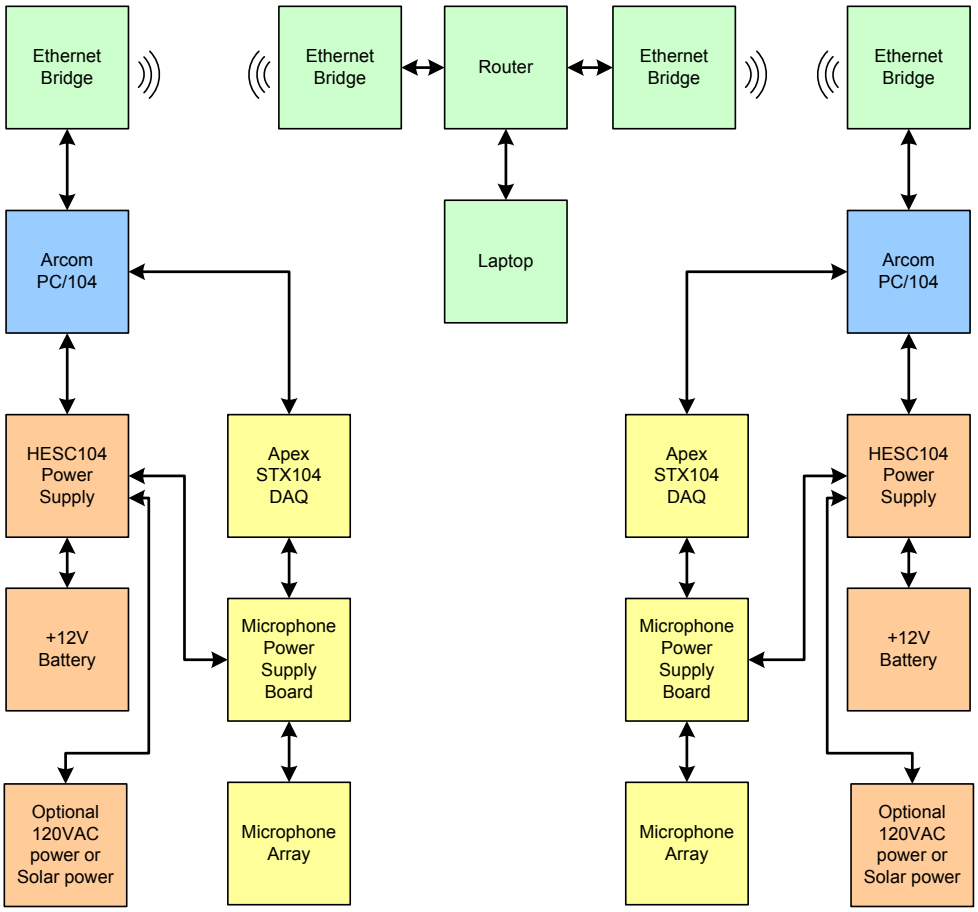

Figure 25. Block Diagram of BAMAS System Configuration.

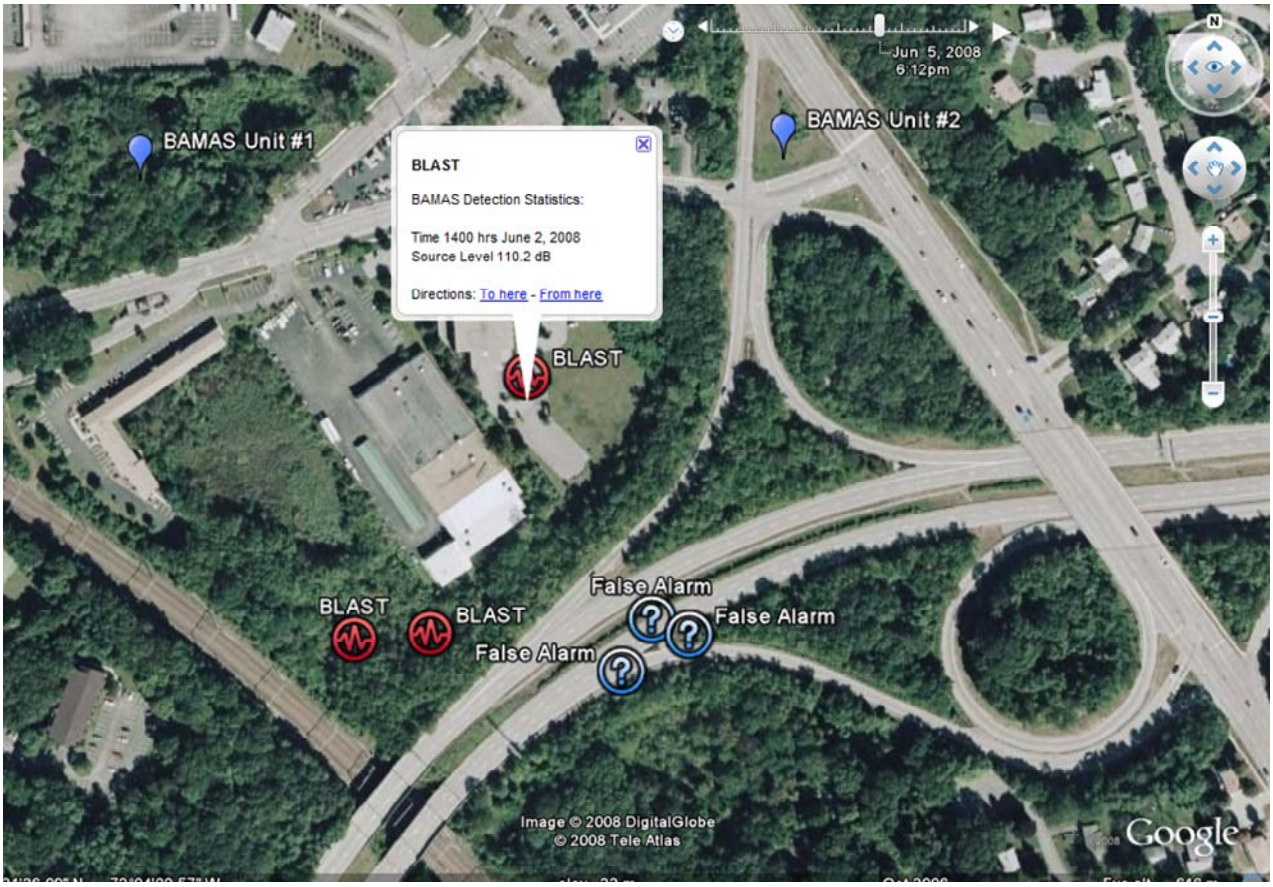

Figure 26. Detected events cataloged by BAMAS System will be displayed in Google Earth. Detailed information regarding detection statistics (i.e. source level, time, and date) and false alarms will appear at each waypoint.

5.6. Results of Field Trials

Two prototypes were constructed by APS and deployed by APS and UPitt at a participating military base during the week of 19-October 2009. Both units continue to operate today. It was desired to install prototypes by existing BLAM monitors that receive a lot of events in order to make comparisons. This was not possible since some of these BLAM monitors were not working. One prototype was installed near a distant BLAM unit which receives few blasts. The other was installed near an observation point where there was no BLAM monitor. The observation point represents a somewhat atypical place to install a monitor, given that it is nearer the impact zone instead of the perimeter of the base. This makes the monitor susceptible to operational noise (especially vehicles and electronic noise) and small arms fire that more distant noise monitors won't receive. However, the number of blast events recorded will be much higher. Soon after deployment of the monitor, the ANN was retrained on a limited set of early recorded data to improve performance. Retraining was required for two reasons. One was that

the classifier needed to be trained on metrics computed by the C codes. Outputs were very slightly different, yet produced somewhat different results with the ANN. The second reason was to train the classifier on the new sources of sound associated with close monitoring.

Each prototype monitor records data on a compact flash card. Data from 29-Oct to 11-December 2009 were obtained to assess operation. A laptop was sent to MCBCL to download the data, which in turn was then sent to APS, who shared the data with UPitt. Unlike the data collection trips, the data on the CF had to be classified after the fact. A person listened to each recorded waveform, plotted it along with its spectrum, and looked at detected output from BAMAS (could better see acoustic events buried within the noise). In most cases the person could tell what the source was, but 240 of the 9,168 total waveforms were unknown and thus not used in the analysis. Of the remaining 8,928 useful waveforms, 3,741 were blasts and the remaining 5,187 were not. The data were processed to assess and compare operation of the algorithms. From the onset, the intention was to compare the performance of the UPitt classifier, APS detector, BLAM monitors, and combination of APS detector and UPitt classifier.

5.6.1. Performance of UPitt Classifier

Table 6 shows the accuracy of the UPitt Classifier for eight (8) different classes of noise: larger blasts, smaller blasts, mixed blasts, machine gun, aircraft, electronic, vehicle, and wind noise. Most of these are self-explanatory. The boundary between larger versus smaller blasts is probably around 25-40 mm. Electronic noise was likely due to base communications (radios). Vehicle noise consisted of some loud personal vehicle (car with performance muffler and a Harley Davidson motorcycle with the same), but mostly tanks driving by (could hear the gas turbine engines). Machine gun was chosen as "not blast" since study only focuses on ordnance that is 20mm and up. Mixed blasts included two types of waveforms. The first is partial events that were not pretriggered properly. This reduces the SNR, since the whole waveform is not captured in the file. The second type of waveform was those that had significant, less typical background noise (as a result of being so close to the operations). By putting these types of waveforms in their own category, their effects on the system performance can be separated.

Table 6. Classifier Results for various noise sources (Overall Accuracy=89.2%, Error=10.8%, FPR=2%, FNR=8.8%).

Noise Source	**Total # of Waveforms**	**Blast Sources**		**Nonblast sources**	
		True Pos. Rate (%)	False Neg. Rate (%)	True Neg. Rate (%)	False Pos. Rate (%)
Larger Blast	2510	90.0	10.0		
Smaller Blast	480	89.2	10.8		
Mixed Blast	751	35.6	64.4		
Machine Gun	648			78.4	21.6
Aircraft	940			99.6	0.4
Electronic	128			99.2	0.8
Vehicle	1739			99.1	0.9
Wind	1732			99.1	0.9

Results in Table 6 show that the accuracy has dropped from the 99% observed during the development phase to about 89% overall. However, rejection of nonimpulse noise (aircraft, electronic, vehicle, and wind) remains quite high (>99%) and thus the goal listed in the statement of need of the project to substantially reject wind noise was met. Accuracy suffered when compared to the development stage in categorizing impulse noise (90%). As suspected, the mixed noise posed problems for the classifier (35% accuracy). A challenge was also posed by categorizing machine gun as non-blast noise. It was observed that when several machine gun shots are recorded in a single waveform, the blast characteristics become stronger, which is reflected in the classification as blast noise. Again, most noise monitors would not be exposed to machine gun fire, but there may be ways to ameliorate this problem. Although the total false positive rate (FPR) is 2%, most of that is due to machine gun noise, which ordinarily would not be in the vicinity of a noise monitor. The false negative rate (FNR) is 8.8%.

Part of the analysis also assessed whether retraining the ANN could improve performance. Just the waveforms recorded by the prototypes were used in the process. Modest improvements were observed (91.8% accuracy) after retraining using all of the collected data. Further improvements were noted when throwing out mixed (95.7% accuracy) or machine gun noise (97.8% accuracy). When both mixed and machine gun noises were thrown out, the observed accuracy was above 98% (compared to 95.5% without retraining).

All of the above analysis uses a classifier threshold of 0.5 for detection. An optimal threshold can be made based upon the subjective desire to control the FPR. Since this is a binary classifier, a receiver operating characteristic (ROC) curve can be generated to show the performance for various detection thresholds.

Figure 27 shows a set of ROC curves for the classifier. Each line represents a different measurement threshold from 90 dB to 120 dB. Higher measurement thresholds have higher SNR resulting in better performance of the classifier. As a refresher, a perfect classifier would go up the ordinate and across all FPRs at 100% true positive rate (as the 120 dB line does). A chance detector would create a 45 degree line from the bottom left to the top right. The performance is observed to be much better than chance, but less than perfect. Choosing a classifier detection threshold can be associated with an operating point on each curve. For example, for the 90 dB line, a 10% FPR occurs with an 84% true positive rate (TPR). Performance improves with increasing measurement threshold, as expected.

Figure 28 shows the same ROC curve, but with the mixed noise sources removed. Somewhat better performance is noted for all measurement thresholds. If proper pretriggering were implemented and the classifier located away from certain noise sources, this is the performance that could be expected. Removing the machine gun noise further improves the classifier ROC curves (not shown).

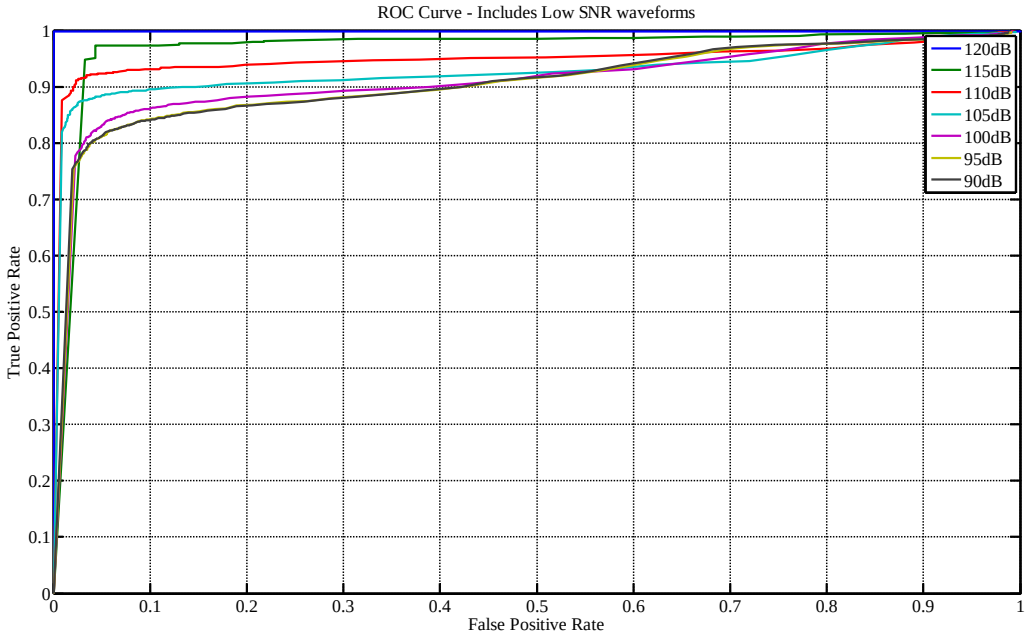

Figure 27. ROC curve of UPitt classifier

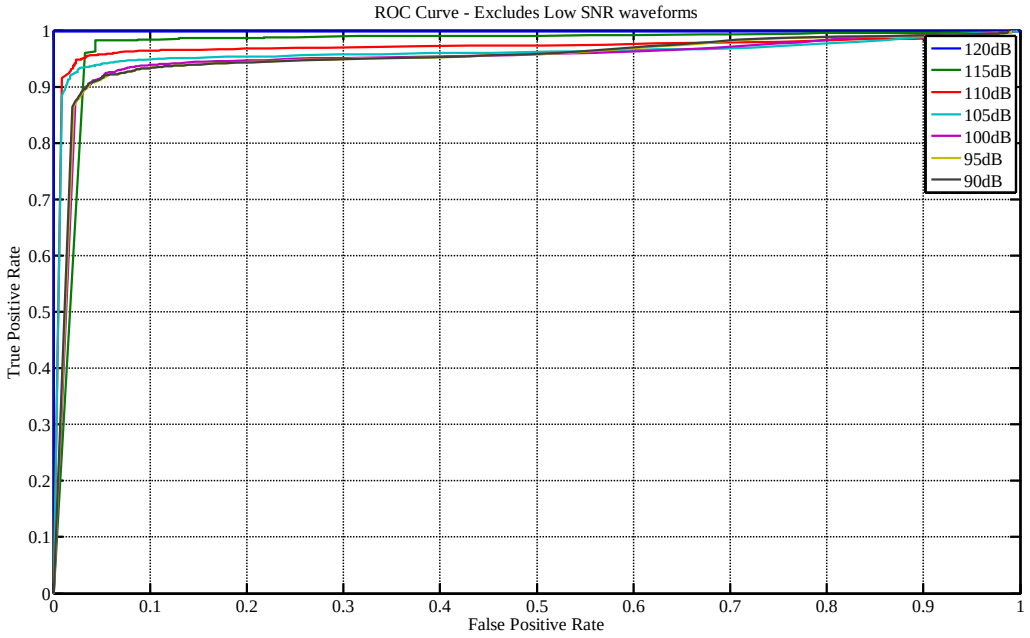

Figure 28 ROC curve with mixed noise removed.

5.6.2. Performance of APS Detector

The APS BAMAS algorithm is not a signal classifier and is not expected to operate well as such. Rather, it is an acoustic detector. The detection algorithms utilize the data generated by the acoustic array. Nevertheless, Figure 29 shows an ROC of signal classification ability. The performance is poor as expected. It can well discern between acoustic (blast, machine gun, vehicle, and aircraft) and nonacoustic (wind, electronic) noise, but can't discern blast noise from all other acoustic sources. As such, when the mixed noise is removed from the waveforms and the new ROC created (not shown), it is nearly identical to Figure 29. In contrast, Figure 30 shows the ROC for its detection capabilities (acoustic vs. nonacoustic), which are considerably better. These are based upon the RMS error of the time delay as the detection variable [Allanach (2009)]. As with the UPitt classifier, acoustic detection performance improves with SNR. It stands to reason that combining the BAMAS detection abilities with the UPitt classification capability would provide a classifier with better performance than either alone.

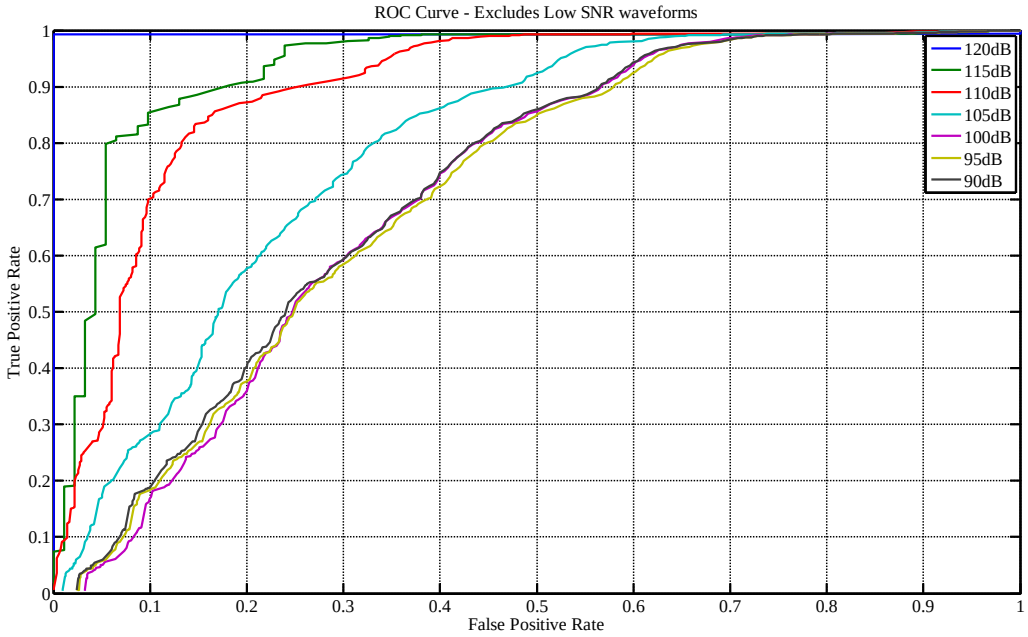

Figure 29 ROC for BAMAS used as a classifier (not its intended purpose).

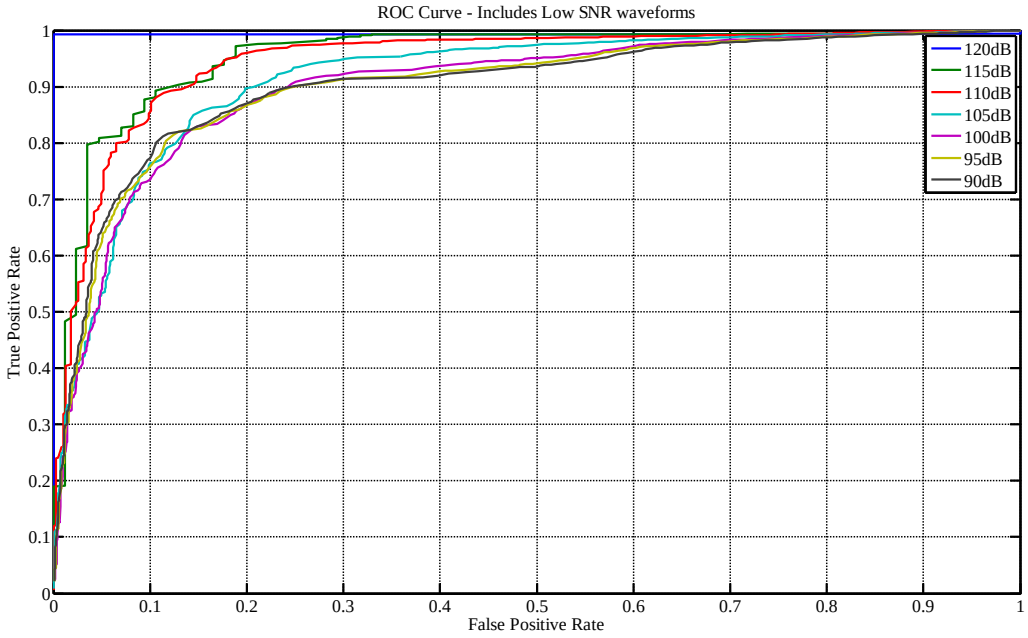

Figure 30 ROC for BAMAS used as an acoustic detector.

5.6.3. Performance of BLAM

A side-by-side comparison of the prototypes and existing BLAM monitors was not possible since the BLAM located next to the prototype wasn't working. However, during some of the data collection, the BLAM performance was observed and presented here. Figure 31, shows a combination of recorded events ("Collected Data") and BLAM triggers from two different neighboring monitors. The data collection window spans from about 11:30 to 14:50 hours. The UPitt recorded events are more plentiful than the BLAM events, owing to the much lower measurement threshold (about 90 dB compared to 115 dB). However, there is a discrepancy between the peak levels measured by the permanent BLAM noise monitor and the UPitt equipment, suggesting that the monitors are in need of calibration. The UPitt hardware was routinely checked for accuracy. An end-to-end calibration was performed by recording 94 and 114 dB calibration tones at the beginning and end of every measurement session. Figure 32 shows BLAM reports for a Sunday on which the largest ordnance fired was shotguns and .50-caliber machine guns. These would not trigger the monitors owing to the comparatively small size and large distance the sound would have to travel. In addition, five monitors are simultaneously triggered, which is also highly unusual. Also plotted in the solid lines is the sustained and gust wind speeds. Note that the BLAM events seem correlated with the wind speed. It appears that the BLAM events are false positives caused by the wind.

5.6.4. Performance of Combined UPitt and BAMAS Algorithms

The UPitt classifier and APS detector were combined to form a classification system with the best of both worlds. In this mode, the UPitt algorithm classified the detected output of the

BAMAS algorithm. The detected output rejects wind noise by averaging the four channels. Overall accuracy did improve by over 3% to 92.4%. However the FPR has more than doubled at 4.8%, while the FNR is a modest 2.8%. The performance for individual types of noise is given in Table 7. Now blast noise identification has improved to over 98% accuracy and mixed blast noise at 73%. These results are not surprising considering the wind rejection. Rejecting the wind apparently also relaxed some of the low-frequency response requirements of the microphones. Performance in discriminating machine and electronic noise is worse, however. The machine gun noise result is not surprising either, since the wind noise actually helps drown out the machine gun making it appear as nonblast noise. It is not known for sure why electronic noise rejection is worse. It is suspected that the microphones act like antennas, picking up the high level RF energy. The noise rejection will strengthen the coherence between channels and possibly make it look more refined. As mentioned earlier, the electronic noise is easily rejected by detecting zero phase between the four microphone channels. Finally, Figure 33 shows the ROC of combined UPitt/BAMAS System. Notable performance enhancements are noted, particularly with the TPRs. Figure 34 shows the ROC when the mixed noise is thrown out, showing even better performance.

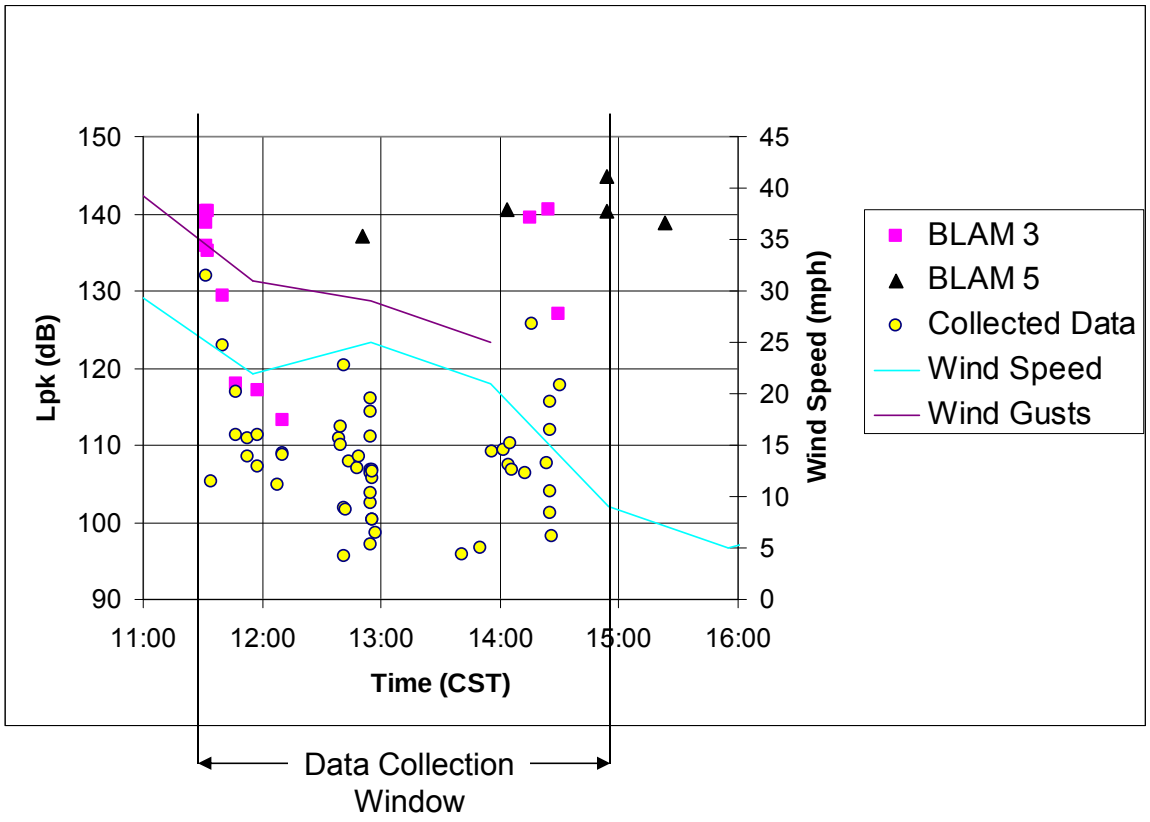

Figure 31. Comparison of field measurments and BLAM events.

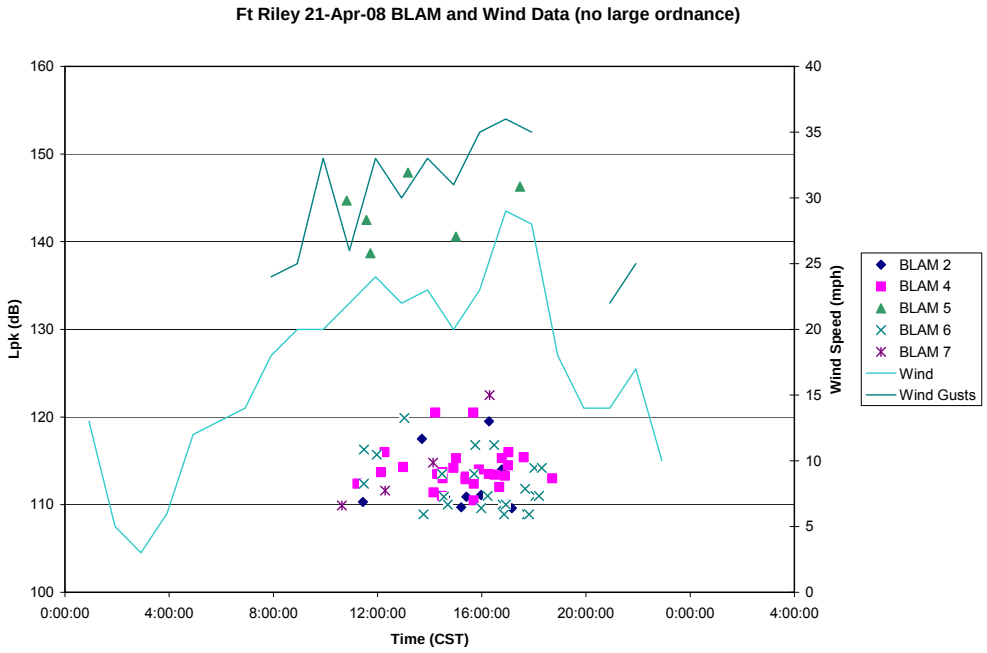

Figure 32. BLAM wind triggers.

Table 7. Accuracy of combined UPitt/BAMAS System (Overall Accuracy=92.4%, Error=7.6%, FPR=4.8%, FNR=2.8%).

Noise Source	**Total # of Waveforms**	**Blast Sources**		**Nonblast sources**	
		True Pos. Rate (%)	False Neg. Rate (%)	True Neg. Rate (%)	False Pos. Rate (%)
Blast	2510	98.1	1.9		
High-f Blast	480	98.8	1.2		
Mixed Blast	751	73.3	2.7		
Machine Gun	648			45.1	54.9
Aircraft	940			99.3	0.7
Electronic	128			85.2	14.8
Vehicle	1739			98.9	1.1
Wind	1732			98.4	1.6

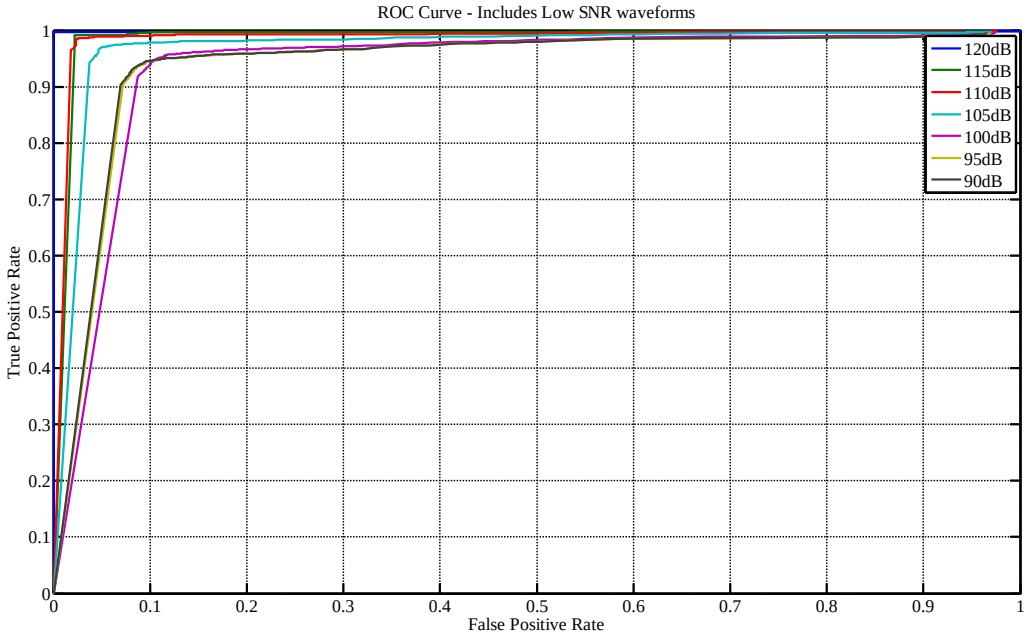

Figure 33. ROC of Combined UPitt/BAMAS System.

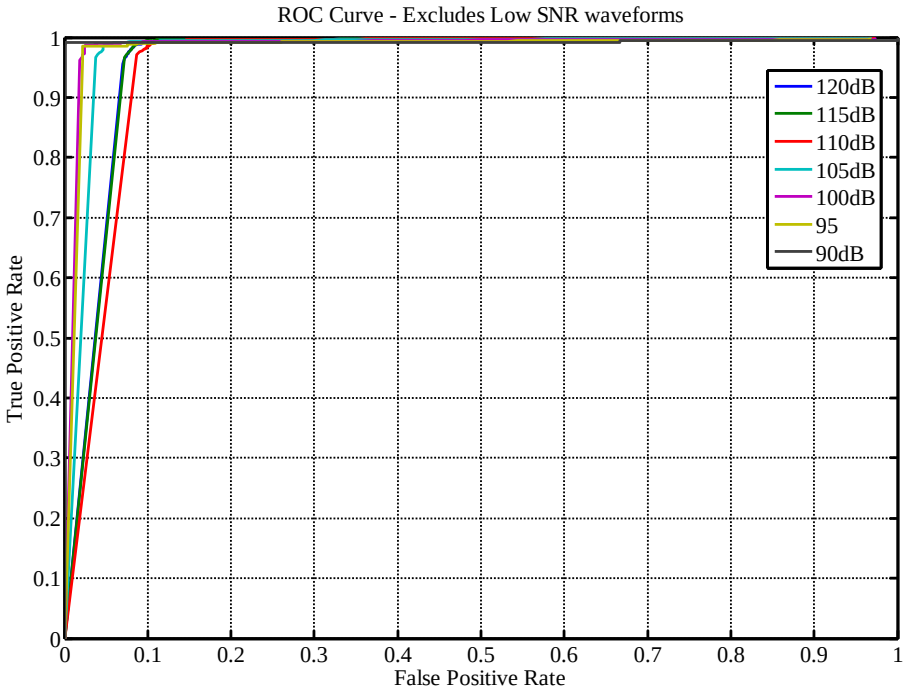

Figure 34. ROC for Combined UPitt/BAMAS System With Mixed Noise Thrown Out.

6. Conclusions and Implications for Future Research/Implementation

In summary, this SERDP project SI-1585 was a 2.9–year continuation of a 1 year SERDP Exploratory Development (SEED) project. The five specific aims or tasks of the project were to:

1) Expand the library of noise measurements started in the SEED effort, which includes all high level noises around a military base
2) Refine the noise classifiers developed in the related SEED effort
3) Establish hardware requirements for algorithms
4) Create a real time laboratory demonstration using the newly developed algorithms
5) Develop and demonstrate a prototype system on a military base.

All five of the project tasks have been completed. For Task 1, trips were made to Fort Hood, TX, Fort Riley, KS, Fort AP Hill, VA, and Marines Corps Base Camp Lejeune, NC (MCBCL), Fort Indiantown Gap, PA, Fort AP Hill, VA, Fort Carson, CO and Fort Benning, GA. Fort Riley is one of the bases reportedly plagued by wind noise. Several false positive triggers in the BLAM monitors were noted for the period during which the measurement trip was made. In FY09, two measurement trips were conducted to Fort Benning, GA. The team also returned to MCBCL to deploy the two prototype systems. The Fort AP Hill data were also especially meaningful, since they were conducted under very controlled conditions. Here, the measurement distance was varied as anti-tank mines were detonated. Measurements were found to follow the inverse-square-law as the distance to the source was increased. To date, more than 3,000 usable waveforms of impulse noise, wind noise, and vehicle/aircraft noise have been collected. Two trips to Fort Benning, GA yielded more recordings of Bradley fire.

In Task 2, it was found that the four metrics (kurtosis, crest factor, spectral slope, and weighted square error) provide a suitable feature set for a military impulse noise classifier. Both the ANN and Bayesian classifiers that are based upon the metrics performed comparably with accuracies as high as 100% observed. It is not surprising that accuracies are similar given that they use the same feature space. The tANN classifier, which investigates the shape of the waveform, was found to not perform as well as the metric-based classifiers (94% accuracy), and hence as abandoned.

Task 3 was largely completed by UPitt in FY08. Originally, Task 3 was to be a joint effort between UPitt and the original subcontractor, since the original contract called for the integration of the UPitt algorithms into an existing noise monitoring product. When the original subcontractor chose not to partner for the project (they were replaced with APS as the subcontractor), UPitt assumed the work of these tasks. In particular, the high fidelity data were degraded to match the specifications of the McQ NoiseWatch product (sampling rate of 1 kHz, 400 Hz 4^{th} order low-pass filter, and 12-bit ADC resolution). The classifiers with the degraded data were found to perform reasonably well, though they lost a little accuracy at lower thresholds (higher SNR). A study to examine the effects of dynamic range was devised. An artificial measurement threshold of 100, 105, 110, and 115 dB was imposed. The resulting data for each case were in turn used to train and test the classification algorithms. It was found that overall accuracy improved as the threshold was increased, with 100% accuracy achieved at 110 dB

threshold and over 99% accuracy at 100 dB threshold. Another significant finding was that the low-frequency response of the measurement hardware should go down to at least 0.5 Hz. In particular, the low-frequency information was found to be very important to discern wind noise (which is pink in quality – that is it has equal energy in all octaves such that the log-log plot of the power spectral density is a downward sloping line) from impulse noise. Also performed in Task 3 was the conversion of the algorithms to fixed point and determining the computational requirements. Accuracies were not found to significantly degrade with fixed-point computation. The metric-based classifiers were found to require about 1 MFLOP to complete, which is a modest computational burden. Since APS has specified a new hardware platform (with input from UPitt), these fidelity issues became less important. The PC/104-based system was scalable and had four array microphones and one high quality microphone that can be used to measure data for the BAMAS and UPitt classifiers, respectively. The more formal subcontracting relationship with APS resulted in better integration of the BAMAS/UPitt algorithms. For example, UPitt was able to also classify the "detected output" from BAMAS, which has inherent wind rejection.

Task 4 saw integration of the C codes. The classifier codes which were implemented in MATLAB required extensive rewriting before porting them to C language. For example, where MATLAB supports vectorized operations, C requires loops. In addition, routines had to be written, obtained, or modified to handle the built in subroutines in MATLAB (FFT, maximum value, linear interpolation, tanh(), etc.) The codes were first developed and tested on a PC. UPitt then ported them to the target PC/104 platform, which runs under Linux. This task culminated with bench testing by UPitt and APS at the APS facility.

Prototype hardware specification and deployment made up Task 5. The prototype has a number of exciting features, including low-power operation, solar/battery power operation, 16-channels of 16-bit ADC, RF communication to a home base, and integration of the data into Google Maps or Google Earth. The devices were remotely managed by APS and UPitt through the cell phone and internet hookups. An anemometer will also collect wind data to help validate wind FPs. In October, 2009, two field prototypes were installed at Marine Corps Base Camp Lejeune, NC. These systems ran continuously and collected data to verify classifier performance.

The performance of the UPitt and BAMAS algorithms was analyzed separately and together. The goal of the project was to classify with 99% accuracy. While the overall accuracy of the UPitt classifier individually was somewhat less (89%), nonblast noise was rejected at over a 99% rate. This meets the military objective in the original SERDP Statement of Need (SON) to significantly reduce the number of false positives, which were estimated to be as high as 10%. The BAMAS array and algorithm was shown to work well at acoustic detection, but not classification. The detection algorithm can discern between acoustic and nonacoustic noise sources, but not whether any acoustic noise is from blasts. When BAMAS is combined with the UPitt algorithm, blast noise was identified at over 98% accuracy, while aircraft, wind, and vehicle noise were rejected at over 98-99%. Classification of mixed noise (lower SNR) improved from about 35% to 73%. However, the ability to reject machine gun and electronic noise diminished slightly in the combined classifier.

The prototypes were successfully implemented and the DoD objectives were met. The system is at Technology Readiness Level (TRL) 7 and is ready for the demonstration/validation phase. ESTCP funds will be sought for this purpose. At this point, a long term implementation is needed at a base in order to further refine the system. For example, additional attention can be focused on handling new sources of noise that are germane to specific monitoring locations. It needs to be determined if retraining can be eliminated from future implementations, since new types of background noise (though atypical) necessitated retraining in the prototypes. The UPitt classifier should be compared to more types of existing classifiers (e.g. CHPPM (US Army Center for Health Promotion and Preventive Medicine), Schomer's fuzzy logic, BLAM, Boom Event Analyzer Recorder (BEAR), and human classification). Although the most important function of the classifier is to distinguish blast noise from all others, some military representatives have expressed interest in expanding the classifier to distinguish between different noise sources. Example sources include large ordnance, small arms, rotary and fixed wing aircraft, trains, various vehicles, storms, etc.). This task would require some structural changes to the classifier, but similar changes have been demonstrated during this project (e.g. it has been demonstrated that the algorithm can distinguish between wind, aircraft, and blast noises). In terms of creating a final product, the user interface could be made even more user-friendly. Classification/detection results should be integrated into existing range management software such as a geographic information system (GIS) and/or range managers toolkit (RMTK). A calibration procedure also needs to be developed. One could also integrate knowledge of noise source locations into the system to provide a statistical measure of the likelihood of an event or misclassified event. In addition, statistics can be compiled for each base location. These suggestions are enhancements to the base technology, which has been successfully demonstrated.

7. Literature Cited

Abe, S. (2001), Pattern Classification: Neuro-fuzzy Methods and Their Comparisons, Springer-Verlag, London, England, 2001.

Alanach, J., Borodinski, J., and B. Abraham, (2009) SERDP (SI-1427 Final Report), "Impulse Noise Bearing and Amplitude Measurement and Analysis System (BAMAS)," http://www.serdp-estcp.org/content/download/6560/86531/file/WP-1427-FR.pdf

Anil, J., P. Duin, J. Mao, (2000) "Statistical Pattern Recognition: A Review," IEEE Trans. on Pattern Anal. and Mach. Intell., 22(1).

ANSI (1993). Methods for measurement of impulse noise. New York, NY: American National Standards Institute, ANSI S12.7-1986 (R1993).

ANSI (1996), Impulse Sound Propagation for Environmental Noise Assessment, American National Standards Institute, New York, NY, ANSI S12.17-1996

Attias, J., A.Y. Duvdevany, I. Reshef-Haran, M. Zilberg, and N. Beni. (2004) Military noise induced hearing loss. In: Handbook of effects of noise on man. Luxon, L (Ed), London.

Bendat, J. and A. Piersol, (1986) *Random Data: Analysis and Measurement Procedures.* John Wiley and Sons, Inc. New York, NY, 2nd ed.

Benson, J. (1996). "A real-time blast noise detection and wind noise rejection system." *Noise Control Engineering Journal*, **44**(6), November-December 1996, p. 307-314.

Bucci, B. A. and J.S. Vipperman, (2006a) "Development of Artificial Neural Network Classifier to Identify Military Impulse Noise," 151st meeting of Acoustical Soc. of Am., Providence, RI, June 5-9, 2006.

Bucci, B.A. and J.S. Vipperman.(2006b) "Artificial neural network military impulse noise classifier." IMECE2006-14065, Proceedings of IMECE 06: 2006 ASME International Mechanical Engineering Congress, November 5-10, 2006, Chicago, Illinois.

Bucci, B. A, (2007) "Development of Artificial Neural Network-Based Classifiers to Identify Military Impulse Noise," MS Thesis, University of Pittsburgh, December 2007.

Bucci, B. A. and J.S. Vipperman (2007a), "Comparison Artificial Neural Network Structures to Identify Military Impulse Noise," 153rd meeting of the Acoust. Soc. of Am., Salt Lake City, June 4-8, 2007

Bucci, B.A., and J.S. Vipperman (2007b), "Development of an artificial neural network-based classifier to identify military noise," *J of Acoust. Soc. of Am.*, **122**(3), pp. 1602-10, Sep. 2007.

Bucci, B. A. and J.S. Vipperman (2007c), "Bayesian Military Impulse Noise Classifier," IMECE2007-41700, ASME IMECE-07 Conference, Nov 11-15, 2007, Seattle, Washington.

Bucci, B. A. and J.S. Vipperman, (2008) “An Investigation of the Characteristics of a Bayesian Military Impulse Noise Classifier,” Proceedings of NCAD2008, Paper NCAD2008-73046, Dearborn Michigan, July 28-30, 2008.

Burr, D. (1986) “A neural network digit recognizer.” IEEE Conference Procedings, Atlanta, GA, Oct. 1986, pp 1621-1625.

Cabell, R.H., Fuller, C.R. and O’brien, W.F., (1992), “Identification of Helicopter Noise Using a Neural Network,”*AIAA Journal*, 30(3), pp.624-630.

Chan, R.W.Y., and Hay. R.D., (1982), “A Case Study of Sensitivity of Some Pattern Classifiers used in Sorting Acoustic Emission Signals,” *Review of Progress in Quantitaive Nondestructive Evaluation*, Vol. 3a, pp. 413-420.

Chong, E. and S. Zak (2001), An Introduction to
Optimization. John Wiley and Sons, Inc. New York, NY, 2nd ed., 2001.

Cortes, C. and V. Vapnik. (1995) “Support-Vector Networks.” Machine Learning, **20**(3), September 1995, p.273-297.

Fidell, S. editor, (1996), “Community Response to High-Energy Impulsive Sounds: An Assessment of the Field Since 1981,” CHABA, National Research Council, National Academy of Sciences, Washington, DC.

Ganesan, R., T. Das, V. Venkataraman, (2004) “Wavelet-based multiscale statistical process monitoring: A literature Review,” IIE Transactions, 36, pp. 787-806.

Greene, R. (1997), “Noise source identification using acoustic signature and predicted magnitude,” Noise-con-97, Part 2/2, University Park, PA, pp. 109-114.

Guo, H. and C. S. Burns, (1997) “Wavelet Transform Based Fast Approximate Fourier Transform”, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-97, Munich, April 20–24, 1997, vol. III, pp. III:1973–1976.

Haykin, S. (1999), Neural Networks: A Comprehensive Foundation, Prentice Hall, Upper Saddle River, New Jersey, 1999.

Horvath, P., and Cook, F.J., (1982), “Establishing Signal Processing and Pattern Recognition Techniques for in-flight Discrimination Between Crack-Growth Acoustic Emissions and Other Acoustic Waveforms,” Review of Progress in Quantitaive Nondestructive Evaluation, Vol. 3a, 1982, pp. 463-473.

Jagota, A. (1990), “Applying a Hopfield-style network to degraded text recognition.” International Joint Confernce on Neural Networks, San Diego, CA, June 1990, pp. 27-32.

Jaynes, E.T. (1985), “Bayesian Methods: General Background,”Maximum Entropy and Bayesian Methods in Applied Statistics, J. H. Justice, Editor, Cambridge University Press; pp. 1-25.

Jaynes, E.T., (2003), “Probability Theory: The Logic of Science,” Cambridge University Press.

Kohonen, T., (1990) “The self-organizing map,” Proc. IEEE 78(9), 1464–1480.

Kosko, B. (1992), Neural Networks and Fuzzy Systems, Prentice Hall, Englewood Cliffs, NJ
Luz, G. A. (1980), "Suggested procedures for recording noise complaints at army installations." United States Army Environmental Hygiene Agency, Aberdeen Proving Ground, MD 21010-5422. HSE-08/WP Technical Guide, April 1, 1980.

Luz, G.A. (2004), Evaluatino of the Performnace of the Blast Analysis and Measuring (BLAM) System, J. Acoust. Soc. Am. 115 2497, 2004.

McQ Associates, "NoiseWatch Automated Remote Noise Monitor," http://www.mcqinc.com/products/noisewatch.php

MIL-STD-1474D, (1997), "Department of Defense Design Criteria Standards, Noise Limits"AMSC A7245.

Kohonen, T. (1990), "The self-organizing map." Procedings of the IEEE, **78**(9), p. 1464-1480.

Moser, J.M., and Aunon, J.J., (1986), "Classification and Detection of Single Evoked Brain Potentials Using Time-Frequency Amplitude Features," IEEE Transactions on Biomedical Engineering, Vol. BME-33, No. 12, pp. 96-1106.

Moukas, P., Simpson J., and Norton-Wayne, L., (1982), "Automatic Identification of Noise Pollution Sources," IEEE Transactions on Systems, Man, and Cybernetics, SMC-12(5), 1982, pp. 622-634.

Norton, M. and D. Karczub (2003), Fundamentals of Noise and Vibration Analysis for Engineers, Cambridge University Press, Cambridge UK, 2003.

Nykaza, E., L.L. Pater, and R.H. Melton, (2009) "Minimizing Sleep Disturbance Fom Blast Noise Producing Training Activities For Residents Living Near a Military Installation, JASA 125(1), pp. 175-84, Jan 2009.

Office of Economic Adjustment, Office of Assistant Secretary of Defense, and Economic Security, (1993), "Joint Land Use Study." November 1993.

Quatieri, T. (2002), Discrete-Time Speech and Signal Processing Principles and Practice. Prentice Hall, Upper Saddle River, New Jersey, 2002.

Pearl, J., (1986) "Fusion, Propagation, and Structuring in Belief Networks," Artificial Intelligence, 29, pp. 241-88.

Ripley, B. D. (1996), Pattern Recognition and Neural Networks. Cambridge: Cambridge University Press.

Russell, W.A. Jr., G.A. Luz, *et al.*, (2001), "Environmental Noise Management: An Orientation Handbook for Army Facilities," U.S. Army Center for Health Promotion and Preventive Medicine (USACHPPM).

Sachs, D., J. Benson, and P. Shcomer (1990), "CERL Noise Monitoring and Warning System 98." CERL Technical Report 99/99, December 1999.

Scharf, L.L., (1991) "Statistical Signal Processing, Addison-Wesley Publishing Co., Reading MA.

Schomer, P., et al., (1994) "Human and Community Response to Military Sounds: Results from Field-Laboratory Tests of Small-arms, tracked-vehicle, and Blast Sounds," NCEJ, 42(2), pp. 71-84, Mar-Apr.

Schomer, P., M. Bandy, J. Lamb, and H. Van Slooten (2000) "Using fuzzy logic to validate blast noise monitor data." *Noise Control Engineering Journal*, **48**(6), November-December 2000, p. 193-205.

Shapiro, S.C., ed., (1987), "Encyclopedia of Artificial Intelligence," John Wiley and Sons, NY, NY

Stark, H. and J. Woods (2002), Probability and Random Processes with Application to Signal Processing Prentice Hall, Upper Saddle River, New Jersey, 2002.

Vapnik, V. (1992), "Principles of risk minimization for learning theory." *Advances in Neural Information Processing Systems*, Vol. 4, p. 831-838, 1992.

Vipperman, J.S. and B. Bucci, (2005a) "Development of a Real-Time Military Noise Monitor," SERDP/ESTCP Partners in Environmental Technology Symposium and Workshop, Washington, DC, November 28-30,.

Vipperman, J. S., (2005b) "Development of Metrics to Identify Military Impulse Noise," 149th Meeting of the Acoustical Soc. of Am., Vancouver, BC Canada, May 16-20, also JASA 117, p. 2448.

Vipperman, J.S. and B. Bucci, (2006), "Algorithm Development for a Real-Time Military Noise Monitor," Final Report, SERDP SI-1436.

Vipperman, J. and B. Bucci (2006), SERDP (WP-1436). "Development of Metrics for Identifying Military Impulse Noise Sources," http://www.serdp-estcp.org/content/download/6561/86539/file/SI-1436-FR.pdf.

Widrow, B. J, and S. D. Stearns, (1985), *Adaptive Signal Processing*, Prentice--Hall, Englewood Cliffs, NJ 07632

Appendix A. Publications

Peer Reviewed Journal Articles

Brian A. B., J. S. Vipperman, (2007) "Development of an artificial neural network-based classifier to identify military noise," *J of Acoust. Soc. of Am.*, **122**(3), pp. 1602-10, Sep. 2007.

Technical Reports

Bucci, B. A, (2007) "Development of Artificial Neural Network-Based Classifiers to Identify Military Impulse Noise," MS Thesis, University of Pittsburgh, December 2007.

Vipperman, J.S. and B. Bucci, (2006), "Algorithm Development for a Real-Time Military Noise Monitor," Final Report, SERDP SI-1436.

Conference Papers

Bucci, B.A. and J.S. Vipperman, (2006) "Artificial neural network military impulse noise classifier." IMECE2006-14065, Proceedings of IMECE 06: 2006 ASME International Mechanical Engineering Congress, November 5-10, 2006, Chicago, IL.

Bucci, B. A. and J.S. Vipperman, (2007), "Bayesian Military Impulse Noise Classifier," IMECE2007-41700, ASME IMECE-07 Conference, Nov 11-15, 2007, Seattle, WA.

Bucci, B. A. and J.S. Vipperman, (2008), "An Investigation of the Characteristics of a Bayesian Military Impulse Noise Classifier," Proceedings of NCAD2008, Paper NCAD2008-73046, Dearborn Michigan, July 28-30, 2008.

Rhudy, M., B.A. Bucci, J.S. Vipperman, B. Abraham, and J. Allanach, (2009), "Microphone Array Analysis Methods using Cross Correlations," ASME IMECE 2009 Conference, Lake Buena Vista, FL, Nov 13-19, 2009.

Published Technical Abstracts

Vipperman, J.S., B.A. Bucci, M. Rhudy, (2008), "Characterization of a Bayesian Classifier to Identify Military Impulse Noise, Partners in Environmental Technology Technical Symposium and Workshop, Washington DC, Dec 2-4, 2008.

Vipperman, J.S. and B. A. Bucci, (2008), "An image processing based neural network method of waveform classification," 156th Meeting of the Acoustical Society of America, Miami, FL, Nov 10-14, 2008 (abstract published in JASA, **124**, p. 2597, 2008).

Bucci, B. A., J.S. Vipperman, (2007) “Bayesian Classifiers to Identify Military Impulse Noise,” Partners in Environmental Technology Technical Symposium and Workshop, Washington DC, Dec 4-6, 2007.

Bucci, B. A., J.S. Vipperman, (2007), “Artificial Neural Network Classifiers to Identify Military Impulse Noise,” Partners in Environmental Technology Technical Symposium and Workshop, Washington DC, Dec 4-6, 2007.

Bucci, B.A. and Jeffrey S. Vipperman, (2007) “Comparison of artificial neural network structures to identify military impulse noise,” *J. Acoust. Soc. Am*., **121**, 3112 (2007)

Bucci, B. A. and J.S. Vipperman, (2006a) “Development of Artificial Neural Network Classifier to Identify Military Impulse Noise,” 151st meeting of Acoustical Soc. of Am., Providence, RI, June 5-9, 2006, *J. Acoust. Soc. Am*. **119**, 3384.

Vipperman, J. S., (2005b) “Development of Metrics to Identify Military Impulse Noise,” 149th Meeting of the Acoustical Soc. of Am., Vancouver, BC Canada, May 16-20, 2005, J. Acoust. Soc. Am., **117**, p. 2448.

Appendix B. Accuracy of Classifier Based on DFR Waveforms.

A study was conducted on data that had been decimated, filtered, and re-sampled (DFR) to match the hardware specifications of existing hardware. First, the data were passed through a 4th order low-pass Butterworth filter with cutoff frequency of 400 Hz. Then the data were re-sampled at a frequency of 1 kHz (data were initially sampled at 10 kHz). Finally, the data were decimated to reduce the resolution from 16 bits to 12 bits. This new data was then used to train and test the ANN classifier. Decreasing fidelity was found to degrade the performance when no threshold was imposed on the measurements. However, at higher thresholds, the performance was comparable to the cases that used high-fidelity data (owing to a higher signal to noise ratio (SNR)).

Figures B-1 through B-5 represent the results of the DFR analysis for the ANN classifier (see section 5.3.3). Each figure represents a different trigger threshold, e.g., for a 100 dB trigger threshold, only waveforms with Lpk values greater than 100 dB were included. As the trigger threshold increases, the performance was found to improve significantly. Retraining the ANN can produce slightly different results. For this network, imposing no threshold (Figure B-1) yields poor results due to the fact that the signal to noise ratio will be high for waveforms with small Lpk values.

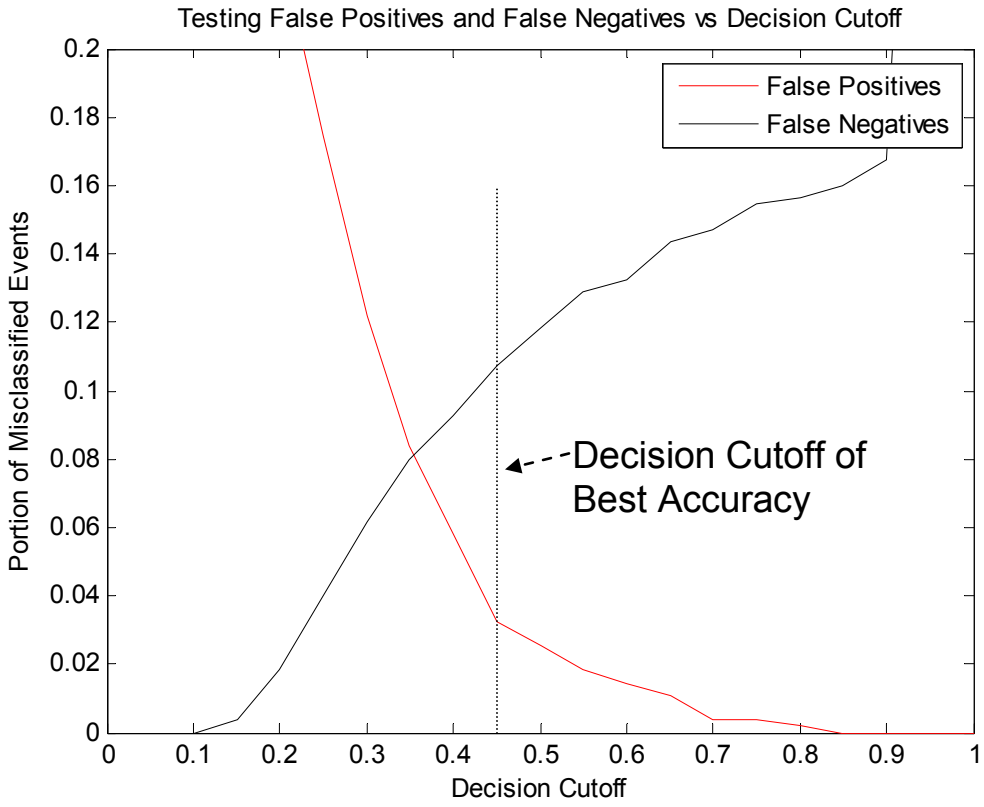

Figure B - 1 ANN Classifier Results for DFR data (no trigger threshold).

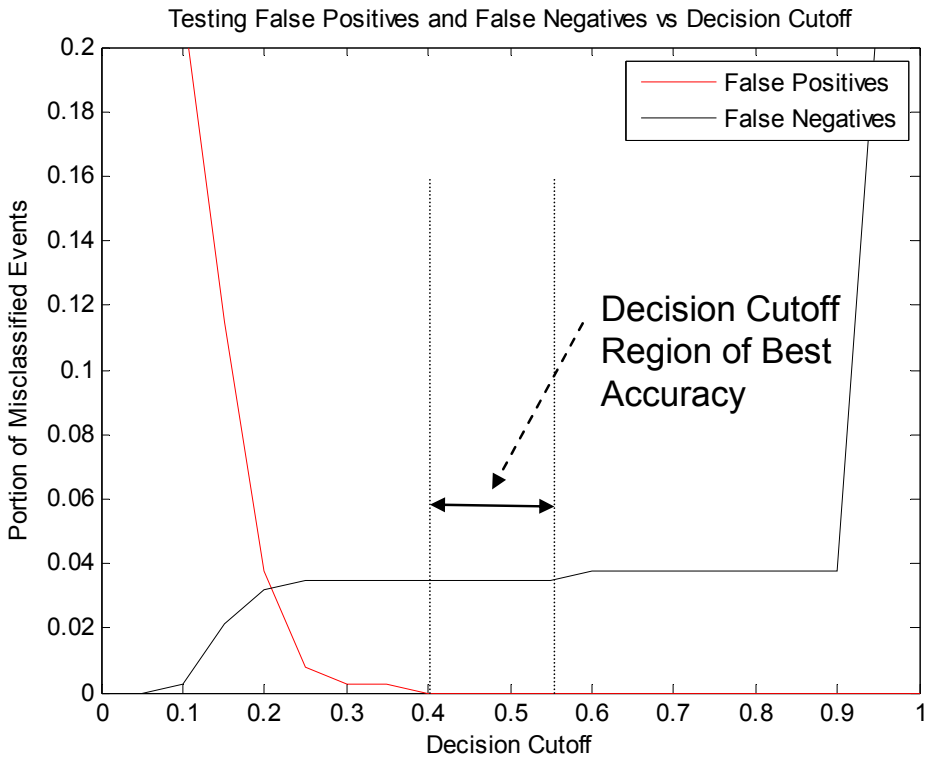

Figure B - 2 ANN Classifier Results for DFR data (100 dB trigger threshold).

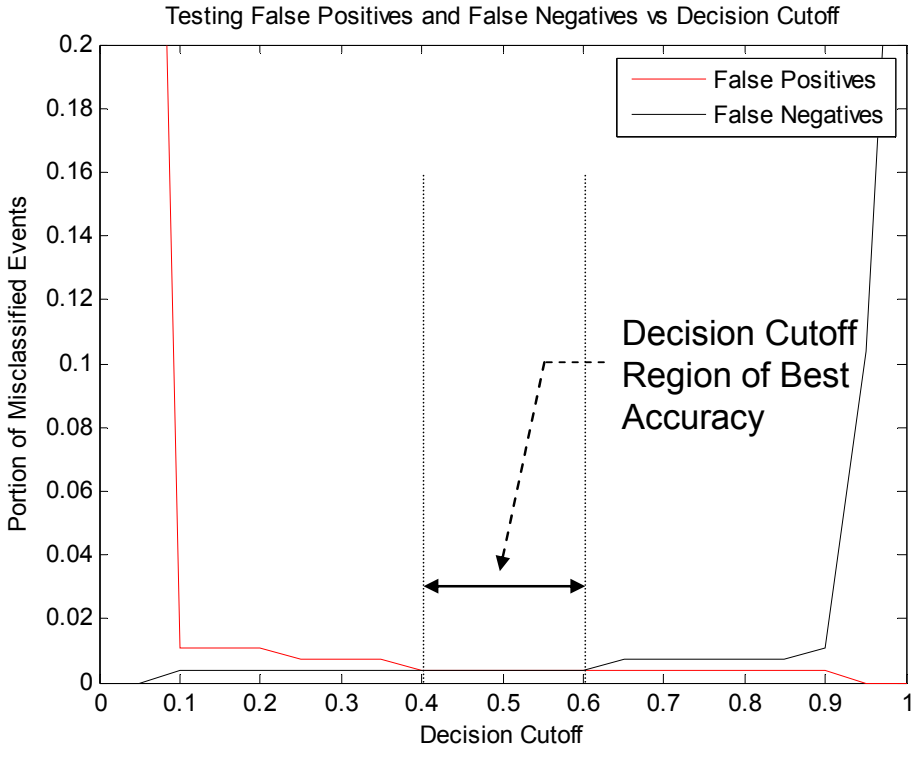

Figure B - 3 ANN Classifier Results for DFR data (105 dB trigger threshold).

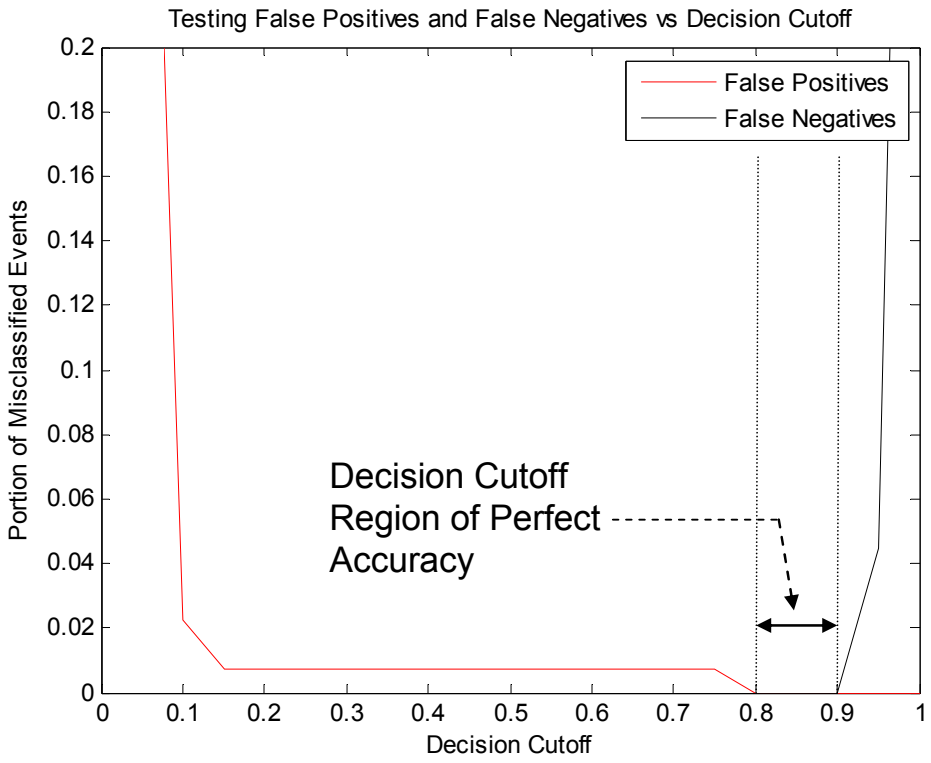

Figure B - 4 ANN Classifier Results for DFR data (110 dB trigger threshold).

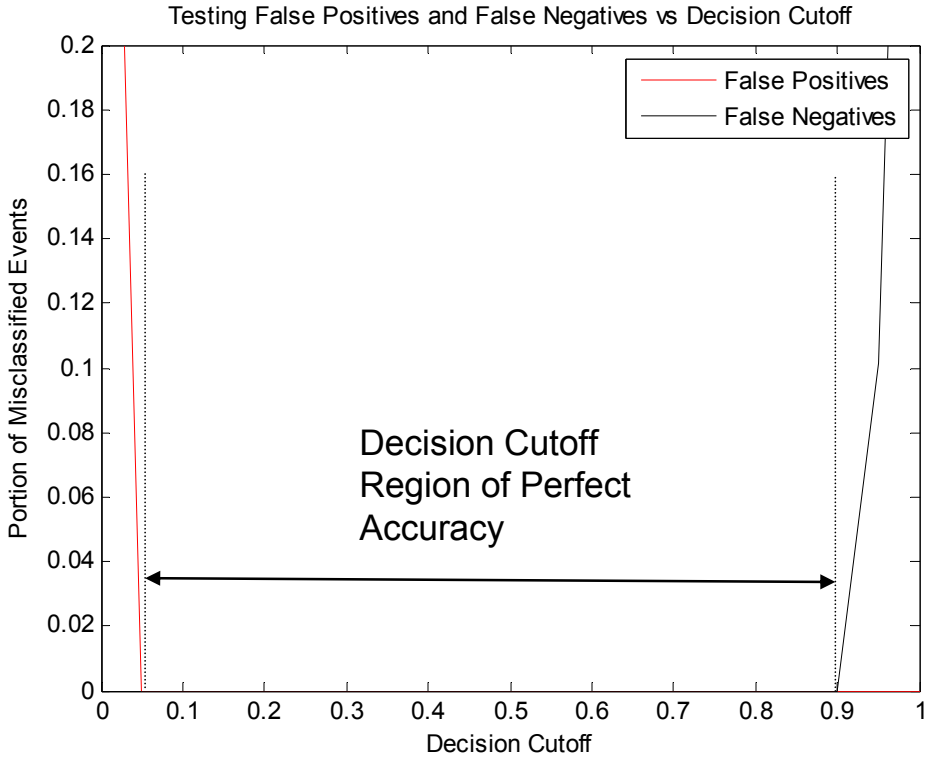

Figure B - 5 ANN Classifier Results for DFR data (115 dB trigger threshold).