

Engine Noise Signature for Boat Identification

A Major Qualifying Project

Submitted to the Faculty
Of the

WORCESTER POLYTECHNIC INSTITUTE

In Partial Fulfillment of the requirements for the

Degree of Bachelor of Science

By

James Curry

Jonathan Ross

Date: April 26, 2007

Approved:

Professor Peder Pedersen, Primary Advisor

Professor Fabio Career, Co-Advisor

Abstract

This project, conducted within the framework of the European Union's *Civitas Mobilis*, aims to develop an instrument capable of gathering data on passing motor boat traffic in a Venetian canal. The instrument uses neural networks to analyze the underwater sound produced by the boat engine, and thereby identify boat type. It is hoped that this data can be applied by the city of Venice toward measures to reduce the impact of "*moto ondosso*," or motor boat wakes, on city infrastructure.

Table of Contents

Abstract.....	2
Table of Contents.....	3
Table of Figures.....	5
Authorship.....	6
1 Introduction.....	7
2 Background.....	9
2.1 Boat Wake Damage.....	10
2.2 Project Context.....	11
2.3 EU Civitas Mobilis Project.....	11
3 Systems Design and Requirements.....	13
3.1 Feature Extraction: Requirements.....	13
3.2 Recognition Algorithm: Requirements.....	13
4 Technical Literature Review.....	14
4.1 Sonar signals.....	14
4.2 A new Technique for Diagnosing Piston Engine Aircraft.....	16
4.3 Sound and vibration measurement.....	18
4.4 Conclusion of literature review.....	20
5 Overview of Signal Analysis Techniques.....	21
5.1 Fourier Transform.....	21
5.2 Time-Frequency Analysis.....	22
5.2.1 Gaussian Windowing.....	22
5.3 Energy Normalization.....	25
5.4 Feature Identification Using Spectrograms.....	25
5.5 Techniques of Feature Extraction.....	27
5.5.1 Peak Detection.....	27
5.5.2 Energy Analysis.....	28
5.5.3 Smoothing.....	29
6 Overview of Neural Networks.....	30
6.1 Neural Network Theory.....	30
6.2 Neural Networks in Practice.....	31
7 Methodology.....	35
7.1 Selecting a Hydrophone.....	35
7.2 Acquiring Recordings.....	38
7.3 Feature Identification.....	39
7.4 Feature Extraction.....	41
7.5 Neural Network Development.....	42
7.5.1 Network Properties.....	43
7.5.2 Training Using Non-Boat Data.....	44
7.5.3 Format for the Training and Test Data.....	46
8 Testing & Results.....	47
8.1 Post Screening.....	47
8.2 Generalization.....	47
8.3 Testing Phase 1.....	49

8.4	Testing Phase 2	50
8.5	Improving Accuracy	52
9	Conclusion	54
10	Appendix A: Student Recording Manual.....	55
10.1	Initial Setup.....	56
10.2	Recording a Sound.....	58
10.3	Final Steps.....	60
11	Appendix B: MATLAB Code.....	61
11.1	Column Generator (columnngen.m).....	61
11.2	Frequency Domain Analysis (fdanalysis11.m).....	62
11.3	Identification Function (identify.m).....	64
11.4	Mass Identification Function (massid.m)	65
	Works Cited	68

Table of Figures

Figure 1: Current and Past Projects	11
Figure 2: Top level block diagram.....	13
Figure 3: Sonar block diagram (Mill, 2005)	15
Figure 4: Four Stroke Cycle (Thio, 2001)	16
Figure 5: Filtering for the aircraft algorithm (Thio, 2001)	17
Figure 6: Vibration measurement algorithm (Bruel and kjaer, 1998)	18
Figure 7: A Gaussian Function	23
Figure 8: Original Signal	24
Figure 9: Gaussian Windows Applied	24
Figure 10: Fast Fourier Transform Applied.....	24
Figure 11: Spectrogram, Baby Crying.....	26
Figure 12: Spectrogram, Harley Davidson	26
Figure 13: Peak detection Example	28
Figure 14: Frequency bands used in energy analysis.....	28
Figure 15: Smoothing applied.....	29
Figure 16: Input Data	31
Figure 17: Target Data Set.....	32
Figure 18: Test Data.....	33
Figure 19: Neural Network Test Results	33
Figure 20: Weights.....	34
Figure 21: Overall project block diagram.....	35
Figure 22: Hydrophone comparison chart	37
Figure 23: Recording setup used by Venice staff	39
Figure 24: Spectrogram, cargo boat.....	40
Figure 25: Spectrogram, personal boat	40
Figure 26: Spectrogram, taxi boat.....	40
Figure 27: Legend for Figures 28-30	44
Figure 28: "Baby Crying" Neural Networking Results	45
Figure 29: "Harley" Neural Networking Results	45
Figure 30: "Orchestra" Results	45
Figure 31: Format of the training and test data.....	46
Figure 32: An example of overfitting	48
Figure 33: Early stopping.....	49
Figure 34: Phase 1 test data	50
Figure 35: Phase 2 test data	51
Figure 36: Phase 2 overall results	52
Figure 37: Downloading the GoldWave software	56
Figure 38: Creating a New Sound File in GoldWave.....	57
Figure 39: GoldWave, Ready to Record.....	58
Figure 40: A Recorded Sound Wave in GoldWave.....	59

Authorship

Abstract.....	Ross
1 Introduction.....	Ross
2 Background.....	Curry
2.1 Boat Wake Damage	Ross
2.2 Project Context.....	Ross
2.3 EU Civitas Mobilis Project	Ross
3 Systems Design and Requirements.....	Ross
3.1 Feature Extraction: Requirements.....	Ross
3.2 Recognition Algorithm: Requirements.....	Ross
4 Literature Review.....	Curry
4.1 Sonar signals	Curry
4.2 Speech Recognition	Curry
4.3 A new Technique for Diagnosing Piston Engine Aircraft	Curry
4.4 Sound and vibration measurement.....	Curry
4.5 Conclusion of literature review.....	Curry
5 Overview of Signal Analysis Techniques.....	Ross
5.1 Fourier Transform.....	Ross
5.2 Time-Frequency Analysis.....	Ross
5.3 Energy Normalization.....	Ross
5.4 Feature Identification Using Spectrograms	Ross
5.5 Techniques of Feature Extraction.....	Ross
5.5.1 Peak Detection	Ross
5.5.2 Energy Analysis.....	Ross
5.5.3 Smoothing.....	Ross
6 Overview of Neural Networks.....	Curry
6.1 Neural Network Theory	Curry
6.2 Neural Networks in Practice	Curry
7 Methodology.....	Ross
7.1 Selecting a Hydrophone	Curry
7.2 Acquiring Recordings	Ross
7.3 Feature Identification	Ross
7.4 Feature Extraction.....	Ross
7.5 Neural Network Development	Curry
7.5.1 Network Properties	Curry
7.5.2 Training Using Non-Boat Data.....	Curry
7.5.3 Format for the Training and Test Data	Ross
8 Testing & Results.....	Ross
8.1 Post Screening.....	Ross
8.2 Generalization.....	Ross
8.3 Testing Phase 1	Ross
8.4 Testing Phase 2	Ross
8.5 Improving Accuracy	Ross
9 Conclusion	Both
10 Appendix A: Student Recording Manual.....	Ross

1 Introduction

One of the great challenges mankind faces in the 21st century is to find means of transportation to meet the needs of the present without jeopardizing the ability of future generations to meet those same needs. The concept of this so-called sustainable transportation has received an increased amount of attention in the past two decades, as a reaction to problems with transportation policy which rendered many modes of transportation no longer sustainable (Wikipedia, 2006.) Good planning can result in sustainable transportation systems that meet the needs of the people, while saving resources in the long term.

This problem is of an especially serious nature in Venice, Italy. Venice presents a special challenge in that the vast majority of its transportation, both for humans and cargo, is by boat, along Venice's many canals. This is a difficult situation for city planners, because many of the considerations for sustaining the city's transportation infrastructure differ greatly from more common systems, such as roads. Many factors threaten the integrity of the canal system, including rising water levels over time and the degradation of the canal walls themselves over time. This damage is quite costly to the city to repair, and substantially raise the cost of living in Venice as these expenses are passed down to the taxpayer (Chiu et al., 2002).

One of the foremost contributors to canal wall damage are motor boat wakes, or "*moto ondosso*," which, over time, have a deleterious effect on city structures such as streets, bridges, and docks, as well as buildings and areas of artistic and architectural heritage (Carrera, 2005). Flooding and wet-dry cycles associated with tides can compound this problem.

In order to better understand and analyze the impact that these boats are having on some of the smaller canals in Venice, data on boat traffic is needed, specifically information on the distribution of the types of boats passing through the canals, the speed at which they travel, and the turbulence their wakes produce. *Mobilis* is a project within the European Union's *Civitas* initiative, which is aimed at providing clean solutions for sustainable urban mobility (Civitas 2006). *Mobilis* has already implemented several measures in Venice, in areas ranging from boat parking management to promotion of

bicycle use, however there is still work that needs to be done in order to properly quantify and assess boat wake damage.

The purpose of this project is to develop a device that can be placed in a canal and can read the acoustic signals produced by passing boats. This signal may then be analyzed to extract its important spectral features. These spectral features collectively form a distinct acoustic signature. At that point, a recognition algorithm in the form of a neural network can identify the boat's specific type based on this signature. The boat type data accumulated by this process over time may help experts better frame a plan to protect the Venice canal system for generations to come.

2 Background

Despite being a city afloat in the middle of a bay, Venice still has many of the same major logistical traffic problems as every other city around the world. Being a city hundreds of years old it was not built around modern technology and has had to readjust as new inventions have been introduced. Boats used to be exclusively paddle or pike driven, but modern technology took over and motorboats became the rule, rather than the exception. (Chiu et. al, 2002)

There are three major types of boats that traverse the canals of Venice. One type is the cargo boat, which carries the supplies that the city needs to survive, another is the taxi boat, which carries the population itself throughout the city, and the last are the personal motor boats used by the general population of Venice (Chiu et. al, 2002). There are several other types of boats that populate the canals, however, those boats can usually be safely grouped into one of the above two categories, or represent an insignificant percent of the total daily traffic, and thus, can be ignored.

The cargo boat is any boat that is used for a commercial or industrial purpose. These can include construction, garbage collection and disposal, as well as food and supplies for the entire city. These boats tend to be much heavier in order to carry the increased load, however are not necessarily that much bigger than taxi boats due to the size restrictions in some of the canals. (Duffy et. al, 2003)

The taxi boat category encompasses all boats that carry passengers as a major means of their usage. These boats are usually much lighter than the cargo boats, however since the owner makes money based upon how many passengers they can transport, they often travel faster. Because of Venice being such a major tourist attraction there has been an increased demand for the taxi/public boats over the year resulting in them representing a much bigger faction than either the private boats and cargo. (Duffy et. al, 2003)

The final category which will be discussed is the private boat. Because the taxi boat and cargo boat captains generally have to report to a company or are representative of some franchise, they are will not be the ones speeding in the canals. If the captains of such boats were caught speeding they may suffer repercussions besides just a ticket, such

as being suspended or even losing their jobs. However this does not stop the private boat captains. Private boats generally have outboard motors, while taxis and Cargo will have inboard or inboard-outboard. Taxis represent 46% of the total boat traffic in the canals, cargo represents 36, and private boats represent about 18% (Chiu et. al, 2002).

2.1 Boat Wake Damage

The effects of water damage upon structures in Venice have been chronicled ever since records have been kept (Carrera, 2005). Indeed, Venetians are reminded constantly of the damage to their city's infrastructure, with related articles appearing in local newspapers multiple times per week (Chiu et al. 2002). Recently, several attempts have been made to analyze this damage scientifically and analyze its root causes. With regard to canal walls specifically, a prior WPI project determined that the root cause of this damage to be high sediment levels in the canals, a problem which is exacerbated by high boat traffic (Babic et al., 1998).

One recent effort to study the effects of boat wakes on stones in Venice included a 2002 project conducted by a group of WPI students. They designed and tested a device that could map hot spots for wake turbulence, and thereby predict the most likely places for future damage (Chiu et al., 2004). Knowledge such as this is very useful in order to address the problems caused by *moto ondosso*.

One such application of city knowledge was put forth in a project entitled *Re-Engineering the City of Venice's Cargo System* (Duffy et al., 2003). This award winning IQP contained a plan for revolutionizing the distribution of merchandise throughout the city so that boat drivers organized their trips by destination, rather than by product. This approach would greatly increase efficiency and reduce the total amount of boat traffic accordingly. Of course this would also have the beneficial side effect of a large reduction of *moto ondosso*. This plan is currently in the implementation phase. Some estimates predict that a 90% reduction in wall damage is possible (Carrera, 2005).

2.2 Project Context

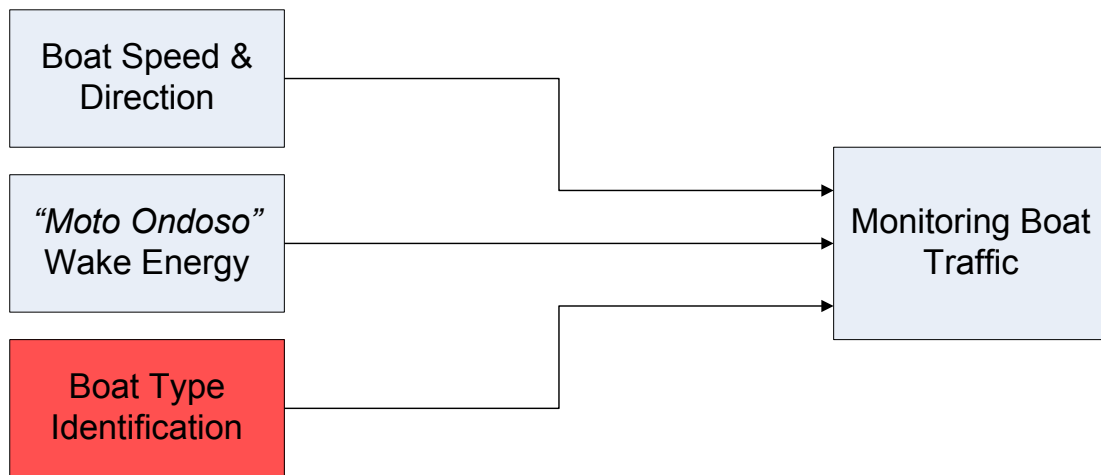


Figure 1: Current and Past Projects

This project is one of several WPI Projects over the past several years relating to information gathering to aid the City of Venice. These projects have approached the problem in multiple different ways, including ultrasound monitoring of boat speed and direction and wake energy measurement. This project’s angle is to gather data on passing boat types so that the City of Venice can better target its measures to reduce the effects of *moto ondosso*.

2.3 EU Civitas Mobilis Project

This project was created within the framework of the European Union’s project *Civitas Mobilis*. The Civitas Initiative is a creation of the European Council (EC). Civitas is a loose acronym which stands for City-VITALity-Sustainability. Its aim as stated on its website is: “to generate a decisive breakthrough by supporting and evaluating the implementation of ambitious integrated sustainable urban transport strategies that should make a real difference for the welfare of the European citizen” (Civitas, 2006). Mobilis is a partnership within Civitas between the cities of Toulouse (France), Debrecen (Hungary), Ljubljana (Slovenia), Odense (Denmark), and Venice (Italy) to address common sustainable transportation issues.

Mobilis is focused upon eight so-called “technical themes.”

- Clean Vehicles and Alternative Fuels
- Access Management
- Integrated Pricing Strategies
- Stimulation of Collective Transport Modes
- New Forms of Vehicle Use and Ownership
- New Concepts for the Distribution of Goods
- Innovative “Soft Measures” for Managing Mobility Demand
- Telematics

Venice is currently implementing Mobilis measures in each of the above themes (Civitas, 2005). These measures range from the introduction of LPG as a fuel for boats, to changes in parking management strategies, to promotion of bicycle use.

An increase in knowledge about current traffic models could be invaluable to the city as they implement some of these measures, in particular those relating to urban logistics. The plan proposed by Duffy et al. is just one example of a possible solution that could utilize this data.

3 Systems Design and Requirements

This project aims to recognize boats based on their unique sound signature. First, the analog sound signal must be digitized and preprocessed. Next, the digital sound clip must be analyzed and its identifying features identified. These features must then be used by a recognition algorithm to identify the boat. Figure 2 below is a block diagram for this process.

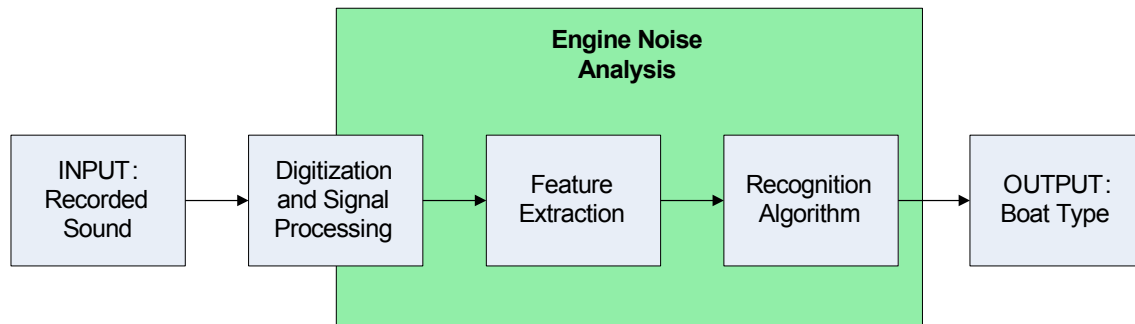


Figure 2: Top level block diagram

3.1 Feature Extraction: Requirements

The first major block to consider is feature extraction, which is defined as the process of finding and extracting the most useful identifying information from a given sound signal. The input to this block is a short digital sound clip of the boat at its closest point to the hydrophone. This block must analyze the signal and pull out the relevant identifying information. Its output must be a one-dimensional column vector because this is the format that can be used by a neural network.

Feature extraction in this project was accomplished primarily through the use of MATLAB code.

3.2 Recognition Algorithm: Requirements

The next major block is the recognition algorithm. In this project, the recognition function was performed by neural networks. This block takes as input the column vector from the feature extraction block, and outputs a boat determination. This process is accomplished by simulating a pre-trained neural network (e.g. “Cargo” or “Taxi”). The goal is to identify each passing boat with the best possible accuracy.

4 Technical Literature Review

Several articles were reviewed for content relevant to the project. It was decided that background research was needed to search for prior content that had already been done to see what strategies for sound recognition were already in use. With the knowledge of the articles reviewed below in mind, a plan of attack for boat identification could be made. Key features discovered were the ability to detect tonals transients and noise, feature enhancement and extraction, effective windowing techniques, and specific engine noise feature analysis.

4.1 *Sonar signals*

Passive sonar signals are used in water to determine the pressure signature of boats and other marine objects. In the past it has been humans who have manually listened for certain unique sounds to determine what was in the water, and take actions based upon what they heard. With the rise of computers and sensitivity of noise monitoring devices, as well as advanced sonar arrays, it has become increasingly difficult for humans to pick up and identify all of the sounds that a computer can hear. This is because there are important features at low amplitudes and frequencies outside the range of the human ear. Sound recognition equipment will step in here and create a much more advanced system that can be superior to a human ear. For all practical purposes a boat's sound signature will be found in the range of frequencies between 0 and 2 kHz (Mill, Auditory-Based Time-Frequency Representations and feature Extraction Techniques for Sonar Processing). Tonal signatures (usually the periodic signal when a boat passes) are produced by a boats engine, and by bubbles forming behind the propeller, these noises are fairly continuous and appear over a large area of time. Transient features are caused by a person walking around the boat, or the boat colliding with a dock. These noises are easy to pick up, however they are not predictable because they could happen at any time.

To describe figure 3, first the signal must be broken into three parts: Tonals, Transients, and noise such that the three parts combined, equal the input signal. First the overlap-add analysis, which is when you take individual frames of the signal over time,

and overlap them slightly to allow for continuity, was employed such that a windowed analysis could be taken. This is similar to the windowing which has been discussed earlier. Then a FFT of each individual frame is taken of the signal. In this form, the signal was put through a peak selecting algorithm to determine tonals and non tonals in the signal. The discrete Cosine Transform (real half of the Fourier transform) was then applied to the signal to separate the noise from the transients, that is, spikes in the time dimension are converted to cosines in discrete time. The Fourier transform was taken out of the signal again resulting in the cosines showing up as peaks which could be selected out as the transients.

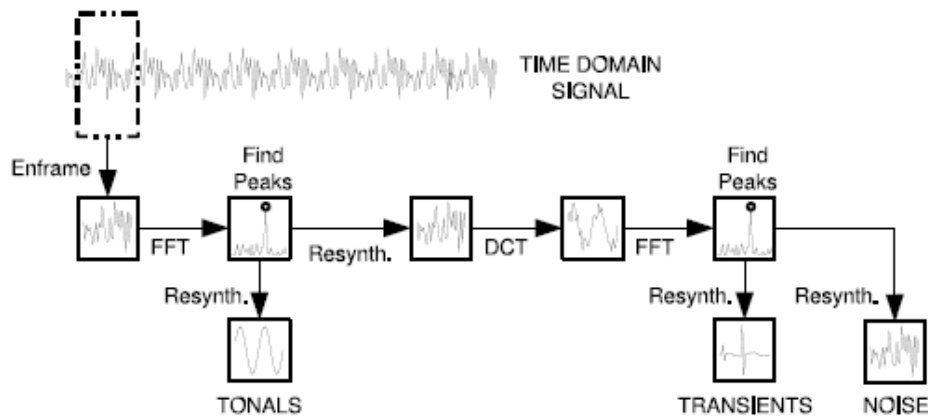


Figure 3: Sonar block diagram (Mill, 2005)

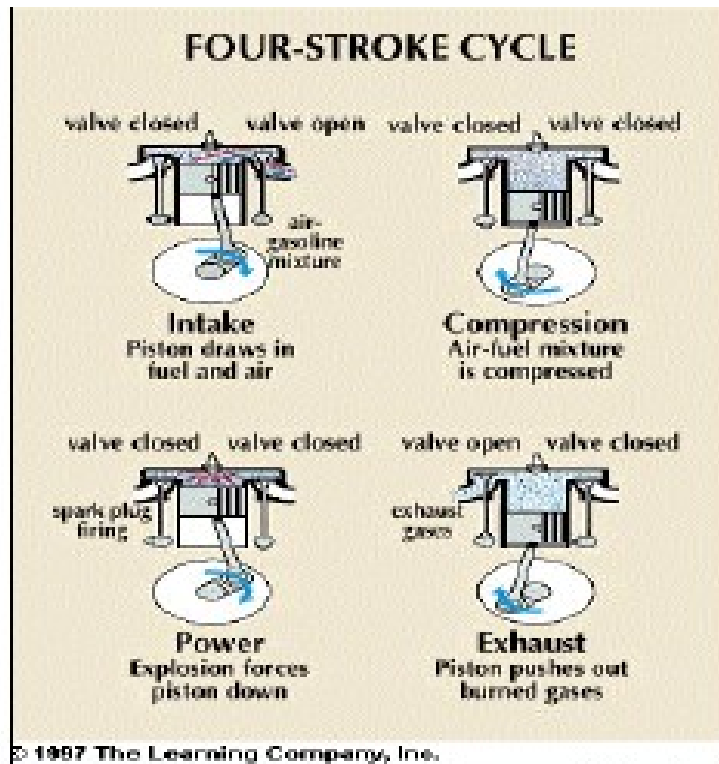
Peak detecting can be done by taking the energy in one window of the frequency analysis and comparing it to the energy in the surrounding windows.

$$P[k] = 20 \log(X[k]) - 20 \log\left(\frac{1}{c} \sum_{j \in M} X[k + j]\right)$$

$P[k]$ is a set threshold in decibels. This threshold is determined arbitrarily by the accuracy of the readings, (around 3-5 decibels). Peak detecting looks for energy spikes in a signal and creates a wave that rides along the top.

4.2 A new Technique for Diagnosing Piston Engine Aircraft

This project experimented with the auditory sensing for abnormal vibrations in 4 piston aircraft engines. Aircraft engines are similar to boat engines in that they both consist of multiple cylinders and use the same principle of internal combustion. It was assumed that because of this, there would be similarities in both of their sound structures and in the way they could be analyzed. First the four stroke piston cycle was observed and recorded (see figure 4).



*Intake Valve on left, exhaust valve on right in diagram.

Figure 4: Four Stroke Cycle (Thio, 2001)

To begin; the sound was put through 3 different filters before it was analyzed. The first was the Reduction Envelope filter. This reduced the noise level across all spectra of frequency and allowed for an easier signal to analyze.

The next filter applied was a low-pass filter, with a cutoff frequency around 3k - 3.5 kHz, which cut of all of the high frequencies that were producing a lot of noise. Much of this noise appeared to be coming from air cooling as it passed the microphone on the plane. It is unclear as to whether or not this filter will be necessary for the

purposes of this project do to the fact that the recordings will be underwater, and the microphone will be stationary with respect to the surrounding objects.

The last filter was a band pass filter with a static range around 1 to 4 kHz. It was unclear why a low pass and a band pass filter were needed because they seemed to both cut off the same high frequencies.

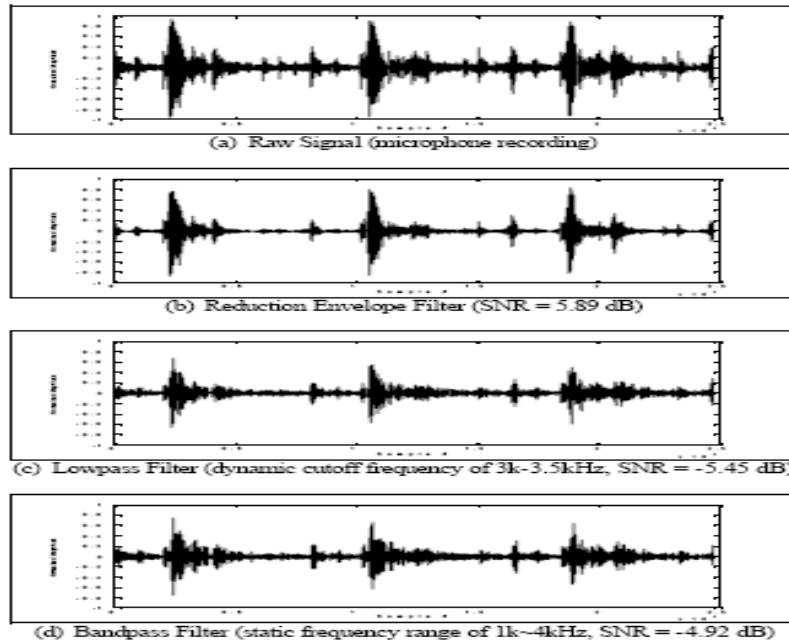


Figure 5: Filtering for the aircraft algorithm (Thio, 2001)

The signal was then enhanced to increase the signal to noise ratio. This was done by averaging the signal with itself shifted it in time by one sampling point. This created peaks that were much more distinctive than before. By doing this around 15-20 times a signal that was very distinctive was created.

A period for the engine cycle was determined by setting a threshold for significant peaks and ignoring all other sound, then taking those peaks and setting the period for every 4 peaks, starting with the highest of the first 5. This established period can then be used for the frequency spectrum analysis with confidence that all significant patterns are retained.

After the FFT of the signal was taken, all amplitudes were normalized to take into account any dampening which may have occurred during recording. This was an important step, and is something that must also be done when recording boats to try and

take into account the fact that engines of some boats will be louder than others. In order to find any significant frequency patterns the frequency spectrum had to be compared to the time domain to observe specific frequencies at given peaks. The project looked at the four different engine pistons individually and observed any major differences between them. Looking at the 4 different recording of the 4 pistons (see figure 5) it can be seen that the first signal seems to have more “noise” in it than the other signals. For the purposes of boat identification this process is unavailable, and what the overall picture was has to be looked at. The process of an engine stroke involves the piston moving more quickly after the mixture is ignited resulting in higher frequencies, and then slower as it comes around during the venting stage, resulting in lower frequencies, observing these changes during a single engine stroke may be the key to finding key features in a signal.

Although the above techniques are specifically for detecting a broken, or faulting engine, the method used still provides insight on feature extracting from engines in general, and those techniques learned can be applied to this project.

4.3 Sound and vibration measurement

The method of signal analysis and feature extraction has 5 major steps in the chain. These steps will be discussed and analyzed individually. It is assumed that the transducer has already been fed a sound signal for analysis. At the end of the last step there could be an algorithm and a database blocks, however the purpose of this section is to focus on the analysis, and what enters the output stage.

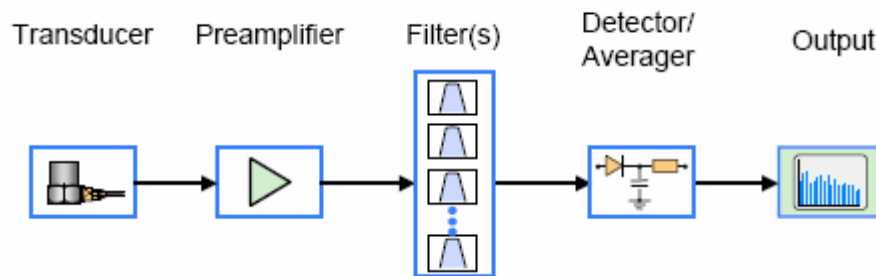


Figure 6: Vibration measurement algorithm (Bruel and kjaer, 1998)

The first step is the transducer, which would be the signal input stage. The purpose of a transducer is to convert energy from one form to another. In this case it takes in sound energy, and converts it to electrical current.

The next step is the preamplifier which pulls the voltage of the signal up to line level while leaving the current the same. The third step of the process is the filters. The purpose of the filter system is to eliminate any extraneous noise present. The frequencies which are filtered out must be determined before the construction of the signal. This means that some experimentation has to be applied beforehand to determine which frequencies contain relevant information and which are background noise.

The detector, also known as the averager, looks for a set peak at the desired frequencies and then creates a more linear output from the changing peaks. This can be done by taking an energy-envelope of the signal and sending it to the output.

Detecting extraneous vibrations from a motor can change depending on what device is being studied. For example if a fan is malfunctioning, the change of vibration occurs at the same frequency of the spinning motor due to the nature of an imbalanced piece of equipment vibrating, and resonance.

Conversely if we look at a gearbox or other machine that has various moving mechanical parts, a clashing gears or other mechanical failure will result in a possible vibration at lower or higher frequencies than the engine resonance, in fact high or low pitch hums coming from machinery can often be detected by ear, and determined to be a problem with no analysis.

The passing of boats in canals may create similar scenarios with mechanical parts rubbing together, specific observations can be looked for if one type of boat is more prone to have a specific fault, then that fault can be detected and used as a parameter in the identification algorithm.

The lesson learned from this is that the fault in the fan can be found by looking at the amplitude of the signal while the machine requires a spectral analysis to determine if there is a fault. In relevance to this project, the importance of using both a time domain, and frequency domain signal is clear. An engine signature has more than just a unique frequency spectrum. It also has a unique time domain. By using time domain analysis a more accurate detection system can be created.

4.4 Conclusion of literature review

These 4 projects have provided insight to many different strategies for boat analysis. None of the other project directly work with boat signature analysis, nor can any one of them be used in its entirety for such, however taking key pieces, as well as lessons learned from the above, an overall plan has been obtained.

5 Overview of Signal Analysis Techniques

The purpose of the signal analysis in this project is primarily to create and implement an algorithm by which a microprocessor can take an input sound signal and manipulate it to determine the type of boat that produced that signal.

Although some features of the input signals may be determined through analysis of the signal in the time domain, it will also be necessary to analyze the signal in the frequency domain in order to see how the frequency spectrums of the engine noises differ. Since the raw recorded sound signals will be in the time domain, several methods of converting such a signal to frequency domain must be considered to determine which best suits the project.

5.1 Fourier Transform

One basic tool that can be used to analyze a signal is the Fourier transform. Using the Fourier transform it is possible to convert a signal that is described in the time domain into a function in the frequency domain. This is defined by:

$$f(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt$$

Of course, this integral applies to continuous time signals only. Since this project is concerned with signals that will be recorded digitally and analyzed in discrete time, the discrete Fourier transform, or DFT, would be more appropriate for consideration:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \quad k = 0, \dots, N - 1$$

One way to cut down on the amount of computations required to generate this sequence is to use the FFT algorithm (Fast Fourier Transform.) For a signal with N_0 samples, the FFT divides the sequence into two separate sequences with length $(N_0/2)$ consisting of the even- and odd-numbered samples, respectively. The DFTs of these signals can then be recombined into a N_0 point DFT (Lathi, 1998). Implementations of this algorithm are readily available for MATLAB.

5.2 Time-Frequency Analysis

When analyzing a complex periodic signal such as a motor boat engine noise, it is clear that a traditional discrete Fourier transform is inadequate. While a Fourier transform would provide an intuitive spectrum for a sine wave of infinite length, when provided with a signal that is not of infinite length the Fourier spectrum yields instead an imprecise band of frequencies around the intuitive fundamental frequency (Fulop and Fitz, 2006). This so called “smearing” might make it quite a bit more difficult for a computer algorithm to differentiate between two engine types.

It is much more useful to determine the “instantaneous frequencies” of the signal at different points in time. Such an analysis can provide an intuitive understanding of the dynamics of the sound wave that is completely independent of the length of time that the signal persists (Fulop and Fitz, 2006).

This model can be said to be more intuitive because it seems to be the same way that the human ear perceives sound. For example, a human being can tell exactly what frequency a sine wave is playing at whether it plays for 1 second or 4 seconds, while a Fourier Transform does not have that sort of flexibility (Fulop and Fitz, 2006). In much the same way, it is expected that an instantaneous time-frequency analysis can better differentiate between two finite samples of boat motor noise that the human ear can easily tell apart.

5.2.1 Gaussian Windowing

The original signal must be broken up into partitions for the purpose of instantaneous frequency analysis. One’s first impulse when considering such an algorithm might be to simply multiply by a rect function. Such a window would zero out the signal for all but the desired window of time, and leave the signal intact for that window.

The problem with this windowing method becomes apparent when considering the behavior of the rect function under a DFT. The frequency spectrum of this function is a sinc, which introduces artifacts in the spectrum. These may, in turn, reduce the accuracy of the analysis.

A better alternative is a Gaussian window. A Gaussian function is a function of the form $f(x) = ae^{-(x-b)^2/c^2}$ for some real constants $a > 0$, b , and c . If c^2 is chosen to equal 2, the resulting function is an eigenfunction of the Fourier transform. What this means is that after a Fourier transform on a Gaussian function, the result is another Gaussian function that is a scalar multiple of the first one (Wikipedia, 2006).

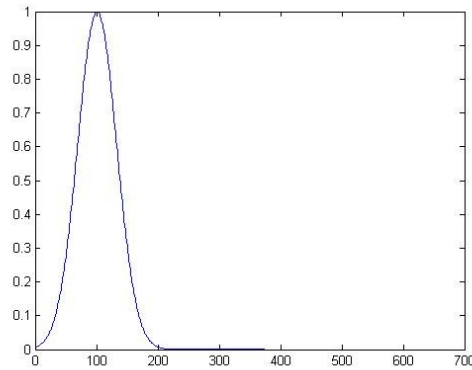


Figure 7: A Gaussian Function

This procedure may be illustrated with an example. The simple input function shown in Figure 8 may be multiplied by a series of time shifted Gaussian functions. For the purpose of this example 3 windows will be used.

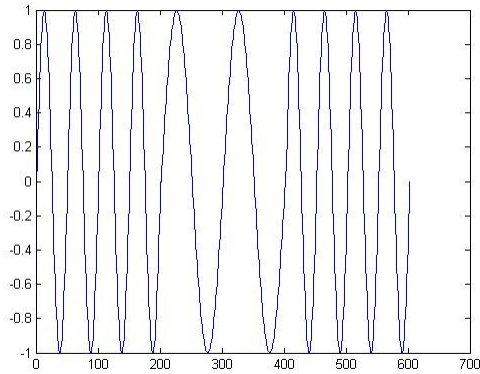


Figure 8: Original Signal

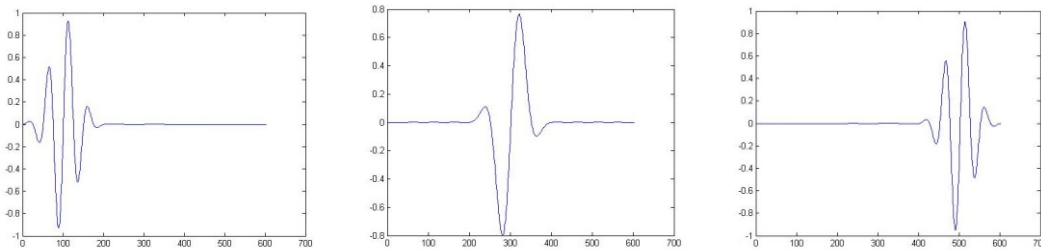


Figure 9: Gaussian Windows Applied

MATLAB's built in *FFT function* may then be used on the resulting windowed signals. The raw results of the FFT are in complex form, so it is necessary to find the magnitude of the points in the output vector. This provides a meaningful representation of the frequency spectrum at each respective point in time.

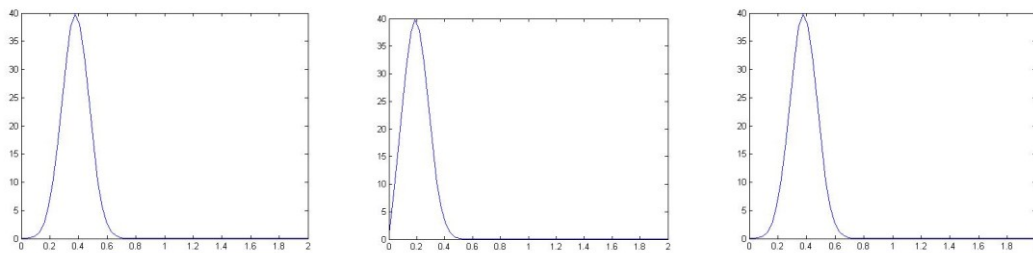


Figure 10: Fast Fourier Transform Applied

Using a series of Gaussian windows to partition the original acoustic signal should provide a much more accurate depiction of the frequency spectrum of a short segment of a signal.

5.3 Energy Normalization

For accurate boat identification, it is preferable to analyze recordings of similar amplitude. Since distance from the microphone is likely to vary between boats, it was deemed necessary to devise an energy normalization algorithm. Each boat's signal energy was calculated, compared to a benchmark, and adjusted accordingly prior to the feature extraction stage.

5.4 Feature Identification Using Spectrograms

The first step necessary in order to identify a thing is to describe it in such a way that a recognition algorithm can make the necessary differentiation. This "description" may take the form of a set of features. By finding the most distinctive features of a given subdivision, it becomes far easier to make that identification. "Feature identification" here is defined as the process of finding those features. For this project, the bulk of the feature identification was done by carefully studying the spectrograms of various types of boat noise.

A spectrogram is a graphical way of representing a signal in which the intensity of a signal is denoted in terms of both time and frequency. In the following spectrograms, the x axis represents time and the y axis represents frequency. The amplitude of the signal at a given point is denoted by a color scale, with blue representing the least intense and red representing the most intense.

These two example spectrograms represent a baby crying and a Harley Davidson revving up. The Harley spectrogram exhibits a low concentration over 1.5 kHz, while the baby cry has frequency components well over 4 kHz for most of the duration of the signal. In this example problem, these two very disparate sounds could be easily identified by simply extracting the energy over 4 kHz. For a description of frequency identification applied to actual boat noise, please see section 7.3.

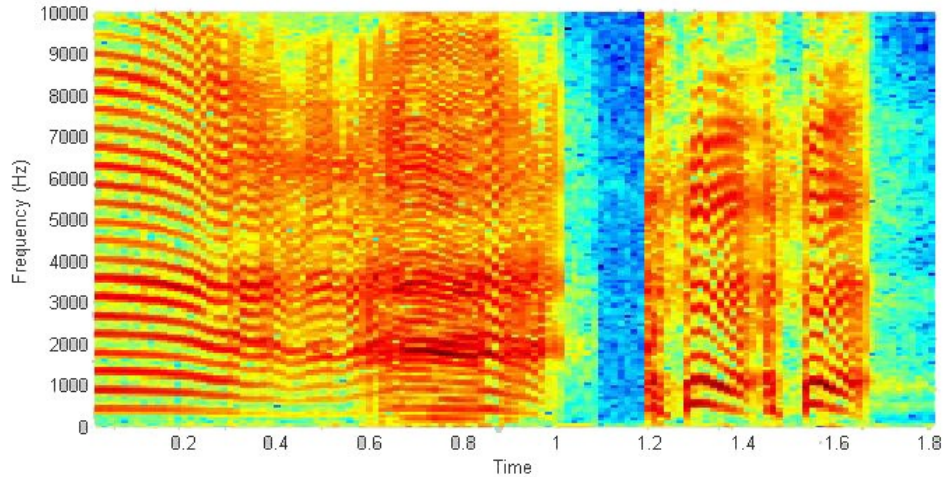


Figure 11: Spectrogram, Baby Crying

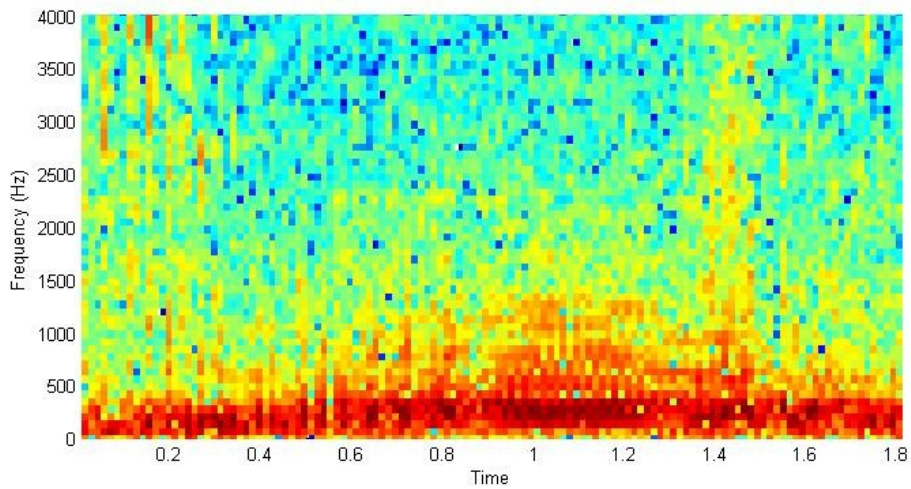


Figure 12: Spectrogram, Harley Davidson

5.5 Techniques of Feature Extraction

Feature extraction is the process of finding and extracting the most useful identifying information from a given sound signal. Over the course of this project, three main methods of feature extraction were employed.

5.5.1 Peak Detection

The goal of the peak detection method is to identify the distinct peaks of a frequency spectrum and thus to come up with some identifying information for that signal. To do this, the spectrum was provided as an input to “extreme.m,” a function written by Carlos Adrian and Vargas Aguilera and available at www.mathworks.com. The output of this function is an exhaustive list of the location and magnitude of each peak in the spectrum. It was then necessary to narrow down the list of extrema. Two conclusions were drawn by viewing the spectra generated by “fdanalysis.”

- Very low peaks contain little identifying information and may safely be ignored.
- Peaks that are closely clustered are probably redundant and might have an adverse effect on the weighting of the neural network matrix.

In light of these observations, some additional code was added to “fdanalysis” to parse the raw data from “extreme.m.” First the highest peak is identified, and then it steps through the peaks in descending order by amplitude, ignoring all peaks within 20 samples of a higher peak. This is to eliminate “ancillary” peaks that contain little information and may obscure the results. Additionally, all very low peaks, defined as those peaks that fall below a threshold percentage of the amplitude of the max peak are ignored completely.

The figure below is a frequency spectrum of a cargo boat over a 5000 sample segment in which the peaks are identified in red. Some key features that can be extracted from this include the frequency of the highest peak, and the number of peaks with amplitude of over 0.75 the maximum amplitude.

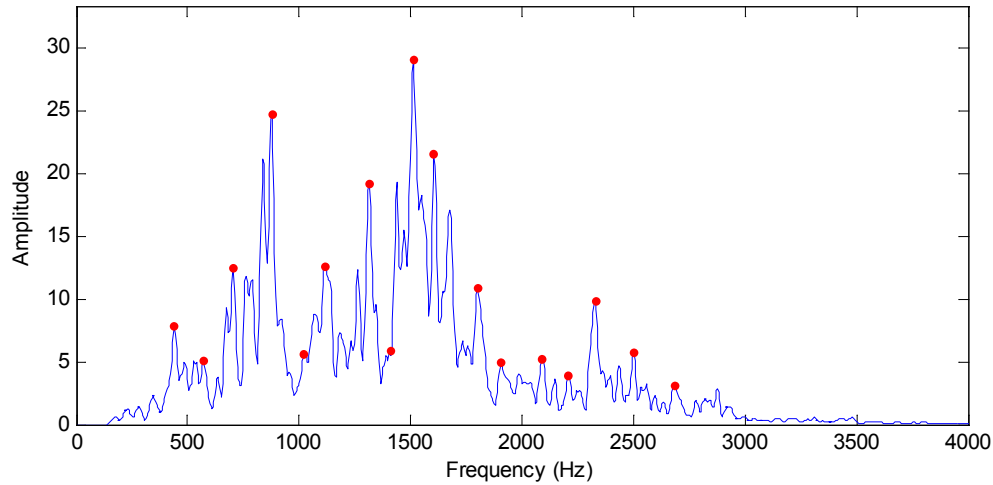


Figure 13: Peak detection Example

5.5.2 Energy Analysis

Energy analysis is a very useful method of feature extraction, in which the base spectrum is broken down into frequency bands. The energy of each band can then be calculated and used as a feature. Below is the same frequency spectrum from Figure 15 partitioned into 500 Hz segments.

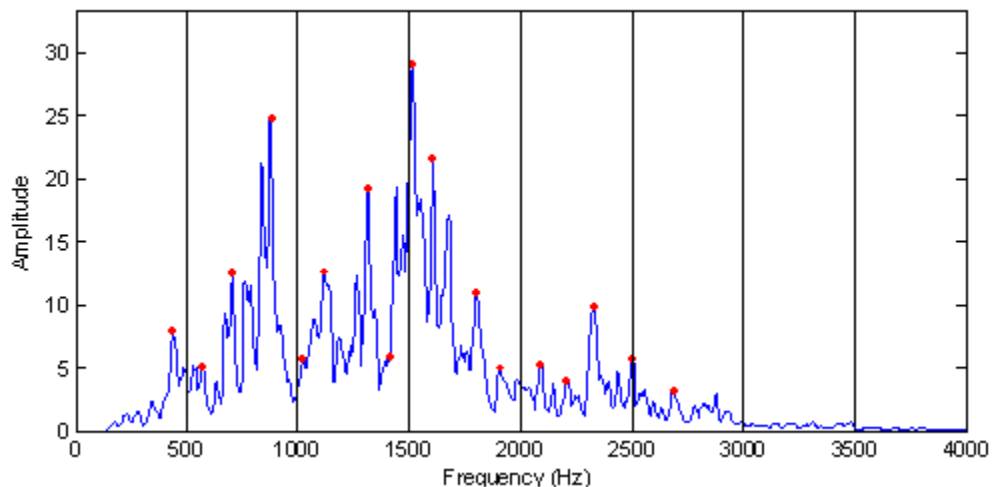


Figure 14: Frequency bands used in energy analysis

For a description of energy analysis as applied to boat noise, please see section 7.4.

5.5.3 Smoothing

The main purpose of using smoothing is to bring out the underlying shape of the frequency spectrum. The smoothing function used was “smoothts,” a built in MATLAB function. A very coarse window length of 150 was specified as a parameter for this function, and the result was considerable smoothing. The smoothing method used was the MATLAB default ‘box’ which is an equal weight window.

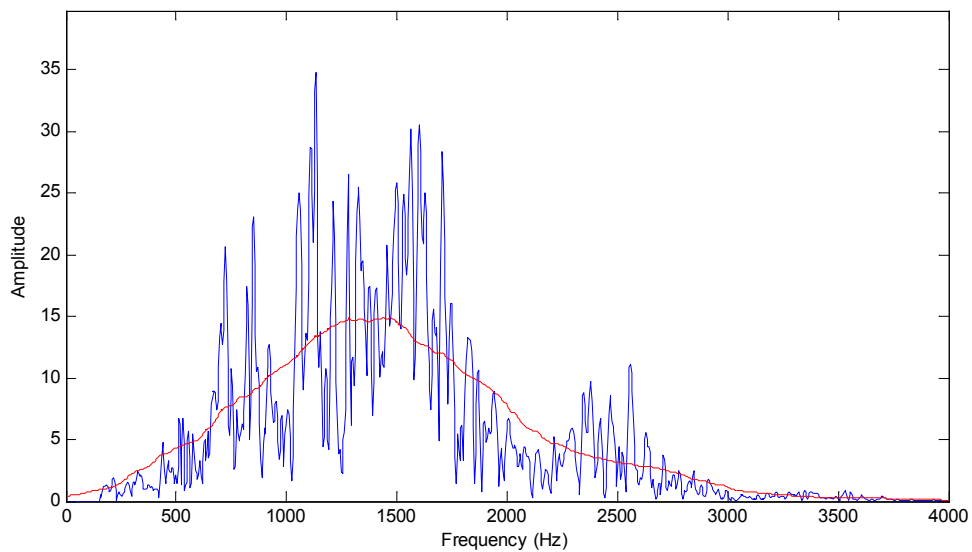


Figure 15: Smoothing applied

The above figure shows a spectrum from a cargo boat (blue) and the resulting smoothed function (red.) The smoothed function follows the general shape of the original spectrum. One important feature that could be extracted from this is the “central” frequency, defined as the maximum value of the smoothed function.

6 Overview of Neural Networks

Neural networks play a major part in the identification algorithm of this project. The sections below describe what this idea is and how it is used in this project. First will come a discussion on what a neural network is and the theory behind it, followed by a discussion of how MATLAB handles the problem and why MATLAB is useful, and finally the progress in MATLAB this term.

6.1 Neural Network Theory

A neural network is a computing algorithm that is modeled loosely after the human brain. Unlike many computers of today it uses parallel processing rather than series. In the past the term “neural network” was used to refer to a biological neural network, which had biological neurons. Biological neurons are what make the brain function in modern vertebrates. Modern terms now include the category of artificial neural networks. This field involves complex mathematical algorithms to describe something, and is what will be talked about here.

Artificial neural networks are designed to have interlocking neurons, which means that the individual neurons work together to provide the desired result. This output produced relies on the neurons to work together based on a series of weights that they are assigned to produce the output of the network. One advantage to neural networking is that most can function even if several of the neurons nodes are not functioning. (Wikipedia, 2007)

Neural network theory is considered a branch of computational science which models or simulates a theory that is hard to study otherwise. One of the main functions is to produce artificial intelligence (AI) in computers. Being self sustaining is one of the key advantages to a neural network. After the initial training, it can be left to its own devices and can be relied upon to complete the tasks given to it, in a dependable fashion.

Neural networks are able to train themselves after being given a concrete set of target data. This ability to self-adapt is what researchers are hoping will allow the breakthrough in a computer that is able to think on its own. Neural networks have at least 2 layers of neurons, and traditionally have a third. The first is the input layer, where raw

data is fed into the matrix. The final layer is where the output data is streamed, and this is where the results can be observed. In the middle can be any number of layers depending on the complexity of the data. These middle layers are what process the data, and can even function to adapt the network. (Nissen, Neural Network Theory.)

6.2 Neural Networks in Practice

The first step of the process was to train how to construct and run the “nntool” function in MATLAB. A simple tutorial was found online that taught how to create a simple XOR gate using nntool (Jiang, 2001.)

Creating a matrix for the input was the first step in this tutorial.

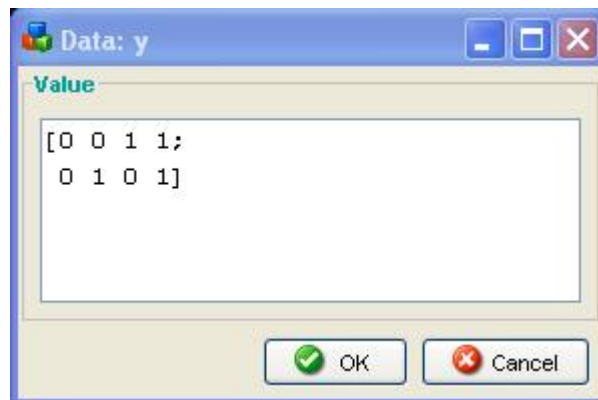


Figure 16: Input Data

From Fig 16 it can be seen that a simple 4x2 matrix has been created using every combination of binary inputs possible. After the inputs have been created, a target data set needs to be created.

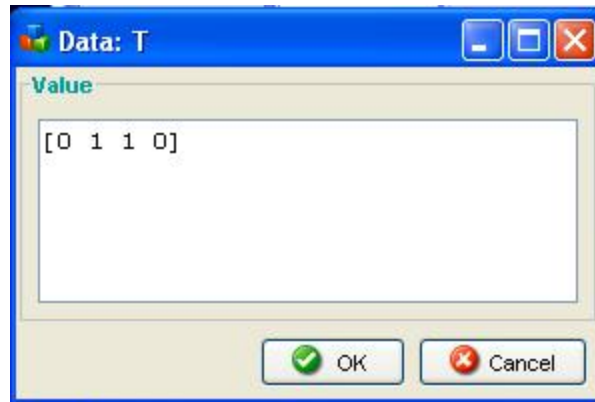


Figure 17: Target Data Set

The Boolean functionality of an XOR gate is to output a one, when the two inputs are different. Figure 17 have a 4x1 matrix that match the columns of Fig 16 and demonstrates this.

Now that an input and a set of target data were created, a network to implement them was needed. This was done with the following parameters. The network was a feed forward backprop loop, which is the standard network to input raw data, provide target, and train the network. The input ranges were simply retrieved from the input y in Figure 16. The next important parameter is the Training function which is the TRAINLM algorithm. The adaptation learning function and Performance functions were left at their default settings. Because of the small size of the input matrix there were only two layers required for this matrix, with the second one only requiring one node due to the fact that the expected output only required one binary digit. The next step in the process was to actually train the network to perform the desired operation. This was done by opening up the network properties and setting the inputs to Y from Fig 16, and the target data to be T from Fig 17. The function was then trained, but still needed to be tested. A new set of data was then created in the form of a test, see Fig 18. This data is in the same size as the input, however could have had more columns if needed.

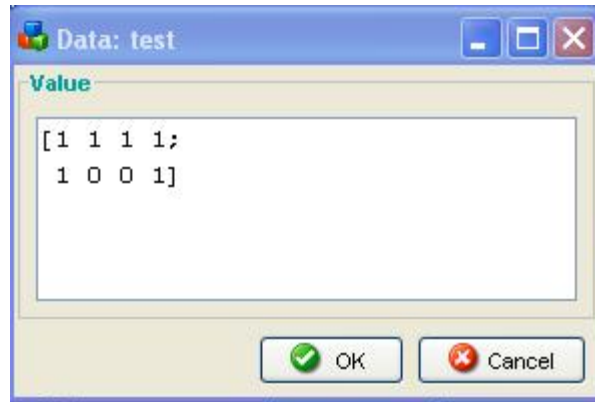


Figure 18: Test Data

After simulating the network using train from figure 14, the output results were observed, (see Figure 19).

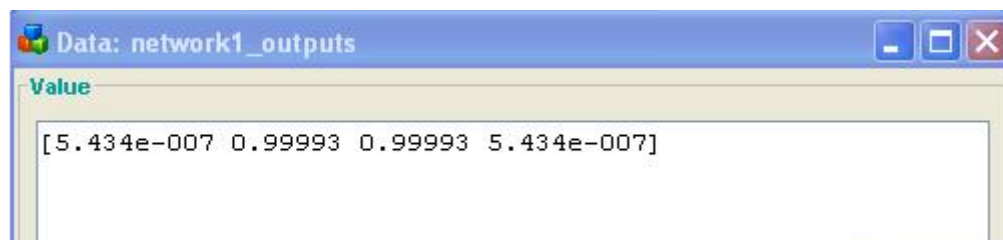


Figure 19: Neural Network Test Results

As can be seen from Fig 19 the results were very close to the expected $[0 \ 1 \ 1 \ 0]$ matrix expected. There will always be a certain degree of error in the matrix, however this can be retrained and adapted to converge to within a desired margin of error.

The data manger is what most projects will look like, with certain inputs, test sets, and training. Along with Target data, a network, and the generated output networks as well.

Getting more technical, the weights that were created in the training of the network, based off of the input of the initial training matrix can be viewed under the network properties. Next to it is the pre-designated input ranges. If an input matrix deviates from these values, there will be an error, and the matrix will be unable to correctly process the information. The weights that the matrix gives each individual column can be seen in Figure 20.

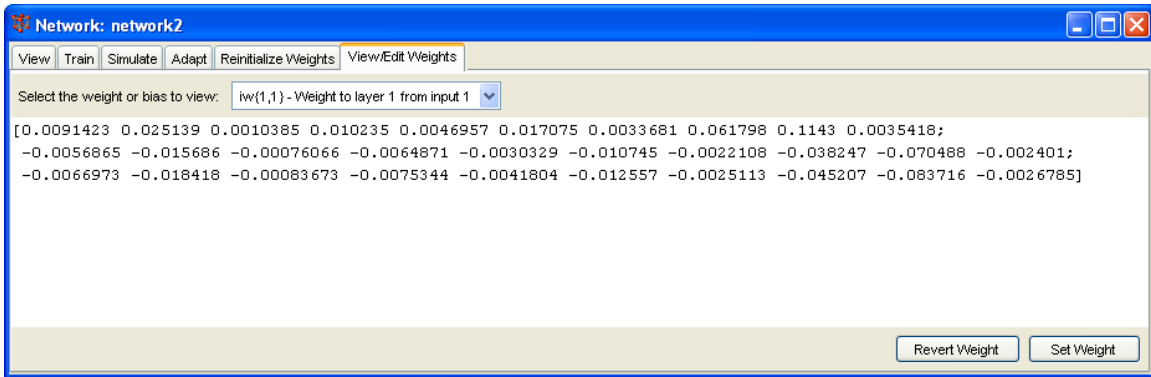


Figure 20: Weights

For the purposes of this project a neural network was decided to be used because of the ease of use. All the functions were in place, a network was only needed to be created and trained. The alternative would have been to create a separate matrix identification algorithm for taking the results of the feature extraction and identifying which boat it was. The actual implementation of how neural networks were used will be discussed further in section 7.5.

7 Methodology

This project's goal is to take as input an analog sound signal from a hydrophone, digitize and process this signal, analyze it in order to extract its features, and finally use this information to determine the type of boat passing.

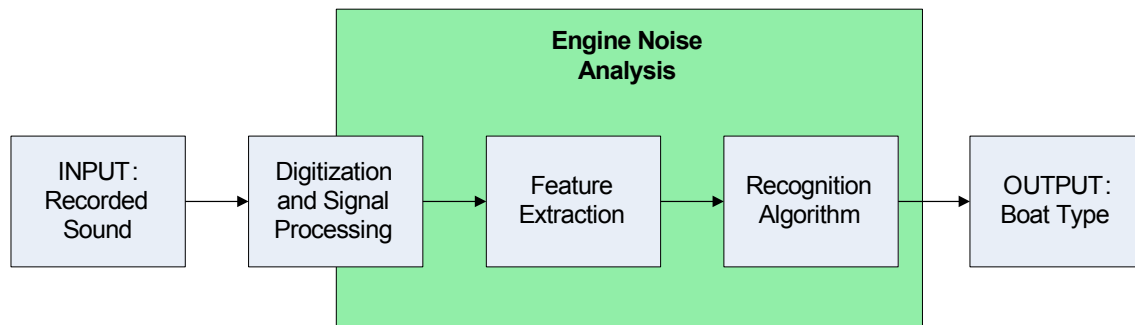


Figure 21: Overall project block diagram

The above figure shows the process by which this goal can be reached. First the recorded sound is digitized and pre-processed. Next, features are extracted from the resulting digital sound clip using frequency analysis code. These features may then be input to a recognition algorithm (in this case the neural network) in order to produce the ultimate output of a boat type determination.

This section describes in detail the methods used in order to accomplish these tasks, and describes some of the design choices that were made to that end.

7.1 Selecting a Hydrophone

A hydrophone will be necessary for this project in order to record engine sounds from an underwater vantage point. From Wikipedia:

Hydrophones are an important part of the [SONAR](#) system used to detect [submarines](#) by both surface vessels and other submarines. A large number of hydrophones were used in the building of various fixed location detection networks such as [SOSUS](#) (Wikipedia 2006).

The basic purpose of the hydrophone is to pick up sound energy from an underwater source, essentially an underwater microphone. Many hydrophones use as a sensing element a very thin, flexible piece of piezoelectric material. Its first major usage was in military applications to detect noise emitted from enemy ships and submarines in the two world wars. Hydrophones can be used individually, however they are more often than not used in large arrays containing into the hundreds of hydrophones. Such massive arrays are often used to search for undersea life such as whales at great distances, directional hydrophones are much more powerful over great distances, so using them in an array allows for both power and versatility

There are four major factors to consider when purchasing a hydrophone's specifications needed. These are the frequency response, the operating depth, and the overall sensitivity of the hydrophone. Frequency response describes the range of frequencies that over which the hydrophone can accurately detect sound waves. The operating depth of the microphone describes how deep in the water it can be placed and still operate correctly, note this is not the max depth it can go before succumbing to pressure, which is usually much deeper. The sensitivity of the Hydrophone determines how accurate it is when describing amplitude and frequency of a signal. The average hydrophone has a sensitivity from -220 V/uPa to -150 V/uPa, smaller negative numbers represent a more sensitive hydrophone. The last factor is whether it is omni directional or directional. That is, it can pick up sounds from all directions, or just the one it is pointing, respectively.

For this project it was determined that a hydrophone with a range of frequencies of at least 20Hz to 20kHz is wanted the average sensitivity of -220 to -150 V/uPa as well, the operating depth of the device is not at all important because it is expected to use them with a few feet of the surface, therefore an operating depth of 6 feet should be plenty to keep our device working properly, because at such an insignificant depth, it was assumed all hydrophones considered would be well within this specification. Because the boats will be approaching from at least two possible directions in the canals an omni direction hydrophone will be needed. One more requirement was that it be capable of plugging directly into a regular microphone jack so that it is compatible with the laptops we will be using it with.

Upon researching the internet, several possible devices came up. The first device looked at was the dolphin ear set which was a complete package, costing \$250. It also did not provide the exact specification of the hydrophone so it was not considered.

The next possible hydrophone looked at was the Aquatic Hydrophone system. This had a frequency response of 20-50k Hz, and was omni-directional. It is designed to be plugged into a standard 3.5mm jack, the same size as standard microphones, and receives all necessary power from the same jack. Inside it has an FET buffer preamp which should allow us to use the direct signal from it, without having to go through a secondary amplifier, or other sound enhancement system. It came standard with a 3 meter cable, which was determined to be sufficient from our dockside usage of it. After all of these statistics were taken into account it was determined that this hydrophone was what we were looking for, with a price tag of \$119 we went ahead and purchased it. (Aquarian Hydrophone system, 2005).

Hydrophones	Frequency		Interface	Cord	Operating depth (m)	Sensitivity	Transducer type	Directionality
	(Hz)	Cost		Length (m)				
Dolphinear C-10	7-22000	\$259	3.5mm					
	25-25000		stereo	8			Piezoelectric	Omnidirectional
AQ3	20-100000	\$154	3.5mm					
			stereo	3				Omnidirectional
AQ9	20-100000	\$175	3.5mm					
			stereo	9				Omnidirectional
AQ15	20-100000	\$196	3.5mm					
TC4013	1-170000		stereo	15				Omnidirectional
TC4032	5-120000			6	700	-211 +-3	Piezoelectric	Omnidirectional
				6	600	-170 +- 1	Piezoelectric	Omnidirectional

Figure 22: Hydrophone comparison chart

This chart compares all of the different hydrophones considered for our project. The missing fields represent specifications which were not readily available from the distributor. The hydrophone which was eventually chosen for this project was the AQ3, do it its known interface and price. The Dolphin ear was also an excellent runner up, but was turned down due to the excessive cost.

7.2 Acquiring Recordings

In order to acquire the necessary sound recordings to work with the first step taken was to contact WPI students doing their IQP in Venice. *The CanalSpace Project* and the *Taxi Transportation Project* were selected specifically because those teams were working closely with the canals already.

A correspondence was maintained over the course of B-term. The team's main contact in this was Greg Bukowski. After the initial recordings were received, concerns about their integrity necessitated a request for additional recordings, in addition to information about the precise methodology used in making the recordings.

One thing that became clear over the course of C-term was that the recordings made by the Venice IQP students were quite conflicting and in many cases difficult for the human ear to differentiate, let alone a computer algorithm. As a result, it was requested that recordings to be made by Professor Carrera's staff in Venice.

The project team provided several recommendations to maximize the worth of the data received, including:

- A larger data set than previously requested
- Digital pictures to accompany each recording, in order to better determine what might be causing differences in the sound signatures or similar boat types.
- Alterations in the recording procedure to ensure clean recordings

Below is the setup used by Daniela Pavan and Andrea Novello to make the second and third set of sound recordings. A wooden boom was used to orient the hydrophone to face outward into the canal, and GoldWave recording software was installed on the laptops that were used in order to make the recordings. Details on the suggested recording procedure can be found in Appendix A: “Student Recording Manual.”



Figure 23: Recording setup used by Venice staff

7.3 Feature Identification

To identify the most descriptive features to extract, spectrograms were created for each sound file received from Venice and carefully scrutinized. The figures that follow are spectrograms from a cargo boat, a personal boat, and a taxi boat respectively.

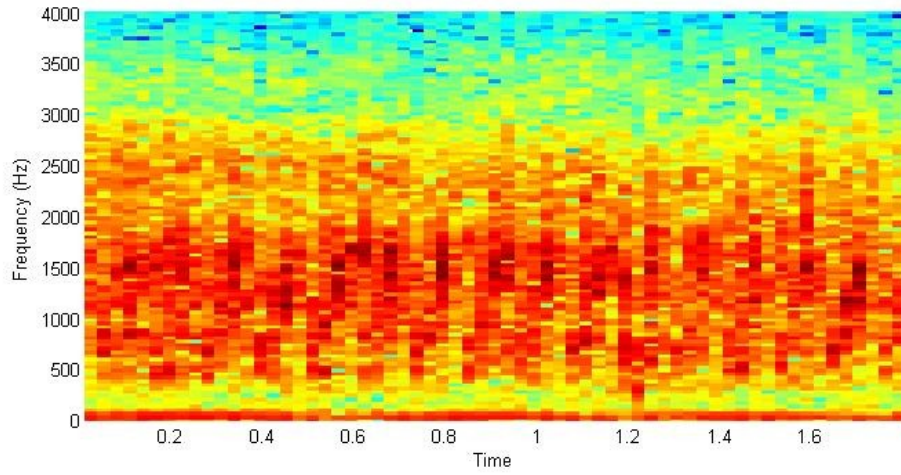


Figure 24: Spectrogram, cargo boat

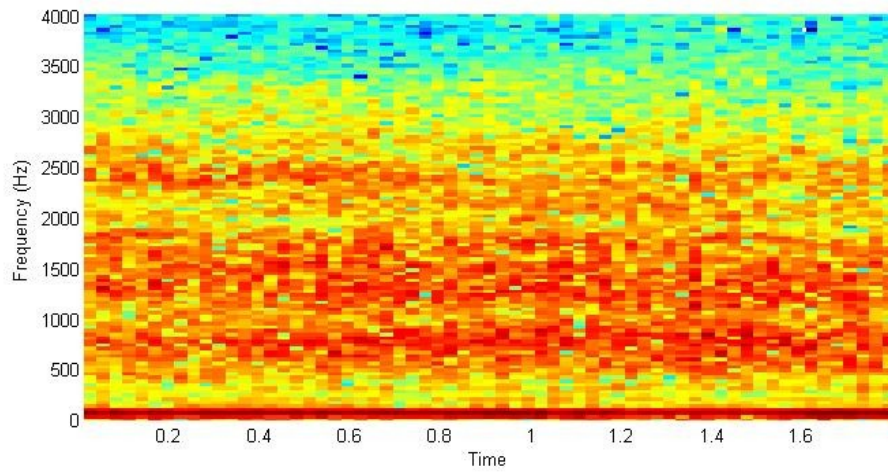


Figure 25: Spectrogram, personal boat

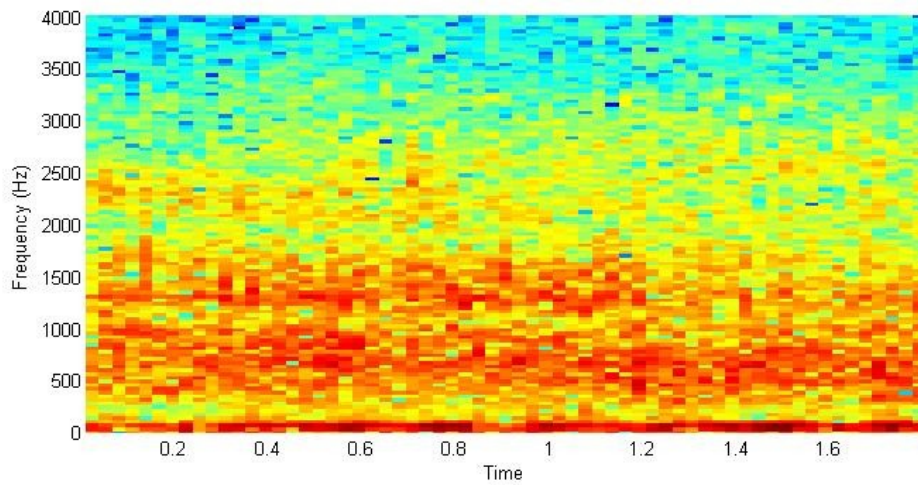


Figure 26: Spectrogram, taxi boat

Although the differences between the above three example spectrograms may not be readily apparent, some key strategies for differentiating between boats were identified.

- Many spectra have characteristic frequency “bands”
- Many spectra can be described in terms of a “central” frequency

7.4 Feature Extraction

In light of the findings from the feature identification phase, the final frequency domain analysis function for this project, “fdanalysis11,” was tailored to extract the information most important to boat identification. Please see section 4.6.2 and section 4.6.3 for some of the theory behind the techniques used in the boat feature extraction method. The MATLAB code for “fdanalysis11” may be found in Appendix B.

Each sound signal was broken up into 5000 sample segments (~0.25 seconds at 22050 Hz.) Each of these were analyzed in turn. At the time of this writing, the current implementation is not able to detect dynamic features between the segments; however this is an improvement that may be made at a later time.

The energy analysis method was employed. The base spectrum was divided into eight frequency bands, each 500 Hz wide. The energy within each band was calculated, resulting in a vector that contained a discrete representation of the frequency information in the signal that the neural network could use to differentiate between boat types. Additionally, a central frequency was estimated as the frequency at which the energy of the lower frequencies equals the energy of the higher frequencies.

Finally, a coarse smoothing function was applied to the spectrum in order to approximate the rough shape of the spectrum. The maximum of this smoothed function specifically was useful in identification.

The final output produced by this process is a 1x10 column vector. The first 8 entries contain the energy levels of the eight frequency bands respectively. Entry 9 was the central frequency, and entry 10 was the frequency of the smoothing function maximum. For more information on how this data was formatted for the training and test sets, see section 7.5.3.

7.5 Neural Network Development

The purpose of using neural networking in MATLAB with regard to this project is to provide the network with raw data, and have it correctly identify a boat. The theory is that an input matrix is passed through a pre-constructed network with defined various weights, and the network is able use the parameters that it was given, to correctly identify what the expected result of each column of the matrix is. For instance, if a 4x2 matrix is used as input to the matrix, the matrix will look at each of the 4 columns, and give a 4x1 matrix output. Neural networking should prove to be the easiest method available for boat identification, which is why it was chosen for this project.

The next step of this project was to create a network more similar to what would actually be input received from a boat engine. To do this a large training matrix was needed, as well as a target data set to match. The following MATLAB code creates a 10x10 matrix and stores it as the variable x to simulate the many spectral peaks that may be picked up from the passing of a boat. The amplitude of the signals have all been set to one in order to try and keep the neural network from getting confused due to the low number of samples required to train it. After that a target set T1 was created which followed a simple guideline to try and teach the neural network what the output should be.

This became the first intermediate step to learning neural networking, and preparing for the actual input that would be used to identify boats. The next step was to create a series of input data points that include many rows and a small number of columns. The intention of these matrices was to simulate individual data packets that would be fed into the system. Each row represents a separate frequency. The amplitude of these points are what is then fed into the matrix. As can be expected it will take a large amount of data to fully train the neural network and force the training curve to converge. The more data that can be fed into the neural network, the more accurate it can be, and therefore the more reliably it can identify a passing boat.

The next large step was to take this idea, and apply it to a simulated set of boat data. Therefore, three sets of data were fabricated to simulate three different types of boats. First, 3 sets of distinct data was created using random data points to simulate three sets of boats. Then three targets are created for each set of data. The idea was to

train the network for each set of data, giving them the respective targets. The P series of data was trained to the target of T_p , which equaled 2, the Q series of data was trained to the target of T_q , which equaled 1, and the R series of data was trained to the target of T_q , which equaled 3.

The final step was to actually train the network. The network was trained by importing each matrix into the neural network and then training it to the target goal. The training parameters had several key factors to take into account. The number of epochs used was set to 1000. This meant that the data from the input matrix would be presented to the network 1000 times in the training. The `Min_grad` was set to $1e-010$. This parameter dictates the acceptable error that is allowed. After the matrix has achieved this level the training will stop.

7.5.1 Network Properties

A layer recurrent neural network was created for this project. The type layer recurrent was chosen because of its ability to accept matrices with a large number of rows, and target a single number per column. Since the matrices used have one column and 10 rows, this fits the criteria. A layer recurrent matrix was also chosen because it is hoped that it will be able to handle back propagation when previous inputs need to be taken into account. The input ranges were assigned to the input ranges of Q1, and it was assumed that this would be sufficient for all inputs. In the training of the matrix there were no errors as a result of input going beyond the bounds so this assumption appears to be correct. The training function used was `TRAINLM`, and learning function used was `LEARNGDM`, and the performance function used was `MSE`. These functions were chosen as the result of reviewing several online sources which all use those parameters. Experimenting with other functions did not seem to give noticeable differences, and so the network was created using the known parameters. Two layers were needed for the network, with the first layer having 3 neurons using the `TRANSIG` transfer function, and the second layer having 3 neurons due to the fact that there are three numbers in the output, but using the `PURELIN` transfer function. These figures were determined by using trial and error.

7.5.2 Training Using Non-Boat Data

For training, the first set of data entered would be trained to the network, and the rest of the sets of data would be entered into the matrix through the adapt function. It was originally attempted to simply enter all of the data into the matrix through the train function, however it appears that each time the train function is run, it resets the neural network to the most recently trained data. After the initial training, the matrix seemed to converge (see figure 19), so subsequent adaptations proceeded.

Now that much of the hard work has been accomplished in MATLAB, it is only a matter of reapplying the lessons learned thus far to new sets of data and analyzing it. To test the data on a more realistic set of data, three actual sounds were taken and fed into the neural network. These three sounds were a baby crying, a Harley Davidson motorcycle, and an Orchestra. The network was trained to a small set of data for each one, and then simulated. The results can be seen below in figures 20, 21, and 22.

The following three figures show what the results of the training were. Each output was a 3x1 matrix for each set of data. The criteria for a successful identification were that one of the values of the data was converging to one and the other two were converging to zero. From these figures it can be seen that it was able to tell exactly what a baby was, it had a good idea what an orchestra was, and it was confused at first as to what a motorcycle was, but then towards the end it was able to tell.

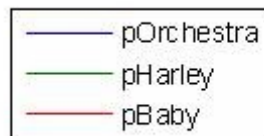


Figure 27: Legend for Figures 28-30

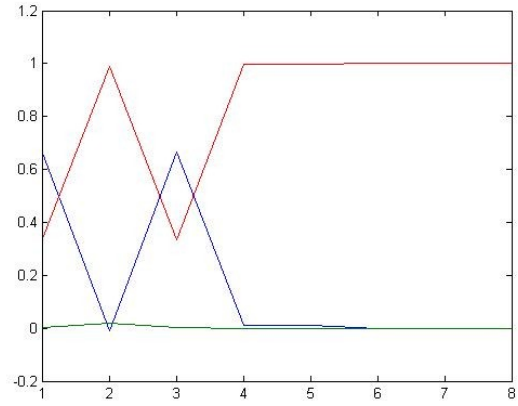


Figure 28: "Baby Crying" Neural Networking Results

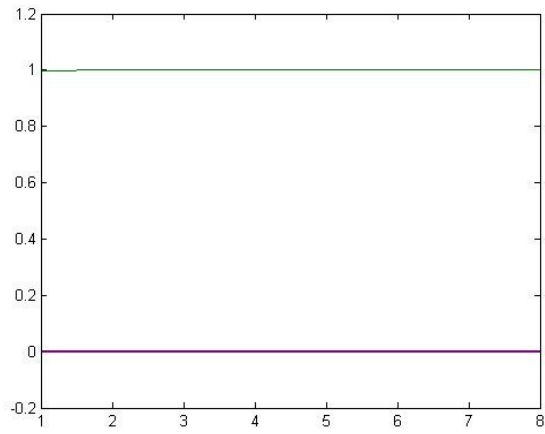


Figure 29: "Harley" Neural Networking Results

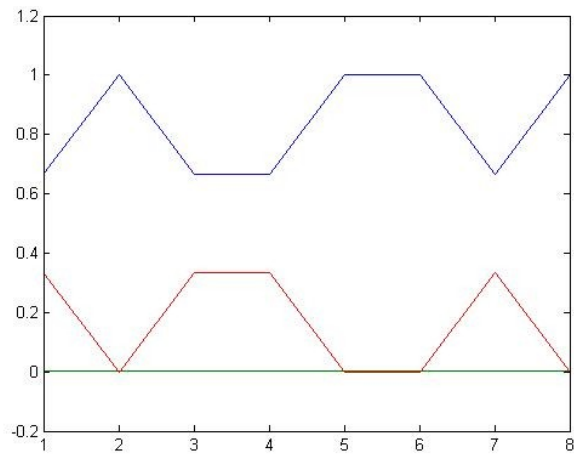


Figure 30: "Orchestra" Results

7.5.3 Format for the Training and Test Data

The final method used for training a neural network to identify boat types utilized a series of 1x10 column vectors as described in section 7.4. Where multiple training samples were used, each sample's feature information comprised a column of the training matrix. For example, for a 6 sample training set, a 6x10 matrix was used to train the neural network. Figure 31 illustrates the format of the training and test data used.

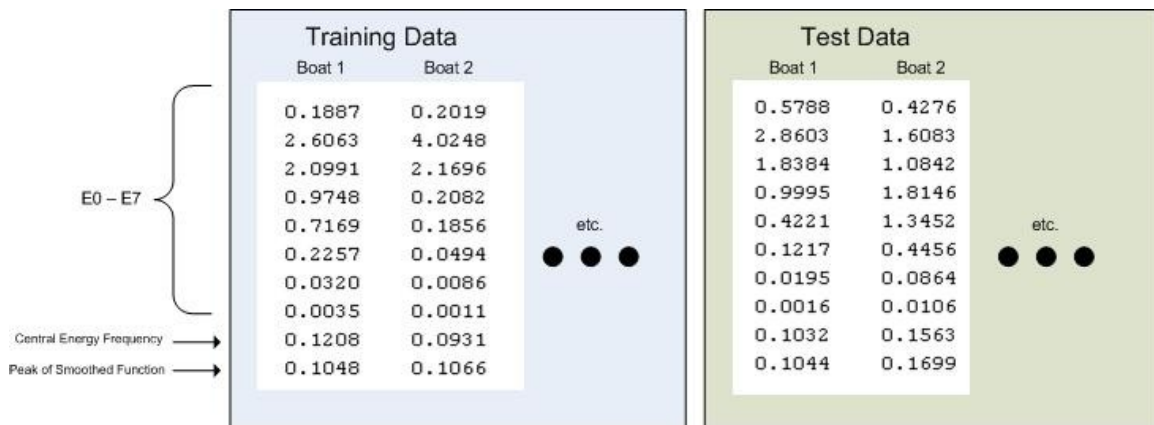


Figure 31: Format of the training and test data

8 Testing & Results

Testing for this project was somewhat difficult due to the small sample set provided to work with. In all, merely 28 samples were recorded well enough to gather information from, including 9 that were identified as cargo boat recordings, 10 as personal boat recordings, and 9 as taxi recordings. Additionally, it was established quite early that for testing to be valid, there must be no overlap between samples used for training the neural network and for testing that same network. Due to these strict considerations, testing needed to be designed very carefully.

Testing for the recognition algorithm was done in two phases, which shall be explained in the following section.

8.1 Post Screening

One major consideration necessary in testing was the issue of post-selection. Due to different initializations, two different neural networks given the same training inputs might result in very different weight matrices. For best results, it was discovered that it was often necessary to reinitialize and retrain the weight matrix several times until a satisfactory result was achieved.

For the purposes of testing, this post-screening was done in phase 2 but not in phase 1.

8.2 Generalization

For a neural network to be useful it must “generalize” well. That is, it must be able to work for *all* inputs, and not only the training set it was given. Our testing reflected this by separating the training set from the test set.

If allowed to train for too long, it is typical for neural networks to become so refined that they recognize *only* the training set, and may fail at identifying inputs that differ from the training set by even a small amount. This phenomenon is known as “overfitting,” and is very bad for generalization. The figure below shows an example of a neural network that has been trained for too long and exhibits overfitting. Note the extremely low performance after only 90 epochs. Such a network would have a difficult time identifying anything but its training set.

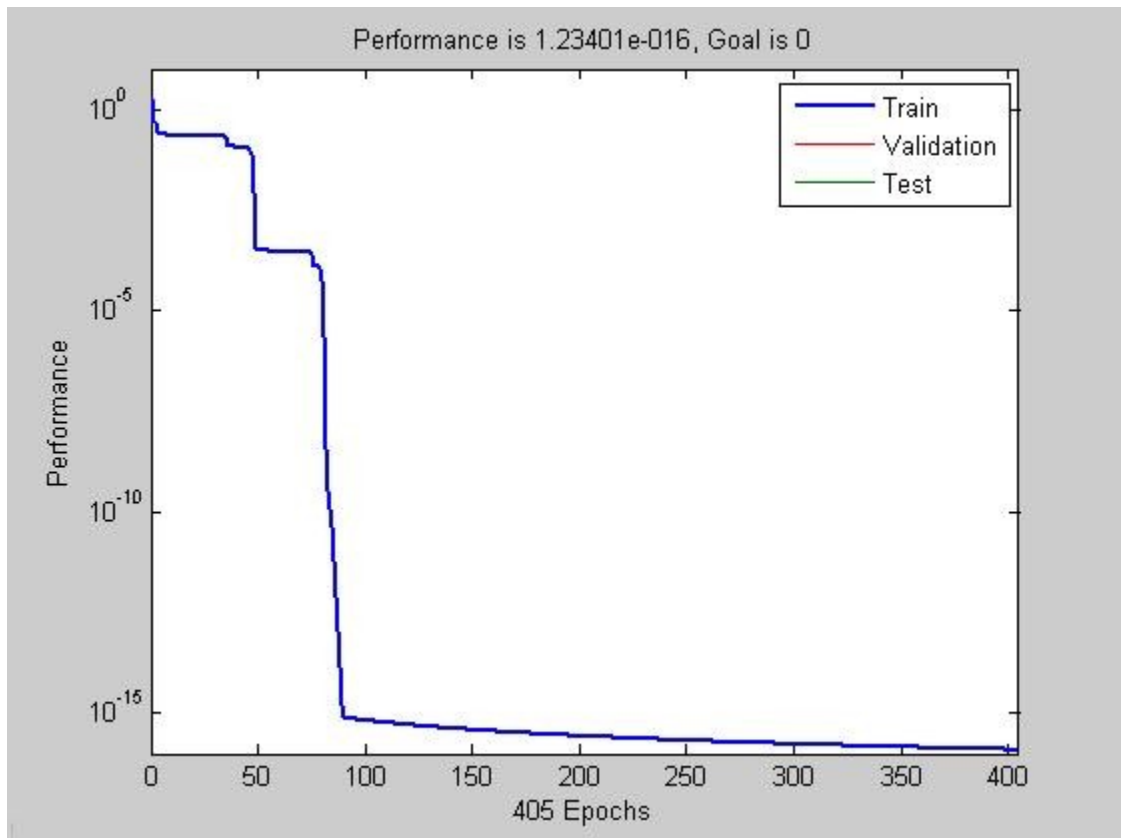


Figure 32: An example of overfitting

One strategy to counter overfitting is known as “early stopping.” As the name suggests, this provision halts training once a given “goal” is reached. This helps prevent overfitting.

The following figure shows a training session in which early stopping was utilized. The black line at 10^{-2} is the goal performance. Note that training halted at 327 epochs when the training reached the goal. Had the training gone on longer, it is probable that overfitting would have occurred.

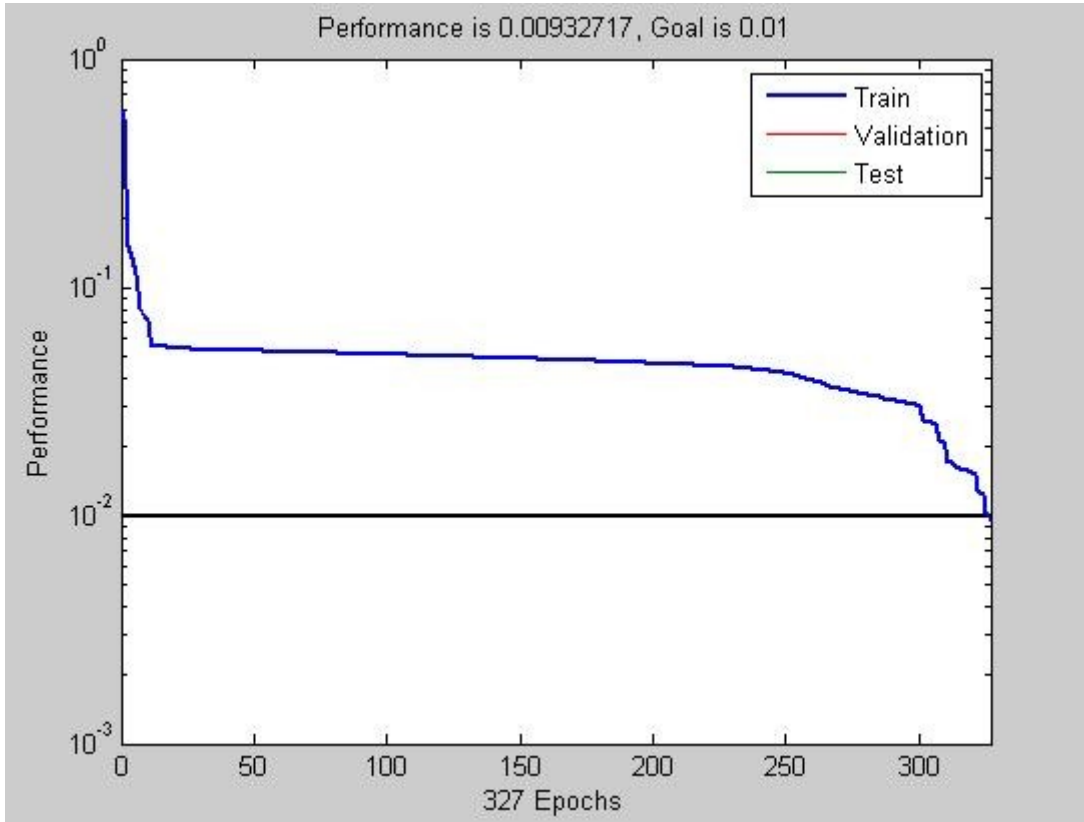


Figure 33: Early stopping

For the purposes of testing, early stopping was used in phase 2 but not in phase 1.

8.3 Testing Phase 1

For the first phase, training sets were picked by a random number generator. No special consideration was given for outliers. Training sets of size 6, 12, and 18 were used, with equal numbers for each of the three types of boats. The remainder of the sound samples became the test set. Furthermore, no consideration was given to post-screening matrix performance; the first network generated was used.

	Training Samples	Test Samples	Correct IDs	Incorrect IDs	Not Recognized	Success Rate (%)
Cargo	2	7	1	5	1	14
	4	5	0	1	4	0
	6	3	0	1	2	0
Personal	2	8	5	2	1	63
	4	6	0	1	5	0
	6	4	1	1	2	25
Taxi	2	7	1	5	1	14
	4	5	1	2	2	20
	6	3	0	3	0	0
OVERALL	6	22	7	12	3	32
	12	16	1	4	11	6
	18	10	1	5	4	10

Figure 34: Phase 1 test data

This data clearly shows that the methods used in phase 1 were very inaccurate. In many cases, the results were worse than random chance (~33.3%). These results would contrast greatly with those from phase 2. Some discussion on that later.

8.4 Testing Phase 2

For the second phase a more thorough process was used with an aim for maximum accuracy. Training sets of size 6, 9, and two 12s were hand picked and included no outliers. Additionally, the test sets were divided into “good” samples and “outliers” so that the effects of outliers as inputs might be gauged. Early stopping was used to prevent overfitting and post-screening was used liberally to ensure the best possible results.

	Rec. Quality	Training Samples	Test Samples	Correct IDs	Incorrect IDs	Not Recognized	Success Rate (%)
Cargo	good	2	3	0	2	1	0
	outlier		4	0	2	2	0
	total		7	0	4	3	0
	good	3	2	2	0	0	100
	outlier		4	1	3	0	25
	total		6	3	3	0	50
	good	4 (1)	1	0	1	0	0
	outlier		4	2	2	0	50
	total		5	2	3	0	40
	good	4 (2)	1	1	0	0	100
outlier	4		1	0	3	25	
total	5		1	0	4	20	
Personal	good	2	3	0	0	3	0
	outlier		5	1	1	3	20
	total		8	1	1	6	13
	good	3	2	0	2	0	0
	outlier		5	3	2	0	60
	total		7	3	4	0	43
	good	4 (1)	1	1	0	0	100
	outlier		5	4	1	0	80
	total		6	5	1	0	83
	good	4 (2)	1	1	0	0	100
outlier	5		0	1	4	0	
total	6		1	1	4	17	
Taxi	good	2	3	1	2	0	33
	outlier		4	2	0	2	50
	total		7	3	2	2	43
	good	3	2	2	0	0	100
	outlier		4	0	4	0	0
	total		6	2	4	0	33
	good	4 (1)	1	1	0	0	100
	outlier		4	2	1	1	50
	total		5	3	1	1	60
	good	4 (2)	1	1	0	0	100
outlier	4		0	0	4	0	
total	5		1	0	4	20	

Figure 35: Phase 2 test data

	Rec. Quality	Training Samples	Test Samples	Correct IDs	Incorrect IDs	Not Recognized	Success Rate (%)
OVERALL	good	6	9	1	4	4	11
	outlier		13	3	3	7	23
	total		22	4	7	11	18
	good	9	6	4	2	0	66
	outlier		13	4	9	0	18
	total		19	8	11	0	42
	good	12 (1)	3	2	1	0	66
	outlier		13	8	4	1	62
	total		16	10	5	1	63
	good	12 (2)	3	3	0	0	100
	outlier		13	1	1	11	8
	total		16	4	1	11	25

Figure 36: Phase 2 overall results

These results show that accuracy clearly increases with the size of the training set, especially when considering only “good” recordings. The 6 sample training set appears to be too small to properly train the network, while the 9 sample set is a bit better. The first 12 sample training set provided solid overall performance for both good samples and outliers. The second 12 sample training set was very effective at recognizing good recordings, but had trouble recognizing outliers. Also note that the second 12 sample training set had only 1 false identification, while the first had 5. This is likely because the second 12 sample set was more closely fitted.

Comparing the results of phase 2 with the results of phase 1, it is fairly clear that the procedures of early stopping and post-screening are effective to improve network accuracy. Furthermore, pre-screening for outliers is very useful because it ensures that the network is not confused by conflicting data.

8.5 Improving Accuracy

As stated previously, it is clear from this data that increasing the number of samples in the training set goes a long way toward making the neural network more accurate. This was expected to some degree because many of the recordings differed even within the same category. To maximize the effectiveness of this strategy, it is clear

that more samples are needed and of higher quality. If this process is to be attempted again, every effort should be made to get as many samples as possible.

Samples for these tests were drawn from multiple different dates and people doing the recordings. Further attempts should be made to reduce discrepancies in the recording procedure if this is to be attempted again.

Refinement of the feature set could also improve accuracy. The feature set could possibly be narrowed down to eliminate insignificant frequency bands like the one from 3500 to 4000 Hz. Also, research might be done toward additional features to extract, especially those in the time domain.

Since the percentages above might be greatly influenced by a number of outliers in the sample set, a function to pre-screen for these outliers based on a number of characteristics might also be beneficial.

Finally, it might be useful to investigate the viability of above water recordings for identification. If significant new features are noticeable in above water recordings, then they might be a valuable source of identifying information that could improve the accuracy of this approach.

9 Conclusion

The final product of this project is an algorithm that can identify with a fair degree of accuracy the three main different types of motor boats that are likely to pass in a Venetian canal. This process is not perfect, however, and many improvements may still be made to the procedure in order to improve its accuracy, including the collection of more samples and some refinements to feature extraction.

Additionally, this project was capable only of implementing the software algorithm. Were further time available, the next step for this project might be to design a prototype for an instrument that might be placed in a canal to gather data using this algorithm.

Future groups, if they choose to continue work on boat identification through engine noise signature, should first work to understand the methods used in this project including neural networks and feature extraction. Once this understanding is reached, attempts may be made to improve the algorithm and attain a greater level of accuracy. Once this is done, effort may be directed toward development of the hardware solution.

It is hoped that the data gathered by such an instrument might be useful to *Civitas Mobilis* and the City of Venice to better target measures to reduce the negative impact of *moto ondosso* on city infrastructure and help preserve the city for years to come.

10 Appendix A: Student Recording Manual

This project, sponsored by Civitas Mobilis of the European Union, aims to design and prototype an instrument that is capable of gathering data on passing motor boat traffic in a Venetian canal. The instrument will analyze the sound produced by boat engines, from which information about the boat type and engine RPM can be deduced. It is hoped that in turn, this data can be applied toward a measure to reduce the impact of “*moto ondosso*,” or motor boat wakes, on city infrastructure.

An important step in the development of this device is the initial recording of the actual boat noise. Unfortunately, since we the MQP project team will not be traveling to Venice, we need somebody else to do these recordings for us. If you are reading this, you have been selected to do the job. Try to follow the instructions in this manual as closely as you can to ensure that the recordings are as clean and usable as possible.

Thank you for your cooperation, and good luck!

10.1 Initial Setup

We strongly recommend that you use the GoldWave sound recorder for your recordings. This software is freely available at GoldWave Inc's website. You will need to install this on a laptop first.

- Go to www.goldwave.com and click the “Download” link.
- Download the installer from one of the provided download links. We recommend the second link, “direct from GoldWave Inc. primary download server.”
- Run the installer to install the software on the laptop.

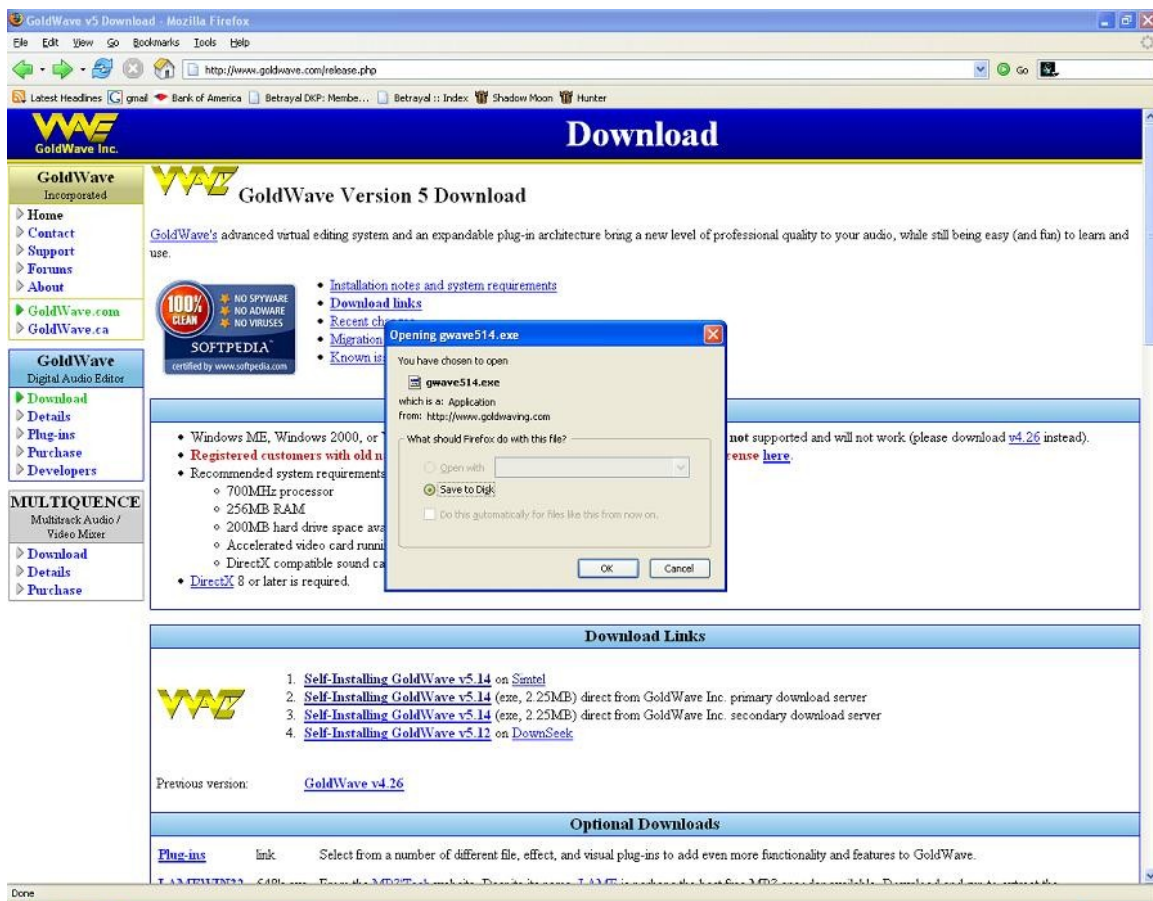


Figure 37: Downloading the GoldWave software

When you open up GoldWave, you first need to create a sound file to record to. You will go through this process for every sound your wish to record.

- Click the “New” button on the toolbar. You should see a window similar to the one in Figure 2.
- Set the number of channels to “1 (mono)”.
- Set the sampling rate to 22050 Hz.
- Set the initial file length. We recommend 30 seconds, which should be plenty of time to get the boat passing.

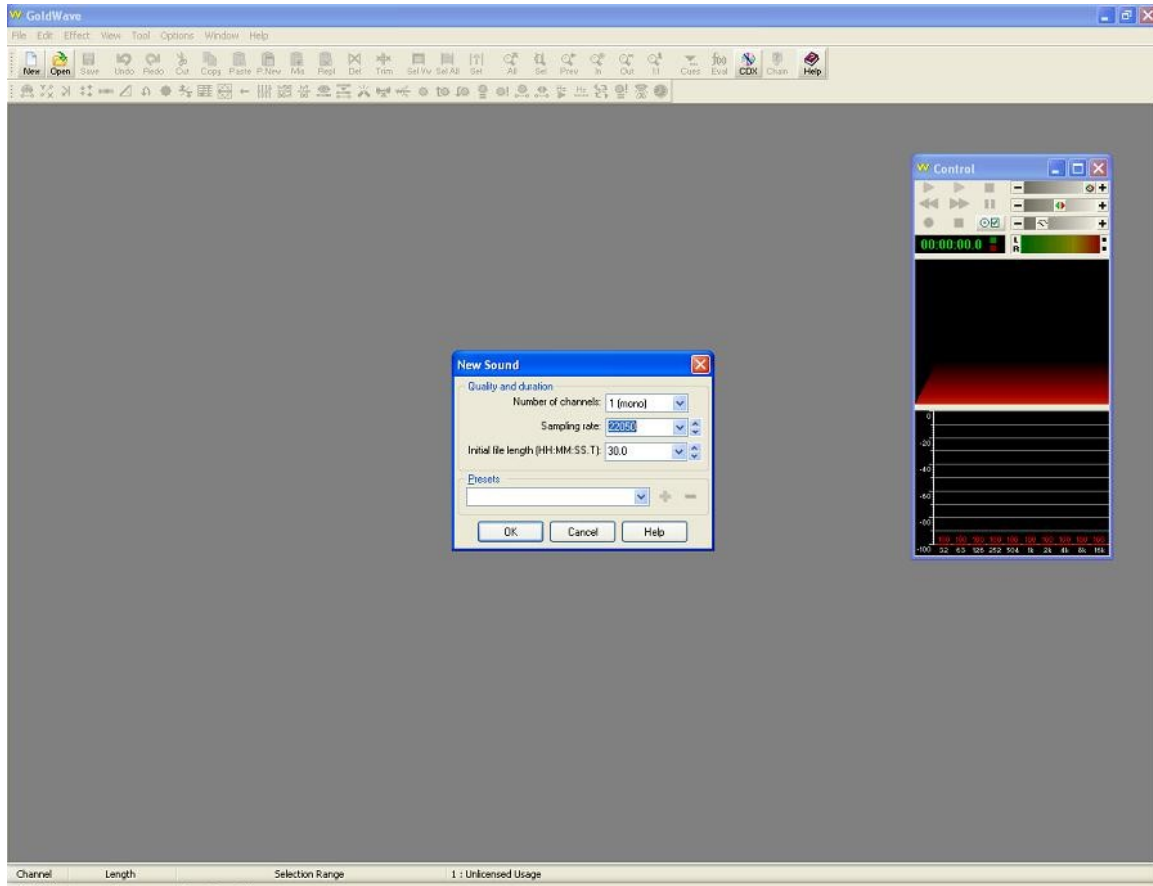


Figure 38: Creating a New Sound File in GoldWave

10.2 Recording a Sound

Find a good spot to record boats passing by. This part is up to you, but we recommend finding a spot with easy access to the water that is narrow enough so that you are most likely to have only one boat passing at a time.

Plug the hydrophone into the microphone input jack of the laptop, and submerge it 1 to 2 feet underwater. Make sure somebody is holding the cord at all times to prevent accidental loss of the hydrophone.

Start recording when the boat is approximately 40 feet away by hitting the red record button in the GoldWave (circled in red in Figure 3.) You may stop recording when the boat is about 40 feet past you. Make a note of the type of boat and of how fast the boat was traveling in comparison to other boats.

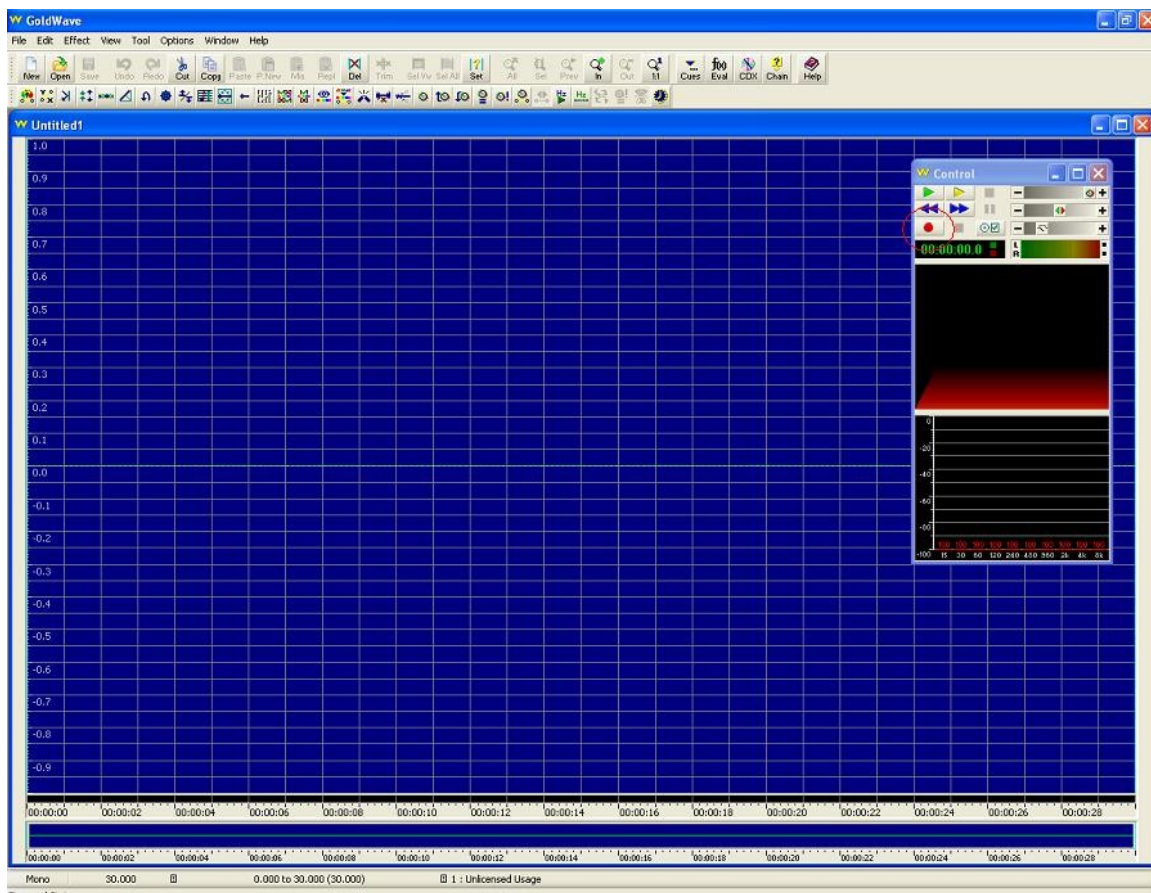


Figure 39: GoldWave, Ready to Record

We need at least 5 samples of cargo boats, 5 samples of taxi boats, and 5 samples of personal boats, although more are certainly welcome. While recording, please refrain

from talking in loud voices or otherwise making noise in order to reduce the amount of noise in the recording. This will make it much easier for us when it comes time to analyze the signal.

After a successful sample taking, save the file with a descriptive file name that makes note of the type of boat recorded, rough speed of the boat, and the number of the recording, for example, “Cargo-medium-1.wav” or “Taxi-fast-1.wav”.

When the recording is finished, you should see a waveform that looks something like Figure 4. You might want to bring headphones with you to verify that the sound file is of good quality. You want to make sure that the recording is neither too soft nor so loud that there is audible distortion (this so called “clipping” happens when the input saturates the sound card’s quantizer.) If there are problems of this sort, you should attempt to adjust the microphone input in the laptop’s sound control. If you are still having problems, please contact us as soon as possible so that we can come up with a solution.

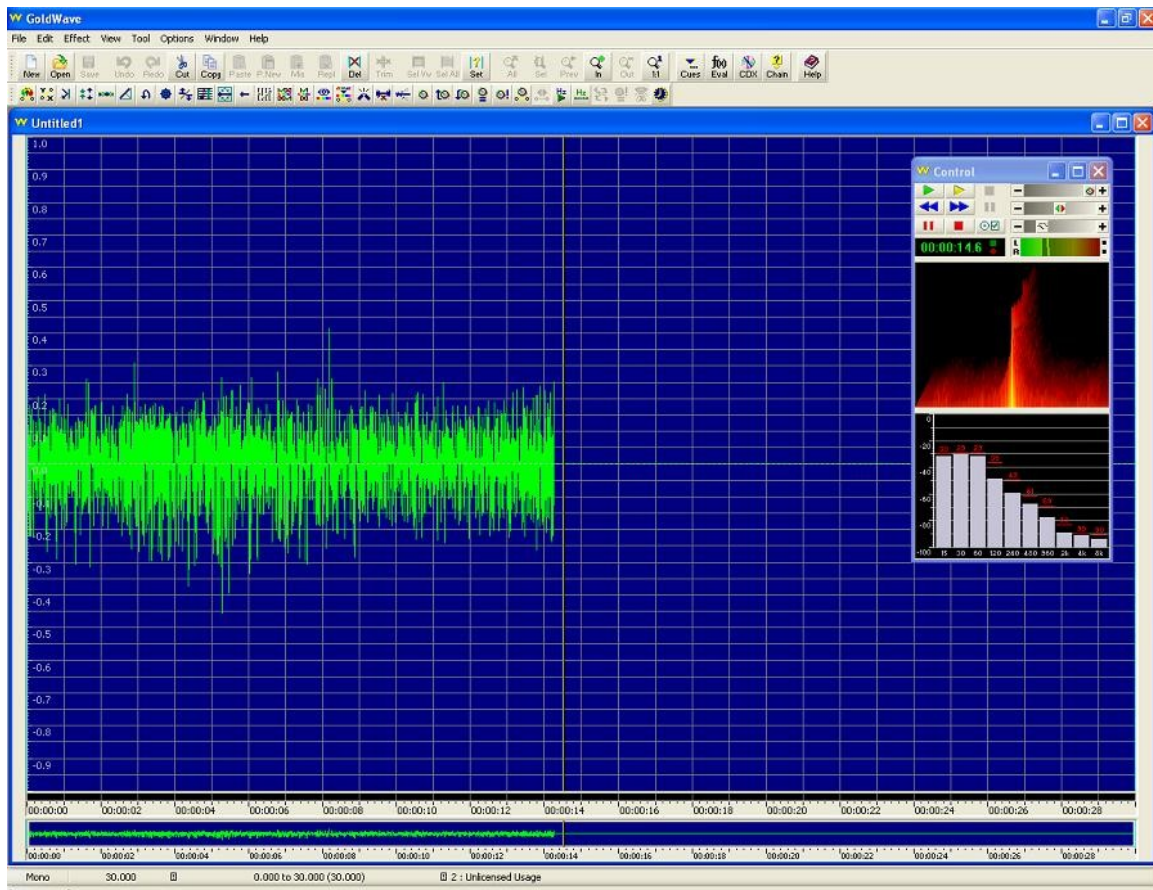


Figure 40: A Recorded Sound Wave in GoldWave

10.3 Final Steps

Finally, you will need to get files to us in Worcester. Uploading them to your WPI web space will probably be the easiest way to do this. We will analyze these signals as quickly as possible and then provide you with some feedback. At that time we'll let you know if anything further is necessary on your end.

11 Appendix B: MATLAB Code

11.1 Column Generator (*columnngen.m*)

```
function p_ave = columnngen(y)

% columnngen.m - by Jon Ross
% -----
% Takes an input sound clip y with a length of 40000 samples.
% Outputs a 10x1 column vector with the neural network vector
% for that clip.

index=1;

for row = 1:8;
    nnmat = fdanalysis11(y(index:index+4998));
    index = index + 4999;
    nnm(row,:) = nnmat;
end

p_ave=mean(nnm);
p_ave=p_ave';
```

11.2 Frequency Domain Analysis (fdanalysis11.m)

```
function nnmat = fdanalysis11(y)

% fdanalysis11.m - by Jon Ross
% -----
% Takes an input vector y with a length equal to the buffer size.
% Outputs a vector describing relevant features of the input signal.

bufferlen = 4999;           % Size of the input buffer (should be odd)
Y = y';                    % Transpose y
Fs = 22050;                % Sampling frequency
T = bufferlen/Fs;         % Length of the buffer in seconds
t = 1/Fs:1/Fs:T;          % Time vector
f = -(Fs/2-1/T/2):1/T:Fs/2-1/T/2; % Frequency vector
lowcutoff = 150;          % Sets the lowest frequency considered (Hz)
g_width = 42.9;           % Gaussian width factor
Enorm = 6.8524e+004;      % Energy to normalize to

if length(y) ~= bufferlen
    error('bad input buffer')
end

g = exp(((g_width*(t-T/2)).^2).*-1./2); % Gaussian Curve defined (center of buffer)
y_windowed = g.*y; % Windowing
Y = abs(fft(y_windowed)); % Fast Fourier Transform
Y = fftshift(Y); % shifts the spectrum to center at zero
Yt = Y;
Yt(1:(bufferlen/2+0.5+(round(lowcutoff*T)))) = 0; % Get rid of negative and low frequency trash

E = 0; % Initialize Energy
for i = 1:bufferlen
    E = E + Yt(i)^2; % Sums the energy for the whole signal
end
Yt = sqrt(Yt.^2 .* (Enorm/E));
```

```

p = bufferlen/2+0.5; % Initialize pointer to the first positive frequency value
for i = 1:8
    nnmat(i) = 0;
    while p < (500*i*T + (bufferlen/2+0.5))
        nnmat(i) = nnmat(i) + Yt(p)^2; % Step through the frequency spectrum calculating energy
        p = p + 1; % for each 500 Hz segment
    end
end

nnmat(9) = bufferlen/2+0.5; % Initialize a pointer to 0 Hz.
Eh = 0;
while Eh < Enorm/2 % Step up until central frequency is found
    Eh = Eh + Yt(nnmat(9))^2;
    nnmat(9) = nnmat(9) + 1;
end
nnmat(9) = (nnmat(9) - (bufferlen/2+0.5)) / T; % Convert to Hz and output to vector

sY = smoothts(Yt, 'b', 200); % Smoothing applied
[maxs,imax] = max(sY); % Find the max of the smoothed function
nnmat(10) = (imax - (bufferlen/2+0.5)) / T; % Convert to Hz and output to vector

% plot(f,Yt,f,sY);
% axis([0 4000 0 max(Yt)+max(Yt)/7]);
% pause

```

11.3 Identification Function (*identify.m*)

```
function P = identify(net,y)

% identify.m - by Jon Ross
% -----
% Takes an input vector y with a length of 40000 samples.
% Displays the most probable type of boat and outputs a vector with the
% neural network outputs in the format: [pTaxi; pPersonal; pCargo].

index=1;

for row = 1:8;
    nnmat = fdanalysis11(y(index:index+4998));
    index = index + 4999;
    nnm(row,:) = nnmat;
end

nnmat=mean(nnm);
nnmat=nnmat';

P = sim(net,nnmat);

if P(1) > 0.7 && P(1) < 1.2
    disp('Taxi: ');
    disp(P(1));
elseif P(2) > 0.7 && P(2) < 1.2
    disp('Personal :');
    disp(P(2));
elseif P(3) > 0.7 && P(3) < 1.2
    disp('Cargo :');
    disp(P(3));
else
    disp('Cannot be identified');
end
```


11.4 Mass Identification Function (massid.m)

```
% massid.m - by Jon Ross
% -----
% Given a neural network in the variable "net",
% quickly identifies every boat sample using that
% network.

r=0;
w=0;
u=0;
disp('Cargo:');
disp('-----');

for i = 1:9;
P = sim(net,cargomaster(:,i));
if P(1) > 0.5 && P(1) < 1.2
    disp('Taxi: ');
    disp(P(1));
    w=w+1;
elseif P(2) > 0.5 && P(2) < 1.2
    disp('Personal :');
    disp(P(2));
    w=w+1;
elseif P(3) > 0.5 && P(3) < 1.2
    disp('Cargo :');
    disp(P(3));
    r=r+1;
else
    disp('Cannot be identified');
    disp(' ');
    u=u+1;
end
end
disp('Right:');
```

```

disp(r);
disp('Wrong:');
disp(w);
disp('Unidentifiable');
disp(u);
pause

r=0;
w=0;
u=0;
disp('Personal:');
disp('-----');

for i = 1:10;
P = sim(net,persmaster(:,i));
if P(1) > 0.5 && P(1) < 1.2
    disp('Taxi: ');
    disp(P(1));
    w=w+1;
elseif P(2) > 0.5 && P(2) < 1.2
    disp('Personal :');
    disp(P(2));
    r=r+1;
elseif P(3) > 0.5 && P(3) < 1.2
    disp('Cargo :');
    disp(P(3));
    w=w+1;
else
    disp('Cannot be identified');
    disp(' ');
    u=u+1;
end
end
disp('Right:');
disp(r);
disp('Wrong:');
disp(w);

```

```

disp('Unidentifiable');
disp(u);
pause

r=0;
w=0;
u=0;
disp('Taxi:');
disp('-----');

for i = 1:9;
P = sim(net,taximaster(:,i));
if P(1) > 0.5 && P(1) < 1.2
    disp('Taxi: ');
    disp(P(1));
    r=r+1;
elseif P(2) > 0.5 && P(2) < 1.2
    disp('Personal :');
    disp(P(2));
    w=w+1;
elseif P(3) > 0.5 && P(3) < 1.2
    disp('Cargo :');
    disp(P(3));
    w=w+1;
else
    disp('Cannot be identified');
    disp(' ');
    u=u+1;
end
end
disp('Right:');
disp(r);
disp('Wrong:');
disp(w);
disp('Unidentifiable');
disp(u);

```

Works Cited

- Aquarian Hydrophone System. 2005. Great Lakes Marine Specialties. Oct 2006. <http://www.navstore.com/aquarian_hydrophone.asp>.
- Babic, Kristopher, et al. *Analysis of Sewer Holes and Canal Wall Damage in Venice, Italy*. Worcester Polytechnic Institute. 2003. <http://www.wpi.edu/Academics/Depts/IGSD/Projects/Venice/Center/Projects/IQP_public/E98/Sewers/E98_Report-Sewers.pdf>.
- Bruel and kjaer. Sound and vibration measurement A/S. 1998. Online. Available: <http://www.bksv.com/lectures/BA767612.pdf>
- Carrera, Fabio. “[City Knowledge as key to understanding the relation between waters and stones in Venice](#)” in C.A. Fletcher and T. Spencer, eds., *Flooding and Environmental Challenges for Venice and its Lagoon: A State of Knowledge 2003*. Cambridge, UK: Cambridge University Press, 2005.
- Chiu, David, et al. *The Moto Ondoso Index: Assessing the Effects of Boat Traffic in the Canals of Venice*. Worcester Polytechnic Institute. 2002. <http://www.wpi.edu/Academics/Depts/IGSD/Projects/Venice/Center/Projects/IQP_public/E02/MotoOndoso/E02_Report-MotoOndoso.pdf>.
- Civitas-Mobilis. Oct 2006. Civitas. <http://www.civitas-initiative.org/project_sheet?lan=en&id=2>.
- DolphinEAR Hydrophones. Oct 2006. DolphinEAR. <<http://www.dolphinear.com/>>.
- Duffy, Jill, et al. *Reengineering the City of Venice’s Cargo system for the Consorzio Transportatori Veneziani Riuniti*. Worcester Polytechnic Institute. 2003. <http://www.wpi.edu/Academics/Depts/IGSD/Projects/Venice/Center/Projects/IQP_public/E01/Cargo/>.
- Dysarthric Speech Database. Oct 2006. Speech Research Labs. <<http://www.asel.udel.edu/speech/Research-dys.html>>.
- Fulop, Sean A., and Fitz, Kelly. “A Spectrogram for the Twenty-First Century.” *Acoustics Today*, July 2006: 26-32.
- Hydrophone Product Range. Oct 2006. Underwater Acoustic Solutions. <<http://www.reson.com/sw233.asp>>.
- Hydrophone Systems and Accessories. Oct 2006. Aquarian Audio products. <<http://www.aquarianaudio.com/products.htm>>.

Lathi, B.P. *Signal Processing & Linear Systems*. New York: Oxford University Press, 1998.

Mill, Robert. Auditory-Based Time-Frequency Representations and feature Extraction Techniques for Sonar Processing. 2005. Online. Available: <http://www.dcs.shef.ac.uk/intranet/research/resmes/CS0512.pdf>

N.A. N.D. Tutorial on speech recognition. Newcastle education. Online. Available: http://murray.newcastle.edu.au/users/staff/speech/home_pages/tutorial_sr.html

Nissen, Steffen. Neural Network Theory. 2003
<http://fann.sourceforge.net/report/node4.html>

Smaragdīs, Paris and Raj, Bhiksha. “Sound recognition.” Oct 2006. Mitsubishi Electric Research Laboratories. <<http://www.merl.com/projects/SoundRecognition/>>.

Thio, Tzer. 2001. Development of Inexpensive Acquisition and diagnostic technique for piston-engine aircraft. Online. Available: http://sun.library.msstate.edu/ETD-db/theses/available/etd-02072001-140439/restricted/Thesis_Report.pdf

Wagner, David. “A Speech Recognition Project.” 1999. Dartmouth college. Oct 2006. <<http://www.cs.dartmouth.edu/~dwagn/aiproj/speech.html>>.

Watkins, Steven. Advanced Materials and Smart Sensors for Civil Infrastructures. 1999. <http://campus.umar.edu/smartengineering/EducationalResources/Neural_Networks_Lab.pdf>

Wikipedia. “Hydrophone.” Wikipedia.com 2006.
<<http://en.wikipedia.org/wiki/Hydrophone>>.

Wikipedia. “Sustainable transportation” Wikipedia.com 2006.
<http://en.wikipedia.org/wiki/Sustainable_transportation>.

Wikipedia. “Magnetic Amplifier” Wikipedia.com 2006.
http://en.wikipedia.org/wiki/Magnetic_amplifier

Carrera, Fabio. “[City Knowledge as key to understanding the relation between waters and stones in Venice](#)” in C.A. Fletcher and T. Spencer, eds., *Flooding and Environmental Challenges for Venice and its Lagoon: A State of Knowledge 2003*. Cambridge, UK: Cambridge University Press, 2005.

Wikipedia. “Artificial Neural Networking” Wikipedia.com 2006.
<http://en.wikipedia.org/wiki/Artificial_neural_network>

Woo, Joeseph. "A short History of the development of Ultrasound in Obstetrics and Gynecology." 2002. Sclinks from NSTA. Oct 2006. <<http://www.ob-ultrasound.net/history1.html>>.

Y. Michael Jiang, City College of New York, Feb. 4, 2001, <<http://www-cs.ccny.cuny.edu/~csolb/csc59944/xor.htm>>