

Project Number: EE-PCP-0202

Ultrasound Boat-Monitoring System

Venice Canal Boat Traffic and Damage Monitoring System

A Major Qualifying Project Report:

submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Mark Daniel Johnson

Jonathan Lovisolo

Yasuhiro Okuno

Date: April 24, 2003

Approved:

Professor Peder C. Pedersen, Major Advisor

Professor Fabio Carrera, Co-Advisor

- 1. ultrasound**
- 2. Venice**
- 3. boat monitor**

Table of Contents

Table of Contents.....	2
Table of Figures.....	4
Abstract.....	6
Acknowledgements.....	7
1. Introduction.....	8
2. Background Information.....	10
2.1 Canals of Venice.....	10
2.2 Current Situation.....	12
2.3 Past Projects: Analysis of Sewer Holes & Canal Wall Damage in Venice, Italy....	13
2.4 Past Projects: Remote Monitoring System.....	16
2.5 Existing Systems: Waterway Expert Traffic System.....	17
2.6 Project Goals.....	18
3. Technical Background.....	20
3.1 Ultrasound.....	20
3.1.1 Basic Properties.....	20
3.2.2 Relevant Aspects of Pulse-Echo Ultrasound Systems.....	22
3.2 Signal Processing.....	24
3.2.1 Signal Processing Overview.....	25
3.2.2 Correlation Definition.....	26
3.3 Embedded System and Data Storage.....	27
3.3.1 Hardware Configuration.....	29
3.3.2 Data Accessibility.....	30
3.3.3 Software Configuration.....	31
3.3.4 Digitizer Availability.....	34
3.3.5 Operating Environment Considerations.....	36
3.3.6 Database Format.....	36
4. Methodology.....	39
4.1 Development Process.....	40
4.2 Ultrasound.....	42
4.2.1 Components.....	42
4.2.2 Signal Multiplexing.....	45
4.2.3 Testing.....	47
4.2 Signal Processing.....	48
4.2.1 Overall System Design.....	49
4.2.2 Receiving an Echo.....	49
4.2.3 The Fourier Transform Algorithm.....	52
4.2.4 Measuring the Signal Amplitude Using the Analytical Signal Magnitude.....	57
4.2.5 Compensating for Time Shift.....	59
4.2.6 Measuring Arrival Time.....	63
4.2.7 Measuring Signal Width.....	63
4.2.8 False Positives.....	65
4.2.9 False Negatives.....	66
4.2.10 Exporting Data in the MatLab Environment.....	67
4.3 Boat Detection and Data Logging.....	67

4.3.1 Basic Boat Detection	67
4.3.2 Boat Velocity and Length Calculation	68
4.3.3 Special Cases	70
5. Testing and Results	71
5.1 Transducer Tolerance Testing	71
5.2 Lab Tank Testing	79
5.3 Second Generation Development and Testing	83
6. Future Implementation	84
7. Conclusions	86
Appendix A: Software Operation Manual	87
A.1 Current System Capabilities	87
A.2 Implementation Details	87
A.3 Known Issues	91
A.4 Possible Improvements	93
A.5 Conclusion	94

Table of Figures

Figure 1: Traditional Gondola in a Venetian Canal	8
Figure 2: Map of Central Venice	10
Figure 3: A Venetian Ambulance	11
Figure 4: Canal Wall Damage and Repair Work	12
Figure 5: Base MOI for Regions Studied During the Summer	14
Figure 6: A Parked Venetian Taxi	15
Figure 7: Ultrasound System and Wave Reflection	21
Figure 8: Wave Reflections off Different Surfaces	24
Figure 9: Signal Processing Block Diagram	26
Figure 10: Example of Correlation	27
Figure 11: System Block Diagram	39
Figure 12: Embedded System Development Diagram	41
Figure 13: Multiplexer Schematic	45
Figure 14: System Synchronization Timeline	46
Figure 15: Extraction Parameters	49
Figure 16: Analog Echo Waveform	50
Figure 17: Digital Echo Waveform	51
Figure 18: Known 6 Hz Echo	52
Figure 19: Sample Echo	53
Figure 20: FFT of Echoes	53
Figure 21: Result of FFTs multiplied	54
Figure 22: Correlation Magnitude Function	54
Figure 23: Sample echo at 12 Hz	55
Figure 24: Result of Multiplication of FFTs	55
Figure 25: Correlation Magnitude Function of Different Frequency Echoes	56
Figure 26: Analytical Signal FFT	57
Figure 27: Final Envelope Output	58
Figure 28: Final Envelope Output for a 12 Hz Sample	59
Figure 29: Two Echoes Received Consecutively with Equal Offsets	60
Figure 30: Different Points on a Boat at Different Distances	61
Figure 31: Two Echoes Received Consecutively with Different Offsets	62
Figure 32: Measuring Arrival Time	63
Figure 33: Signal Width Echo	64
Figure 34: Signal Width Rectified Signal	64
Figure 35: Signal Minus 85%	65
Figure 36: Boat Traffic Information Measurement	69
Figure 37: Experimental Setup - Surface Fixture	72
Figure 38: 1.0 MHz, 1 inch Diameter Using Fine Grain	73
Figure 39: 1.0 MHz, 1 inch Diameter Using Medium Grain	74
Figure 40: 1.0 MHz, 1 inch Diameter Using Rough Grain	74
Figure 41: 2.25 MHz, 1 inch Diameter Using Fine Grain	75
Figure 42: 2.25 MHz, 1 inch Diameter Using Medium Grain	76
Figure 43: 2.25 MHz, 1 inch Diameter Using Rough Grain	76
Figure 44: 5.0 MHz, 0.25 inch Diameter Using Fine Grain	77
Figure 45: 5.0 MHz, 0.25 inch Diameter Using Medium Grain	78

Figure 46: 5.0 MHz, 0.25 inch Diameter Using Rough Grain	78
Figure 47: Test Setup Using X-Y Plotter	80
Figure 48: Sample Arrival Time	81
Figure 49: Sample Correlation Amplitude	81
Figure 50: Sample Log File	82
Figure 51: Future Improvements	85

Abstract

The city of Venice relies heavily on canals and boats to provide transportation within the city. It is suspected that there is a correlation between this boat traffic and damage to the canal walls. There is interest in studying this relationship and determining the extent of the correlation. Our system is designed to use ultrasound to collect data on this boat traffic, such as boat type, velocity, wake and other information that can be used to explore this relationship.

Acknowledgements

On behalf of the Venice Boat Monitoring Team, we would first like to thank our two advisors, Professor Peder Pedersen and Professor Fabio Carrera. Without their unwavering patience, guidance, support and administration, the goals we accomplished wouldn't have been possible. We would also like to thank Mr. Tom Angelotti and the ECE Department, for supporting us with our hardware needs. Our final gratitude go to the many professors of the Worcester Polytechnic Institute Electrical Engineering Department, for providing us with 4 years of dedication, enthusiasm, and instruction, whose tireless efforts provided the solid foundation for this project, and for all future challenges that lay before us in our professional career.

Thank you.

1. Introduction

Venice, unlike almost all other cities in the world, relies on the use of canals and boats to provide its residents with transportation. In the days when the gondolas¹ (see Figure 1) and other human-powered craft were the mainstay of this transportation, Venice was known as “the silent city.” However, as technology advanced, motorboats replaced the gondolas and the noise rose. In addition to increased noise levels, the wakes produced in the canals by these motorboats have been a cause of serious destruction of canal walls.



Figure 1: Traditional Gondola in a Venetian Canal

The wakes, left unchecked, have caused serious damage to the canal walls causing the erosion and collapse of some. Currently, there is no efficient way to monitor the boat traffic. This problem was recognized by Venice officials, and brought to the attention of Prof. Fabio Carrera, who is the director of the Worcester Polytechnic Institute Venice Project Center. A project idea was formed to develop a device to help Venetian officials monitor the boat traffic in Venice.

The goal of this project is to develop a system capable of measuring the frequency of boats that pass through a given canal, their type, the height of the wake the boat produced, and the time the boat passed. This project will work in three phases. The

¹ This photograph was taken from Durant Imboden’s Venice for Visitors website located at: http://europeforvisitors.com/venice/articles/gallivanting_by_gondola.htm

first phase will involve developing the system in a controlled tank on the WPI campus. This phase will develop the fundamental theory behind the applications, the methods for determining parameters, such as transducer placement, and a prototype system. The second phase will develop a working model of the system on a laptop, to be tested in the canals. The final system will be an embedded standalone system for final deployment in the canals of Venice. This system will provide Venetian officials with an effective and reasonable way to monitor and process this important data.

2. Background Information

Before technical research began, investigation into the current situation and past projects was performed. This was done to help achieve a better understanding of the problem and to avoid any duplicate efforts. The following chapter details this work.

2.1 Canals of Venice

The city of Venice is located in northeastern Italy, in the north of the Adriatic Sea. It is composed of over 100 different islands in the Lagoon of Venice. The only form of travel, besides walking, is by boat on canals that run between the islands. There are over 170 different Venetian canals, the biggest being The Grand Canal, which snakes its way through the center of the city (see map below²).



Figure 2: Map of Central Venice

² Taken from Lonely Planet at <http://www.lonelyplanet.com/mapshells/europe/venice/venice.htm>

The original Venetian boat is the gondola, however most Venetian residents use motorized waterbuses, called vaporetti, or their own private boats. There are four main types of boats: taxis, cargo boats, pleasure boats, and big taxis. Below is a picture of a Venetian ambulance, similar to a cargo boat. The wake generated by the boat is visible in the bottom left corner of the picture. A boat of this magnitude is considerably noisier and creates more wake than a human powered gondola.



Figure 3: A Venetian Ambulance³

The canals are constructed of two different types of materials, depending on location. In the wealthy sections of the city, the canal walls are made of a limestone called Istria, named after the location the stone is quarried from. Underneath this limestone is wood piling, supporting the wall in the earth. The top portion is made of brick. In the poor sections of the city, the walls are made of all brick, because it is cheaper. The walls are still supported on the same wood piling. The canals are subjected

³ This photograph was taken from David Goldsmith's homepage at <http://www.dgolds.com/photos/Italy/TheBoatsOfVenice.htm>

to tidal activity, with a variance of ± 1 meter relative to normal sea level. The average current in the canals is approximately 30cm/s.

2.2 Current Situation

The current problem with the canal system in Venice has to do with the damage caused by boat traffic. Each boat moving in a canal causes disturbances in the water. These disturbances are energy in the form of visible wakes or invisible underwater turbulence. This energy can cause damage to the walls of the canals, necessitating repair on a regular basis. This repair is important, as the walls support many Venetian buildings. If they are not cared for,



Figure 4: Canal Wall Damage and Repair Work

serious wall decay, eventually leading to collapse, can occur. This picture is an example of canal walls being damaged, and the repair process to fix them. Here, a dam was put in place to dredge this particular canal.⁴

The only method available to monitor boat traffic is by counting boats manually. This involves having a person wait on the side of the canal, counting and classifying the boats as they pass. This is not a very effective procedure, as it is hard to watch a large number of canals for an extended period of time.

⁴This photograph was taken from Durant Imboden's Venice for Visitors website located at: http://europeforvisitors.com/venice/galleries/blg-v_canal_maintenance_cofferdam_by_fondamenta_della_misericordia_cannaregio.htm

The current mayor of Venice is in charge of the canal preservation project. This includes determining the cause of damage to the canal walls and implementing measures to minimize this damage. This will in turn reduce the amount of repairs that the canal walls need, as well as preserve the buildings that rest on them. Our project will help in collecting info in order to locate and manage the sources of this damage.

2.3 Past Projects: Analysis of Sewer Holes & Canal Wall Damage in Venice, Italy

Under the advisory of Fabio Carrera, Natalie Mello, and David DiBiasio, WPI students David Chiu, Annand Jagannath, and Emily Nodine completed this Interdisciplinary Qualifying Project during the summer of 2002. This project worked to collect information on boat traffic and relate it to *moto ondosso*, which roughly translates to “wake impact”. They developed a *Moto Ondoso Index* (MOI) based on the energy released into the canal water from the wake of the boat. A greater MOI correlates to more damage being caused. The following table was taken from their report:

Segment	Boat Code	Payload	Summer MOI at 4 km/h	Winter MOI at 4 km/h	Summer MOI at 5km/h	Winter MOI at 5 km/h	Summer MOI at 20 km/h	Winter MOI at 20 km/h	Summer MOI at Avg km/h	Winter MOI at Avg km/h
ALBO	1	High	3.2	2.6	6.0	4.9	100.9	82.2	41.4	33.7
ALBO	1	Low	63.0	51.3	85.4	69.5	625.4	509.2	326.9	266.2
ALBO	1	Medium	10.1	8.2	13.9	11.3	97.6	79.5	29.8	24.3
ALBO	2	High	1.1	0.9	3.0	2.5	204.2	166.3	204.2	166.3
ALBO	2	Low	5.7	4.7	15.2	12.4	1021.2	831.6	968.8	788.9
ALBO	2	Medium	1.0	0.8	2.6	2.1	175.1	142.6	175.1	142.6

Table 1: Moto Ondoso Index

In this table, boat codes 1 and 2 refer to large and small cargo boats respectively, the payload column represents the cargo amounts in the boats measured, and the rest of the numbers represent the calculated MOI for different parameters. The MOI was

calculated using the energy released and the wake height. ALBO refers to Rio de l'Aboro, the region of Venice where this data was taken. The MOI index was calculated for a number of regions in Venice. The team combined this data, and produced the following map:

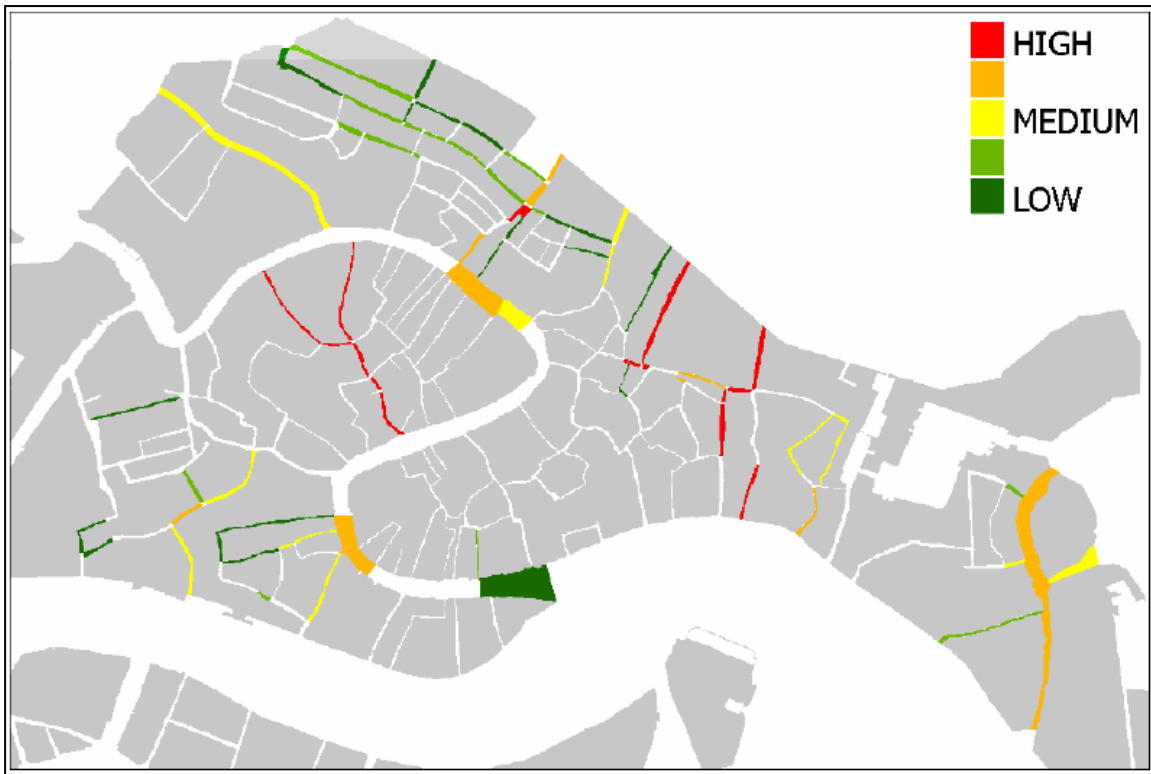


Figure 5: Base MOI for Regions Studied During the Summer

The color of the section correlates to the level of the MOI in that area. They divided the index into two seasons, winter and summer, because of the great fluctuations in traffic caused by tourism. The project gathered data about past traffic trends and projected traffic trends for the year 2007. When they applied the MOI calculation to this, they concluded that the MOI would continue to increase unless something is done. This will cause further canal damage.

They made a number of recommendations to help improve this. First, they concluded that excessive speeding was the primary cause of the wall deterioration. Although speed limits were implemented in early 2002, only 3% of the motorists obeyed them⁵. Reducing damage meant enforcing stricter speed limits.

Another recommendation they made involved a cargo warehouse proposal. This cargo warehouse would serve as a central distribution center, eliminating 90% of cargo boat traffic, which would in turn reduce the total MOI of all boats in Venice by 68%. This would be a major reduction in canal damage.

Taxis had a very high MOI rating. The group recommended looking at ways to improve the efficiency of taxi routes using Global Positioning Satellite (GPS) and Geological Information System (GIS) to coordinate the shortest routes, routes with the least traffic to reduce congestion, and the fastest routes. In addition, taxis have restricted areas that are not adhered to. Enforcing this restriction would reduce the number of boats in the inner canals, reducing the MOI.



Figure 6: A Parked Venetian Taxi

The final recommendation they made to help reduce canal wall destruction was to improve traffic data. The traffic data they used in their report was incomplete, had gaps,

⁵ Moto Ondoso project

and did not cover enough of Venice. This sets a strong foundation for this boat-monitoring project.

2.4 Past Projects: Remote Monitoring System

Another project done by WPI students in the past has been an attempt to create an automated traffic monitoring and wake data gathering system. The system was to monitor traffic at different sections of the canals, and measure the amount and level of wake generated by that traffic. Their systems comprised of three devices; a boat identification device, a boat-monitoring device and a wake-measuring device. The boat identification device used in this project made use of a radio signaling technology similar to the highway toll collection systems such as *EZ-Pass* and *FastLane*. The wake measuring device used a pressure sensor in a cylinder immersed into the water. The boat-monitoring device and the wake measuring device were to be installed together in different sections of the canals, and the boat identification devices were to be installed in individual boats in Venice. When a boat with the identification device installed passed through a section of the canal with the monitoring device, the monitoring device recognized the boat and measured the wake created. The collected information was recorded and made accessible for later analysis. The system was completed and functional. However, the city of Venice did not implement it for two reasons: installation difficulty and logistics of the boat identification device. The installation difficulty was on the calibration of the wake measurement system. The wake measurement system required some technical knowledge to install and configure at each installation location before it could gather any useful information. Because the boat-monitoring device was self-contained, installation was easy. However, the city of Venice decided they could not

force the device to be installed on boats because it could potentially be used to track movement and issue penalty fines. This would cause a problem with the boat owners.

Currently, Venice has several stations in their canals that could house different equipment for measuring and monitoring the canal usage and its conditions. The stations are currently only equipped with a generic device that continuously monitors the water levels, but there is no system to monitor the traffic at the location.

2.5 Existing Systems: Waterway Expert Traffic System

Research was done on existing designs of similar products in the market. Although the project team did not expect many findings from a market limited to canal boat-monitoring systems, the research has turned out one very similar product accomplishing the same goal as our project. The project was sponsored by Ocean Systems Development Corporation and was done by the Oceanographic Center at Nova Southeastern University. The aim of the project, named Waterway Expert Traffic System (WETS), was to develop and test new methods to monitor watercraft usage and Sea State in an urban canal. The system is capable of calculating speed, heading and distance of the boats and the calculated data is made available via the web along with the wakes and other monitored data. The monitor unit uses laser traffic counters and several wave sensors to obtain information on traffic. An array of monitoring units is used to obtain information for processing the information regarding the boat's speed, heading, and distance in a real-time manner. The data is recorded and made available through a website. Through the website, it is possible to query the database to look for certain boat traffic meeting different criterion as well as obtain statistical information of overall traffic. A CD-ROM report of the project is made available for free by the research institute and

may become a great resource for the project although the project aims to use ultrasound technologies instead.

Although the WETS system provides much of the basic capabilities of our system, there is some lack of functionality for the use in Venice. WETS does not classify the boats automatically, which makes things harder for the researchers to analyze the obtained information. Also, some information such as pressures exerted to canal walls by the boat traffic is not measured or collected in WETS.

2.6 Project Goals

The goal of this project is to develop a system to monitor boat traffic and the wakes generated. The system will log all boats passing a given point with time-stamps and record their type, speed, and the level of wake they leave behind. The data gathered will then be organized into one central database. This data will then be used to assess the sources of destructive wakes and formulate methods to minimize the destruction.

The system will use pulse-echo ultrasound measurements from two ultrasonic transducers to gather information about the boats passing through the canals. A signal-processing unit will interpret this data, and use it to determine the specific characteristics of the traffic at each point of the canal, and to quantify the character of the traffic. This includes type and size of the boat, speed of the boat, wake generated by the boat, and energy discharged to the canal wall by the boat. The signal processing unit is also responsible for rejecting any false echoes from debris and other false signals obtained through the ultrasound receiver. The information gathered will be transferred to a data

collection unit. This will organize the data into a database accessible to the officials in charge of maintaining the canals in the form of a database viewable through a computer.

Upon completion, the system created by this project will be useful for analyzing the traffic in the Venetian canals. This will greatly increase the information gathered about the canals. As mentioned earlier, the canals of Venice are now under direct control of the mayor, who will be in charge of taking any means necessary to improve the conditions of the canals. To assess the most effective measure to improve the canal's condition, and to reduce damage, the mayor and his committee will require a considerable amount of information. By installing the system created by this project, a lot of this data collection process will be automated. This will allow for a larger collection of more complete information. This will in turn help reduce canal damage.

3. Technical Background

In the technical background section, the implications of designing the system will be discussed in three different sections. Each section will discuss one of the three major technologies being used in the system design and implications regarding the implementation of the technology in this application.

3.1 Ultrasound

The measurement technology we are using in this project is pulse-echo ultrasound. Ultrasound consists of high frequency sound waves that are reflected off of objects in order to get information about those objects, such as range information.

3.1.1 Basic Properties

Ultrasound frequencies consist of frequencies greater than 20 kHz. The device used to transmit and receive ultrasound signals is called a transducer. A transducer is a passive piezoelectric device. Whenever the transducer disk is subjected to stress it creates an electrical signal and whenever an electrical signal is applied to the transducer it causes a corresponding variation in the thickness of the transducer disk. The transducer is usually connected to a pulser/receiver. In order to produce an ultrasound pulse, the pulser/receiver sends an electrical pulse to the transducer. This causes the transducer disk to move, which in turn creates a sound wave pulse. A higher voltage electrical pulse will cause a shorter duration ultrasound pulse. The pulser/receiver then switches to receive mode where it detects electrical signals produced by the transducer when a sound wave impinges on it. The received data is then passed through a signal processor and either

stored or displayed according to the application. This signal processing will be discussed in more detail in the next section.

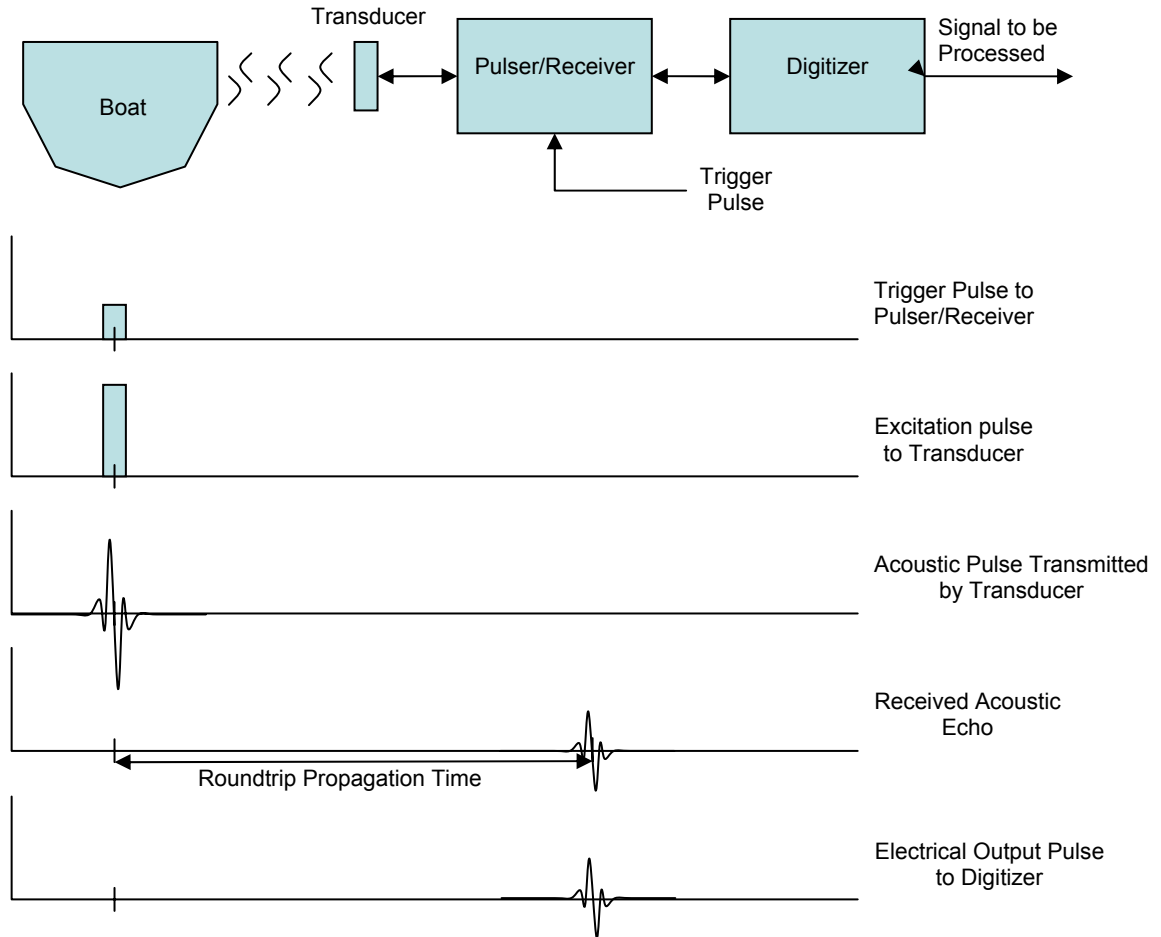


Figure 7: Ultrasound System and Wave Reflection

An ultrasound transducer in receive mode produces electrical signals in response to incoming acoustic signals. These acoustic signals are typically echoes from an acoustic pulse transmitted by the transducer. This is why the system is referred to as a pulse-echo ultrasound system. The acoustic echoes are due to partial reflections of the ultrasound pulse off of different materials. Different materials, such as the hull of a boat or the wall of the canal, will have different acoustic properties. Every material has different acoustic impedance. Any acoustic pulses that reflect off an object and

propagate back to the ultrasound device are received during the receive phase.

Depending on the properties of the materials which the wave has propagated through or reflected off of, parameters such as the amplitude, energy, or phase of the wave may be altered. In addition, the time it takes for a wave to return is dependent on how far it has traveled and the speed of sound in the medium. Taking all of these factors into account, this signal can be processed to determine different kinds of information. In our case we will be looking to determine whether there is an object such as a boat in the canal. In addition to determining the presence of a boat, we want to obtain specific information about the boat including what its length is and how fast it is traveling. The boat monitoring system may also be designed to collect information on the wakes generated by these boats. One way to do this is to set up another ultrasound device to send out pulses to reflect off of the wakes generated by the boat and collect similar information. Another way is to use pressure sensors to measure the lateral movement of the water.

3.2.2 Relevant Aspects of Pulse-Echo Ultrasound Systems

There are a few factors that have to be considered with respect to ultrasound signals though. The first parameter to be determined is the frequency we want to use for the transducers. Testing will be done to determine the highest frequency transducer that can be used that has acceptably small levels of attenuation. We have to be sure that simple propagation across the canal and back through the water will not attenuate the signal too much. The main source of attenuation in our system will most likely be due to scattering. Objects such as bubbles or dirt in the water can attenuate or deflect the signal. Another source of attenuation is beam spreading. It would be preferable if the boats being detected by the system were within range of the near field of the transducer. By

keeping targets within the near field it will minimize beam spreading. In order to do this we will determine a reasonable radius for the transducer using the equation $NF = \frac{a^2}{\lambda}$ where NF is the length of the near field, a is the radius of the transducer, and λ is the wavelength of the transducer. Once we know the frequency we are using for the transducer, we can calculate the wavelength using the equation $f = \frac{c_0}{\lambda}$ where c_0 is the speed of sound in water. One other issue we will have to deal with is the potential for multi-pathing of the signal. We will have to analyze the potential of the wave hitting the top of the water and thus the air and make sure that there will not be a problem. In addition, we have to be sure to design our system in such a way that the reflection of the signals off the boats will not cause the signal echo to miss the transducer and be lost. If too much of the signal reflects away from the ultrasound device we will not get a good return signal. Because the hull of the boats may be rough or smooth and will be angled in different ways, we have to pick an angle for the transducer that will maximize signal return. Smooth surfaces will reflect a wave at an angle equal to the incident angle while rough surfaces will reflect signals at different angles.

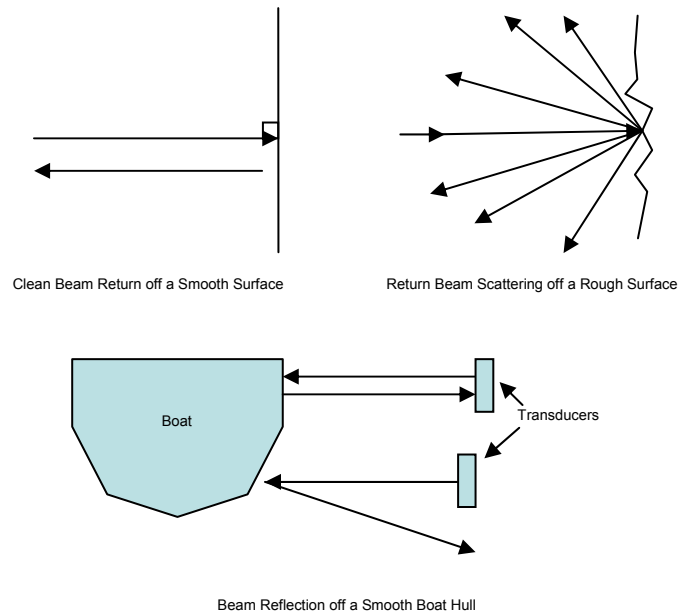


Figure 8: Wave Reflections off Different Surfaces

In order to get maximum signal return, the transducer should be at a normal incidence to the surface being examined. We are currently looking to design our system to send pulses horizontally toward the boat and will test to see if this returns sufficient signal data or if too much of the signal will be reflected away from the transducer.

Keeping in mind all of these factors, our group will design a robust system that will allow us to collect the necessary data as efficiently as possible.

3.2 Signal Processing

The signal processing section will be where almost all of the calculations and work by the system is done. The next sections will outline the responsibilities of the signal processing system, the theory it will rely on, and the goals we wish it to accomplish. There will first be a general overview of the system and what we intend to do with it. Then a treatment of each individual section and related theory will follow.

3.2.1 Signal Processing Overview

The goal of the signal processing unit is to receive information from transducers in the canal, take measurements on this information, and then convert it into useful data. This system can be broken down into a number of smaller modules. The first module in this process receives an echo from the digitizer. Once echoes can be received, the signal processing unit will use correlation to determine if the subject the pulses are bouncing off of is actually a boat. Correlation is important because if a number of pulses received all correlate very highly, then they were reflected off of the same object. The speed of the boat and the height of the wake need to be determined. The signal processing system will be capable of accounting for false positives and negatives. Once all of this information has been gathered, the data must be exported to a user-friendly interface. Finally, testing must occur for each of these individual modules. Each of these modules will be described in more specific detail in the following sections. A block diagram of the system follows:

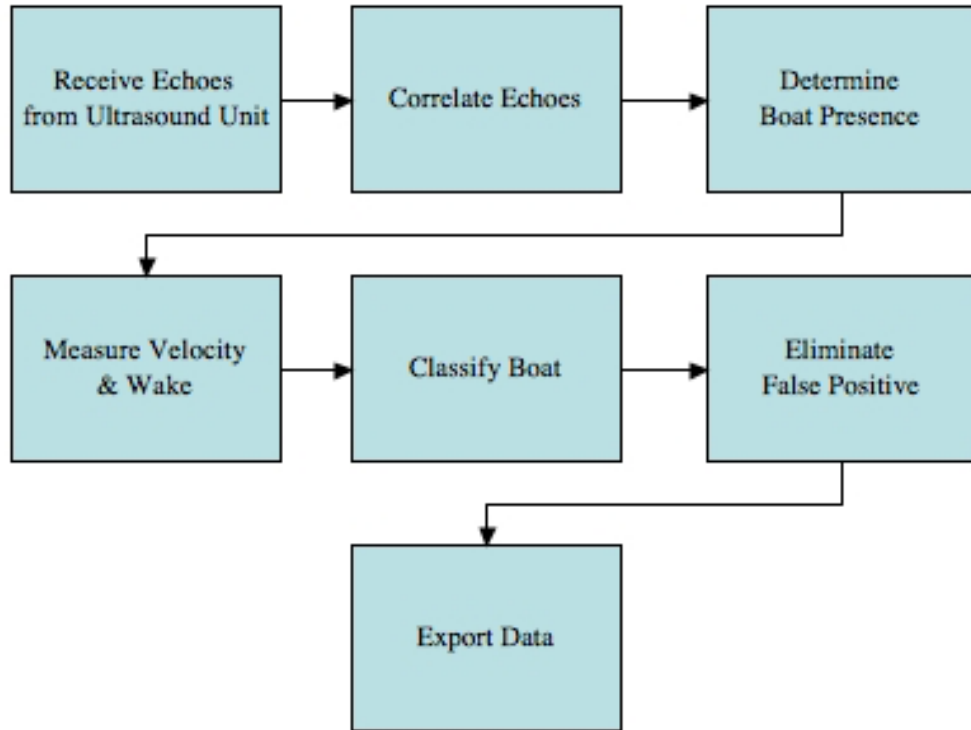


Figure 9: Signal Processing Block Diagram

3.2.2 Correlation Definition

Correlation is a measure of similarity between two objects. Correlation between two signals is a measure of the similarity between the two signals. Correlation is defined by a correlation coefficient, c_n . In order to find the similarity between two signals, the correlation between them would be evaluated. The range of the correlation is:

$$-1 < c_n < 1$$

The correlation of two signals can never exceed unity. If two signals are perfectly the same, then the correlation is at its highest, $|c_n|= 1$. If the two signals are orthogonal, then the correlation between them is 0. The idea of correlation can be seen graphically in the following figure:

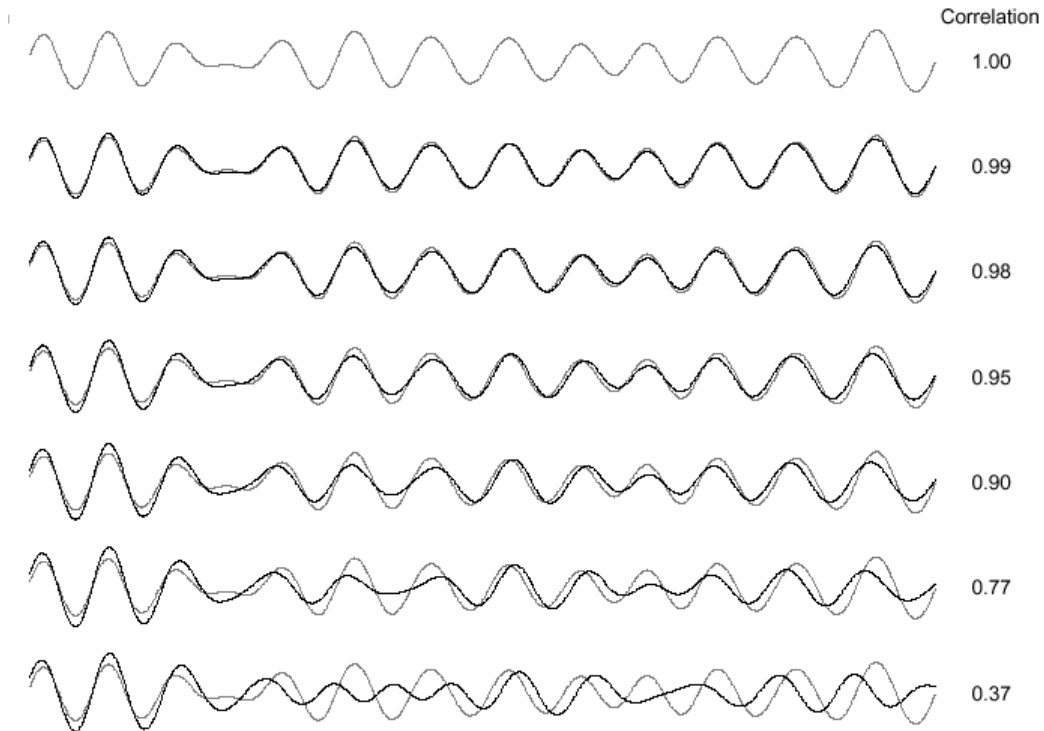


Figure 10: Example of Correlation⁶

When the correlation on the right is 1, the two signals are exactly the same. Hence, when placed on top of each other, it appears as one signal. As the correlation decreases to 0.37, the signals become more and more different, as can be seen in the variances of the two signals on the bottom row.

3.3 Embedded System and Data Storage

The embedded system in this project requires the following attributes: ability to control ultrasound transceivers for data collection; ability to handle the signal processing required to extract the information as stated in the signal processing section; ability to store relevant information generated from the signal processing software/hardware; and

⁶ http://hobbes.ee.virginia.edu/ultra/pdf/SPIE_4325_19.pdf

the ability to make recorded data available to the researchers in a format that is easy to read.

To design this system, the project team could take several different approaches to design the final system. One possible way is to create custom hardware to process the obtained signals and use commercially available data loggers to log the information processed by the unit. Another possible way is to create a signal processing and data logging software that runs on a standardized IBM/PC compatible hardware. The first approach will remove the necessity to create a data logging and access system as long as the data generated by the signal-processing unit complies with the data logger chosen for the project. At the same time though, more effort is required in hardware design of the signal-processing unit. Also, the first approach will limit any possible expandability in the system for future updates. The second approach using a standard IBM/PC compatible hardware requires the project team to design a data collection system and a means to access this data. Such things are possible using readily available software that is generic enough to use without any modifications. Also, this approach will allow additional system components to be added later or different system configurations to be used in different situations. Another advantage to using a standard IBM/PC compatible hardware is that many hardware devices are readily available for different means of data transfer and other utilities.

The following sections will discuss the different aspects of the system that need to be decided when designing the system using an IBM/PC compatible architecture.

3.3.1 Hardware Configuration

The hardware for the IBM/PC compatible system is mostly standardized and is available from many vendors. In this system, a minimum amount of hardware is required to run the operating system used for this project as well as connect to the ultrasound unit.

The following is a list of hardware needed for this system:

- Power Supply
- Mother Board
- Storage Medium
- Digitizer (to receive signals from Transducer unit)
- Some form of communication device (modem/LAN/WAN)
- Removable storage media (for logging data)

The power supply must be able to operate with the power source that is provided in the stations where the system is to be installed. Some things to consider at the final stage of development with regards to the power supply are to consider installing power surge protection or battery backup systems since the power supply to the stations where the system will be installed may not have the most stable power source.

Most IBM/PC compatible motherboards have very similar configurations, with minor differences in supported CPU, Memory, bus technology and size. For the purpose of this project, any modern system configuration will provide all necessary processing power and utility to implement the desired system. The size of the motherboard is also not a major concern, but to create a smaller system it may be desirable to choose a smaller motherboard design such as the micro-ATX standard. The tradeoff to choosing a smaller motherboard is that there may be less space for additional components such as modem, Network Interface Card (NIC) and digitization cards.

Storage mediums will be used to contain the software for the system as well as the operating system to run the system software. Also, a data log may be contained in this storage space. For the purposes of this project, the storage does not need to be large, although the choice of operating systems for the design will greatly affect the minimum required size. Some operating systems can be minimized to run off a few megabytes, while some operating systems require at minimum several hundred megabytes to a gigabyte. The data to be stored will not be very large, and even using some of the more elaborate database systems for logging the data, the information will not be so large as to require a considerably large storage size. Another concern on choosing a storage medium is its reliability. The data stored in this system needs to be safe from any environmental hazards. From this point of view, it may be better to have a backup solid state media such as a flash memory device to reduce the mechanical failure possibilities. Also, to protect the system from power problems such as blackouts and power surges, the system may need some protections such as UPS battery backup to cut off any possible damages to storage media at all times.

Some form of communication device is required to relay the logged data to the researchers. These can be in the form of wired communication, or a form of physical memory device that can be removed and carried. The next section will talk about the implications regarding this subject.

3.3.2 Data Accessibility

It is necessary to develop a method of transferring the logged data from the system. One possible way is via remote communication. Some rudimentary technologies that can be used for this purpose are a modem, an Ethernet, or wireless communication.

All of these technologies are readily available for the IBM/PC architecture, although some may require more infrastructure than others. Using a modem only requires a phone line and is the slowest of all three communications. The major advantage is that it requires the least advanced infrastructure to use. Ethernet or wired internet connection would allow for more robust communication between the monitoring units and the research center, but will require more wiring to be done and is more expensive to create the infrastructure required. A wireless network of any sort will require connection stations that allow connection to a network. Although there is no need for wiring in the stations, connection points are required near the stations to relay the data. Another simple way of transporting the data is to use removable media. This requires no infrastructure to be built around the system for communication other than the engineers who will have to physically replace the media on a regular basis. This could be a simple Flash media, a floppy or other similar alternative device depending on the size of the databases.

For the purpose of this project, no highly advanced networking is required, but a remote access to the data may be a desired feature. A modem system would be a good transfer method in this project since it requires the least infrastructure. Also, it may be wise to keep an alternative option to obtain the data through a removable media.

3.3.3 Software Configuration

There are many different operating systems (OS) available for the IBM/PC compatible architecture, but the most common two types of operating system available are the Microsoft Windows variants and the many UNIX and UNIX-like operating

systems. In this section, the advantages and disadvantages of using a Windows variant OS and a UNIX-like OS are considered.

In using the Microsoft Windows variant OS for an embedded system, the largest advantage is that there is more support for Windows OS on many different hardware products. Many hardware devices created for the IBM/PC compatible system have drivers and software for Windows written by the vendor. This may not directly translate to the software development library availability we may need for developing our software, especially for the signal processing software to use the digitizer inputs. There may be a better chance that these devices work on Windows than on any other OS. The other advantage to choosing a Windows variant is the ease of developing client side software⁷. This software is most likely to be run on Windows, and for this client side software to work with the embedded system, file transfer will be much easier when both systems use the same file system and file types. When using other operating systems, it may be required that the data format and file system format be translated before it can be read from the client Windows environment.

The disadvantage of using Windows is its size and stability of the operating system. The earlier versions of the Windows (95, 98) are smaller in size but are not stable to be run for an extended period of time. Embedded systems need to be able to maintain themselves and stay functional without intervention from any user for extended periods of time. If the OS cannot guarantee stability for an extended period of time when it starts up, it is less desirable for this project. Windows NT and its variants (2000, XP)

⁷ Client side software: software specifically written for the user of the system to read the data obtained from the server side software. Server side software is the embedded system in this case and the users of this system will use the client side software to read any information obtained from the embedded system via network or removable media.

are considerably more stable, but also larger in size (XP requires about 1GB in minimum install, NT versions are much smaller. Embedded variants of Windows NT and XP are also available and are capable of running in smaller disk spaces). This will require that there be a larger storage installed in the system that will increase the cost of the final product. The one last disadvantage to choosing Windows variant is that they are all costly to obtain, and require that each embedded system have its own licensed copy of the operating system.

The advantages to using UNIX-like operating systems are that many are free with source code available to the public domain. Not only is this advantageous for reducing the final product price but the software development may be helped by the fact that the source code is readily available for the development process. Also, by minimization, the UNIX variant OS could be reduced in size to run from a very limited storage space and may help in reducing the size of required storage space for the whole system. The other advantage to the UNIX-like operating system is that it is much more stable compared to the Windows variants. These operating systems are meant to run for an extended period of time and work well for such operations.

The disadvantages to the UNIX-like operating systems is that there is less vendor support for many of the hardware products available for the IBM/PC compatible hardware. This may limit the choice of digitizer cards if a UNIX-like OS is chosen for the project, although the development communities of each UNIX-like OS support many devices. Also, a file system and type conversion may be required in order to properly communicate information from the UNIX-like OS to the client Windows variant operating systems. Although this problem is not a tough problem to solve, since many

other users of UNIX-like OS have done this in the past, it still adds an extra aspect to the system design.

Aside from the operating systems, the final system requires signal processing software. The final system is required to process all the signals obtained from the digitizer and record the information desired by the researchers. The signal processing will be done using some form of a signal processing tool kit in C or C++ such as IT++⁸. The signal obtained from the digitizer will be fed through the signal processing unit written in C/C++ to extract any data. Once the signal processing unit determines that data needs to be recorded, data can be stored using standard system functions to write the data out into a file. The processing power of a modern PC will be more than enough to handle the rejected signals and writing data out to storage. And while there are more useful data being processed, the software can be written to concentrate on processing the signal. This will allow for a real time signal processing on the PC. Also, when there are 100 signals per second (which is enough for our system) a system running at 500MHz will have 5 million CPU cycles to process each signal. Even with the overhead of using an operating system, this is more than enough to process the signal and record necessary data.

3.3.4 Digitizer Availability

Some digitizer cards were found but it cannot be determined which ones are most suitable until the OS and the transducer frequency are chosen for this project. For choosing the digitizer card for the embedded system, the following factors must be considered:

⁸ <http://itpp.sourceforge.net/>

- Supported frequency range and sampling rate ($f_s = 4 \times f_0$ (f_0 is frequency of transmitter))
- Digitization bits (8 bits or more)
- Number of channels (2 to 3 channels in current design)
- Driver Availability (Windows/Linux)
- Software development environment (SDK required for C/C++)

The first thing to note is that the digitizer card must support the desired frequency range after the optimal frequency range desired for this project is determined. Also, depending on the system design, the final product may use multiple channels of transducers to obtain more information such as boat speed, and corresponding wake, and thus may require multiple channels in the system. Installing multiple cards may circumvent this, but since the motherboard will be limited in expansion slots, cards with multiple channels may be desirable.

Driver availability is also important. If the driver is not available for the OS of the embedded system, the card cannot be utilized. This may force the project to use a certain OS for the product. Also, even if the driver is available, if the specifications of the drivers are not given, development of the custom signal processing software for the project application will not be possible in the given time frame, and thus this information has to be somehow obtained if it is not readily available.

The PCI bus version of a National Instruments NI-5102 costs \$1300 and provides the following features:

- 1 GS/s random interleaved sampling
- 15 MHz Digital Oscilloscope for PCI
- 20 MS/s per channel real-time sampling
- 8-bits vertical resolution
- Waveform Memory: 663,000 samples onboard

The card includes two input signals and an external synchronization signal for controlling the Ultrasound Pulsar Receiver. One or maybe two of these cards may be enough to provide the functionality the system requires. The only drawback to this card is the lack of driver support for operating systems other than Windows.

3.3.5 Operating Environment Considerations

In creating the final product for this project, some special cautions must be taken due to the operating environment of the hardware. One most important aspect is that the system is going to be installed in close proximity to seawater. This requires that the hardware be sealed so it is waterproof to a certain extent. Since the product will be installed in a hut, we do not need to worry about rain and other weather hazards, but temperature may have to be considered. Electronic hardware is very sensitive to heat and usually requires a ventilation system to circulate air constantly to keep the system under a certain temperature. Our group cannot expect the product to be installed in an air-conditioned house, so the system must be able to stand up to the temperatures of Venice throughout its operation.

3.3.6 Database Format

For storing the information, the product could use a simple binary file with proprietary format, or use existing database management software available for the PC (such as mySQL, PostgreSQL, MSSQL/Access which complies with Structured Query Language or SQL for data manipulation and transfer).

The advantage of using a proprietary data format (such as CSV comma or tab delimited text file) is that the data size will be kept small. Boat traffic information can be

packed into a data file and may only use several bytes per record. With such data sizes, the embedded system could record continuously for years before the storage device is filled. The obvious disadvantage to using such file format is that it requires a special client program to extract the information to make it readable by the researchers or a training session for the researchers to properly import these files to Excel or other data analysis tools of their choice.

The advantage to using an existing database management system is that the data is easily transferable. The data can be translated to a standard SQL statement that could be transferred over to other SQL compliant databases if designed properly. Also, it helps for developing any sort of analysis tool for the obtained information since much of the querying, data filtering, sorting, and statistical analysis can be done using the software provided by the database management system. Tools such as Microsoft Access could be used to develop an analysis program of this sort with relative ease. The disadvantage to using one of these systems is the fact that there is much more overhead in storing all of the information given, and for the same amount of information, the database will be considerably larger than using a packed proprietary data file approach. The other disadvantage is that, there is no easy way of directly reading the information and requires the development of analysis software.

The choice of the database format will depend on what is required by the product. If storage space is a major concern for the system, it may be unwise to use the SQL based database management system. On the other hand, if storage space allows for using existing database management systems, it may be wise to use such systems as they

supply ample software for transferring and analyzing the obtained data as well as flexibility in the final system design.

4. Methodology

The system was designed to obtain information on boat traffic. The finalized block diagram of the system is shown in Figure 11.

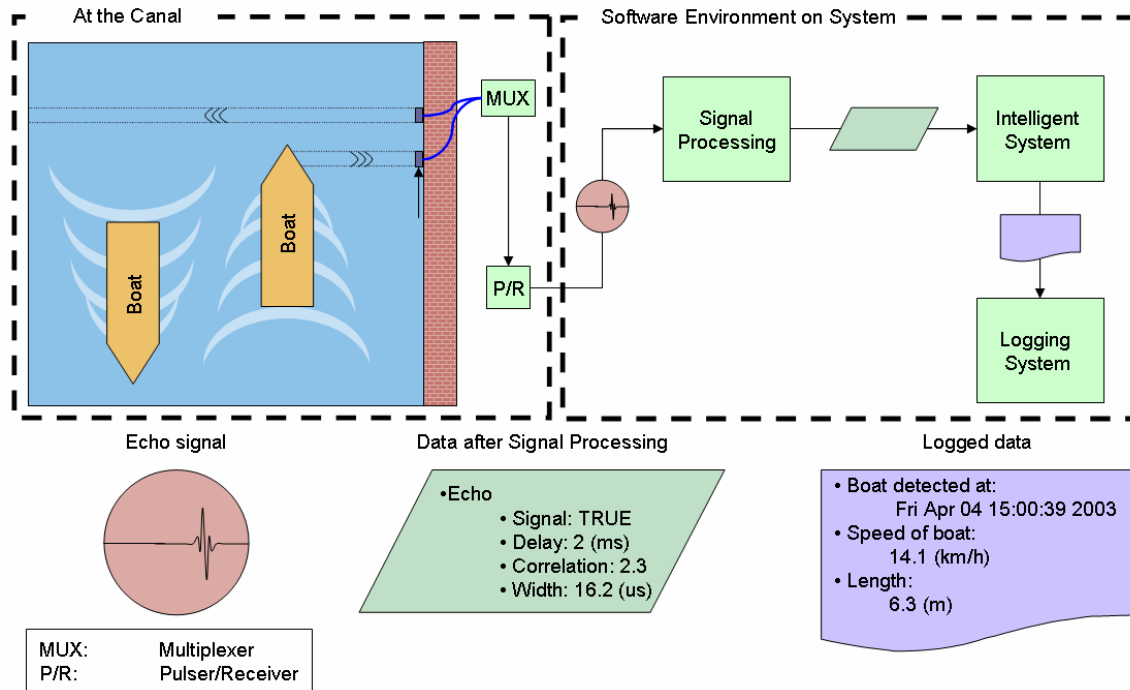


Figure 11: System Block Diagram

The basic flow of the boat detection is as follows:

- The ultrasound pulse-echo system monitors the canal and sends information to the software environment.
- The signal processing software extracts relevant information from the echoes.
- The extracted information is continuously monitored by a boat detection algorithm.
- When the algorithm finds a boat in the canal, the information obtained is logged into a file

In the following sections, the details of the design of the systems will be described. Each module will be described and our group will discuss where it fits into the entire system. Also, the first section of this chapter will be used to briefly go over the development stages setup for this project.

4.1 Development Process

The development of the project was split into three stages (see Figure 12 on next page). The first stage uses the equipment readily available in the WPI laboratory to create a system to prove the concept in a scaled down environment. Once the detection method is solidified in this controlled environment, the system will be ported to a more portable laptop system. The laptop with a digitizer card replaces the LeCroy9400 connected to a PC through a GPIB interface. The second generation system will be used to test the entire system in a real environment with no scaling. Finally, after the system is tested and is ready for deployment in the canals of Venice, it will be integrated into a standalone embedded system for deployment in Venice.

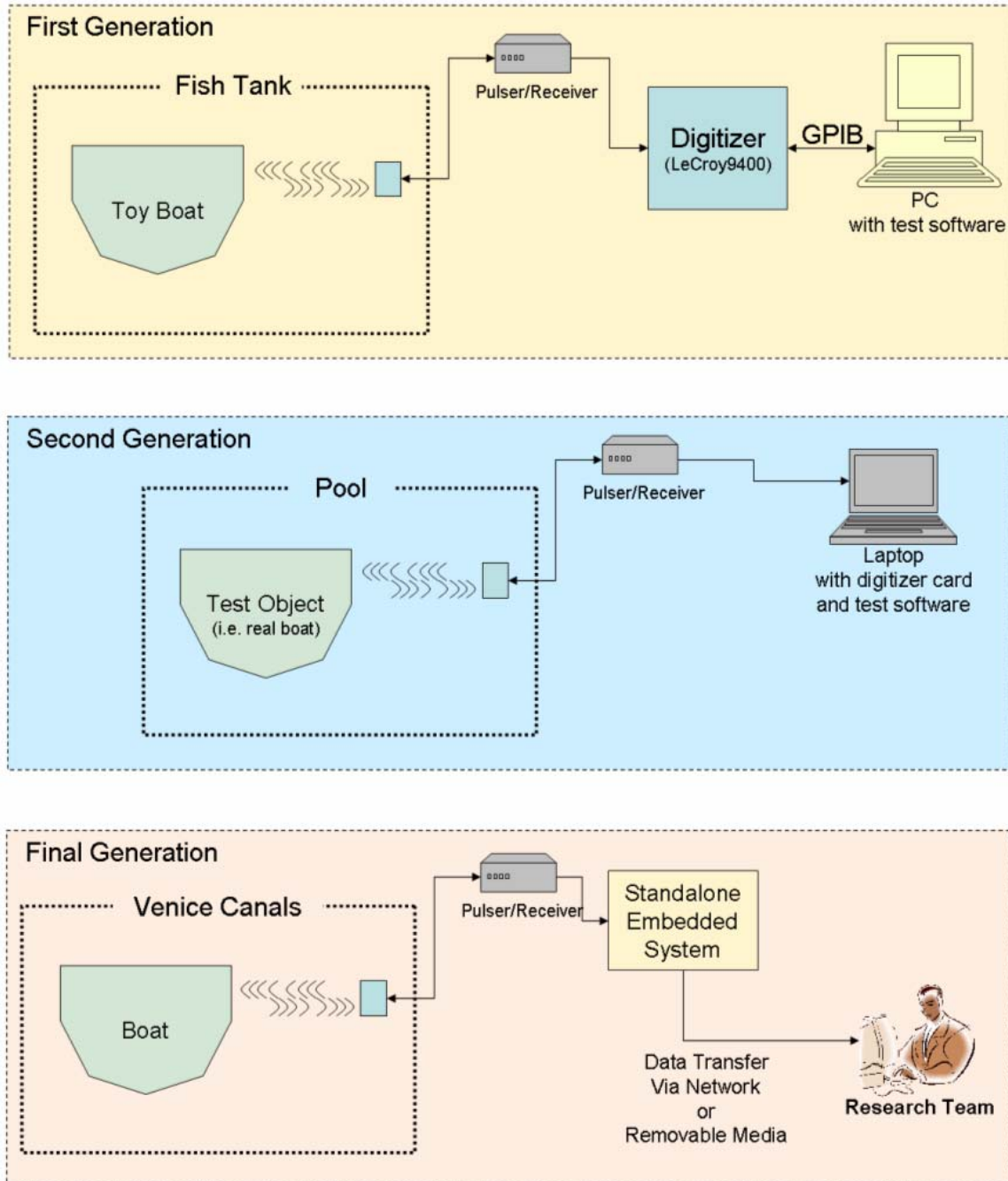


Figure 12: Embedded System Development Diagram

The following sections will discuss the details of each component in the system.

4.2 Ultrasound

The ultrasound portion of the device consists of three components: the transducers, the pulser/receiver, and the digitizer. The use and implementation of both the pulser/receiver and the digitizer is fairly straightforward for both testing and actual deployment of our system. The set-up of the transducers is more complicated.

4.2.1 Components

The pulser/receiver is responsible for controlling the transducers. For the testing stage in the lab our group used a Panametrics 5077PR square wave pulser/receiver to control the transducers. More research will have to be done to determine exactly what kind of pulser/receiver should be used in the final system.

The digitizer is also fairly straightforward. For lab development work we used a LeCroy 9400 oscilloscope as our digitizer. First, our group connected the oscilloscope to the signal received from the transducer. We then connected the oscilloscope to a computer through a GPIB cable in order to get the digitized signal into the computer for testing purposes. In the second generation system and onward, an NI5102 PCMCIA digitizer card is used and is given the responsibility of digitizing the input signal as well as acting as the bridging for synchronization between the pulser receiver and the software environment.

The actual selection of transducers and their positioning is the major ultrasound design hurdle for our system. The design was simplified for the first testing stage and that will be discussed first.

Testing of the system will be accomplished using two transducers. Our group decided to use two 5 MHz transducers for our testing in the tank. We set the transducers to a 90° angle against one wall of the tank pointing directly at the other wall. The transducers were then set at a level below the surface such that the ultrasound waves reflected off of our test objects. The two transducers were spaced only about 0.03 meters apart along the wall. The purpose of having the two transducers is to be able to find the velocity of the boat. The pulser/receiver was originally set to pulse every 500 milliseconds. There was a relay set up so that the two transducers were alternately pulsed. This will be discussed in more detail later. For the first stage this gave us 1 sample per second for each transducer. The system couldn't be set to go any faster in the first stage because of the speed of GPIB transfer to the computer. Once we went to the laptop system the pulser/receiver was set to pulse every 0.01 seconds instead. This was synchronized with the digitizer clock so that we can coordinate the reception of the data with the pulsing of the pulser/receiver.

For the final system a more complex transducer setup will be necessary. The canals in Venice usually have a standard 2 meter difference in water depth between low and high tides. The median water depth is 2 meters as well. Our system needs to be able to detect boats at any time so we need a system that can detect boats at different heights from the canal bottom. In order to accomplish this, our group intends to make use of a separate transducer or other means to measure wake. If a transducer is used, it will be set up in a tube and pointing at a device floating on the surface of the water. The movement of the device will allow us to determine wakes created by boats and the median wake at a given point in time. This device may be able to be eliminated if we can

make use of some of the wake monitors already installed in some of the canals. We will use the movement of the median wake over time to adjust the level of the transducers. One possibility may be to use the median wake to drive a stepper-motor. This stepper motor will move the other two transducers up and down on two stationary poles. This will allow the transducers to be at a constant depth below the current median water level so that we will always be able to detect a boat. Due to the fact that the shallowest boat draft is about 20 centimeters, we always want the transducer to be above this level. For that reason we will put the transducer at a height of at least the radius of the transducer above 20 centimeters below the water's surface. This will most likely be about 10 centimeters below the surface. From the Moto Ondoso project⁹ we know that it is highly unlikely that a wake will have amplitude greater than 18 centimeters. Only half of this amplitude is below the water level so having the transducer more than 9 centimeters below the surface should eliminate any errors of the ultrasound pulse reflecting off of any wakes. By having the transducer closer to the surface it should minimize the possibility of missing a boat that is riding on top of a high wake in front of the transducer. The other option is to have two stationary arrays of transducers on poles to cover the whole range of depths in the canal. We would then set up transducers about 20 centimeters apart in each array. This could be very expensive though as it would likely necessitate the use of 22 or more transducers. The frequency for the final transducers will have to be determined based on the width of the canal and ultrasound wave propagation properties. Transducers with the highest workable frequency should be chosen.

⁹ Moto Ondoso

4.2.2 Signal Multiplexing

In order to obtain enough information to calculate both the velocity and length of the boat, it was determined that there needed to be information from two separate locations in the canal. These locations had to be at a known distance apart and close enough to capture the same boat at some points in time. To acquire signals from two separate transducers, a multiplexing circuit was designed. The multiplexing circuit is designed to switch the connection to and from the pulser receiver to the two transducers alternatively upon a trigger signal. This is accomplished by the following circuit shown in Figure 13.

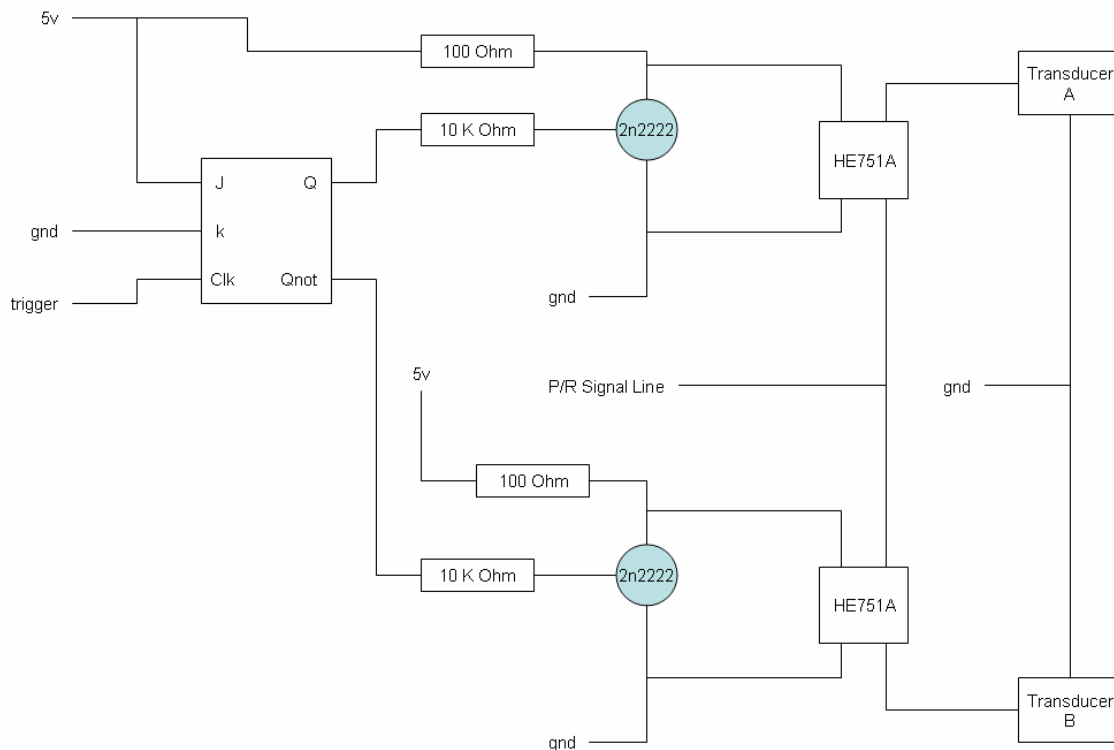


Figure 13: Multiplexer Schematic

The HE751A SPST reed relay was chosen for its operating speed. The specification allows for switching (with bouncing) at 1msec, although our testing showed switching times with bouncing of no more than 300usec. This allows the system to run at

speeds of 100 pulses per second, which will give 50 pulses per second resolution at each transducer.

In the first generation testing, the synchronization was done using an external 555 timer signal. The pulser receiver and LeCroy were triggered using the positive edge of the 555 timer signal while the multiplexer was triggered to switch on the negative edge of the signal. Since the PC software could not be synchronized with the external circuit, the synchronization signal was set to a low enough frequency to allow enough time for the software to capture and process each signal before the next signal was pulsed.

In the second generation and later, the synchronization will also extend to the PC system through the digitizer card. The pulser receiver will be set to fire at the desired frequency (such as 100 Hz), and the software will obtain information at every trigger created by the pulser receiver. The switching circuit will be triggered by an external trigger generated by the digitizer. This pulse will be generated after each time an echo is received. Figure 14 shows the timeline of this process.

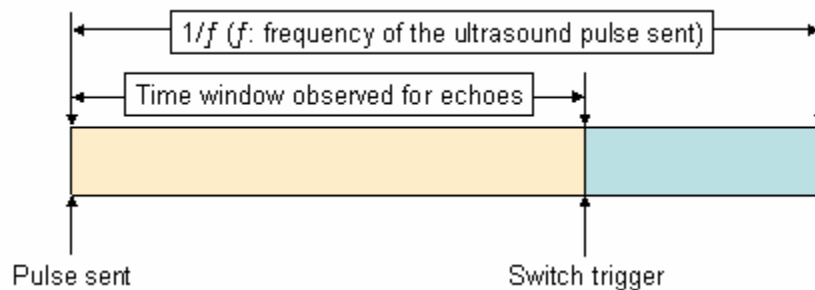


Figure 14: System Synchronization Timeline

Notice that there is a time after the observation. This is due to the fact that the multiplexing circuit does not switch instantaneously. Also considering the time required

to wait for any valid echoes, the maximum frequency the system can be pulsed at is determined by:

$$1/f = \max \text{ echo return time} + (\max \text{ switch delay} \mid \max \text{ software process delay})$$

The second value is the larger value of the two. This is because after an echo has been received by the software; both the multiplexer and the software must be ready to handle the next pulse.

4.2.3 Testing

To test the signal acquisition of the system, there are a few steps common steps to both the first and second generations of the system:

- Provide known signal to input and sample
- Use manufacturer provided software or other available software
- Use custom software (in second generation)
- Connect pulser/receiver with one transducer and sample echoes

The very first step is to test with a known input signal and software that is known to work. In the first generation, since the hardware has a display, the signal should be easily read from the input. The GPIB interface on the other hand does not have any indicators of its own to check whether the digitized signals can be transmitted or not. This can be tested using some software in MatLab to acquire data from the oscilloscope through the GPIB interface. In the second-generation system, we will first use a known signal source such as the signal generator and use the software provided by the digitizer vendor to test whether the card/driver is operational. Once the card and the driver are tested, we can test the software development interface by writing test code to obtain data

and output it in some form. This first set of tests should ensure that the digitizer hardware and software is working correctly and is ready to be used.

In the second test, our group will connect the pulser/receiver to the system with a transducer, immerse the transducer into a body of water and aim it at a solid flat surface at a normal incidence. The digitizer reception should be triggered off of the pulser/receiver trigger pulse. Depending on the distance between the transducer and the wall, the offset to the echo on the digitized signal can be calculated as follows:

$$t_{\text{round trip}} = \frac{2 \cdot d}{c}$$
$$\text{offset}_{\text{discrete time}} = R_{\text{sample rate}} (S/s) \cdot t_{\text{round trip}}$$

There should be a high reading near this range, which indicates that the transducer has correctly transmitted a pulse and received the echo from the flat wall. The readings should also be near zero at most other sections since there should not be any reflections between the wall and the transducer. This test will ensure that the transducer and pulser/receiver are functional. Also, after this test, the system can be used to test the signal-processing unit.

4.2 Signal Processing

The signal processing unit is responsible for taking the data, interpreting it, and converting it into a useful, human-readable form.

4.2.1 Overall System Design

The job of the signal processing unit is to extract correlation amplitude, signal width, arrival time, and signal presence from the incoming echoes. This can be seen in the figure below.

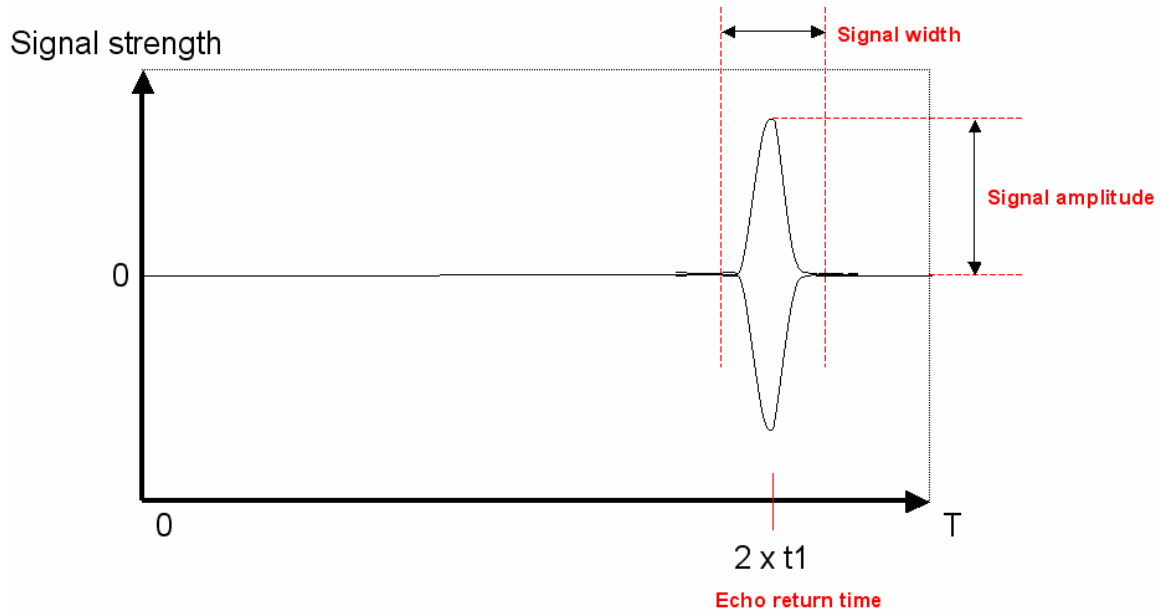


Figure 15: Extraction Parameters

The following section will describe the theoretical process of how this is accomplished. This part of the signal processing was done in a MatLab environment for the first stage and migrated to a MatLab C++ library for the following stages.

4.2.2 Receiving an Echo

The first step in being able to detect the presence of a boat is to be able to receive an echo. The echoes received from the canal are analog signals looking similar to this:

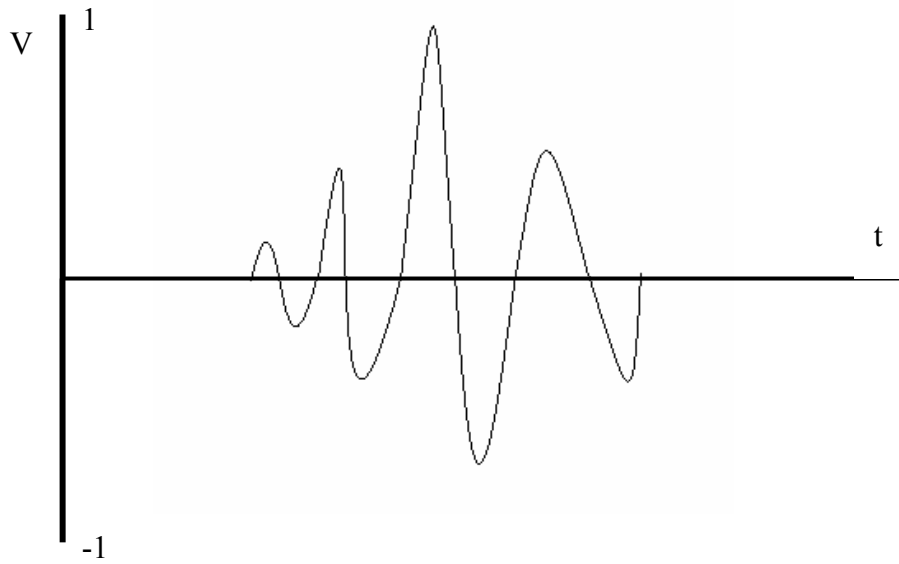


Figure 16: Analog Echo Waveform

Figure 16 shows the raw data received by the digitizer from the canals. After digitization, the analog waveform will be converted into a digital signal, looking like this:

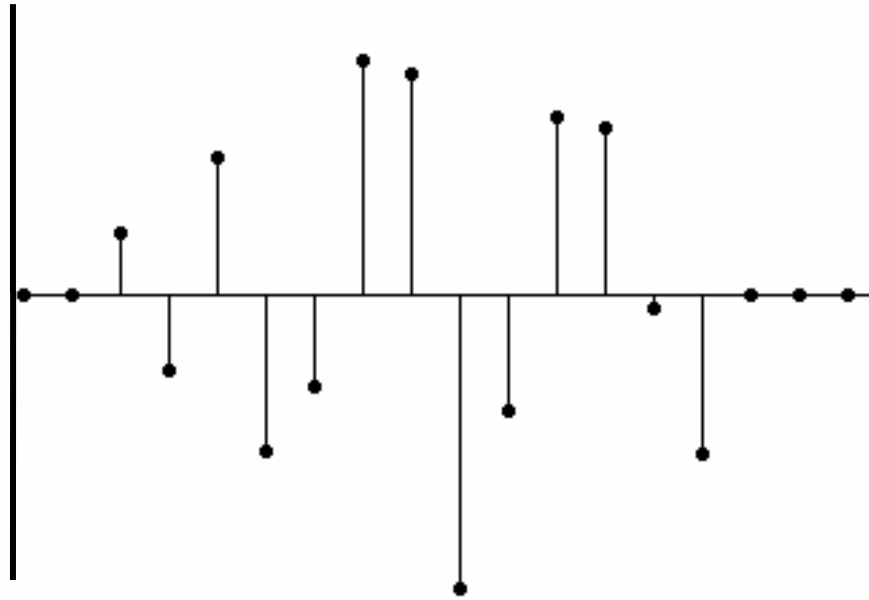


Figure 17: Digital Echo Waveform

Each of the dots represents a sample from the original waveform, correlating to the voltage at that specific sample time. To make use of this data, these voltages will be stored as an array:

$$V_{\text{SAMPLE}} = [0, 0, 0.190, -0.26, 0.48, -0.56, -0.39, 0.82, 0.79, -1.21, -0.47, 0.61, 0.59, -0.03, 0, 0, 0]$$

Given this array, the digital signal can be reproduced at any time. The signal processing unit will receive echoes in this format. For this example a very small sampling rate was used, thus not producing an accurate replication of Figure 16. As discussed in the ultrasound section, the sampling rate (from the frequency) will be much greater, making the actual processed echoes much clearer.

4.2.3 The Fourier Transform Algorithm

Once we receive an echo, the first step is to measure its correlation to a known echo. This known echoes will be acquired ahead of time, with as little environmental noise as possible. This will guarantee the most quantity of desired spectral level content.

The following is a sample execution of the algorithm:

Given a known echo of 6Hz:

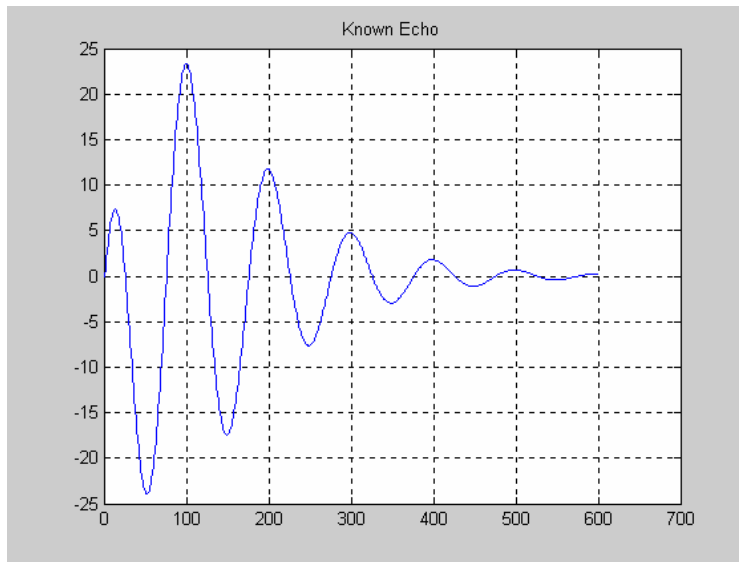


Figure 18: Known 6 Hz Echo

And a similar return echo:

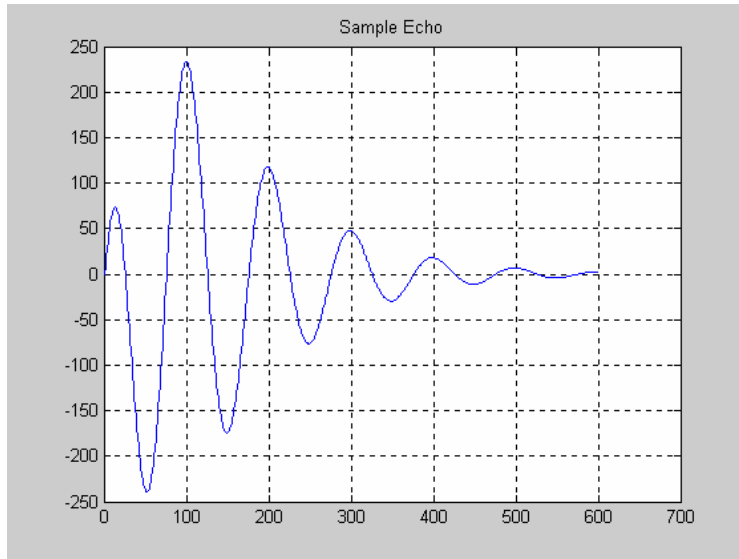


Figure 19: Sample Echo

This sample echo has a center frequency of 6Hz. This represents a strong echo from a solid surface. To calculate the correlation, we first must calculate the FFT of both echoes.

Because both echoes are the same, the FFT of both will look like this

FFT(echo1) & FFT(echo2):

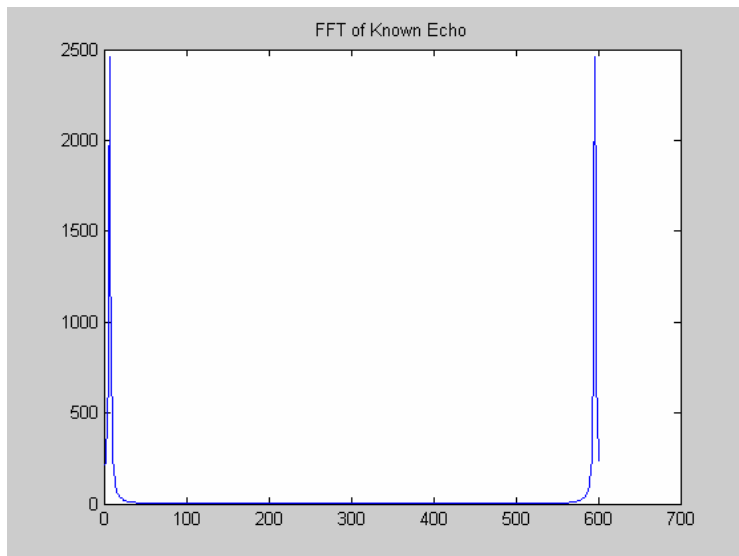


Figure 20: FFT of Echoes

Multiplication of the two FFTs yields

$$FFT_Mul = FFT(echo1) * FFT(echo2):$$

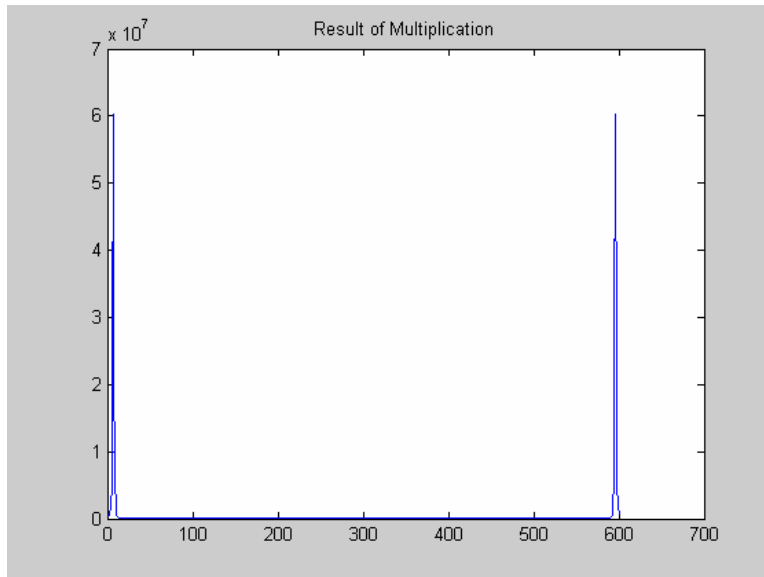


Figure 21: Result of FFTs multiplied

And finally, to find the correlation magnitude function, the inverse FFT of Figure 8 is

$$\text{taken: } FFT^{-1}(FFT_Mul)$$

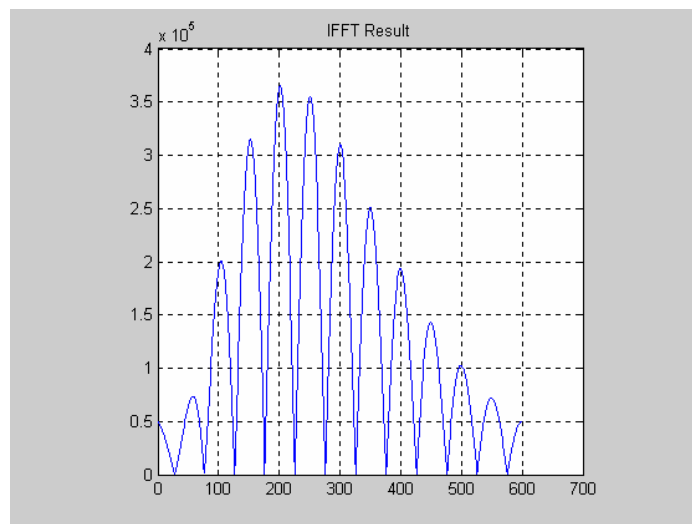


Figure 22: Correlation Magnitude Function

This is the result of the correlation function when the two echoes are similar. When the two echoes are different, for instance when the frequency of the sample echo is 12Hz, the correlation magnitude function changes:

Given a sample echo at 12Hz:

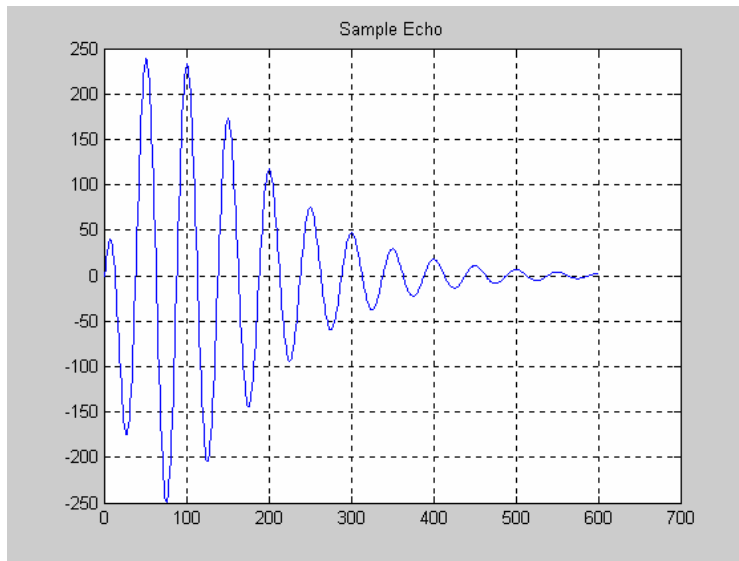


Figure 23: Sample echo at 12 Hz

The multiplication of the FFT of the known echo and the sample 12Hz echo looks like this:

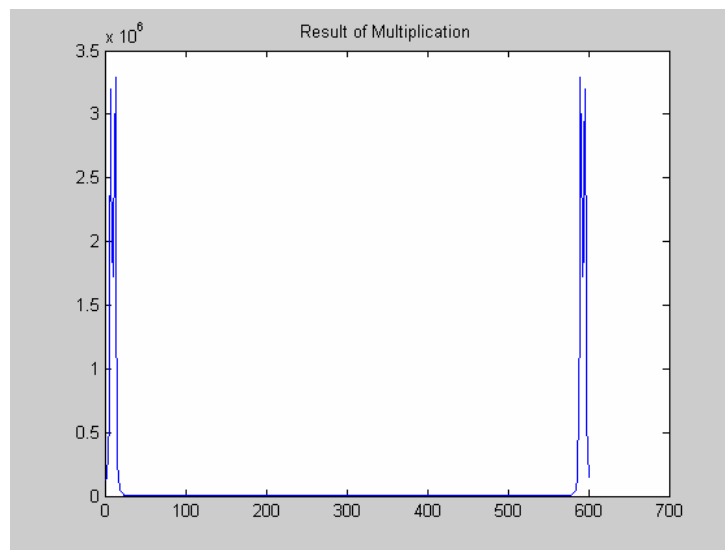


Figure 24: Result of Multiplication of FFTs

Producing a correlation magnitude function that looks like this:

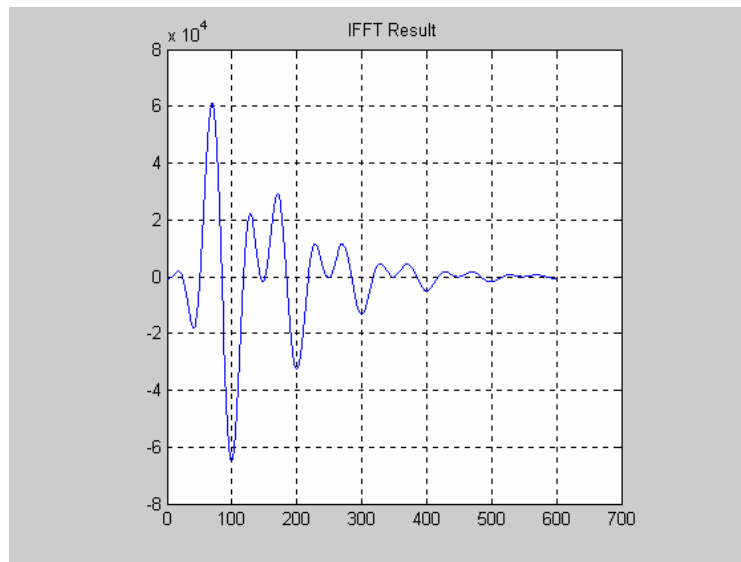


Figure 25: Correlation Magnitude Function of Different Frequency Echoes

This correlation magnitude function is quite different from that of the two like echoes, both in amplitude and shape. It is useful to examine the envelope of this magnitude function to determine similarity. The final decision on correlation levels will be made based on the envelope.

The magnitude of the IFFT corresponds to the level of correlation between the known echo and the incoming sample. If the known echo and the incoming sample are similar, the spectral content of the echoes will line up properly, and when multiplied, the amplitude of the signal after taking the IFFT will increase. If the known echo and the incoming sample have different spectral contents, then the amplitude of the IFFT will decrease. Thus, the correlation between the known echo and the incoming sample is directly proportional to the amplitude of the IFFT. This amplitude is easier to calculate

after taking the envelope of the function. This envelope will be used to calculate the correlation. The final algorithm will be this:

1. Get sample from water
2. Find FFT of sample
3. Multiply FFT of sample with known echo FFT
4. Find IFFT of multiplication result
5. Find Envelope of IFFT
6. Measure amplitude

4.2.4 Measuring the Signal Amplitude Using the Analytical Signal Magnitude

The first step in measuring the signal amplitude is to set the negative frequencies of the FFT of the correlation magnitude function to 0, and double the positive frequencies such that:

$$S^{\wedge}[w] = \begin{cases} 2*S[w], & m \geq 0 \\ 0, & m < 0 \end{cases}$$

Thus, the $S^{\wedge}[w]$ function for the two like echoes will produce the following FFT plot:

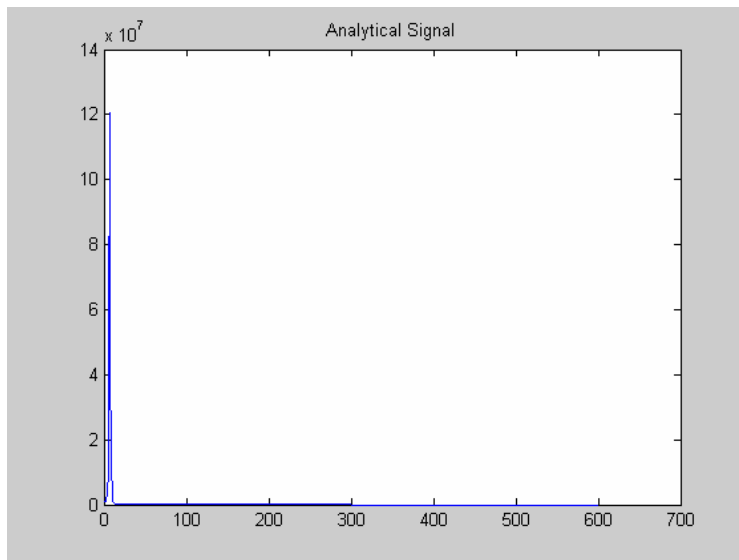


Figure 26: Analytical Signal FFT

Finally, the inverse FFT of this Analytical Signal FFT produces the envelope of the original signal. On this plot, the original signal has been plotted as a dashed line, and the envelope as a solid line:

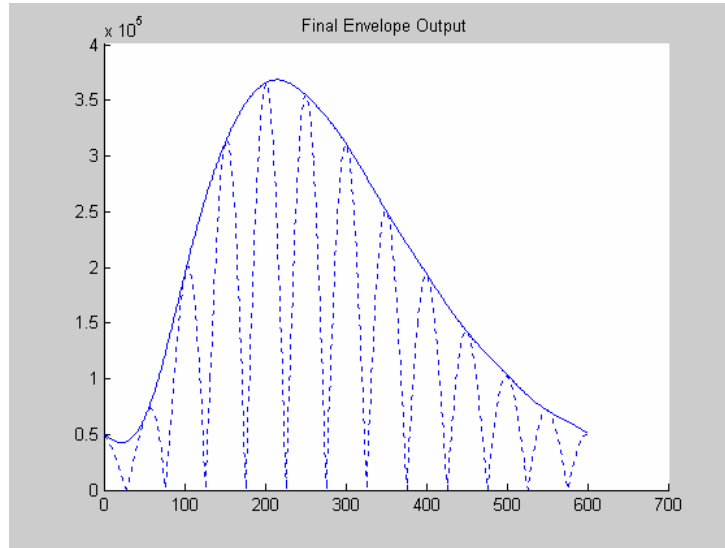


Figure 27: Final Envelope Output

The amplitude of the envelope will be passed to the intelligent system for use in determining boat presence. When the 12Hz sample is used, the final envelope output yields:

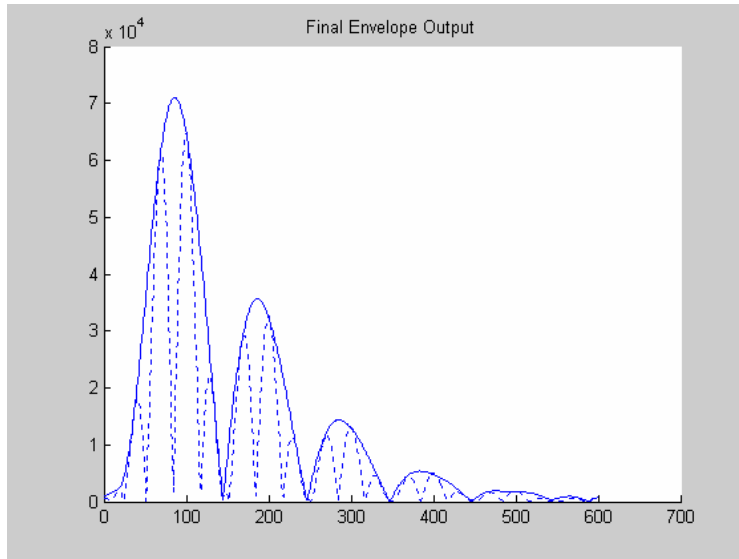


Figure 28: Final Envelope Output for a 12 Hz Sample

The amplitude of this envelope is much less than that of the envelope where the frequencies were the same.

4.2.5 *Compensating for Time Shift*

A problem may arise in that the boat's hull is not completely straight. Thus, there will be arrival time differences in the received samples. In our initial development using MatLab, we employed the Fourier transform algorithm detailed in the previous section. By using this, we can compensate for differences in arrival time. For instance, when two similar echoes are received at exactly the same time, they will align perfectly and have a high correlation.

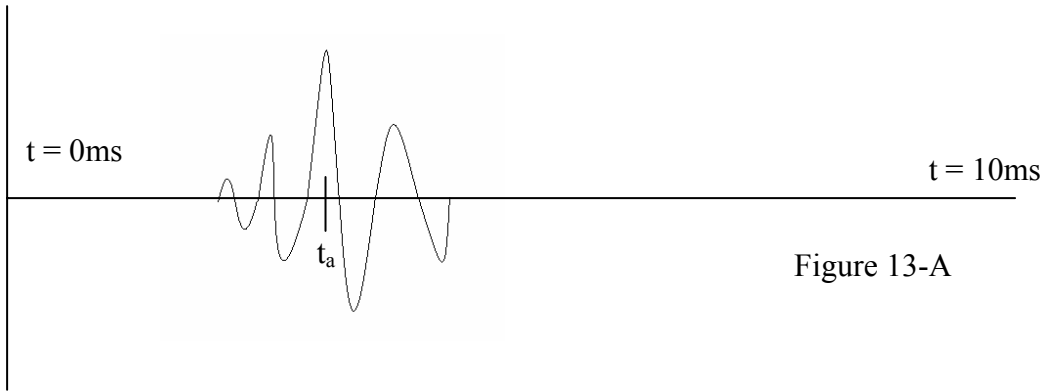


Figure 13-A

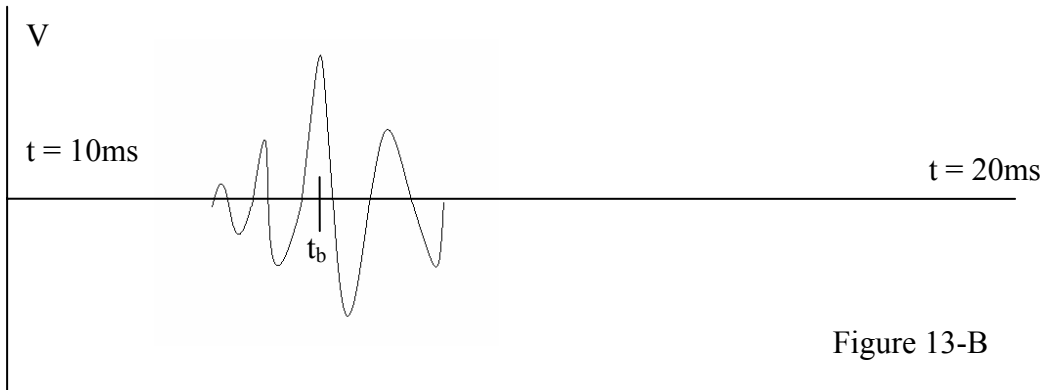


Figure 13-B

Figure 29: Two Echoes Received Consecutively with Equal Offsets

However, in a real-life application, this is not the case. Changes in the shape of a boat's hull will alter the time interval between echoes being received.

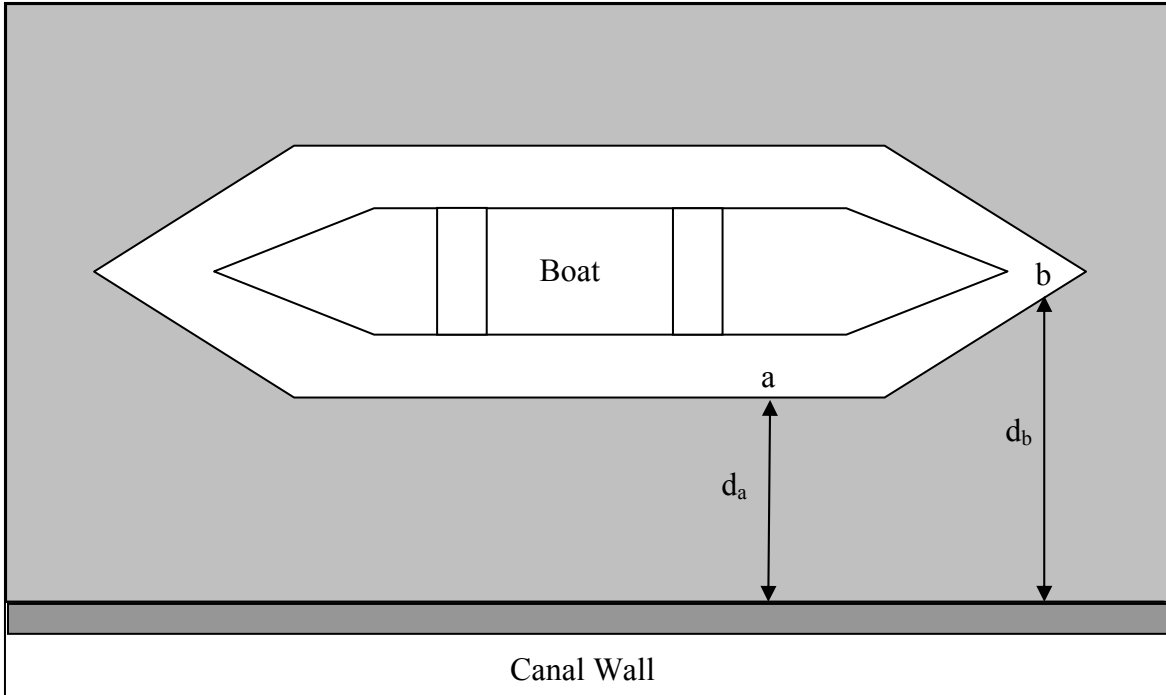


Figure 30: Different Points on a Boat at Different Distances

As can be seen in Figure 14, the distance from the canal wall to the boat is different for points a and b. Thus the echo that bounces off of point “a” will have a smaller travel time than point “b”. If $d_a = 1\text{m}$, and $d_b = 1.3\text{m}$, with the echoes traveling at approximately 1480 m/s , then we see a difference of:

$$t = d/s$$

$$t_1 = \frac{1\text{m}}{1480\text{m/s}} = 6.756 \times 10^{-4} \text{ s} \qquad t_2 = \frac{1.3\text{m}}{1480\text{m/s}} = 8.787 \times 10^{-4}$$

$$\text{Difference} = t_1 - t_2 = 8.787 \times 10^{-4} - 6.756 \times 10^{-4} = 2.031 \times 10^{-4} \text{ s}$$

This delay causes the echoes to line up incorrectly:

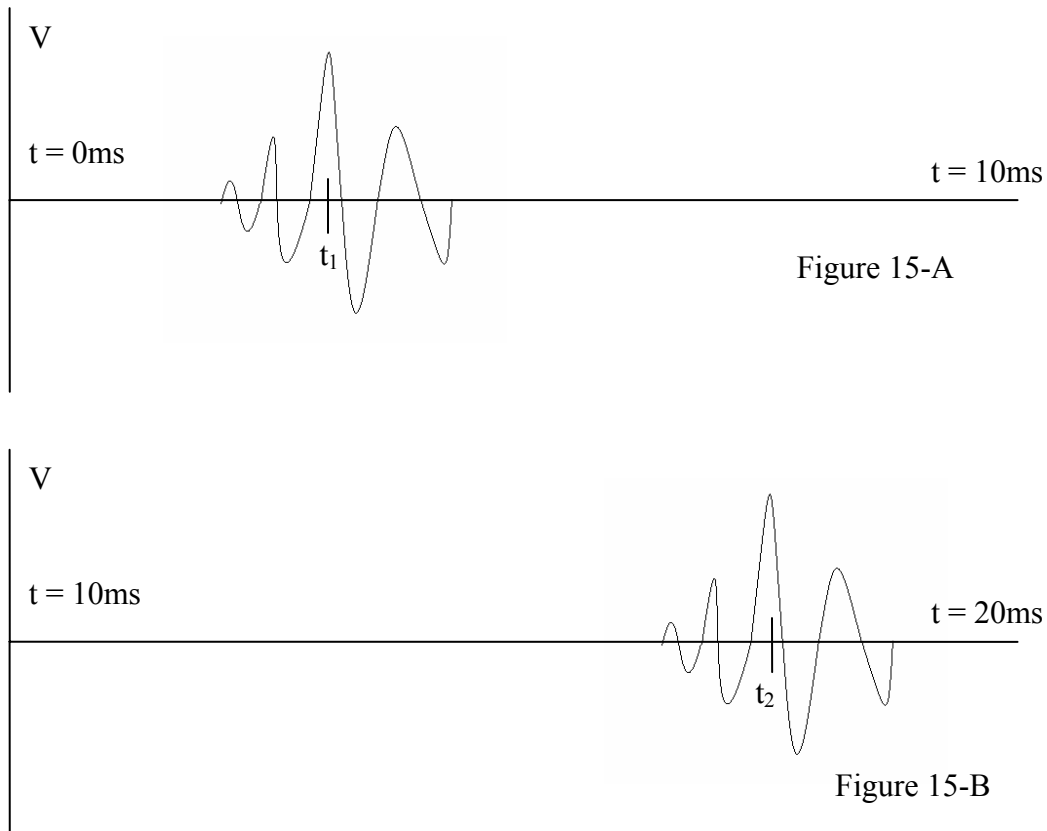


Figure 31: Two Echoes Received Consecutively with Different Offsets

We can see from inspection that both echoes look the same. However, if we used a standard correlation function to calculate the correlation from $t = 0$ to $t = 20\text{ms}$ for figures 15-A and 15-B, it would result in 0. This is because the echoes occur at different times, and we are correlating the signal as a whole. Use of the Fourier transform algorithm will compensate for this. Because we are looking at the entire sample, and examining its spectral content for the known echo, rather than echoes that match time-wise, we eliminate the shift error. The FFT of the incoming echo is independent of time shift. Thus, even if the two echoes aren't perfectly aligned, as long as the echo is contained in the sample being examined, the echo will be correlated.

4.2.6 Measuring Arrival Time

To measure the arrival time of the returning echo, we will look at the location of the maximum point of the signal:

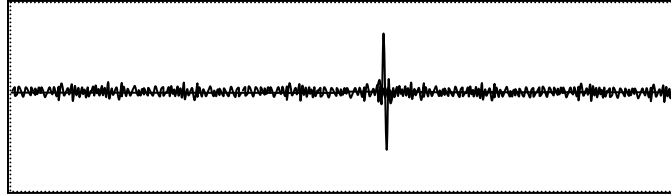


Figure 32: Measuring Arrival Time

In this sample echo, it is clearly seen that the peak of the echo rises much higher than the noise. To find the arrival time, we simply locate the index of the largest sample, and multiply that by the sampling period to find the time in seconds.

4.2.7 Measuring Signal Width

Even though we can find the max peak of the echo, a return sample may have more than one peak, especially if it is bouncing off a piece of debris or a rough surface. Therefore, an algorithm was developed to determine signal width. To measure the signal width of this sample:

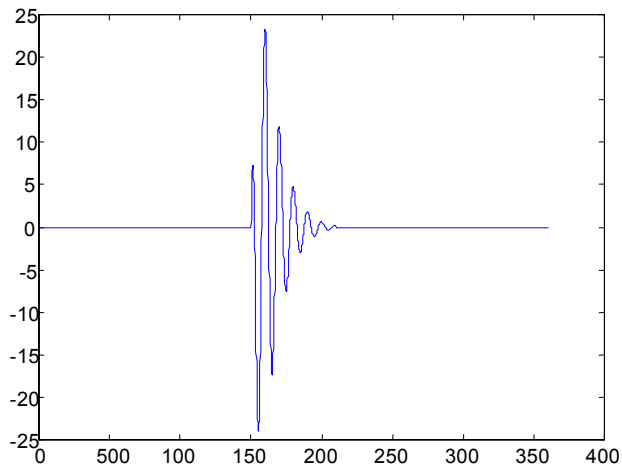


Figure 33: Signal Width Echo

First we rectify the signal:

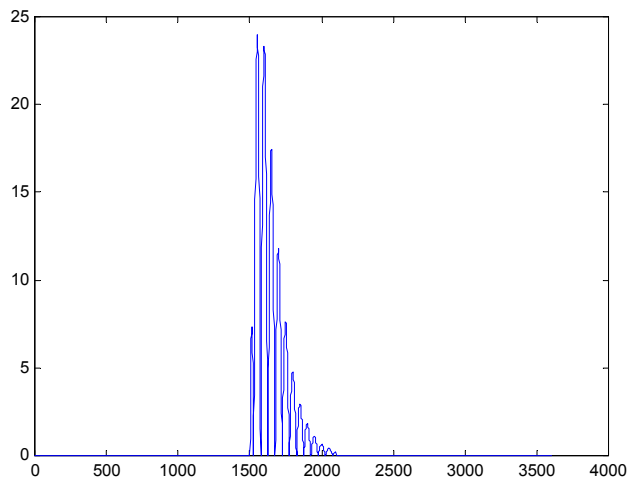


Figure 34: Signal Width Rectified Signal

And then subtract 85% of the peak amplitude to eliminate noise:

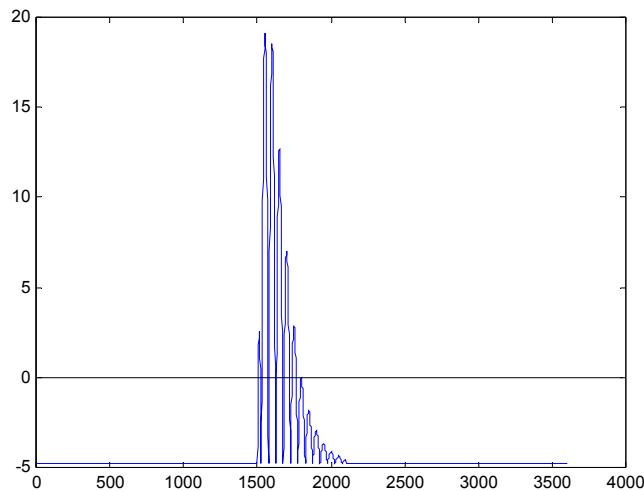


Figure 35: Signal Minus 85%

At this point, we count the number of values greater than 0, and multiply that by the sampling period to determine the width of the signal in seconds. In this example, with a sampling period of 5.71×10^{-6} , we calculate the width as follows:

$$\text{Samples} \times \text{SamplingPeriod} = \text{width(s)}$$

$$347 \times 5.71 \times 10^{-6} = .00198\text{s}$$

Therefore, this signal has a width of .00198 seconds.

4.2.8 False Positives

A false positive is any non-boat object that is interpreted by the system to be a boat. For instance, if a bird swims past the transducers and then stops in front of the second one it could be interpreted as a boat. Because the bird is stationary in front of the transducer, more echoes will reflect off of it than if the bird had been just passing with the current. This will make it look to the system like it is larger than it really is,

interpreting the bird as a boat. Also, because the bird is so close, the amplitude of the echoes will be higher.

One possible way to compensate for this is to adjust the placement of the transducers. If the transducers are just below water level, then the transducer is more likely to detect floating non-boat objects resulting in more false positives. If we lower the placement of the transducer, false positives floating on the surface can be eliminated. The placement of the transducers also must take into account how deep the hull penetrates the surface. If the placement is too low, it will miss the boats.

4.2.9 False Negatives

A false negative is a boat that is not interpreted as such. The hull of a gondola does not penetrate the water's surface very deeply. If the placement of the transducer is lower than the penetration of the boat's hull, the ultrasound pulses will miss the boat, and not record the boat's presence. However, the gondola is rowed, instead of motorized, so it produces little to no wake. We must evaluate the importance of counting the gondolas. If raising the transducer means counting the gondolas, but also counting more false positives, a compromise needs to be made. If too many false positives are detected to count a smaller number of gondolas, then the transducer should be lowered to detect wake-producing boats.

Another possibility of false negatives occurs if the boat is traveling too fast. If a small boat is traveling excessively over the speed limit, the number of pulses that bounce off of it will be reduced. This could make it look like a non-boat object if the pulsing rate is low, causing a false negative.

4.2.10 Exporting Data in the MatLab Environment

In the first generation system the signal processing was done in MatLab. Once the data was gathered, it had to be exported for use for boat detection. The easiest and most straightforward way to do this was to write it to a file in ASCII form and save it in an accessible location

4.3 Boat Detection and Data Logging

The final portion of the system takes input from the signal processing unit and determines the presence of boats passing by as well as its traffic parameters (velocity and length). Then all information about the detected boats is recorded into a log for later retrieval and study.

4.3.1 Basic Boat Detection

The boat detection is done using the information provided by the signal processing unit. The information given is some of the vital meta-data which was contained in each pulse echo sample. The data consists of the following information:

- Echo presence (TRUE or FALSE)
- Correlation Amplitude
- Signal Width
- Signal Delay (distance to the object)

A boat passing in front of the transducers is likely to satisfy some criteria within the series of data obtained. A boat passing the transducers will be detected by the pulse echo system continuously until the boat has completely passed the transducers or another object obscures the boat. A passing boat, which is usually at least 3 meters long, should intersect both transducers at some point because the transducers will be placed about 1

meter apart. All of the echoes coming from one boat should come from about the same distance away from the transducers. A boat should not suddenly move several meters within the scope of two consecutive pulses (each pulse from a transducer is spaced 0.02 seconds apart). All of this can be determined by the information coming from the signal processing unit. In addition, signal width can be used to reject any lengthy signals, which may suggest echoes from bubbles or other debris, and invalidate the echo.

If all criteria are met by a series of echoes, the chance is high that there was a boat, in which case information on the traffic is logged.

4.3.2 Boat Velocity and Length Calculation

If a boat is detected, the system needs to record the boat velocity and the measured length of the boat. Figure 36 shows the information recorded and calculated at each significant step of the boat detection.

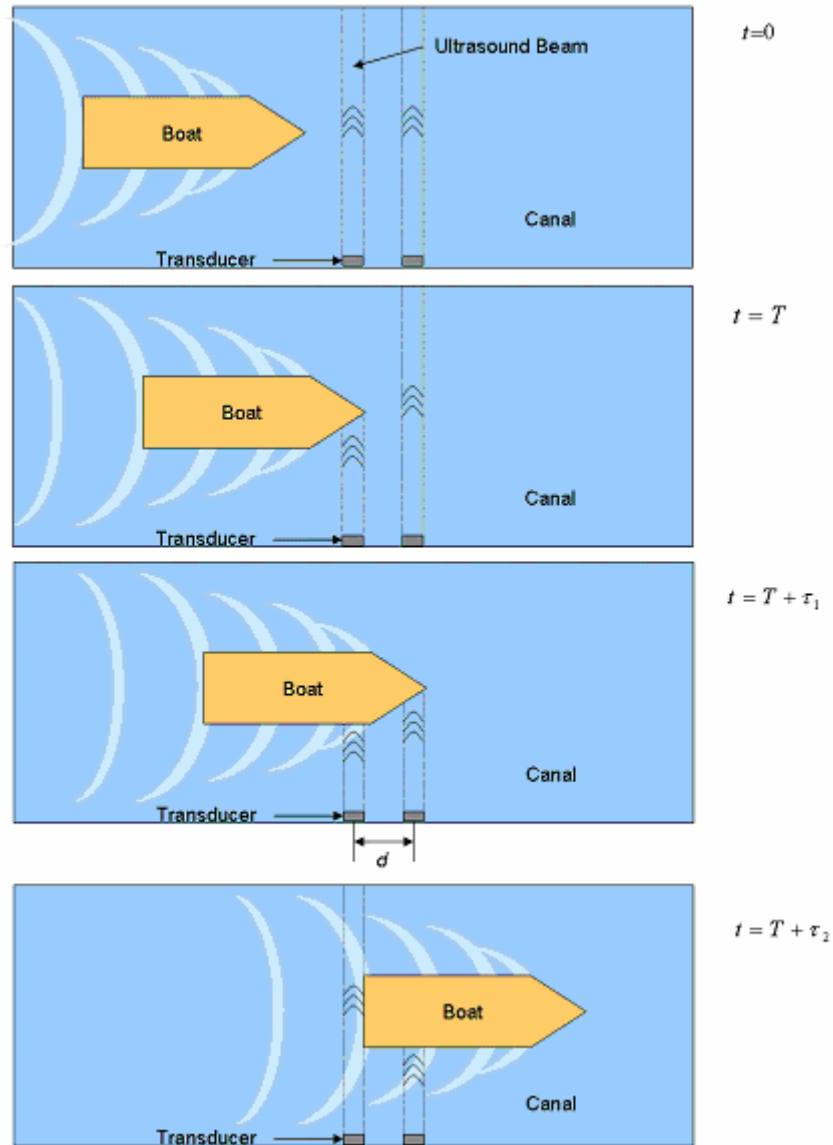


Figure 36: Boat Traffic Information Measurement

At $t = 0$, the boat has not passed any transducers yet. At $t = T$, the boat is detected by the first transducer on the left. A timestamp is taken at this point when the object is first detected. The system starts tracking the object as a possible boat in the canal it is monitoring. After some time, the boat starts to cross the second transducer as well. At this point the velocity of the boat can be calculated by the following equation:

$$v_{boat}(m/s) = d(m) / \tau_1(s)$$

d : distance between two transducers

After some more time, the boat will clear the first transducer. With the previous calculation, the length of the boat can be calculated as follows:

$$l_{boat}(m) = \tau_2(s) \cdot v_{boat}(m/s)$$

Also, at this point, the boat will be logged with the timestamp, calculated velocity and length into the log file.

4.3.3 *Special Cases*

In the canals of Venice it is highly likely that multiple boats may be passing a point in different directions. This can cause problems in the pulse-echo system where a boat which is being tracked can be lost behind another boat passing by. In cases such as these, the information of the boat being tracked will still be logged into the log file as long as the system was almost sure that there was a boat and not some debris. The system decides that the tracking object is a valid boat when there are enough consecutive echoes and has been detected by both transducers. If the boat being tracked becomes hidden by another boat, the boat will still be logged, but it will have a note mentioning the fact that the length calculation may be off due to another boat interfering. This way the system is capable of detecting and logging completely all traffic passing on the near side, and most of the traffic information from the far side of the canal.

More technical implementation details and notes on the system can be found in Appendix A: Software Operation Manual.

5. Testing and Results

This chapter will report the testing that has been conducted on the system and their results. Testing has been done on general feasibility of using ultrasound pulse echo system for such detection, and the actual integrated system performance testing in a lab controlled environment. End of the chapter will be used to report the progress on the porting of the second generation system and its state.

5.1 Transducer Tolerance Testing

A series of tests were conducted to determine the tolerance and boundaries of operation on different transducers and confirm the relation between the transducer type (defined by diameter and frequency) and tolerance towards surface roughness of target objects as well as the target orientation. These tests utilized a fixture that can hold cylindrical objects. The fixture can be oriented at different angles and positions within the tank. As seen in Figure 37, the fixture can change four different orientation variables; distance, vertical angle, horizontal angle, and depth. The red dotted line represents the angle of the surface and the blue dotted line represents the path of the ultrasound beam. For our tests we used three different surface samples with different surface roughness to obtain information on how much effect the surface roughness has on signal reflection.

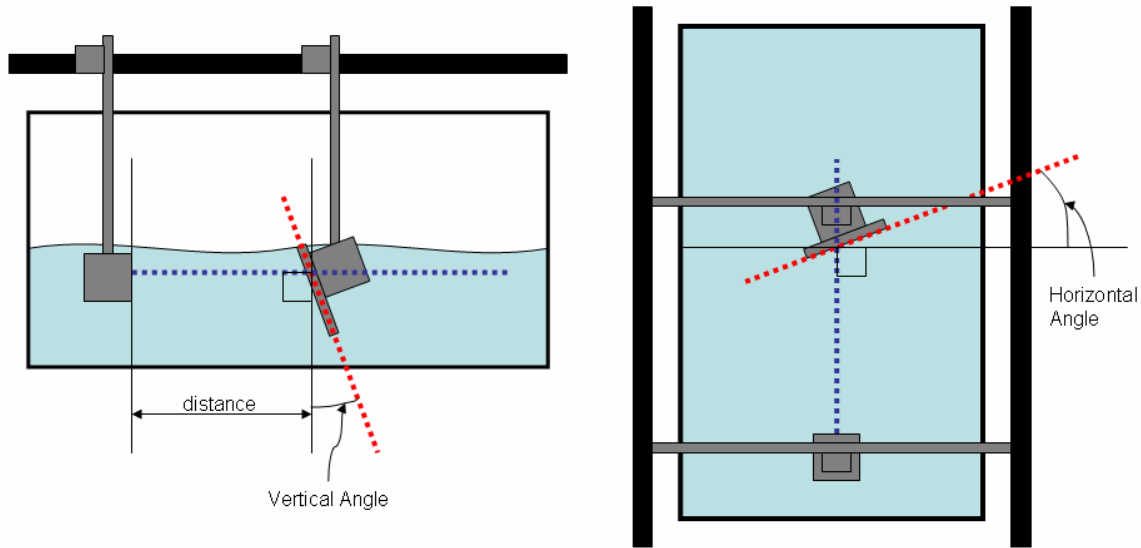


Figure 37: Experimental Setup - Surface Fixture

The test was conducted using the following transducers:

- 1.0 MHz 1 inch diameter
- 2.25 MHz 1 inch diameter
- 5.0 MHz 0.25 inch diameter

Each transducer was tested with the following surfaces:

- Rough grain (36D)
- Medium grain (60D)
- Fine grain (150C)

All of the tests were conducted with 23 different horizontal angles varying from 0 degrees (surface perpendicular to the transducer beam, see Figure 37) to 46 degrees, in 2 degree increments. The first two tests were conducted using 20 different distances, from 1 cm to 20 cm away, but we found that the distance did not seem to have a large effect on return signal strength. With this in mind, the subsequent tests were cut down to 5 different distances (1cm, 5cm, 10cm, 15cm and 20cm). The values listed on the graphs do not directly correspond to the distances or the angles. The values correspond to the

increment on the test sequence, so 1 on the x-axis (angle) will be 0 degrees, 2 would be 2 degrees, and so on until 23, which is 46 degrees. The distance marks for the first two test results are from 1cm to 20 cm in 1 cm increments, while the others were done using the five increments described above.

The test procedure used for this test is as follows:

1. Set angle of the surface to 0 degrees
2. Set distance to 1 cm
3. Measure the peak return echo strength
4. Increment distance to the next position (+1 cm for first two tests, +4 or +5 for all others)
5. Repeat back to 3 until distance is at 20cm
6. Increment angle by 2 degrees
7. Repeat from 2 until angle is equal to 46 degrees
8. Change the surface roughness
9. Repeat from 1 until there are no more surfaces to be tested

The test results for 1.0 MHz, 1 inch diameter transducers are listed in the figures below (Figure 38 to Figure 40).

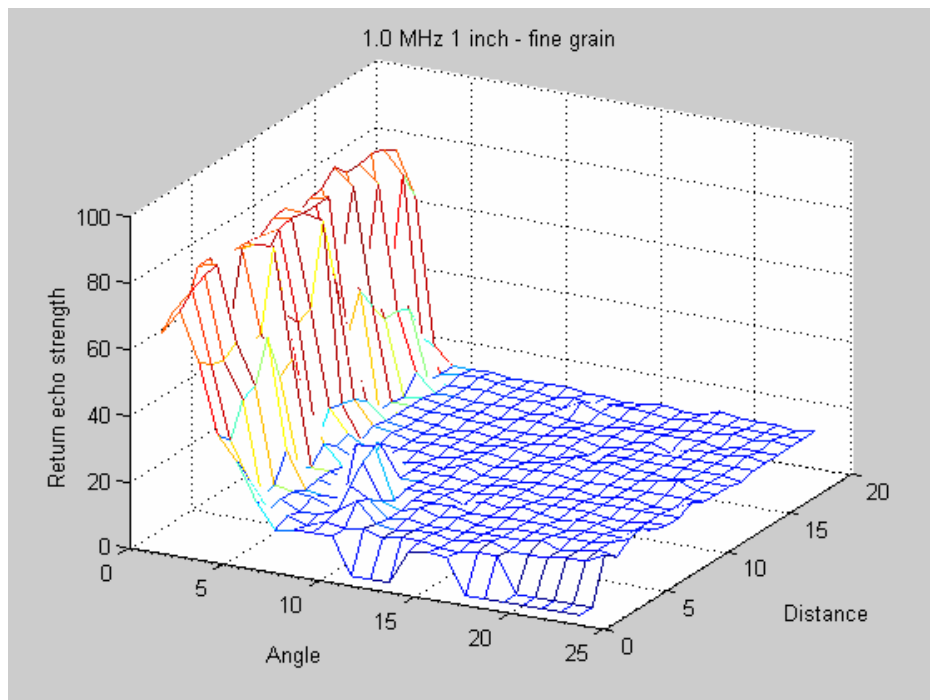


Figure 38: 1.0 MHz, 1 inch Diameter Using Fine Grain

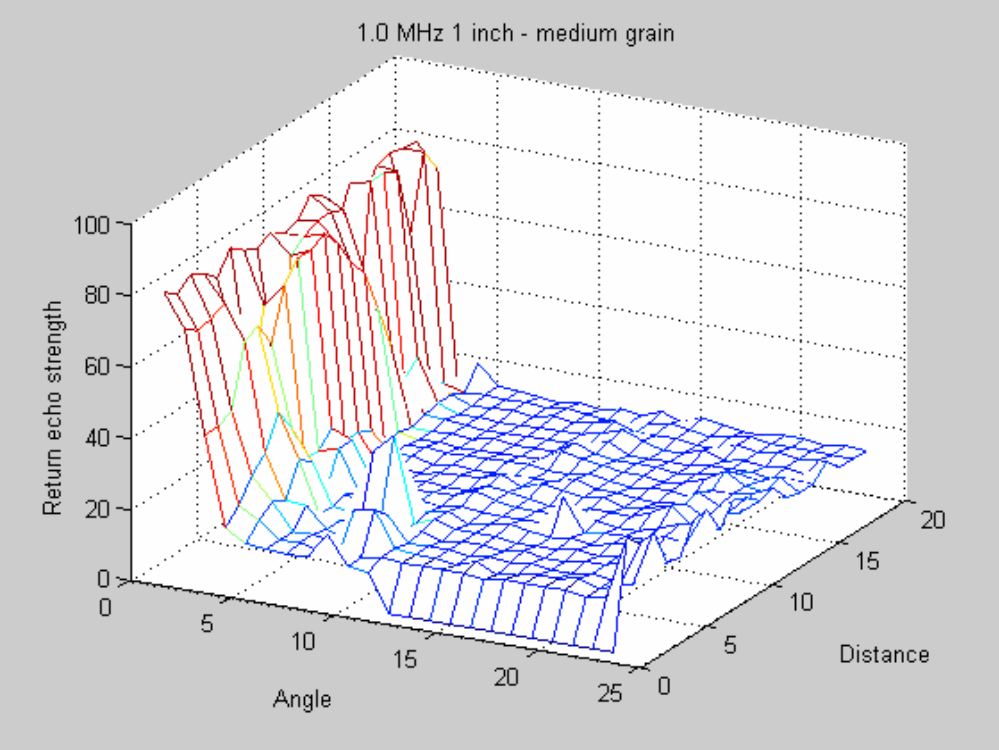


Figure 39: 1.0 MHz, 1 inch Diameter Using Medium Grain

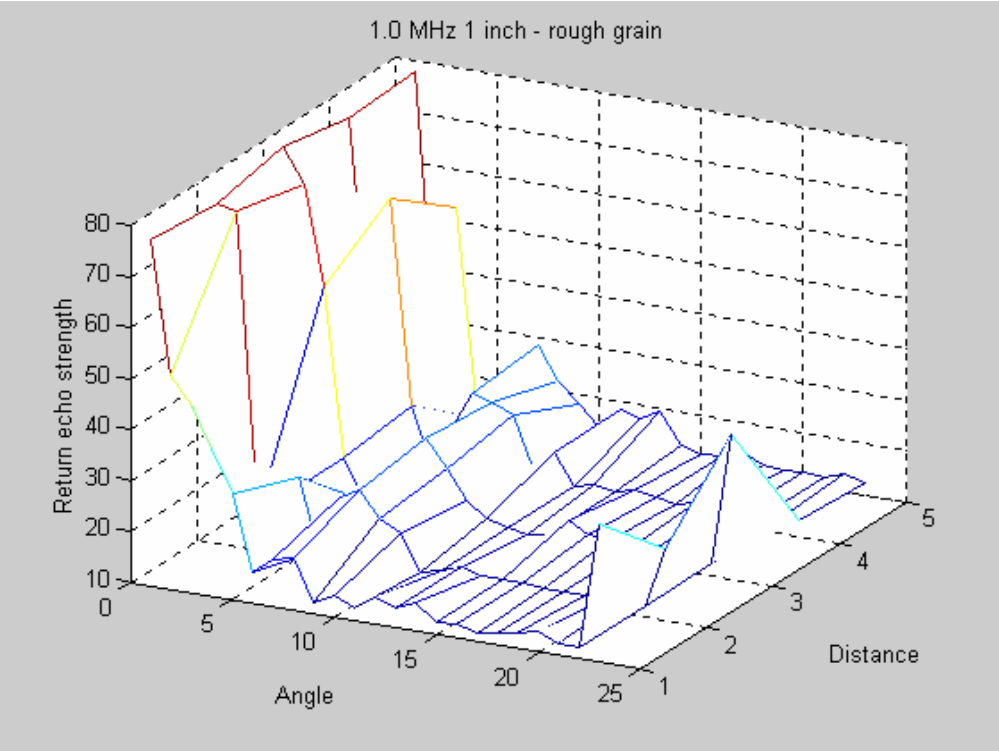


Figure 40: 1.0 MHz, 1 inch Diameter Using Rough Grain

These first two results for 1.0 MHz transducers show how little the distance affects the signal echo strength in the tested range. On the other hand, the angle has a larger effect on the echo strength at this frequency. In addition, greater surface roughness helped widen the range of angles where the return echo was strong. Even so, even with a rough surface, the signal strength is around or below 20 when the angle is greater than 10 degrees. This is due to the fact that the roughness of the surfaces is much smaller than the wavelength of the pulse and thus not affecting the signal to a large scale. This is not desirable since we would like to be able to detect the boats' bows or boats moving in diagonal to the transducers, which will most likely be at an angle.

The test results for 2.25 MHz, 1 inch diameter transducers are shown in the figures below (Figure 41 to Figure 43).

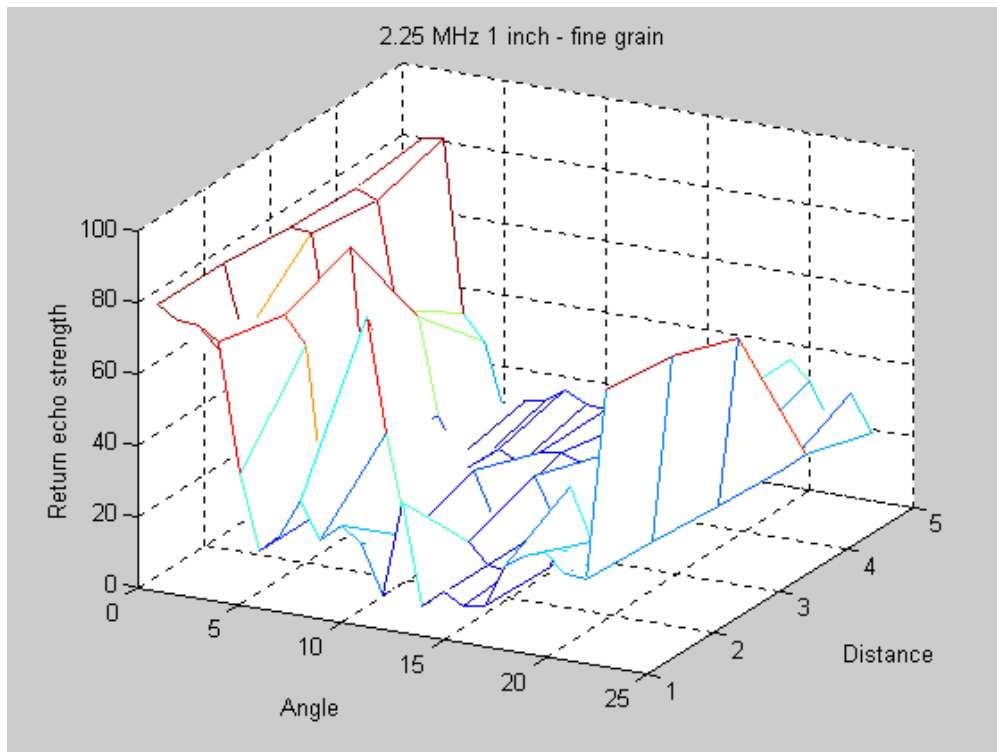


Figure 41: 2.25 MHz, 1 inch Diameter Using Fine Grain

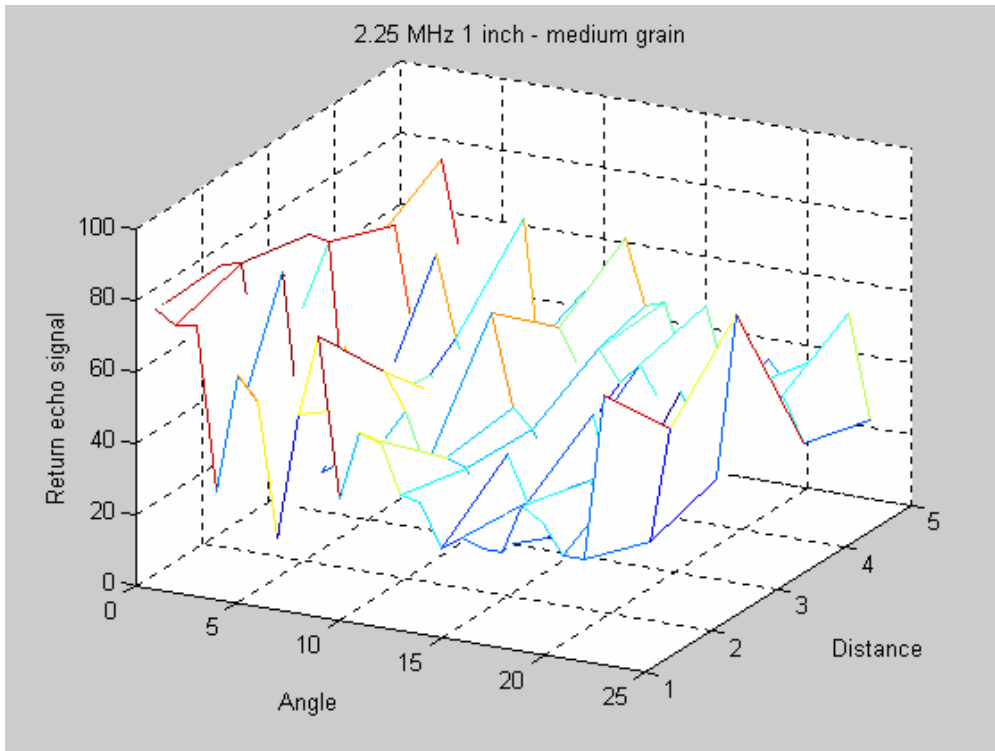


Figure 42: 2.25 MHz, 1 inch Diameter Using Medium Grain

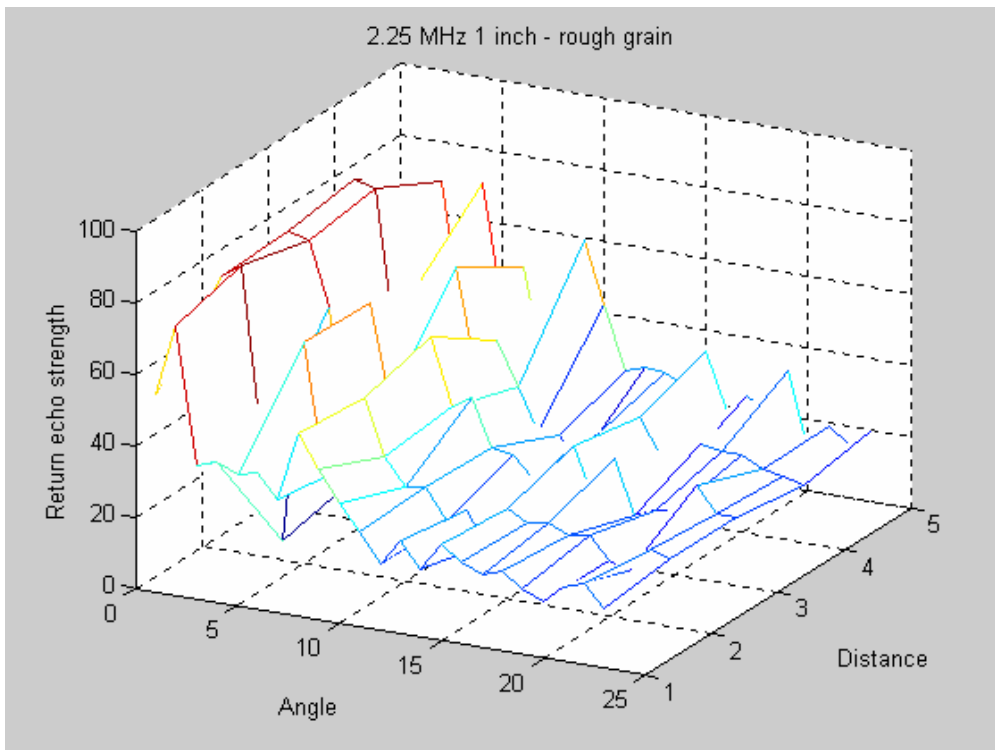


Figure 43: 2.25 MHz, 1 inch Diameter Using Rough Grain

In the test using 2.25 MHz transducers, the results showed more tolerance of larger angles. With the exception of the fine grain test, the results seem to stabilize around 20 or slightly above, which is a more defined echo than the previous test. Although the echoes at angles close to 0 degrees are less than those observed with the 1.0 MHz transducers, over the range, 2.25 MHz transducers were more effective. The distance did not matter in the distance range tested with this transducer either.

The test results for 5.0 MHz, 0.25 inch diameter transducers are shown in the next three figures (Figure 44 to Figure 46).

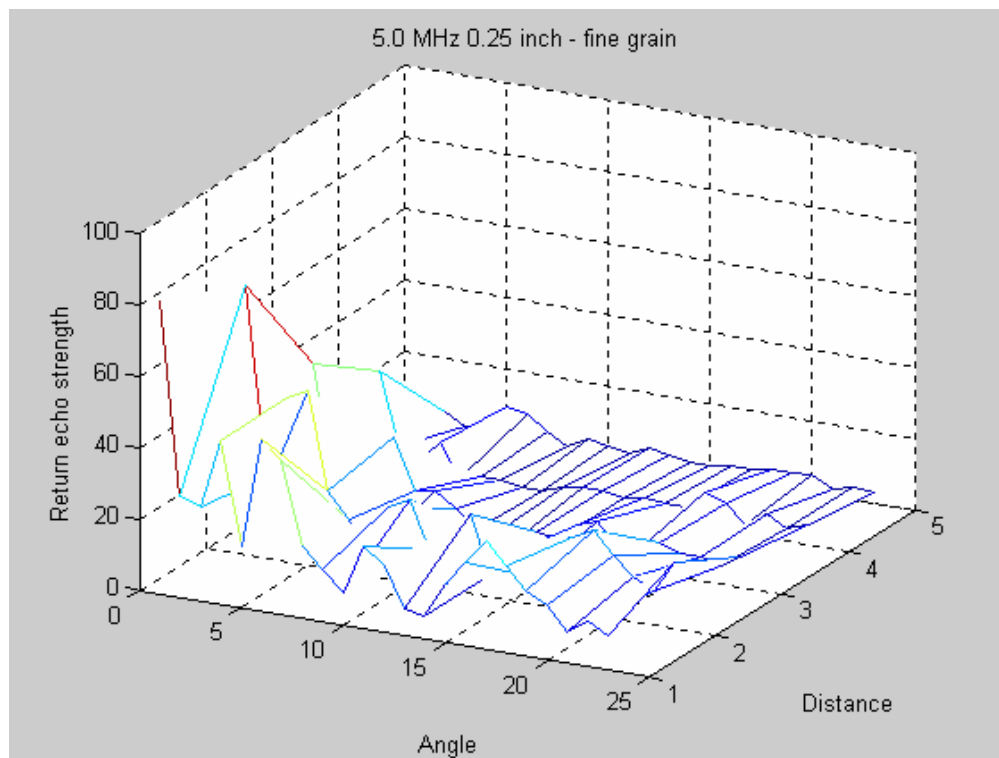


Figure 44: 5.0 MHz, 0.25 inch Diameter Using Fine Grain

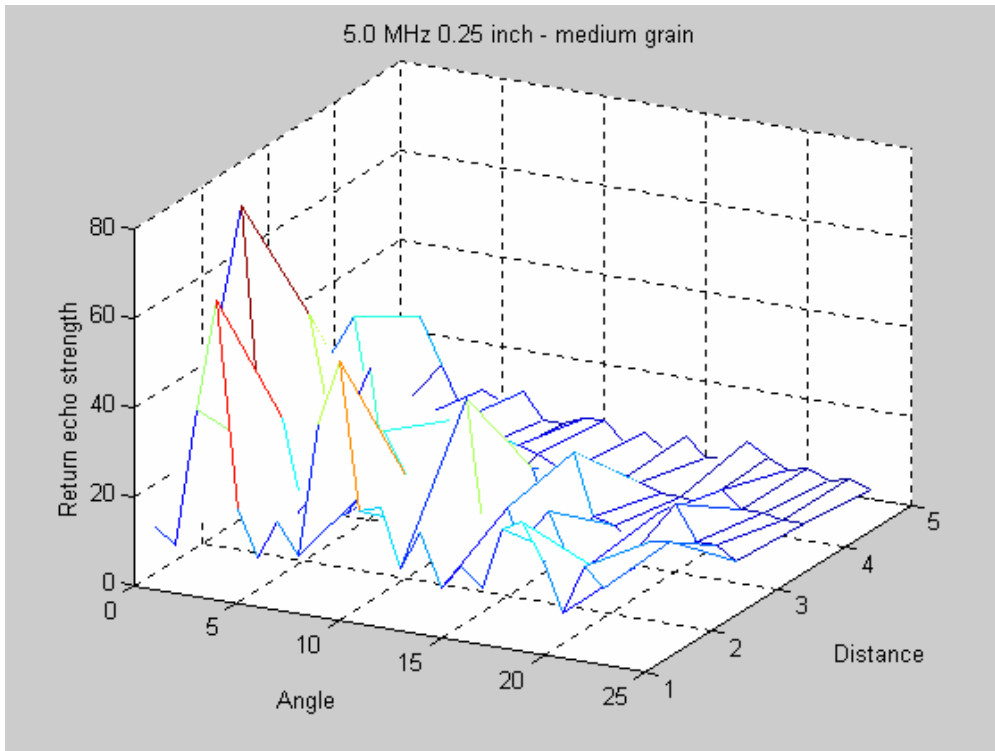


Figure 45: 5.0 MHz, 0.25 inch Diameter Using Medium Grain

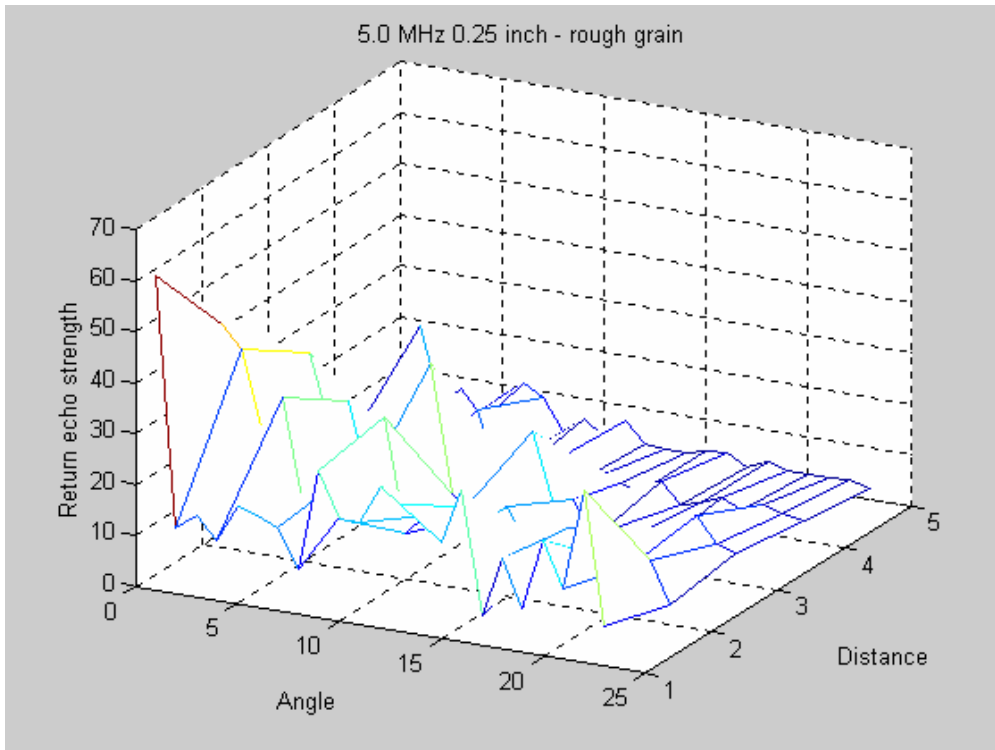


Figure 46: 5.0 MHz, 0.25 inch Diameter Using Rough Grain

With a smaller and higher frequency transducer, both angle and distance of the object affects the echo strength at a considerable amount. The strength of the signal is weaker at further distances due to dissipation of the pulse at further distance (called far field). The far field is affected by the transducer frequency as well as the diameter of the transducer. In this setup, the far field is in effect at distances over 5cm (see also, 3.2.2 Relevant Aspects of Pulse-Echo Ultrasound Systems, Page 22 of this document).

Overall, the test shows that in the scaled down lab tank setup, the system should use a transducer above or equal to 2.25 MHz with the desired diameter to cover enough distance to do a realistic scaled testing.

5.2 Lab Tank Testing

To test the ultrasound unit and to collect data for testing, we needed an experimental setup to continuously collect information from boats passing by the transducer. Once the transducer, pulser/receiver, oscilloscope, GPIB and the PC was set up to collect echo signals from two transducers using MatLab, the transducer was submerged into the tank and fixed on one side. In order to have boats to pass by the transducer's beam automatically, they needed to be pulled around the tank continuously.

The boat passing test was conducted using an X-Y plotter driven by a signal generator. Figure 47 shows the setup.

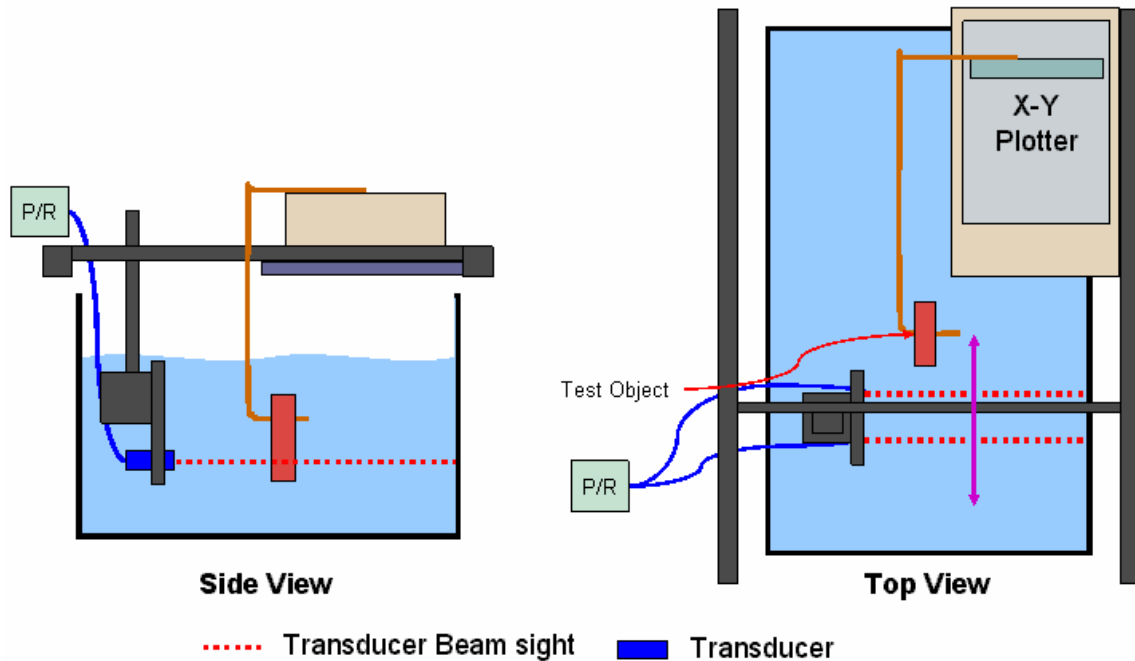


Figure 47: Test Setup Using X-Y Plotter

The pulser/receiver is connected to two ultrasound transducers through the multiplexer. The two transducers are submerged underwater in the tank and fixed at a width of 3 centimeters apart facing sideways to the motion of the X-Y plotter arm. The arm on the X-Y plotter is connected to a bronze arm which extends down into the water where it has a sponge attached to the end. The sponge was used as a model boat and passed back and forth in front of the transducers by inputting a sine-wave into the control of the X-Y plotter. The data obtained from this setup is shown in Figure 48 and Figure 49.

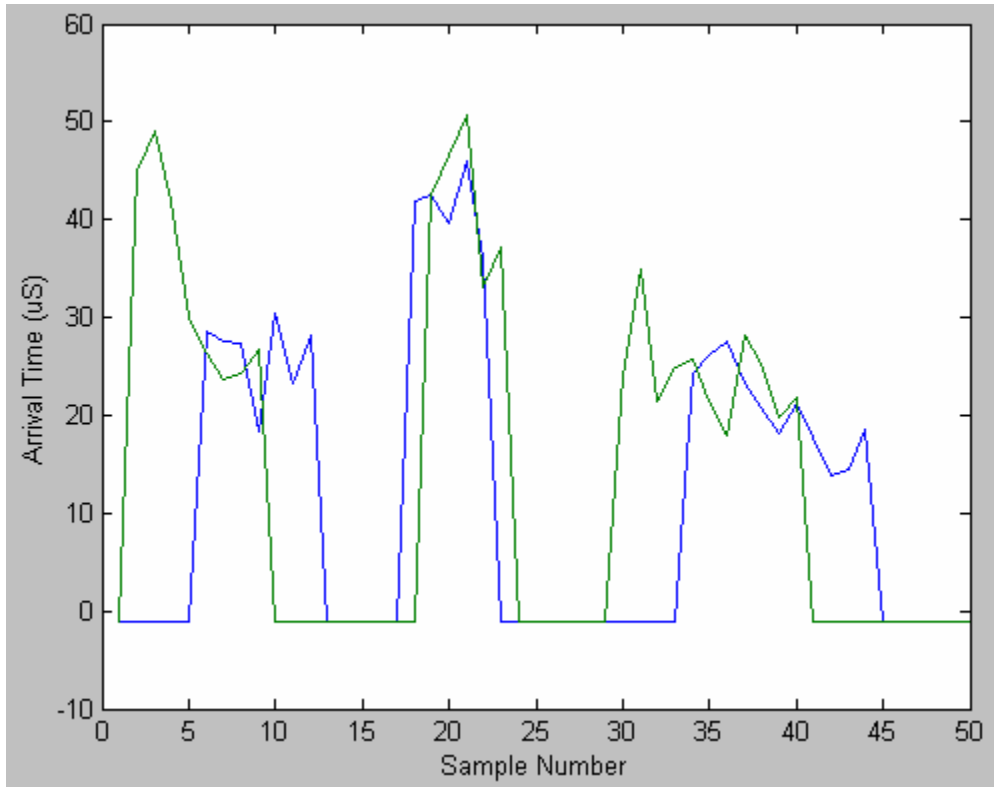


Figure 48: Sample Arrival Time

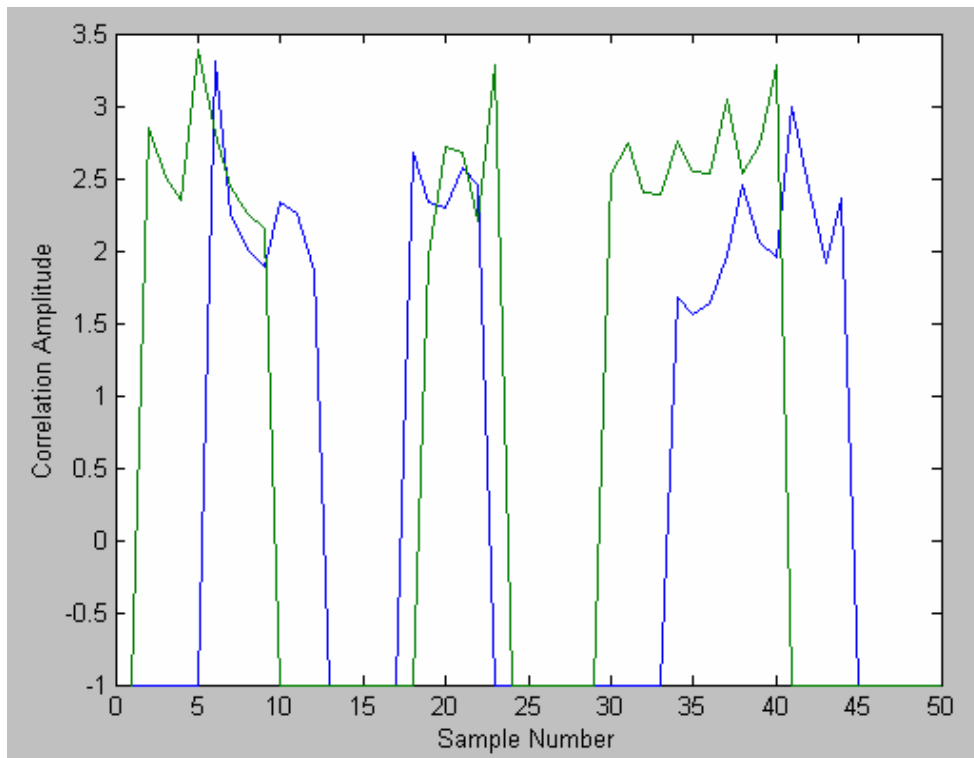


Figure 49: Sample Correlation Amplitude

The green line shows the echo arrival time obtained from one transducer and the blue line shows the echo arrival time obtained from the other transducer. The first pass was first observed by the green transducer, then the blue transducer. A few samples later, the object has passed the green transducer and later passes the blue transducer. Since this sample is a clear sample showing that an object of considerable size has passed in front of the set of transducers coming from one way and going out the other, this is considered as a boat. The second pass comes in faster, noticeable by the time in between the first detection by each transducer, from blue transducer side. The third pass comes from green transducer side at about the same speed as first pass. All of this information is processed by the detection algorithm and a log is created. The log is shown in Figure 50.

```
Timestamp: Fri Apr 04 15:00:39 2003
Velocity: -0.031 km/h Length: 0.064 m

Timestamp: Fri Apr 04 15:00:39 2003
Velocity: 0.072 km/h Length: 0.090 m

Timestamp: Fri Apr 04 15:00:39 2003
Velocity: -0.031 km/h Length: 0.090 m
```

Figure 50: Sample Log File

The data shown in the log file in Figure 50 shows that the first and third objects logged have the same velocity while the second pass is faster and in the opposite direction. This is all consistent with what is shown in the transducer data input plot in Figure 48. Also, the tested object was measured to be 9.0 cm (0.090m), which is also consistent with the logged information except at the first transducer. This inconsistency is due to the fact that the object accelerated after being detected by both transducers. Since the system calculates the object speed at the instant when the object is detected by

both transducers, and the length is calculated after the boat has cleared the first transducer it was detected by, when the speed changes between those time intervals, the length calculation can be thrown off.

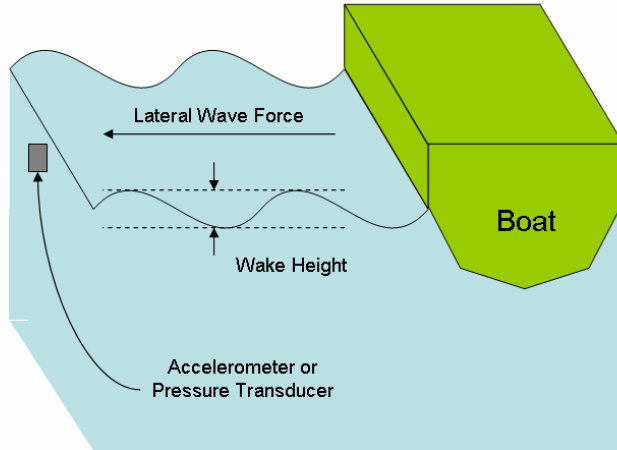
The testing results show that the basic detection of passing objects works as desired in the controlled environment. The test was done using the setup for the first generation. More testing results obtained using the first generation are included on the CD-ROM in the folder “TestResults.”

5.3 Second Generation Development and Testing

The second generation system has been developed to a state where it is almost completely ported from the first generation. Currently the system is capable of capturing signals from the pulser/receiver using the NI5102 digitizer card. Also, the captured signal can be processed to extract echo presence, correlation amplitude, signal width and signal distance. The system, though, has some issues in the multiplexer triggering where the triggering from the digitizer card is not regular, rendering the system unable to consistently detect boat traffic. The detail of the current state can be found in Appendix A: Software Operation Manual.

6. Future Implementation

The system as it currently stands requires minor adjusting and fixes to operate in its intended way for testing in the canals of Venice. The details on required fixes for the laptop system are detailed in Appendix A. There are also some other improvements to the system which would help researchers in Venice to relate boat traffic and canal damage. The one vital improvement would be to include records of wakes and lateral pressure exerted on the canal wall by boat traffic. If such information can be recorded along with boat traffic, the information can be used to correlate different types of traffic to the amount of disturbance it causes to the canal. This could possibly be implemented using some form of accelerometer or pressure transducer. With all of this information, the intelligent system may be able to implement boat type recognition with a fairly accurate recognition level. Figure 51 shows possible information collected in the future system and the illustration of how it could be done.



In the Future...

- Use Accelerometer or Pressure Transducer to measure force exerted on the wall.
- This data can be used to relate traffic and canal damage

With all the information, the log may look like the following for each station:
 (note: the values in the table is a sample and may not resemble real data)

Time	Boat Type	Wake height (cm)	Pressure (?) [*]	Speed (km/h)	Length (m)
2003-7-15-11:23:42	Motor-boat C	35.5	23	14.3	6.1
2003-7-15-11:24:32	Gondola A	9.3	10	6.2	5.5
2003-7-15-11:25:35	Freight D	11.6	18	8	8.2

* The pressure measurement unit is unknown at the point of this writing, and the values may be unreasonably off

Figure 51: Future Improvements

Since the improvements use a system based on waveforms, it is compatible with the current data capturing method and should not be too much trouble to integrate into the existing system.

7. Conclusions

This Ultrasound Boat Monitoring System Major Qualification Project has succeeded in initiating a project to create an automated traffic information recording device. This device is intended for recording boat traffic in the canals of Venice along with information that will help researchers to study the relationship between boat traffic and the damage to the canal walls. The development of the project was split up into three stages (see Figure 12).

This portion of the project concludes the first stage of development for the boat detection and boat information collection. The project has proved the conceptual boat detection using two transducers in a controlled environment. Also, the project has mostly finished porting the code-base to a laptop system where the entire data capture, signal processing and detection are integrated into one machine.

The next step in the second stage of the project is to fix and finalize the porting of the code-base to the laptop based system and test the system in a life-scale environment where more errors are likely to occur.

Appendix A: Software Operation Manual

The Boat Monitoring System is designed to be fully configurable through software and the controls are completely executed by software. This document will cover the current state of the system and how well it accomplishes the above goal as well as give the reader an understanding of how the system works so future modifications can be made easier. This document is intended for the use of any person who wants to understand how the system works and reading it is recommended as the first step for anyone who wants to modify the system.

A.1 Current System Capabilities

The current boat monitoring system accomplishes many of the tasks that the laptop system described in the main document is supposed to do. The current system is mostly configurable and is capable of reading data from multiple transducers through an NI5102 digitizer card. The monitor system is also currently capable of processing raw signals and extracting data about passing objects from the raw signals. There still seem to be some timing issues that prevent the system from fully running properly though. These will be described in more detail later in the document.

A.2 Implementation Details

The system begins in a simple command line startup menu. From this menu the user can configure system parameters before starting the actual signal acquisition and processing. Parameters such as the distance between the two transducers and various thresholds can be configured. A previous version of the software also allowed the user to specify a file with processed signal data to be read into the system to allow for testing of the boat detection logic. This previous version is included in

MonitorTest.zip on the accompanying CD. It can be used in association with the information obtained in MatLab from the first generation system to emulate a full system.

The current system is designed to acquire and process signals on its own using an internal NI5102 digitizer card. The first step taken after configuring the system is the initialization of the digitizer card. This includes setting up the vertical and horizontal parameters for the input channel as well as setting up the trigger. The system currently uses the second input channel as a trigger channel. The Sync Out of the pulser/receiver is connected to this channel to facilitate synchronization of the system. In the current system it is necessary to set the pulser/receiver frequency to the same frequency specified in the software configuration of the system. This trigger signal is used as a positive edge trigger for the purposes of triggering the digitizer. The PFI 1 line is configured to output a pulse every time an acquisition is successfully completed by the digitizer. This leaves only the action of fetching the data from the digitizer at appropriate times. An additional program is included in 5102test.zip that simply initializes the digitizer and constantly reads data from it. This program is useful for testing operations on the digitizer directly.

Once the digitizer has been configured the next step is to acquire a “known echo” to use for later correlation. This echo is expected to be a relatively clean echo off a solid surface to give an example of an expected echo return off of an object like a boat. The system pauses to allow the user to set the system up for the acquisition of this “known echo” and then pauses again after the echo is acquired to allow the user to

reset the system to its standard state. This “known echo” is then stored in memory for later use in correlation functions.

Once the “known echo” is acquired the system can begin running. The system fetches the first signal from the digitizer and processes it to obtain four parameters; signal presence, analytical signal magnitude (correlation), the signal width, and the signal return time. Each of these parameters is extracted using functionality from the MatLab C++ function libraries. This allowed the porting of the code from MatLab to C++ to be performed smoothly and clearly. Directly after the signal has been fetched by the system the digitizer is told to get ready to acquire again as well. The fetch function will block until a signal is received so by setting the digitizer’s trigger to our sampling rate we can use this to control the speed of signal acquisition in the system. In addition, once a signal is fully acquired the MUX circuit is switched to allow for input from the other transducer on the next pulse. This is accomplished through the PFI 1 line that was configured when the digitizer was initialized.

After the signal has been completely processed it is then interpreted by the system to determine the presence of objects such as boats. Each boat in the system is stored as a class in an internal list. As new objects are found in the system they are added to the list and when they are no longer reflecting signals they are removed. Boats in the system are distinguished from each other by their distances from the transducer. One of the configurable parameters modifies the range of distances allowed to be attributed to the same boat. The default is that any signals reflecting off of objects within 50 centimeters of each other are considered to be from the same object. It would be difficult for two boats to have their hulls less than 50 centimeters

apart and the shape of boat hulls dictates that two consecutive signals will rarely return from the exact same distance.

If the system determines that the most recent signal is valid but does not coincide with any boats currently in the system it adds a new boat to the system. At this point in time it also timestamps the boat acquisition for later use. If there is a match to a current boat in the system one of two things happens; if the current transducer is the same as the transducer that first “saw” the boat then the boat’s hit count is increased by one. Each hit essentially signifies that the boat was in front of the transducer for two pulses (as the other pulse was from the other transducer). If the current transducer is the other transducer then we can calculate the boat’s velocity. From a combination of the hit count and the system frequency we can calculate how long it took the boat to move from the first transducer to the second. We can then use the configurable distance between the transducers to determine the speed using the formula $s = \frac{d}{t}$. The velocity is determined by which transducer first “saw” the boat.

Any boats in the system that do not coincide with the current signal automatically have their miss counts updated as well. If, at any time, a boat’s miss count exceeds the configurable threshold, that boat is removed from the system. A few factors determine whether it is logged or not before it is removed though. If the boat does not have a velocity then it is instantly dropped as this indicates that the object never passed in front of one of the transducers and probably was not a boat. Provided the boat has a velocity, its length is calculated using its velocity, hit count, and the system sampling rate. From the hit count and sampling rate the time that the boat was

in front of the transducer is calculated. This is then combined with the speed using the formula $l = v * t$ to calculate the length of the object. The boat is then logged with its timestamp, velocity, and length information to a file on the hard drive. (currently c:\vlog.txt)

The current system also displays some information about boats in the system after every signal is received. This is purely for debugging purposes and may be removed in the final implementation. The system is designed to run until the user wants to stop it. If, at any time, the user hits a button the system will pause. If 'q' or 'Q' is typed and enter is pressed the system will exit. If anything else is pressed it will continue after enter is pressed. This is simply to allow for a method of gracefully exiting the program whenever necessary. The program will also exit if there are ever any errors. Whenever the program is set to quit it will always check the current boats in the system to ensure that any that should be logged are entered into the log file. This ensures that data isn't lost if the program quits. The log file is also opened and closed each time it is accessed so that data won't be lost if the program is terminated and the file isn't closed.

A.3 Known Issues

There are a couple of known issues with the current system. The first is a small issue with the correlation amplitude. Currently a good value for the correlation amplitude threshold has not been found. It seems that a correlation above 2 or so should definitely indicate a good signal but there have been some issues. It seems that the major issue is signal noise. Any noise that is present in the "known echo" and repeated in any way in the subsequent acquired echoes cause the system to have very

similar correlation amplitudes for different signals. The prevalence of the noise ends of hurting the correlation calculation and making a good threshold difficult to determine. In order to eliminate this problem, the “known echo” has to be as clean as possible. This will ensure that any later noise will not correlate with the “known echo” and pollute the correlation amplitude.

The other major problem with the current system centers on the transducer switching. It seems that the current system is not consistently receiving echoes in an alternating fashion from the two transducers. Evidence points to the transducers switching irregularly; most probably switching too often. Our best guess as to the cause has to do with the pulse generated when a signal is fully acquired. Our theory is that this is not only triggering when a full echo is received, but also when a full trigger is received on the second channel. The best solution to this problem would seem to be to manually set the PFI 1 line voltage high and then low again after each signal is received. We did not have enough time to figure out how to do this though. The solution might be in the programming examples on National Instruments’ website (www.ni.com) but we were unable to find the solution in the time we had. Also, the trigger for the digitizer should be generated from the internal clock as opposed to through the second channel. This change may also eliminate the problem.

A.4 Possible Improvements

There are a few possible improvements to this system that we have recognized but did not have the time to implement. The first has to do with the configuration of the system. It might be a good idea to modify the configuration function to allow the user to create a file to specify the configuration information. This way the user would not have to reenter the information each time he/she ran the program. This should not be too hard to implement and would probably save a lot of time in the long run.

The other two improvements are related to each other. The first also relates to the configuration of the system. It might be easier to eliminate the sampling frequency parameter from the configuration and instead have a canal width parameter. From this canal width the program could determine how long it could take a signal to return after reflecting off an object and from this information it could determine what the fastest reasonable sampling rate would be while still allowing signals to fully propagate and return off of objects. The system can determine its best sampling rate and run at that speed without users having to figure it out on their own.

In addition, it might be good if the entire system is synchronized through the software. Currently the pulser/receiver frequency has to be set to match the system frequency. From looking at the manuals for the NI5102 it seems that the PFI 2 line can probably be set to create a pulse at a programmable frequency. If the system could be designed to use this feature to trigger the pulser/receiver externally then the whole system can be synchronized in software. All of the timing will be handled by hardware but all of the hardware could be synchronized by the software using the same sampling frequency. This would make system configuration much easier and straightforward.

In addition, it would solidify the synchronization of the system timing so that everything will happen at the right time.

A.5 Conclusion

The current system is very close to being able to be deployed in an actual environment on a laptop using an NI5102 digitizer. Unfortunately though, the improper transducer switching is holding it from being useful at the current time. In light of this information it is suggested that the trigger from PFI 1 to the MUX circuit is cleaned up and tested before any other modifications are made to the system. This may solve the whole problem, in which case the system would be almost ready for deployment. The other suggestions in the possible improvements section would probably also make for a better system for testing.