Development of a Small Sonar Altimeter and Constant Altitude Controller for a Miniature Autonomous Underwater Vehicle

Jessica Luan

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Dr. Daniel Stilwell, Chair
Dr. William Baumann
Dr. Chris Wyatt

Feb 8, 2005
Blacksburg, Virginia

Development of a Small Sonar Altimeter and Constant Altitude Controller for a Miniature Autonomous Underwater Vehicle

Jessica Luan

ABSTRACT

Miniature Autonomous Underwater Vehicles are a major area of research and development today. Because of their size and agility, they are capable of exploring and operating in smaller bodies of water in addition to areas of the ocean that would be out of reach for a larger vehicle. Being autonomous requires that the system must be capable of performing without the need for human supervision, so use of external sensors such as sonar are needed to ensure the safety of the vehicle during missions. However, since all of the onboard instrumentation and external equipment must also be miniature in size, the implementation of a small sonar system is desirable.

This thesis contains a brief introduction to sound and sonar, leading into a description of the design and development of a small, inexpensive sonar altimeter. Piezoelectric material is used for transduction in the sonar system while a PIC microcontroller processes the return signals from the water. This altimeter was made to be implemented on a miniature autonomous underwater vehicle developed by the Autonomous Systems and Controls Laboratory at Virginia Polytechnic Institute.

In addition to being capable of reporting ocean depths, sonar systems can be used to aid in the navigation of underwater vehicles. A constant altitude controller based on sonar data has been designed, tested, and implemented on the autonomous underwater vehicle. Possibilities for an obstacle avoidance system involving sonar are also discussed in this thesis.

Acknowledgments

First I'd like to express my gratitude to my advisor, Dr. Stilwell, for all his patience and help with research, and for allowing me to be part of his ASCL lab. Thanks to the rest of my committee; Dr. Baumann, for his endless optimism and his help in troubleshooting my circuit, and Dr. Wyatt, who agreed to serve on my committee on such short notice.

There would not be a thesis without the help of Carl Wick, to whom I cannot thank enough for all his guidance and assistance. My complete gratitude goes to him for taking the time and pains to help me debug the circuit, and for always responding so promptly and thoroughly to my ceaseless supply of questions.

A great deal of thanks goes to Irene Robb and the AirMar Corporation for their time, help, and generosity. Thanks also goes to Mr. Carvel Horton at the Fiber & Electro-Optics Research Center for all his useful suggestions for improvement of the altimeter design.

Thanks to everyone in the lab: Bill, my former colleague, for never letting a moment pass without giving me a hard time, and Jan, who is the reigning king of pool and darts. Thanks to Gray for all his good advice, and to Aditya, for the exorbitant amount of help he provided with programming and software, and for never grumbling too audibly about the group lunches I imposed on everyone. Much thanks to Chris for paddling an overstuffed kayak around the murky duck pond during the early stages of sonar testing.

Finally, I'd like to thank Aaron, for his never ending help and support, for always volunteering his time when I needed an extra hand, and for always being there to push me forward when I fell back.

Contents

1 Introduction **1**

2 Sonar and Underwater Acoustics **4**

2.1 Sound properties 5

2.2 Sonar Transducers 7

2.3 The Sonar Equations 9

3 Sonar Design and Development **12**

3.1 Transmitter Circuit Design and Operation 12

3.2 Signal Processor Circuit Design and Operation 17

3.2.1 Microprocessor Program 20

3.3 Layout and Board Construction 23

3.4 Results 25

4 AirMar Triducer and Altitude Controller Design **28**

4.1 Depth Sounder 28

4.2 Controller Design 30

4.2.1 Depth Controller 30

4.2.2 Altitude Controller 32

4.3 Results 32

5 Literature Survey on Sonar-Based Obstacle Avoidance **36**

5.1 Single Transducer OAS 36
5.2 Multiple Transducer OAS 37
5.3 Incorporating an Obstacle Avoidance System in the Control System of an AUV 39
5.4 Working with Mission Goals 40

6 Conclusions and Future Work **43**
6.1 Conclusions 43
6.2 Future Work 44

A Microprocessor Revised Program Source **48**

B Microprocessor Original program source **62**

C Vita **70**

List of Figures

1.1 The Virginia Tech Miniature Autonomous Underwater Vehicle 3

2.1 Basic Sonar Altimeter Operation 5
2.2 Speed of Sound for Varying Temperatures and Salinity 6
2.3 Speed of Sound for Varying Temperatures and Depth 7
2.4 Radiation of Acoustic Energy from Sonar Transducer 9

3.1 Transmitter Circuit 13
3.2 Waterproof Transducer Used For Testing 16
3.3 Impedance Transformation Circuit for Return Signal 17
3.4 Peak Detector Circuit 18
3.5 Amplifier Circuit 19
3.6 Microcontroller IC and Circuit 20
3.7 Flowchart of Original Microcontroller Program 21
3.8 Flowchart of New Microcontroller Program 21
3.9 Top Layer of Altimeter Printed Circuit Board 24
3.10 Bottom Layer of Altimeter Printed Circuit Board 24
3.11 Waterproof Transducer Circuit 25
3.12 Final Sonar Altimeter 26
3.13 Beam pattern of the sonar altimeter 27

3.14 Logarithmic beam pattern of the sonar altimeter and the -3dB point 27

4.1 AirMar Triducer 29

4.2 Five States of the AUV 31

4.3 Illustration of Control Variables 33

4.4 Virginia Tech Miniature AUV with AirMar Triducer Sensor . . 34

4.5 Altitude controller test with altitude set to 2.1m 34

4.6 Altitude controller test with altitude set to 2.5m 35

5.1 An AUV equipped with only a reactive system in a trap-like situation 39

List of Tables

3.1 VT miniature AUV general specifications 23

4.1 Proportional and Differential Gains for Different States of the AUV. 32

Chapter 1

Introduction

Autonomous underwater vehicles (AUV's) are self-powered, submersible systems that require neither a physical connection to the surface nor real-time remote control to operate. They have the potential to revolutionize access to the oceans with the ability to obtain information on charted and uncharted regions. With their instrumentation and their ability to perform without the presence of a human operator, AUV's are capable of solving problems critical to the marine community such as mine detection and countermeasures, climate change assessment, underwater search and mapping, and marine habitat monitoring [14].

Conventional AUV's carry extensive, state-of-the-art instrumentation to perform underwater tasks [10]. As a result, the vehicles can become large, heavy, and costly. Some of the smallest commercial AUV's today measure 2 meters long and weigh over 80 kilograms [17]. Costs typically reach hundreds of thousands of dollars, not to mention the cost of the special equipment needed to deploy and recover these vehicles.

Miniature AUV's are advantageous because of their size, maneuverability, and cost-effectiveness. A platoon of these vehicles could accomplish the same types of tasks as a large AUV in less time and more efficiently. In addition to performing missions in the ocean, these miniature vehicles can also operate in smaller bodies of water such as lakes and rivers. Their light weight allows them to be carried easily, making deployment much more convenient.

Also, in the event of a catastrophic failure in one or a handful of these vehicles, the financial risk would be minimal in comparison to losing a conventional AUV.

One particular use for AUVs is in the field of bathymetry. Bathymetry involves the measurement of the depths of bodies of water, which is still an ongoing process in many uncharted areas of the ocean. Maps of the ocean currently in use contain dated and sparse information. An increased availability of bathymetric data and a better understanding of the seafloor can assist in problems such as deep-ocean positioning and remove some limitations on various naval operations such as mine warfare, amphibious landings, antisubmarine warfare, and salvage and rescue [7].

Bathymetry involves the use of equipment known as SONAR, short for *SO*und *NA*vigation and *R*anging. These systems are instrumental for AUV's in underwater research and can provide various kinds of data depending on the type of sonar used. Some examples of sonar devices are depth sounders, obstacle avoidance sensors, subbottom profilers, side-scan sonar systems, and doppler navigation systems. The depth sounder sends out pings down toward the seafloor and returns the distance based on the return time. Common frequencies used in this application are 100 kHz and 200 kHz. Obstacle avoidance sensors ping the area in front of the vehicle to avoid collisions and potentially dangerous situations. Subbottom profilers use lower frequencies with large impulse-type sources to penetrate and characterize the bottom of the sea. Side-scan sonar are side-ways looking sonar that can map the seafloor orthogonal to the ship's trajectory. In Doppler navigation, two transducers are used to determine speed from the doppler shift of bottom returns [23]. The sonar devices that are of the most interest in this thesis are the depth sounder and the obstacle avoidance sensor.

Due to hard limitations on space in a miniature AUV, the instrumentation used must also be small. Most of the commercial sonar systems today are too large to implement in vehicles of this size. In addition to space constraints, the Virginia Tech miniature Autonomous Underwater Vehicle, seen in Figure 1.1, includes another constraint on cost. The goal of this thesis was to develop a small, inexpensive bathymetric sonar in addition to a controller

Figure 1.1: The Virginia Tech Miniature Autonomous Underwater Vehicle

for navigation at a constant, predetermined altitude above the seafloor u sing sonar data. This sonar and controller is to be implemented in the Virginia Tech miniature AUV developed by the Autonomous Systems and Controls Laboratory. Chapter 2 begins with an introduction to underwater sound and sonar and discusses the formation of the sonar equations. Following this is a thorough exposition in Chapter 3 on the development and construction of a sonar altimeter. The controller used on the AUV for constant altitude navigation is described in Chapter 4. Since there is no human operator involved in controlling these autonomous vehicles, obstacle avoidance systems are an essential element in AUV missions. Chapter 5 discusses sonar-based obstacle avoidance as a future application for the sonar on the Virginia Tech miniature AUV.

Chapter 2

Sonar and Underwater Acoustics

The practice of obtaining underwater information through acoustic signals has actually been around for many centuries. In the year 1490, distinguished artist and engineer Lenoardo da Vinci made an observation on the subject. He wrote [6], "If you cause your ship to stop, and place the head of a long tube in the water and place the outer extremity to your ear, you will hear ships at a great distance from you."

Indeed, sounds can be heard very well underwater. They propagate better in water than they do in air. One of the first measurements of underwater sound was carried out in 1826 by Daniel Colladon, a Swiss physicist, and Charles Sturm, a French mathematician. They carried out an experiment to measure the speed of sound in Lake Geneva. In this experiment, the striking of a bell submerged underwater and a simultaneously triggered flash of light that could be seen by an observer 15 km away was timed with a stopwatch. The timing begins the moment the flash of light is seen and ends when the sound of the signal is heard about ten seconds later through a tube set up in the manner da Vinci had once described. Remarkably, Colladon and Sturm were able to determine the speed of sound with a great degree of accuracy. They measured a speed of $1435\frac{m}{s}$ in 8°C water, which was only $3\frac{m}{s}$ less than the actual value [8].

During each World War, there was a large demand for more underwater research which lead to breakthroughs in sonar development. There are essentially two types of sonar, active and passive. Passive sonar systems are receivers for acoustic energy in the water. They essentially "listen" for sound radiated by a target, such as a ship, submarine, sea mammal, or seismic activity [13]. Da Vinci's listening tube is an example of a passive sonar system. Active sonars are transmitters and receivers. They send out a signal at a specific frequency and then listen for the echoes reflected back from the target. Depth sounders are an example of an active sonar system. Figure 2.1 shows the basic operation of a depth sounder.

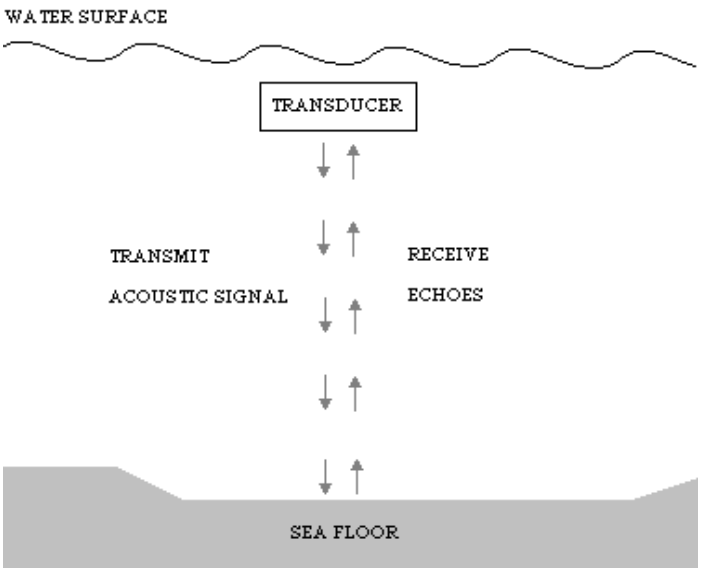

Figure 2.1: Basic Sonar Altimeter Operation

2.1 Sound properties

Why use sonar for underwater operations and not radar? Unlike sound, light and radio waves are types of electromagnetic waves. They can propagate very well in a vacuum, but in mediums with high conductivity like seawater, these signals experience severe attenuation and scattering. Acoustic waves, on the other hand, require a medium and cannot exist in a vacuum. This is because sound is actually a pressure wave that is created through the vibration

of material in a medium. In water, acoustic signals travel as compressional waves that propagate longitudinally at the local speed of sound, independent of the specific characteristics of the signal such as frequency and waveform. The local speed of sound, however, can vary depending on the water temperature, pressure, and salinity.

A simplified formula for the speed of sound relating these quantities was proposed in 1969 by CC Leroy [15].

$$\begin{aligned} c &= 1492.9 + 3(T-10) - 6\times10^{-3}(T-10)^2 - 4\times10^{-2}(T-18)^2 \\ &\quad +1.2(S-35) - 10^{-2}(T-18)(S-35) + Z/61 \end{aligned} \tag{2.1}$$

Here, pressure is expressed as depth Z in meters. This equation yields a speed accurate to $0.1\frac{m}{s}$ for a temperature less than 20°C and in depths less than 8000m. Figure 2.2 displays how the speed of sound changes with temperature and salinity. The values of salinity shown in the graph range from 29-45 ppt. In Figure 2.3 depth and temperature are the variables, with depth ranging from 0-8000 m.

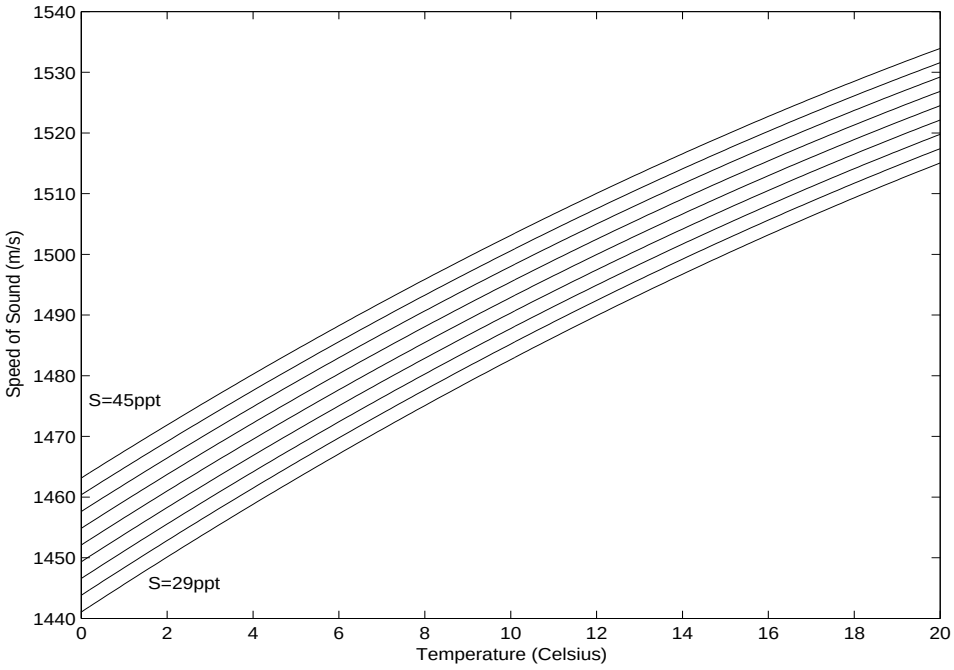

Figure 2.2: Speed of Sound for Varying Temperatures and Salinity

2.2 Sonar Transducers

Sonar Transducers are special devices that can convert electrical energy into acoustic energy and vice versa. Three types of transducer elements are: piezoelectric, electrostrictive, and magnetostrictive.

Piezoelectricity was first discovered in 1880 by the Curie brothers. Materials exhibiting the effect such as quartz and Rochelle-salt started becoming more widely used in sonar since World War I. In piezoelectric transducers, an electric field is applied to crystals that respond by changing dimensions. If the field is alternating, it will cause the crystal to vibrate at the same frequency, radiating acoustic energy. These crystals have a very high impedance, so application of a considerably large voltage will generate only a moderate acoustic signal [22]. The amount of deformation is roughly equal to:

$$\xi \approx rs + kD, \tag{2.2}$$

where s is the amount of mechanical stress, D is the displacement field, and

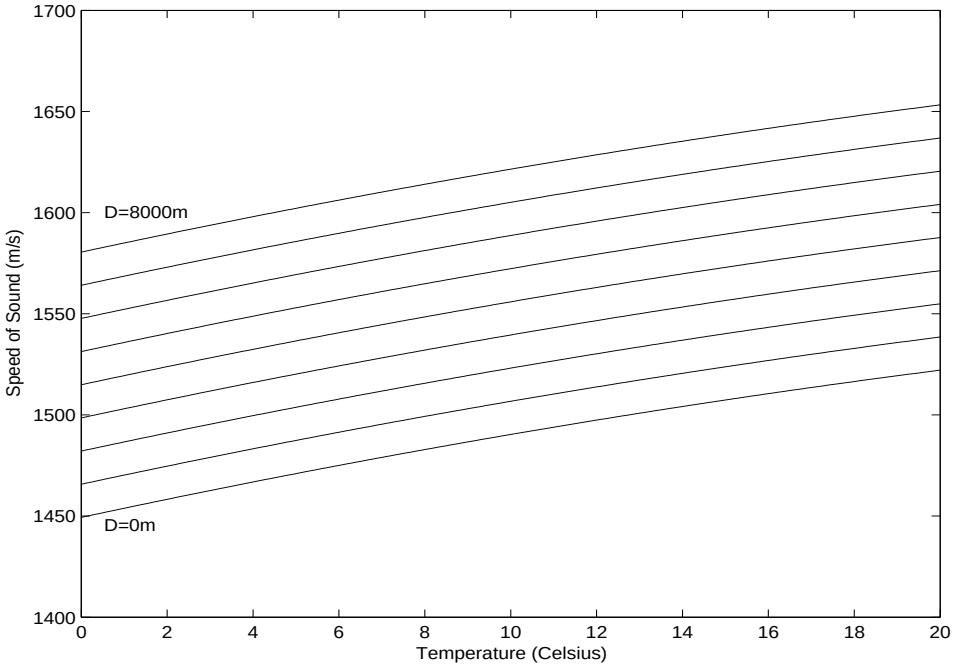

Figure 2.3: Speed of Sound for Varying Temperatures and Depth

r and k are constants specific to the material used. This equation shows that the the deformation is proportional to the field applied.

Most materials exhibit the electrostrictive effect, where an electric field causes a deformation in the material that is proportional to the square of the displacement field. This can be expressed quantitatively as:

$$\xi \approx rs + kD^2 \tag{2.3}$$

However, with the polarization of certain ceramics such as barium titanate and lead zirconate-titanate, the application of an alternating field produces a deformation roughly equal to

$$\xi \approx rs + k(D + \delta)^2 \tag{2.4}$$

where δ is the incremental displacement field. If D is much greater than this value, the incremental deformation can be described as:

$$\begin{aligned} \xi' =& \xi - kD^2 \\ \approx& rs + kD\delta, \end{aligned}$$

creating a linear relationship between the incremental deformation and the incremental displacement field [9]. The only problem is that these materials will not permanently retain a significant remanent pole, so this linearized model will not be able to exactly predict the performance of the transducer.

Magnetostrictive materials change length when a magnetic field is applied. This phenomenon was discovered around 1840 when it was found that musical sounds could be heard when the current in a coil of wire was changed near the poles of a horseshoe magnet. James Joule performed many experiments during this time, measuring the change in length of magnetostrictive materials [23]. Nickel is the most popular as it seems to demonstrate this effect more than any other material. When a coil of wire is wrapped around a nickel rod, a magnetic field is created when a current flows through the wire. The dimension of the rod decreases along the axis parallel to the field and is independent of the direction of the magnetic field. Unless the rod is polarized, the frequency of the motion of the rod will be double that of the oscillating

field [22].

2.3 The Sonar Equations

The central problem in any sonar system is separating the desired signal from unwanted background noise. Noise comes from numerous sources such as waves, other ocean vessels, electronics on the sonar unit, and also any extraneous vibrations. The signal also suffers from transmission loss. As a depth sounder radiates a pulse, the signal spreads conically away from the transducer, as pictured in Figure 2.4. This results in a decrease in the amount of energy per unit area known as spreading loss. There are also attenuation effects on the pulse, or absorption losses. The spreading and absorption losses together make up the transmission loss.

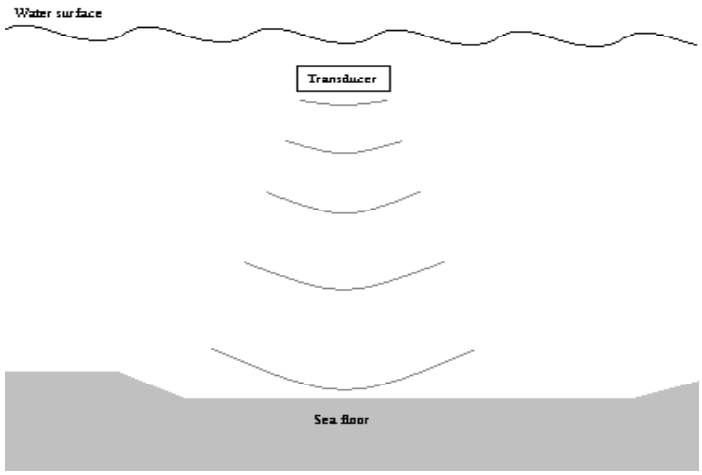

Figure 2.4: Radiation of Acoustic Energy from Sonar Transducer

When the pulse hits the target, the signal scatters. A portion of it is transmitted into the target, and the rest of it is reflected back into the water. The amount that is reflected back differs depending on the characteristics of the target. For example, a soft seabed has a high level of absorption, so the reflection is minimal. Harder substances such as rocks and concrete, however,

will absorb very little energy, so the return signal is much stronger. The amount reflected back is known as the backscattering strength of the target.

The signal-to-noise ratio (SNR) is the ratio of the strength of the received signal to the strength of the noise. In general, the signal may be detected on average 50% of the time through ambient noise if

$$\frac{S}{N} \geq M, \tag{2.5}$$

where M is known as the recognition differential [9]. S is the average power of the signal and N is the average power of the noise measured during the same time period. By expressing each quantity in terms of decibels, (2.5) can be written as

$$S - N \geq M. \tag{2.6}$$

For passive systems, the acoustic energy that emanates from a target to the hydrophone or receiver experiences transmission loss. The signal level received by the hydrophone is now the strength of the signal at the source (S_0) minus transmission loss (T_L). Equation 2.6 now becomes:

$$S_0 - T_L - N \geq M. \tag{2.7}$$

Directional receivers reduce the noise level by a quantity known as the directivity index (D_I). These receivers consist of an array of hydrophones used to cancel out noise coming from other directions. This process is called *spatial processing gain* and is used to increase the main lobe of the beam pattern coming from the direction of the target. Equation 2.8 is the complete passive sonar equation.

$$S_0 - T_L \geq N - D_I + M. \tag{2.8}$$

In an active sonar system, a pulse of energy is sent from the transducer and propagates through the water to the target, accruing a one-way transmission loss T_L. Once it strikes the target, the backscattering strength B_S of

the target (also known as target strength) determines the amount of acoustic energy that will be reflected. The area that the beam strikes is known as the ensonified region. The reflected signal, or echo, coming back towards the receiver suffers another one-way transmission loss. When the transmitter and receiver are at the same location, the equation for an active sonar with noise-limited performance becomes:

$$S_0 - 2T_L + B_S \geq N - D_I + M. \tag{2.9}$$

Reverberation, R, occurs when the acoustic signal is scattered from interfering objects such as bubbles, fish, and any other inhomogeneities in the path of the pulse. This reverberation can either add or subtract from the desired signal. Whereas noise power comes from all directions, R comes primarily from the direction of the transmitted pulse. When the system is reverberation-limited, the detected noise level is the reverberation level, and the directivity index is no longer a factor since R is directional. The equation for an active sonar with reverberation-limited performance is [12]:

$$S_0 - 2T_L + B_S \geq R + M. \tag{2.10}$$

Each one of the factors affect the performance of any sonar system, from altimeters to doppler navigation systems. The next chapter focuses on the design and development of a small altimeter circuit.

Chapter 3

Sonar Design and Development

In July of 2003 two prototype sonar altimeters designed by Professor Carl Wick of the Naval Academy were sent to the Autonomous Systems and Controls Laboratory at the Virginia Polytechnic Institute along with schematics. The goal was to be able to replicate and improve on this design because these altimeters provided erratic data upon testing.

In this sonar altimeter, a 454,545 Hz electrical signal is sent to a transducer which transmits this signal acoustically into the water. This same transducer then stops transmitting to "listen" for the return signal, or echo. Since multiple echoes are reflected back toward the transducer, the first strong return is used to calculate the depth of the water. The following sections will provide a more detailed description of the operation of this altimeter as well as the steps taken to improve and develop the circuit.

3.1 Transmitter Circuit Design and Operation

Figure 3.1 shows a schematic for the transmitter portion of the sonar circuit. A PIC18F252 microcontroller manufactured by Microchip is used to send out 16 PWM waves at a frequency of 454,545 Hz with a 50% duty cycle during the transmit cycle of the sonar unit. This signal drive is sent to the primary of a hand-wound toroidal transformer. To achieve a high drive voltage, the inductance of the secondary of the transformer is tuned to the capacitance

of a square acoustic tile that serves as the transducer. This emits an output voltage just shy of 500 Volts peak-to-peak. During the receiving period, the duty cycle is brought to 0% while the transducer listens for return signals or echoes.

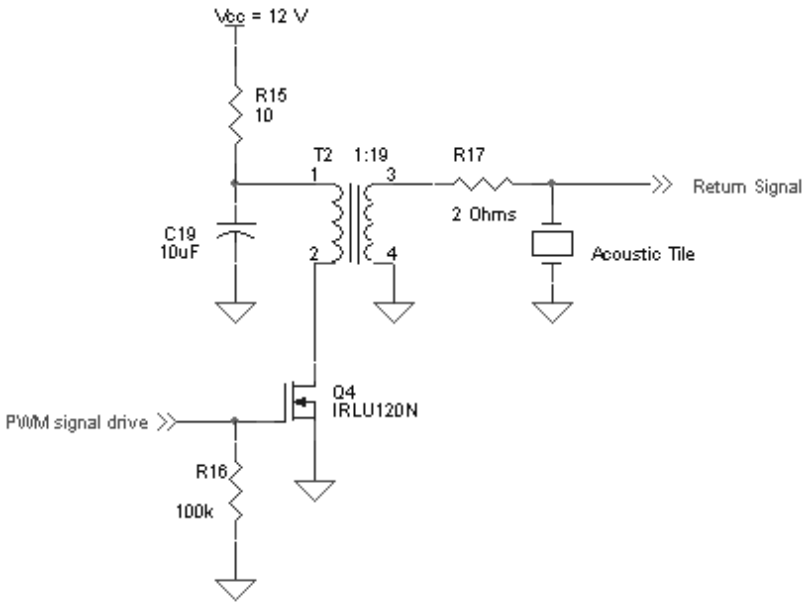

Figure 3.1: Transmitter Circuit

The acoustic tile is a 1" x 1" x 0.04" piezoelectric element from Measurement Specialties, Sensor Products Division. A thin layer of piezoelectric film is placed between two square aluminum conductors. The capacitance across the plates is measured to be about 100 pF. To increase sensitivity, the top conductor is removed and silver ink is painted directly on the face of the tile. The bottom aluminum conductor is connected to ground, while the drive signal is applied to the side painted with silver ink. Piezoelectric elements are used in many sonar based instruments because of its low acoustic impedance, which is only 2.6 times greater than that of water [18]. This makes the transfer of acoustic signals into the water more efficient.

Another useful characteristic of piezoelectric film is its low quality factor, Q. The Q factor is a function of energy stored and power dissipated per cycle, and therefore determines how quickly an object loses its energy. A low Q would denote a system that is highly damped with a short decay period

of oscillation. Due to its low Q, piezoelectric film is capable of emitting a very short pulse, allowing the same device to be used as both transmitter and receiver.

For the toroidal transformer, a ferrite core from Amidon Associates of type "J" material was chosen because of its relatively high permeability and fast switching capabilities. This material is recommended for pulse transformers and has low core loss and volume resistivity at the operating frequency while possessing excellent frequency attenuation for frequencies above 500kHz [4]. The core's high permeability factor allows for a higher inductance value with a fewer number of turns, so a smaller component may be used.

To calculate the number of turns needed, the following equations are used:

$$N = 1000 \times \sqrt{\frac{L}{A_L}} \tag{3.1}$$

where $A_L = 2750$, the inductance index in mH/1000 turns for the particular material and size of the toroid, and L is the inductance in mH needed for the application. The desired inductance for a circuit tuned to 454545 Hz is

$$L = \frac{1}{\omega^2 C} = \frac{1}{(2\pi \times 454545)^2 100 \times 10^{-12}} = 1.2mH$$

Substituting this value for L in (3.1) produces an N of 21. This means that 21 turns of the secondary of the transformer is needed to achieve this inductance. However, it is still necessary to use an LCR meter to measure the real inductance value, otherwise the circuit will not be tuned to the right frequency. One or two turns had to be removed to match the desired value. To maximize the voltage output, a single turn of wire serves as the primary. A considerable amount of current will pass through this single turn, so a larger diameter wire was recommended by Dr. Carl Wick. Upon testing, #27 gauge wire for both the primary and secondary of the transformer seemed to be sufficient.

A 10uF tantalum capacitor is connected to the primary of the transformer to store the large amount of current necessary to drive the circuit. This component must be mounted very near the primary. During testing of a bread-

boarded design, the capacitor had to be soldered directly to the primary in order to achieve the high voltage output because of the parasitic capacitance the breadboard can add to the circuit as well as any resistance that occurs due to dirty contacts.

A logic level FET tied to the primary provides the AC input needed by the transformer. The PWM drive signal from the microcontroller is sent to the gate of this FET which is tied to ground by a 100k Ohm resistor. A high resistance applied at the gate keeps the FET from turning on unless driven by an external logic signal. In choosing a good FET, it is helpful to look at the $I_D - V_{DS}$ characteristics and become familiar with the amount of current being drawn as well as the drain-to-source resistance of the FET when it is on. One problem seen during breadboarding and testing the circuit was that the FET was causing large voltage drops (7 or 8 Volts) in the 12 Volt power bus whenever it turned on. Upon looking at the $I_D - V_{DS}$ characteristic curve, it was seen that whenever the FET was on, it would draw currents greater than 3 Amps, which was the current limit on the power supply. This forces the power supply out of its linear region of operation and the voltage drops significantly. As the FET switches on and off with the PWM input, a large ripple effect is created in the 12 Volt line.

One way to reduce the size of the ripple in the power line is to use a different FET with a higher on resistance and a lower magnitude $I_D - V_{DS}$ curve. The new FET brings a significant improvement to the circuit. It cleaned up the 12 Volt bus and also increased the output voltage by 50 Volts peak-to-peak. There was still, however, a small 2 to 3 Volt ripple that occurred whenever the FET was being switched on and off. As it turned out, this was caused by the breadboard. The problem with using breadboards is that the connections become very unreliable with high frequencies or high current designs because the spring clips inside the breadboard have a small contact area and can overheat quite easily. These white breadboards are only recommended for circuits using less than 100 mA, but this part of the transmitter circuit uses several Amps of current. Because of the high current in this design, the resistance seen from the path connected to the 10uF capacitor became a good amount larger than the path going to the power supply, so instead of tapping

the stored current in the capacitor, the FET was forcing current from the power supply. To fix this problem, a small 10 Ohm resistor is added between the primary of the transformer and the 12 Volt bus to force the FET to use the stored current in the low impedance capacitor before taking it from the power supply. This addition smoothed the ripple from the 12 Volt line.

Connecting the acoustic tile to the rest of the circuit required some work. Long leads were a must at this point in order to test the circuit, and the tile had to be protected from direct exposure to water. Conductive RTV had been recommended but was not strong enough to hold the wires in place. Several attempts were made ranging from soldering the leads directly to the face of the tile to the use of electrical tape. The electrical tape was used to fortify the connection made by RTV and also to waterproof the tile, and although it did help to hold the leads in place, it also severely attenuated the output signal from the tile. Soldering onto the face of the tile did not work either. As high temperatures can be damaging to piezoelectric materials, a low-melting point solder paste was attempted, but it would not adhere to the surface of the tile.

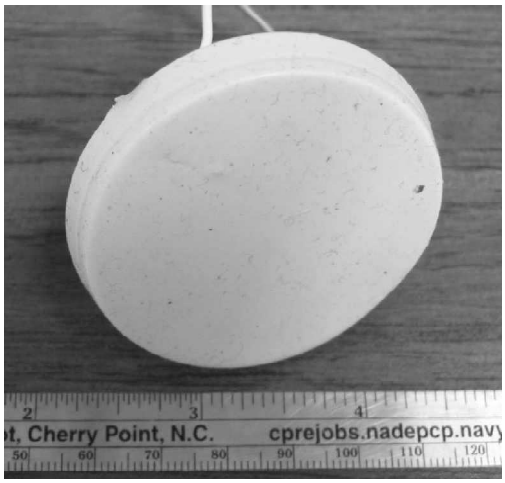

Figure 3.2: Waterproof Transducer Used For Testing

One successful approach was to cut a piece of universal prototyping board about the same size as the tile. Silver ink was applied on one side of the board to connect all the pads and form one large ground plane. Two wires are then soldered to the board. Because of the large voltages, the use of stranded

wire is recommended here to reduce the strain on the connection by increasing the contact area. The negative side of the tile is adhered to the ground plane and connected to one of the wires using a mixture of conductive RTV and regular epoxy and then clamped for 24 hours. The other stranded wire is attached to the positive side of the tile using the same RTV and epoxy blend and allowed to set. A two-part rubber silicone mold is then used to encase and waterproof the board and tile. Figure 3.2 shows the finished product.

3.2 Signal Processor Circuit Design and Operation

A tap is made from the transducer for the return echoes. These echoes typically fall within the range of 50mV to 200mV. A good amplifier is then needed to boost the signal without increasing the noise level. Before the return echoes can reach the amplifier, a buffer is required to convert the high impedance of the piezoelectric tile to match the low input impedance of the amplifier. This transformation also serves to minimize noise and signal loss through the long cable connecting the two boards. The buffer circuit comprises a JFET in a source-follower configuration. Since this circuit is also active during the transmit cycle, a pair of back-to-back Schottky diodes clips the transmitted signal to a safe level for the gate of the JFET. The circuit for the return signal is shown in Figure 3.3.

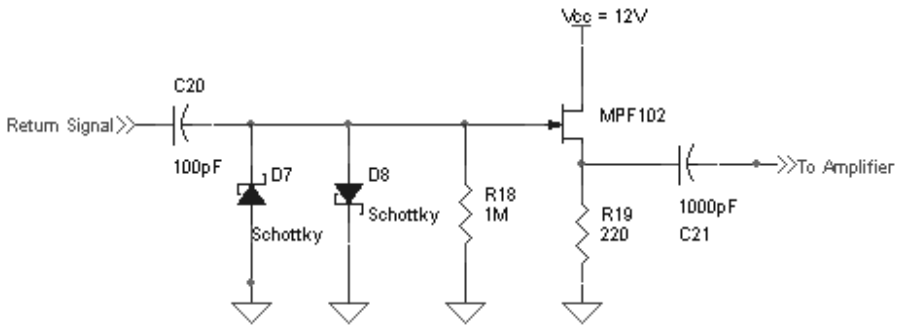

Figure 3.3: Impedance Transformation Circuit for Return Signal

5 Volts is needed to power the amplifier. A voltage regulator not

shown in the schematic is used to convert the 12 Volts used in the power bus to the 5 Volts needed for both the amplifier and the microcontroller. The amplifier chosen for this circuit was the AD605 dual linear-in-dB variable gain amplifier from Analog Devices, which has a high-end gain of 34 dB and very low input noise at maximum gain. This dual amplifier is cascaded with both stages of gain controlled using a 10k Ohm potentiometer. The peak detector circuit in Figure 3.4 rectifies and levels the waveform in order to ascertain the peak values of the return echoes. Samples of the rectified signal are taken at a rate of 28.4 kHz, or once every 16 cycles of the 454545 Hz signal. This corresponds to having a sample every 35.2 μs. At 20°C sound travels in fresh water at the speed of 1,481 $\frac{m}{s}$, so it propagates through one inch of water in 17.2 μs, or 34.4 μs round trip. Thus sampling at a rate of 28.4 kHz is roughly equal to sampling the return waveform every 1 inch round trip that the signal travels in the water.

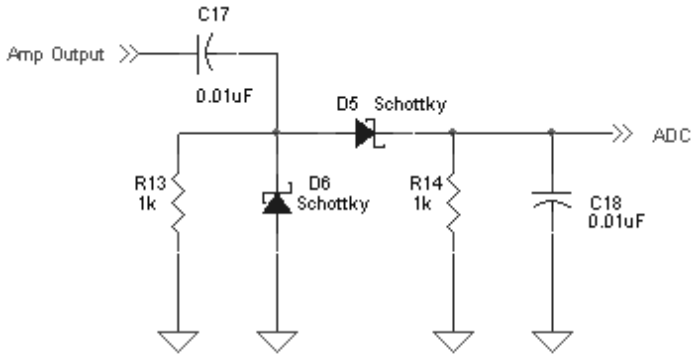

Figure 3.4: Peak Detector Circuit

Noise that enters into this circuit with the echoes will also become amplified. This became a big problem during testing, and when looking at the output of the amplifier through an oscilloscope, the signal could not be detected through all the noise. One method used to reduce the amount of higher frequency noise was to decrease the values of the capacitors used at the inputs and outputs of each stage of the amplifier. One place in particular that this made a difference was at the blanking/gain pins labeled in Figure 3.5 as VGN1 and VGN2. A lower capacitor value here cleaned up the ground and power to the amplifier IC by lowering the cutoff frequency of the low-pass filter circuit and shorting more of the higher frequency noise to ground.

The reference voltage pin was also connected to ground through a lower value capacitor which helps to decrease noise in the same manner. These changes allowed the gain to be reduced and created a cleaner output signal to the A/D converter.

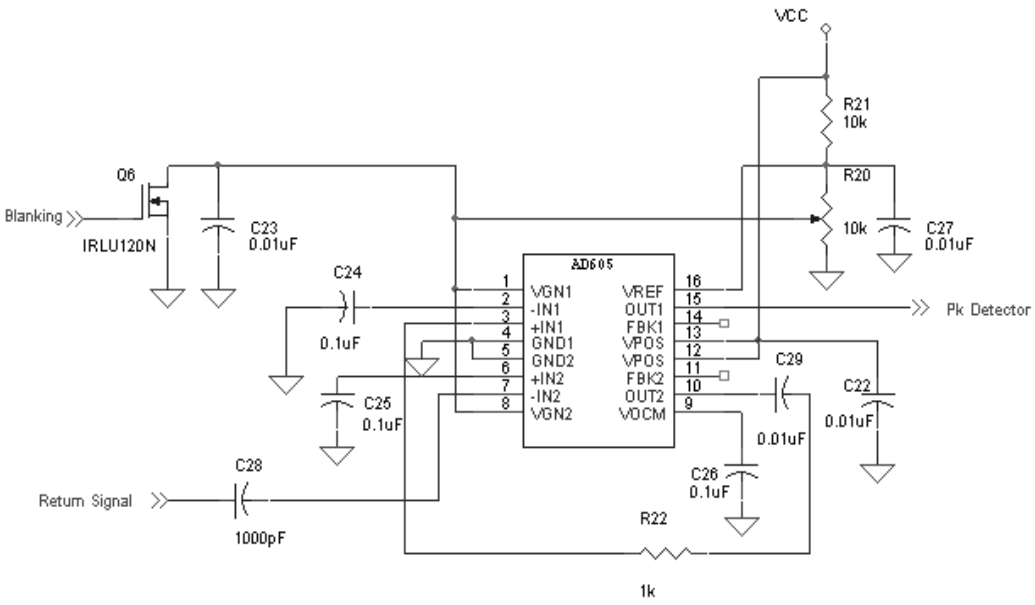

Figure 3.5: Amplifier Circuit

A 10 MHz crystal provides the oscillator frequency for the clock of the microcontroller, which is run by its high speed $\times 4$ phase lock loop, creating an effective frequency of 40 MHz. After some data processing that will be described in more detail in the next section, the depth and signal strength is output through the serial I/O module of the microprocessor at TTL level. In order to be read into a computer serial port, it needs to be converted to RS232. A RS232 driver was used for this purpose, but it was later replaced with a MOSFET and 5k resistor which simplified the circuit. This TTL-to-RS232 circuit is connected to pin 17 of the microcontroller shown in Figure 3.6. Since this conversion is only used in testing and will not be needed once the altimeter is implemented in the vehicle, this portion of the circuit is not included on the printed circuit board.

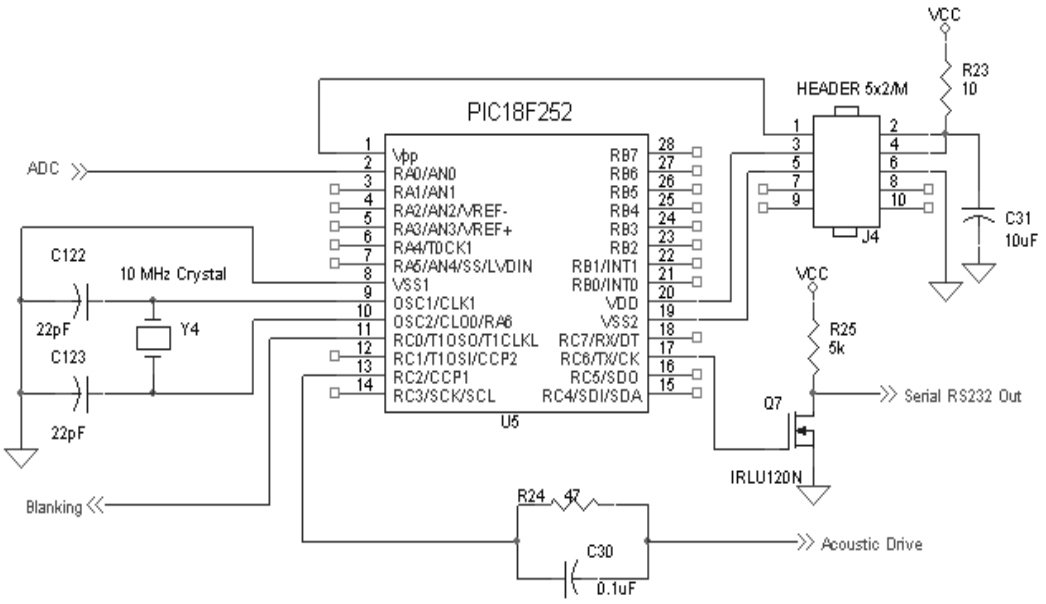

Figure 3.6: Microcontroller IC and Circuit

3.2.1 Microprocessor Program

The original PIC program is written in assembly language. After collecting 256 samples of the return signal, the values are stored in an array and the whole transmit and receive routine repeats 15 more times, collecting 256 samples each time. Each set of new values are added to the old values in the array and then divided by 16 after the last set of samples have been added to form average values. The first four bins of the array are zero because the amplifier is blanked to give the transducer a period of rest between the transmit and receive routines. The values in this array represent the magnitude of the signal, while the bin location represents the distance, since each bin holds values sampled at every inch. The first large bin value and location are then sent to the serial I/O (USART) module of the microprocessor as the signal strength and depth in inches. This I/O module is set at a baud rate of 19,200 bps. Since a total of 16×256 samples are collected with each report at a sampling rate of 29kHz, this results in 7 reports per second. Figure 3.7 shows a simplified flowchart of this program, and the full source code can be found in Appendix B.

Extensive testing was performed with this program implemented in the sonar, the results of which show that it was inadequate in extracting the

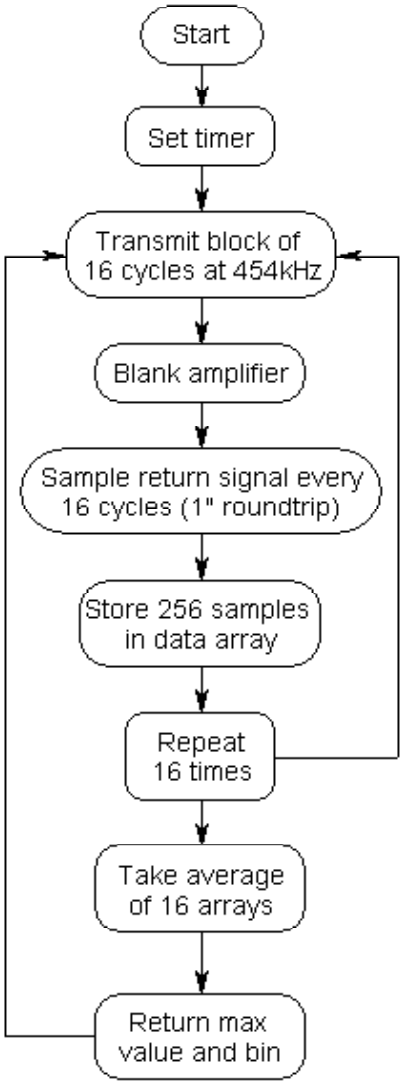

Figure 3.7: Flowchart of Original Microcontroller Program

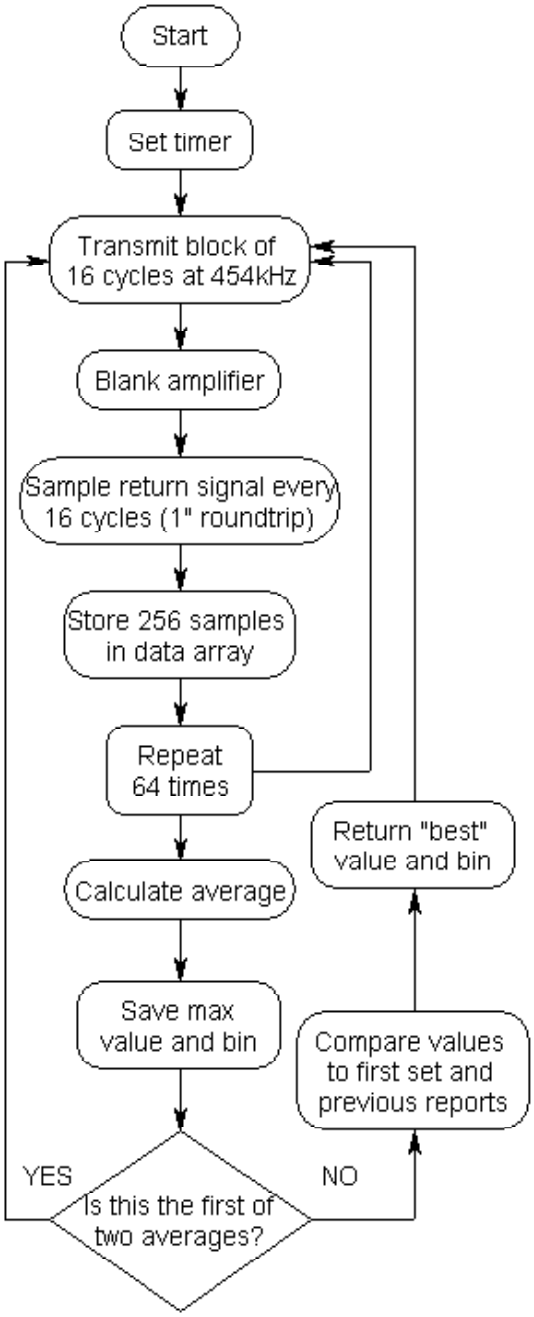

Figure 3.8: Flowchart of New Microcontroller Program

echoes from the high noise level. When the transducer was placed between 5 and 10 inches away from the bottom of a tank of water, the signal processor was able to output reliable measurements. However, any distances outside of this range generated very inconsistent readings from the sonar. Signals would fade in and out, producing erratic and widely fluctuating readings. Movement of the connecting cable or any kind of vibration would generate noise that interfered with signal transmission and resulted in erroneous measurements.

After researching more into the designs of other sonar units and also consulting with Mr. Carvel Holton, a senior research associate in the Fiber & Electro-Optics Research Center in the College of Engineering at Virginia Polytechnic Institute, some changes were made that significantly improved the performance and accuracy of the altimeter. The revised program is written in a PIC programming language similar to C++. In the new code, the blanking period of the amplifier was reduced from four transmission cycles down to two. This changed the minimum reading of five inches to 3 inches. More averaging of the return signal was performed to boost the signal to noise ratio, or SNR. The SNR increases by a factor that is approximately equal to the square root of the number of averages. In the new program, 64 averages of the return signal are made, resulting in roughly 2 reports per second. Then, based on previous outputs, the better of the 2 reports is chosen, output through the serial port, and stored in the pic for use in choosing the next "best" measurement. In most cases, the report closest in value to the previous measurement is chosen as the new reading. During periods of dead time, the last report is held for two refresh cycles, and then decays to zero if no new data is obtained. When there are no previous measurements, which happens when the transducer is first placed into the water or after a long period of dead time, the bin containing the higher signal strength is chosen. If both reports are much greater than the previous one, then the output increases asymptotically towards that value. Because 64 averages are taken instead of only 16, the SNR has approximately doubled. The simplified flowchart can be seen in Figure 3.8; Appendix A contains the full program source.

After implementation of these changes, the readings from the sonar have become very consistent and accurate from a minimum of 3 inches to a

maximum of 18 inches, which is the depth of the test tank. No further testing has been performed in deeper water or with a softer bottom. However, the altimeter is designed to read up to a depth as great as 256 inches, or roughly 21 feet.

3.3 Layout and Board Construction

The circuit for the altimeter is laid out and constructed on a single double-sided board. Components of the signal processing portion of the circuit, such as the voltage regulator, microprocessor, peak detector circuit, and dual amplifier IC, are all situated on the top half of the board. The lower half contains the transmitter circuit, which includes the hand-wound transformer, acoustic tile, and impedance transformation buffer circuit. There are no traces connecting the two halves of the circuit; however, each half contains a four pin header consisting of connections to ground, power, the acoustic drive, and the return signal from the JFET. This is done in order to later separate the two circuits with only a single cable running from one to the other, since only the transmitter circuit needs to be underwater.

To get an idea of the size and space constraints on the miniature AUV, some parameters are listed in Table 3.1 [20].

Parameter	*Specification*
Length	78 cm
Hull Diameter	9.5 cm
Mass	5.195 kg with mission sensor
Power	22.2 volt, 88.8 watt-hour LiPoly battery stack
Maximum speed	3 knots
Max depth	200 meters (anticipated)

Table 3.1: VT miniature AUV general specifications

In order to fit with all the other electronics inside, the board had to be less than 63mm in width, 120mm in length, and 13.5mm in height. Due to these space constraints, the smallest components such as the resistors and

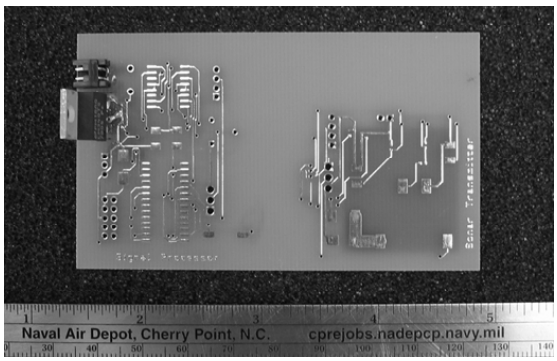

Figure 3.9: Top Layer of Altimeter Printed Circuit Board

Figure 3.10: Bottom Layer of Altimeter Printed Circuit Board

ceramic capacitors were placed on the bottom layer of the board. Large traces are needed to handle the higher currents, especially for connection between the tantalum cap, the primary of the transformer, and most of the ground connections. A one-inch square ground plane was situated on the bottom side of the transmitter circuit half of the board where the acoustic tile was to be placed. Figure 3.9 and Figure 3.10 show the top and bottom layers of the board before it was fully populated. The final dimensions of the board are 52.58×88.14 mm, or 2.07×3.47 in.

Once the board is manufactured, populated, and tested, it is cut into two parts, separating the transmitter and signal processing circuits. To waterproof the transmitter, a box was constructed from PVC material to house the circuit. The inside of the box is milled out, leaving only the outer edges in the shape of a frame. The front side of the frame is left with thicker edges than the back side in order to support the circuit card. Epoxy is then applied inside the frame to secure the board in place. The same two-part rubber silicone mold that was used in testing is used again to protect the face of the tile from elements in the marine environment.

Several different methods were tried in waterproofing the top layer of the transmitter circuit inside the frame. The most obvious was to use the same two-part mold. However, once this compound was poured in and allowed

to set, it severely degraded the performance of the circuit. Another method used was to cut a thin plastic cover for the box and apply sealant around the edges, but after a length of time with the box at the bottom of the tank, water still found a way to seep in around the seal. A special electronic grade silicone was then attempted and used to fill in the box from the back end. This successfully kept the electronics safe and dry while maintaining the integrity of the transmitting signal.

3.4 Results

The final sonar altimeter is shown in Figures 3.11 and 3.12.

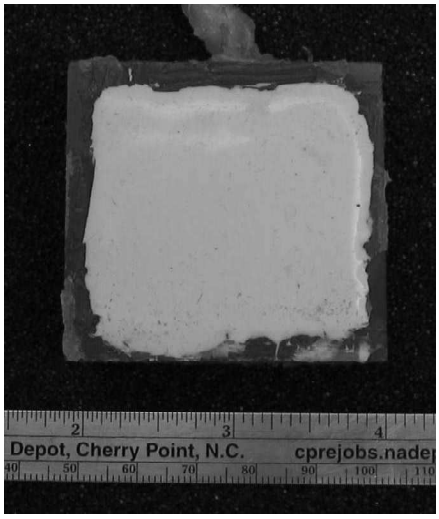

Figure 3.11: Waterproof Transducer Circuit

The equation used to calculate the beam pattern of a square $L \times L$ transducer element is:

$$BP = \frac{sin(w_1 L)}{w_1 L}, \tag{3.2}$$

where

$$w_1 = \frac{\pi}{\lambda} sin\theta. \tag{3.3}$$

where θ is the angle in radians, and λ is the wavelength, which is 3.26mm. Since height and width are equal, the beam pattern will be the same in both

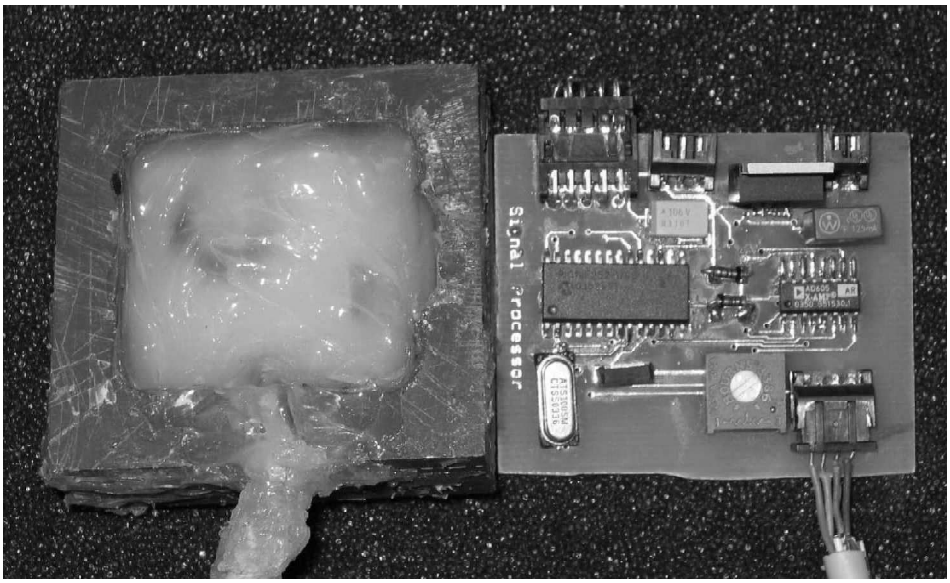

Figure 3.12: Final Sonar Altimeter

directions, vertical and horizontal.

Figure 3.13 shows the beam pattern for this altimeter. Values displayed above the main lobe represent degrees, while the numbers below are magnitudes. The first two nodes from the main lobe are located at 7.4° and 14.8°, and they denote places where the acoustic energy is zero. Most of the energy is located in the main lobe, while very little energy is projected in the side lobes, where the amplitudes are much less than unity. The beam pattern shows that the major lobe is very narrowly directed, so the beamwidth will be small. In general, beamwidth is determined by the $-3dB$ or half-power point on this lobe. A plot of the beam pattern in cartesian coordinates in terms of decibels is shown in Figure 3.14. The $-3dB$ points are located at $\pm 3.4°$, resulting in a horizontal and vertical beamwidth of 6.8°.

Results of testing showed that the altimeter was highly directional, indicating that the beamwidth of the system is too narrow and it unable to detect echoes unless it falls within that 6.8° band. Based on equation 3.2, there are two ways to increase beamwidth. The first method would be to decrease the operating frequency of the transmitter. This would result in a redesign of the entire system. The second, easier method would be to decrease the length of the transducer element, which would involve carefully cutting

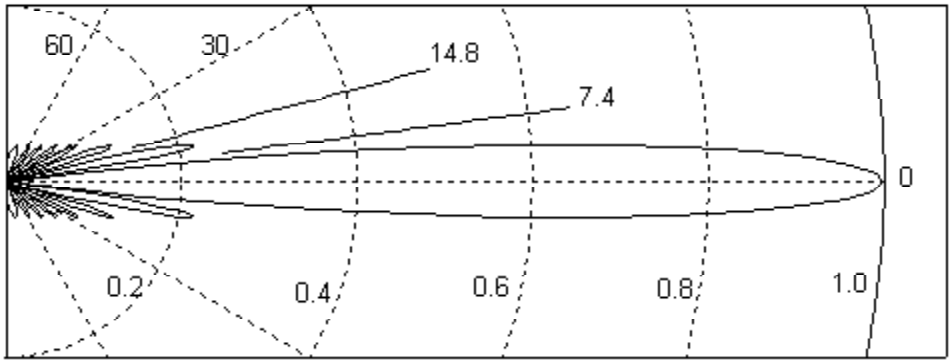

Figure 3.13: Beam pattern of the sonar altimeter

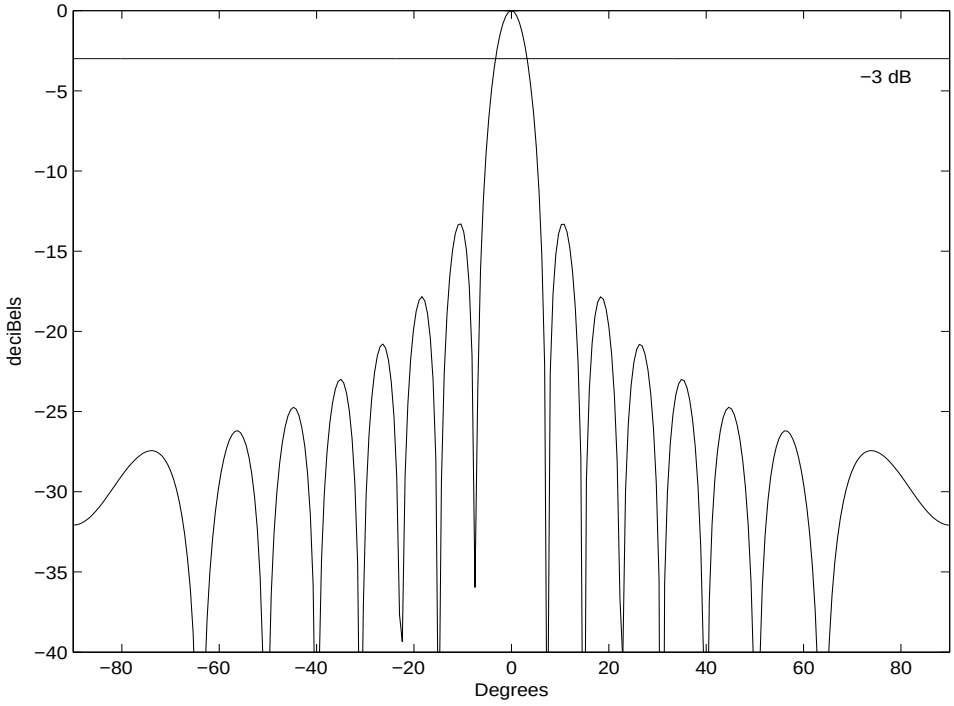

Figure 3.14: Logarithmic beam pattern of the sonar altimeter and the -3dB point

the piezoelectric element to the desired size.

Chapter 4

AirMar Triducer and Altitude Controller Design

To design and develop a good sonar-based controller, one first needs to obtain reliable long-range sonar readings. Since the altimeter developed and described in chapter 3 has not been tested extensively, a commercial sonar was used instead. There are a variety of choices available, so a search was carried out for a reliable, high performance depth sounder with serial output. Many of the devices on the market now have a digital display and LCD driver. This would make the signal output too difficult to decode and read into the serial port of a computer.

4.1 Depth Sounder

In [1], nine digital depth sounders were tested and reviewed. Among these, only three had the ability to communicate using standard NMEA protocol, which is RS-232 compatible and can be read easily into a computer serial port. One of them is the DS150 from Standard Horizon, and the other two were Raymarine's ST40 and ST60. Range, reliability, and ease of use were tested with each sonar device. According to the article, the two Raymarine depth sounders surpassed all others in performance, giving steady, accurate readings and having a range twice as large as the DS150 with a minimum

depth reading of 3 feet and a maximum of 400 feet.

Figure 4.1: AirMar Triducer

The large digital display of the Raymarine depth sounder would not be used when the device is implemented in the AUV. Therefore, only the transducer for the Raymarine depth sounder was needed. This was manufactured by a company called AirMar Technology Corporation based out of Milford, New Hampshire. After consulting with Irene Robb, the technical sales support coordinator at AirMar, the sonar chosen for use with the AUV was the DST800 Retractable Triducer, seen in Figure 4.1. Operating at 235 kHz, this product is the newest in AirMar's line of marine sonar. It incorporates three different sensors into one cylinder of radius 51 mm and length 125 mm. Temperature, speed, and depth are measured with these sensors. Depth is measured in feet and in meters with a resolution of 1/10. The device outputs all the data once a second at a baud rate of 4800 bps using standard NMEA0183 output protocols. The triducer is rated at 100W RMS power, has a vertical beamwidth of 10° and a horizontal beamwidth of 44° [2].

Although this device is very compact, it is still too large and awkward to mount on the AUV. This is why a smaller device would be desired.

4.2 Controller Design

The constant altitude controller is actually based on the depth controller already implemented in the AUV, so the depth controller will be discussed first.

4.2.1 Depth Controller

The depth controller is a proportional-differential (PD) controller. It takes on this form:

$$r = K_P(d_{cmd} - d_{act}) - K_D(p_{cmd} - p_{act}) \tag{4.1}$$

In equation 4.1 d_{cmd} is the desired or commanded depth, d_{act} is the actual depth obtained from the depth sensor, and K_P and K_D are the proportional and differential gains. Since the time-derivative of depth is related to the pitch of the vehicle, this angle is used as the derivative term in the equation, with p_{cmd} as the desired pitch and p_{act} as the actual pitch. For the depth controller, the desired pitch is 0°. In the water, the vehicle could be in one of five states: entry, dive, cruise, rise, and surface. The depth controller is used only when the vehicle is in cruise state. For the dive and rise states, proportional control is used for pitch. This is because a constant pitch is desired to attain a smooth, controlled dive or rise. The desired pitch is set to -15° for dive and +15° for rise.

When the desired depth is reached, the AUV enters into cruise state. A threshold of 1 meter centered around the commanded depth is set for the cruise state; that is, if $d_{act} < d_{cmd} - 0.5$ meters, then the vehicle enters the dive state until the actual depth of the AUV is within the threshold. If $d_{act} > d_{cmd} - 0.5$ meters, then the AUV enters the rise state until it is once again within threshold, at which point the vehicle switches back to cruise. Figure 4.2 illustrates the states of the AUV. The vehicle begins at the water surface in the entry state. During entry, the vehicle gathers speed to break the surface of the water when diving. After reaching a threshold of 0.25 meters, it enters the dive state until reaching the commanded depth of 3 meters where it enters cruise state. A forced rise state can be applied to raise the vehicle back to the

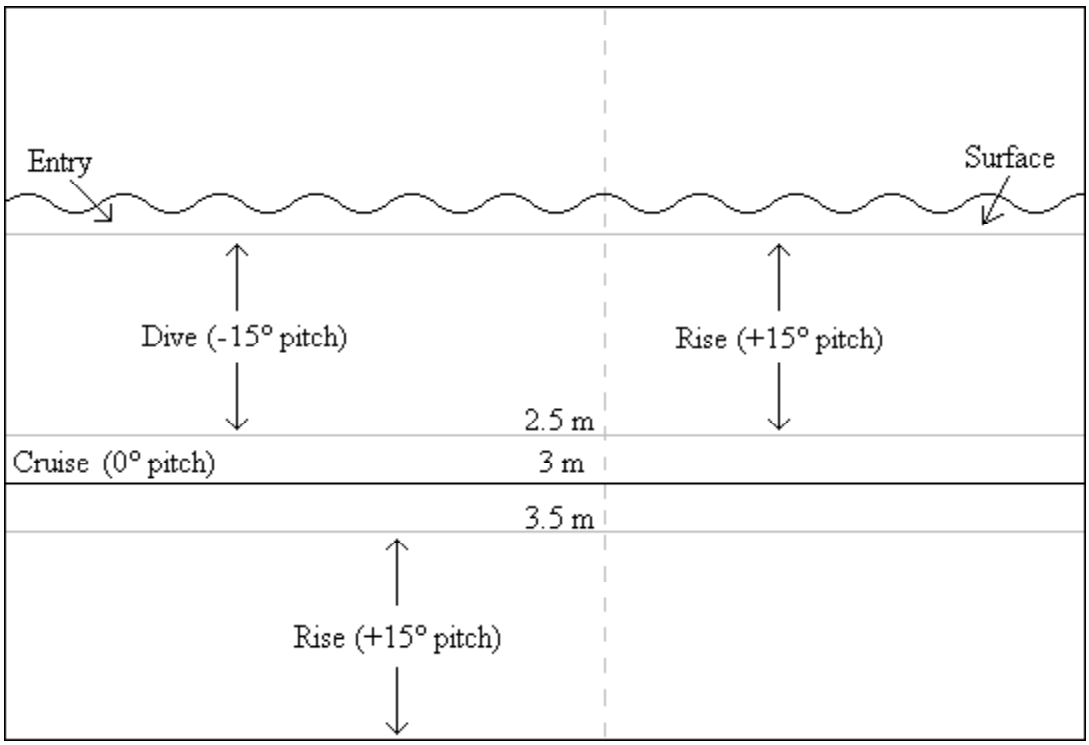

Figure 4.2: Five States of the AUV

top where it switches to the surface state after reaching a threshold of 0.25 meters.

In equation 4.1, r refers to the amount of fin adjustment. There are two side fins, labeled b and c, that are located toward the back end of the vehicle that are controlled using a servo motor. Values from the servo motor range from 3000 to 5000, with 3000 and 5000 being positive and negative 90 degrees. The fin adjustments are always opposite in sign and are added to 4000, the zero degree position for the fins. For example, if r is 500, then the command given to fin b would be 3500 and the command given to fin c would be 4500.

To tune a PD controller, first the proportional gain is tuned to allow more overshoot than normal, then the differential gain is slowly increased to improve transient response. Upon testing, the gains in Table 4.1 were found to work best.

Table 4.1: Proportional and Differential Gains for Different States of the AUV.

States	K_P	K_D
Cruise	150	5
Dive	100	0
Rise	50	0

4.2.2 Altitude Controller

For the altitude controller, the AirMar Triducer is used to measure the altitude of the vehicle from the seafloor, while the commanded height is input by a user. This height is converted to a desired depth by subtracting it from the actual altitude and adding the depth sensor measurement of the depth of the AUV underwater. If the vehicle wanders into shallow water and the commanded altitude exceeds the water depth, the desired depth value is changed to zero. Once a desired depth is obtained, the same depth controller described in the previous section is used to control the altitude, the only difference being that the desired depth is now a variable instead of a constant. Figure 4.3 depicts these variables.

4.3 Results

The altitude controller was tested in Claytor Lake on November 19, 2004. Figure 4.4 shows a picture of the sonar unit mounted on the AUV. Due to the height of the Triducer sensor, the only place to mount it and still have it point downward was against the nose of the vehicle. There were also problems with the depth sensor failing to work properly for most of the day, and the AUV was never able to successfully break the surface of the water and enter the dive state. In order to stay underwater, the vehicle had to be initially pushed down into the water and forced into the dive state, where it could operate without any further assistance.

Despite the problems encountered, three runs were executed, but the first one was too short to see any conclusive data. The results of the second two

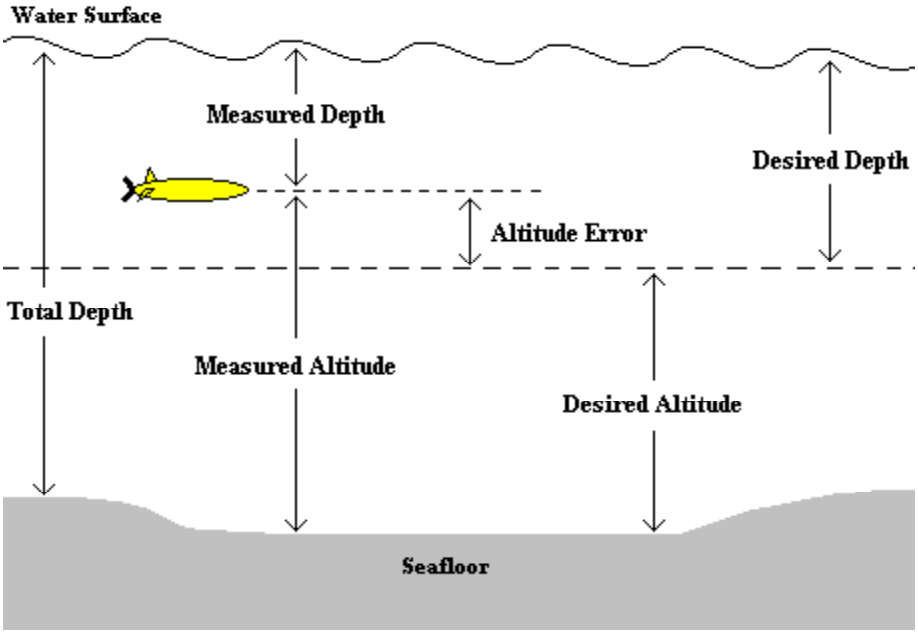

Figure 4.3: Illustration of Control Variables

runs are shown in Figure 4.5 and Figure 4.6. The depth sensor and transducer measurements are almost exact inverses of each other. The sum of the two is the total depth of that portion of the lake, which was about 12 feet, going by the reading from the fishfinder on the boat. The first run lasted 45 seconds, the second one for 60 seconds. Longer runs were preferred and would have provided more conclusive data, but the risk of losing the vehicle was too high with the amount of fallen trees and refuse at the bottom of the lake.

In Figure 4.5, the AUV was programmed to operate at a constant 2.1m above the bottom of the lake. Based on the altitude data obtained, the system experienced 12.9% overshoot with a minimum altitude of 1.829m. The altitude at the final time is 2.1946 m yielding an error of 4.5%. An altitude of 2.5m was input into the controller for the next trial. This time, the vehicle traveled to a minimum altitude of 2.347m with an overshoot of only 6.1% which is half the percentage obtained from the first run. The altitude at the final time is 2.4994, yielding a steady-state error of 0.024%. Both settling time

Figure 4.4: Virginia Tech Miniature AUV with AirMar Triducer Sensor

and rise time can not be calculated accurately from this data since the vehicle could not dive without assistance. However, it does seem as though it would take longer than 45 seconds for the controller to settle to within 1% of the final value since neither system had quite settled yet at this time.

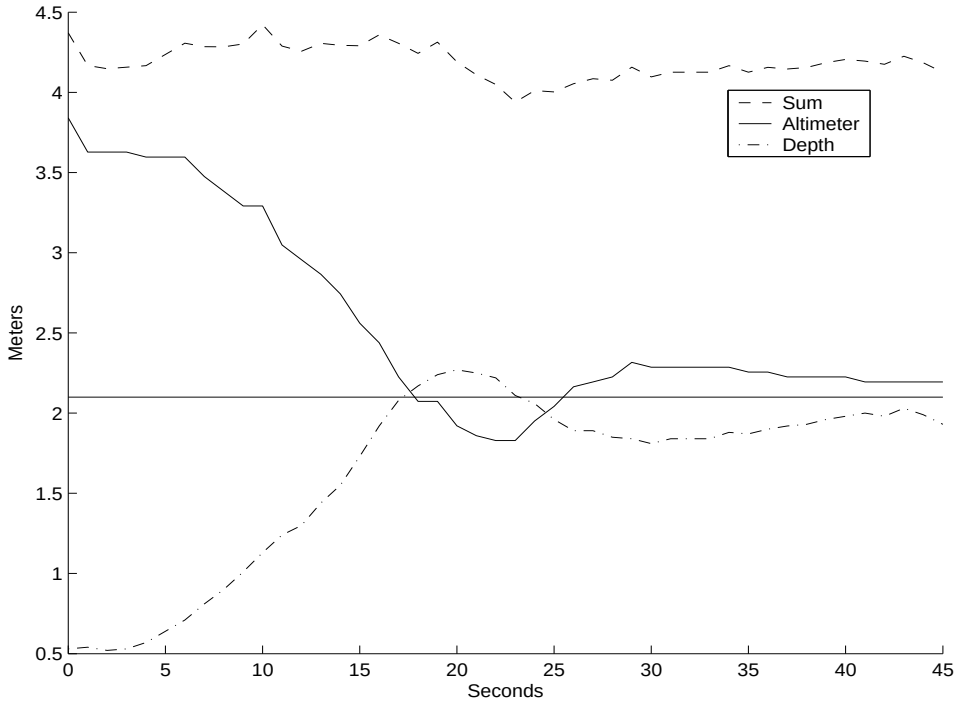

Figure 4.5: Altitude controller test with altitude set to 2.1m

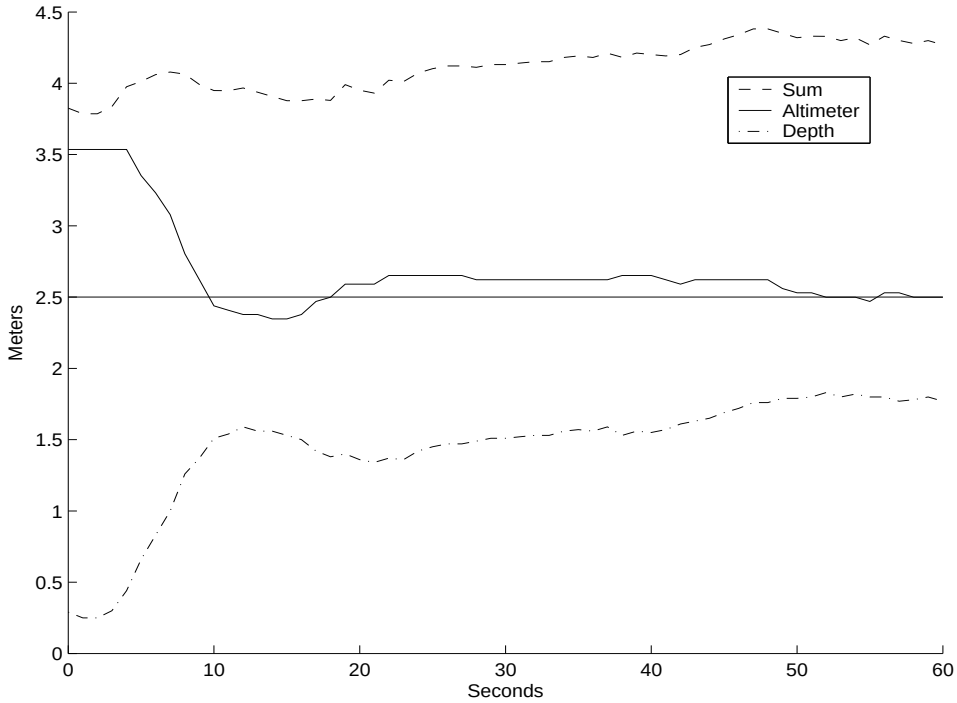

Figure 4.6: Altitude controller test with altitude set to 2.5m

Based on these results, the controller was successful in keeping the vehicle at a constant altitude above the bottom of the lake. Further tuning of the controller could improve its operation and lower the settling time and percent overshoot.

Chapter 5

Literature Survey on Sonar-Based Obstacle Avoidance

The purpose of using an automated system is to eliminate the need for human supervision. This makes obstacle avoidance an essential feature for AUV's, especially if the vehicle is small in size. It is one of the most important safety considerations for AUV missions, since AUV's should be able to safely contend with all the hazards and the unpredictability of the underwater environment. This means that the AUV must be able to not only accurately sense the immediate, surrounding area, but it must also maneuver and react to it accordingly.

An obstacle avoidance sensor (OAS) returns obstacle related data to the navigation and control system of the underwater vehicle. Devising a strategy for obstacle avoidance consists of acquiring data from an OAS, processing this information to obtain obstacle data, and designing algorithms to react to the data and also accomplish mission goals.

5.1 Single Transducer OAS

For obstacle avoidance on the Virginia Tech miniature AUV, a sonar altimeter would be placed in the nose of the vehicle, pointing in the horizontal direction.

With only one simple sonar device as the obstacle avoidance sensor, since it only reports the distance of the nearest object, an obstacle would be considered within reach any time several continuous readings are returned that fall within the same range. However, depending on the depth and pitch of the vehicle, the sensor could be obtaining readings off the seafloor or the surface of the water instead of an actual object. Since this is not obstacle data, it should not be treated as important information. In this case, an algorithm can be developed that considers vehicle attitude, depth, and bathymetric data obtained from the depth sounding sonar and uses this information to filter out range returns from the OAS that are most likely echoes reflected from the surface or floor [11].

In this algorithm, a range value R should be calculated based attitude, altitude, and depth information. This is the value that the OAS should return if the signal was reflected off the seafloor. If R is greater than the range of the sonar, then any value returned from the OAS should be considered a return from a possible obstacle. For a calculated value of R that is within the sonar range, a comparison would have to be performed regarding the OAS report with respect to R.

An alternative method to determine the presence of an obstacle would be to use a scaling coefficient [11]. If the reported distance is less than the expected value multiplied by the coefficient, then that would indicate an obstacle. The value of this coefficient would be left up to the designer. If a number close to unity is chosen, then the algorithm would be highly sensitive, and any slight inclines in the terrain would be reported as an obstacle. Decreasing the coefficient would decrease sensitivity, but it would also leave less time for the AUV to react to obstacles when they are reported.

5.2 Multiple Transducer OAS

With a single sonar, there is no way of knowing the height or width of an obstacle or its orientation. There is not enough information to know which way would be the best way to move. Instead of only one single altimeter, multiple transducers can be used, each angled slightly differently, that can ping one after

another in sequence to map out the seabed. This is called multi-preformed beam sonar [11]. This, however, requires more time and decreases the refresh rate of the sonar. An alternative is to use a separate transducer that acts only as a projector to ensonify the entire region. Several receivers then record the echoes along their particularly aimed beams [16]. This could be implemented on the Virginia Tech miniature AUV utilizing the same acoustic tile from the sonar altimeter [24]. The tile would be divided into three sections, with the left and right sections each being a quarter-inch in length and a half-inch middle section. The middle section would act as the transmitter; the two sides are the receivers. Each receiver could be bent back at a 10° angle. This would create a larger ensonified area by effectively increase the beamwidth of the transmitter, since beamwidth is inversely proportional to the diameter of the transducer element [3]. In this way, 2-D information may be obtained: distance to the object as well as bearing.

If size, cost, and processing power were not an issue, then data obtained from the implementation of a 2-D transducer array can be used to make a 3-D map of the space. This is highly useful in path planning and maneuvering the vehicle around obstacles. Constructing a good map requires tracking of these obstacles from ping to ping in order to obtain information on how the obstacle is moving with respect to the AUV. This is performed by continually updating the position and characteristics of the obstacles from prior frames. However, a problem arises in deciphering whether the objects in the new frame are the same ones of the old frame, or whether the positions of the objects in the old frame wrong, or if the objects are new obstacles altogether. Deciphering techniques such as optical flow involving interpolation of pixels from frame to frame are very computationally intensive and yields poor results in noisy environments [19].

One solution is to implement a feature extraction algorithm and then apply a Kalman filter to update frames with the new data [11]. Some of the advantages of using the Kalman filter are that it is a fast algorithm when the state vector is small, and it possesses a covariance matrix that provides the uncertainty of the estimated states. This is greatly beneficial for path planning purposes and increases the safety of the system when incorporated into the

obstacle avoidance algorithms. In addition, the application of an extended Kalman filter would allow a model of the dynamics of the AUV to be easily included [19].

5.3 Incorporating an Obstacle Avoidance System in the Control System of an AUV

Incorporate obstacle avoidance without straying from mission goals is an important issue in the control of an AUV. There are low level and high level goals that exist in an system. The lower level functions are called reactive; these functions are made to react quickly to incoming data to ensure the safety of the vehicle regardless of the state. A deliberative or higher level system uses vehicle states to achieve higher level goals, such as maneuvering in complex situations [11]. An example of these two types of systems would be in a situation where the AUV encounters a U-shaped obstacle. A purely reactive system could drive the system to travel in circles as seen in Figure 5.1. However, a deliberative system would be able to identify it as a complex obstacle by detecting a local minimum and switch to a different state that executes an escape plan. In this case, it would attempt to follow the wall to find a way out [5]. The system could perhaps also keep a record of these areas and avoid it in the future.

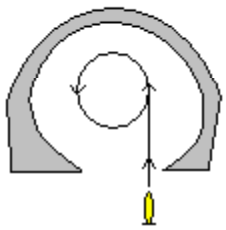

Figure 5.1: An AUV equipped with only a reactive system in a trap-like situation

For a single transducer OAS, the algorithms are going to be primarily reactive since only simple obstacle avoidance can be achieved. The sensor only detects whether an obstacle is present without indicating a good direction

to turn, so the direction selected should be dependent on the mission. For example, if the mission involves track following, it would be best to move vertically and go up over the obstacle so that the vehicle does not stray from the track. If the object is too large to successfully navigate around and the sonar returns hit a low threshold value in terms of distance, then the single transducer obstacle avoidance system should interpret this as a dangerous situation and either remove power from the thrust, or engage in a tight spiral ascent to the surface [11]. A spiral ascent will provide a maximally steep ascent to the surface while only requiring a lateral area equal to the turning radius of the vehicle.

For multisensor operation that involves 2-D data, more complexity can be added, as the vehicle now has enough information to make a decision about the direction to turn to avoid collision and also how strongly to react to the data. Deliberative functions can now be added to the system. However, this also introduces more goals, with each goal bearing its own motor recommendation. Arbitration must be employed to make a decision from the multiple recommendations [11].

One decision method would be the winner-take-all scenario. A hierarchy of all the goals is formed, and control is awarded to the goal with the highest priority. Another arbitration approach would be to scale all the recommendations according to priority and then take the sum. This creates a result that blends together all the goals of the system, where avoidance of an object would also contain awareness of other goal points [11].

5.4 Working with Mission Goals

One possible mission for AUVs is seafloor survey which would involve high detail sonar mapping and bathymetry. This involves staying at standard low altitudes of 5-15 m [11]. For these near-bottom missions, accurate measurements are critical as reaction times become shorter [21]. There are four main objectives of the obstacle avoidance system during low altitude operation: reverberation level reduction, determination of seafloor slope, obstacle detection, and maintaining clear field-of-view to minimize blind maneuvers.

Reverberation can mask the presence of potentially dangerous targets in the path of the vehicle. Two methods to reduce reverberation are discussed here. The first one involves reducing the horizontal and vertical components of the beamwidth [21]. This will decrease the size of the ensonified area which is proportional to reverberation. However since beamwidth is inversely proportional to the size of the transducer, decreasing beamwidth is achieved only by increasing the area of the transducer. This is impractical due to the small diameter of the vehicle.

The second method of minimizing the amount of reverberation is to shorten the length of the pulse which reduces the amount of backscatter. However, decreasing the pulse length will reduce the target strength within the ping, so the amount of data processing required increases linearly with decreasing pulse length [21].

Being able to determine the slope of the terrain ahead allows the vehicle to remain at a low but safe, constant altitude above the floor. However, reverberation level is the only acoustic measure of seafloor position [21]. One good solution is to modify the pulse length. Transmission of a single long pulse periodically amoung regular pulses yields improved results for receiving strong echoes from the seafloor while keeping the reverberation level relatively low [21].

Field-of-view involves the vertical and horizontal components of the ensonified area. For a multibeam OAS, beams should overlap enough to allow for small changes in pitch of the AUV [21].

For obstacle detection in this system, the resolution of the object location and size is limited by the angular resolution of the sonar. Angular resolution refers to the smallest angle separating two obstacles that would enable the OAS to interpret them as separate obstacles. The resolution of the size of an obstacle depends on the angular resolution of the sonar, as well as the distance to the obstacle. Higher angular resolution is accomplished with a narrower beam [11].

Many of the techniques discussed in this chapter could be implemented in the VT miniature AUV without a great deal of effort, and would vastly increase the safety of the AUV during missions. Additional research is highly

recommended in this area.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Good sonar systems are instrumental in mission success for underwater vehicles, especially AUVs. In addition to the capable of reporting ocean depths, they can also aid in vehicles navigation. A small, inexpensive sonar altimeter designed for the VT miniature AUV has been developed and improved, becoming less capricious and less susceptible to noise while providing steady and accurate readings. However, it is not quite ready to be implemented in the vehicle, as the range of the system is yet unknown. Testing in non-ideal environments with low signal returns is also recommended.

The controller designed for flying at constant altitudes was successful, but further testing is required involving a longer time underwater. Results from testing the controller in a lake showed that a faster settling time could be achieved with more tuning. This system could be very useful for gathering detailed information about areas near the seafloor by allowing low altitude or shallow water operation. An obstacle avoidance system should be used in conjunction with the controller during this type of mission to ensure the safety of the vehicle while operating underwater without supervision.

6.2 Future Work

The altimeter circuit has been improved in reliability and robustness. Added components to the design have drastically cut down the amount of noise received by the processor. This improved physical performance coupled with increased data processing has provided a more reliable depth reading which gives confidence in the system's operation. However, there is always room for improvement. The level of noise at the output of the amplifier is still rather high. Future improvements for the sonar altimeter circuit could be to research and find a better amplifier for the design. Increasing beamwidth by using a smaller transducer element is also recommended for future work.

Although the PD controller for constant altitude control showed promising results, it had not been tested in areas with sudden steep changes in terrain. Implementing an adaptive controller would allow the system gains to be changed depending on the terrain, which would be highly advantageous and also safer for the vehicle. Future work could also include evolving this into a controller for tracking contour features of the seafloor, allowing basic mapping possibilities.

Implementation of an obstacle avoidance system would increase the types of areas in which the AUV can operate, adding endless mission possibilities. The multi-receiver, single projector configuration discussed in Section 5.2 could possibility be the best choice for an OAS in the miniature AUV with further research and development.

Bibliography

[1] Just the numbers, please: Digital depth sounders. *Powerboat Reports*, 16(7), July 2003.

[2] AirMar Technology Corporation. *Retractable Tri Flush Countersink: DST800 Series.*

[3] AirMar Technology Corporation. *Sensor Design Fundamentals.*

[4] Amidon Associates, Inc. *Toroidal Cores: Ferrite Cores*, 1989. http://www.amidoncorp.com/aai_ferritecores.htm.

[5] G Antonelli, S. Chiaverini, R. Finotello, and R. Schiavon. Real-time path planning and obstacle avoidance for rais: An autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 26(2):216–227, Apr 2001.

[6] T.G. Bell. Sonar and submarine detection. *U.S. Navy Underwater Sound Laboratory Report*, 545, 1962.

[7] P.M. Cohen. *Bathymetric Navigation and Charting.* United States Naval Institute, 1980.

[8] D.J. Creasey. Underwater acoustics. *Physics Education*, 16(4):244–250, July 1981.

[9] Department of the Navy, Bureau of Ships. *Introduction to Sonar Technology*, December 1965.

[10] S.J. Fantone, O. Leitermann, J. Austin-Brencman, A.S. Eastment, and E. Crumlin. The autonomous underwater vehicle "pipsqueak". In

OCEANS 2002 MTS/IEEE Conference, volume 4, pages 2497–2501, Oct 2002.

[11] E.G. Jones and Nathaniel Fairfield. Sensor selection and behavior strategies for obstacle avoidance in small-diameter auvs. In *Unmanned Untethered Submersible Technology*, Aug 2003.

[12] L.E. Kinsler, A.R. Frey, A.B. Coppens, and J.V. Sanders. *Fundamentals of Acoustics*. John Wiley & Sons, Inc., 4 edition, 2000.

[13] L-3 Communications SeaBeam Instruments. *Multibeam Sonar Theory of Operation*, 2000. http://www.ldeo.columbia.edu/res/pi/MB-System/formatdoc.

[14] J.J Leonard, A.A Bennett, C.M. Smith, and H.J. Feder. Autonomous underwater vehicle navigation. Technical report, Massachusetts Institute of Technology Department of Ocean Engineering, 1998.

[15] C.C. Leroy. Development of simple equations for accurate and more realistic calculation of the speed of sound in sea water. *Journal of the Acoustic Society of America*, 46:216, 1969.

[16] C.D. Loggins. A comparison of forward-looking sonar design alternatives. In *OCEANS 2001 MTS/IEEE Conference*, volume 3, pages 1536–1545, Nov 2001.

[17] J.J. Mach. Toward auto-calibration of navigation sensors for miniature autonomous underwater vehicles. Master's thesis, Virginia Tech, 2003.

[18] Measurement Specialties, Inc. *Piezo Film Sensors Technical Manual*, Aug.

[19] Y. Petillot, I.T. Ruiz, and D.M. Lane. Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar. *IEEE Journal of Oceanic Engineering*, 26(2):240–251, Apr 2001.

[20] W.F. Seely. Development of a power system and analysis of inertial system calibration for a small autonomous underwater vehicle. Master's thesis, Virginia Tech, 2004.

[21] M.T. Shaw. Obstacle avoidance sonar: An analysis of system requirements and detection performance during near bottom operation. In *Autonomous Underwater Vehicle Technology*, pages 223–228, Jun 1990.

[22] D.G. Tucker. *Underwater Observation Using Sonar*. Fishing News (Books) Limited, 1966.

[23] R.J. Urick. *Principles of Underwater Sound*. McGraw-Hill, Inc., 1975.

[24] C.E. Wick. Personal correspondence. 2004.

Appendix A

Microprocessor Revised Program Source

```
// Jessica Luan
// Sonar Pic Program - Updated to one reading per second
// 11/09/04


#include <p18f252.h>
#include <stdlib.h>
#include <string.h>
#include <pwm.h>
#include <adc.h>
#include <usart.h>
#include <timers.h>
#include <delays.h>
#include <reset.h>
#include <ctype.h>
#include <math.h>

void isrhandler(void);
void isr_functions(void);
```

```
////////////////////////////
// variables declarations //
////////////////////////////
static int flags = 0;
static int tcnt = 0;
static int pcnt = 0;
static int adcresult = 0;
static int i = 0;
static int num_readings = 0;
static int prev_depth = 0;
static int prev_depth2 = 0;
static int prev_mag = 0;
static int prev_mag2 = 0;
static int max_bin1 = 0;
static int max_value1 = 0;
static int max_bin2 = 0;
static int max_value2 = 0;
static int zero_reading = 0;
// a count of the number of zero readings
static double testcond = 0;
static signed int change1 = 0;
static signed int change2 = 0;
static char word;       //used just for testing

#pragma udata SumArrays1 = 0x100
static int sum1[128];
#pragma udata SumArrays2 = 0x200
static int sum2[128];
#pragma udata DataArrays1 = 0x300
static int data1[128];
#pragma udata DataArrays2 = 0x400
static int data2[128];
```

```
//////////////////////////////
// Interrupt Service Routine //
//////////////////////////////

#pragma interrupt isr_functions save = PROD

// Locate ISR handler code at interrupt vector

#pragma code isrcode=0x0008

void isrhandler(void)
// This function directs execution to the
// actual interrupt code
{
_asm
goto isr_functions
_endasm
}


#pragma code


void isr_functions(void)
{
    PIR1bits.TMR2IF = 0;

    // return to program if processing not done
    if (flags==1)
        return;

    // transmit routine - 16 times (*4)
```

```
if (pcnt == 0) {
    PORTCbits.RC0 = 1;      // turn blanking on
    if (tcnt == 16*4) {     // return and process data
        tcnt = 0;
        flags = 1;
        num_readings++;
        return;
    }
    else

    tcnt++;
    // PWM 50% duty cycle = (CCPR1L:CCP1CON<5:4>)*Tosc
    //                                    *(TMR2 prescale)
    CCPR1L = 0x0b;
    pcnt++;
    ADRESH = 0;
}

// receiving routine - do this for 255x16x4 times
else if (pcnt > 0) {
    CCPR1L = 0;         // 0% duty cycle
    if (pcnt == 2)      // turn off blanking after first 2 counts
        PORTCbits.RC0 = 0;

    adcresult = ADRESH;
    ADRESH = 0;
    ADRESL = 0;
    PIR1bits.ADIF = 0;
    //ConvertADC();               // start conversion

    if (pcnt < 129)  {
        data1[pcnt-1] = adcresult;
        sum1[pcnt-1] = sum1[pcnt-1]+data1[pcnt-1];
    }
```

```
        else {
            data2[pcnt-129] = adcresult;
            sum2[pcnt-129] = sum2[pcnt-129] + data2[pcnt-129];
        }



        if (pcnt == 255)
            pcnt = 0;   // reset pcnt

        else
            pcnt++;
        adcresult = 0;

        ConvertADC();   // start conversion after waiting Tacq

    }

}



void main(void)
{
    int max_sum = 0;
    int max_s2 = 0;
    int max_b2 = 0;
    int max_value = 0;
    int max_bin = 0;
    char str[19];
    char str1[8];
    char str2[8];
    char str3[8];
    char str4[8];
```

```
char str5[8];
char str6[8];
int len1 = 0;
int len2 = 0;
int len3 = 0;
int len4 = 0;
int len5 = 0;
int len6 = 0;
int len = 0;

for (i = 0; i < 128; i++) {         //  clear values
    data1[i] = 0;
    data2[i] = 0;
    sum1[i] = 0;
    sum2[i] = 0;
}

ADCON1bits.ADFM = 0;    // left justified-6 LSB of ADRESL=0
ADCON1bits.ADCS2 = 1;   // Fosc/64 -> Tad = 64*Tosc = 1.6 us
ADCON1bits.PCFG3 = 1;   // A/D port config control bits
ADCON1bits.PCFG2 = 1;
ADCON1bits.PCFG1 = 1;
ADCON1bits.PCFG0 = 0;

TRISAbits.TRISA0 = 1;
TRISB = 0;
TRISC = 0;

T2CONbits.TOUTPS3 = 1;      // TMR2 output postscale 1:16
T2CONbits.TOUTPS2 = 1;      // for testing, set to 1:4
T2CONbits.TOUTPS1 = 1;
T2CONbits.TOUTPS0 = 1;
T2CONbits.TMR2ON = 1;       // timer2 enabled
T2CONbits.T2CKPS1 = 0;      // TMR2 clock prescale
```

```
T2CONbits.T2CKPS0 = 0;

PR2 = 0x15;          // PWM period = (PR2+1)*4*Tosc
//                                    *(TMR2 prescale)
// PWM 50% duty cycle = (CCPR1L:CCP1CON<5:4>)*Tosc
//                                    *(TMR2 prescale)
CCPR1L = 0x00;
CCP1CON = 0x0c;

PIE1bits.TMR2IE = 1;        // enable timer2 interrupts
INTCONbits.GIE = 1;         // enable all interrupts
INTCONbits.GIEL = 1;        // enable low priority intrpts

//Initialize ADC module
ADCON0bits.ADCS1 = 1;
ADCON0bits.ADCS0 = 0;

ADCON0bits.CHS2 = 0;        // AN0 channel (pin 2)
ADCON0bits.CHS1 = 0;
ADCON0bits.CHS0 = 0;

ADCON0bits.ADON = 1;        // turn on module

//setup the USART for asynchronous communication with Host

SPBRG = 129;                // high baud (Async), BR=19.2k
TXSTAbits.BRGH = 1;         // high baud rate (Async)
TXSTAbits.SYNC = 0;         // Asynchronous
TXSTAbits.TX9 = 0;          // 8-bits
RCSTAbits.RX9 = 0;          // 8-bits
TXSTAbits.TXEN = 1;         // transmit enable
RCSTAbits.CREN = 1;         // receive enable
RCSTAbits.SPEN = 1;         // enable serial port
```

```
//mainloop
while (1) {
    while (!flags);
    while (flags) {
        //process values
        max_sum = 0;
        max_s2 = 0;
        max_b2 = 0;
        max_value = 0;
        max_bin = 0;
        max_value2 = 0;
        max_bin2 = 0;

        for (i = 0; i < 128; i++) {
            data1[i] = sum1[i]/64;
            data2[i] = sum2[i]/64;

            if (data1[i] > max_value+10) {
                max_value = data1[i];
                max_bin = i;                // i-1
            }
            if (data2[i] > max_s2+10) {
                max_s2 = data2[i];
                max_b2 = i + 128;           //i+128-1
            }
        }
        if (max_s2 > max_value+20) {
            max_value = max_s2;
            max_bin = max_b2;
        }

        if (num_readings == 1) {
            max_value1 = max_value;
            max_bin1 = max_bin;
```

```
    flags = 0;
    break;
}


num_readings = 0;   //reset
max_value2 = max_value;  //save 2nd set of readings
max_bin2 = max_bin;

testcond = 2*prev_depth;


//Choose best value based on previous values

//if no previous values (values are less than 3),
//then choose bin based on strength
if (prev_depth < 3 && max_bin1 > 2 && max_bin2 > 2)
{
    if (max_value2 >= max_value1) {
        max_bin = max_bin2;
        max_value = max_value2;
        zero_reading = 0;
        word = 'a';
    }
    else {
        max_bin = max_bin1;
        max_value = max_value1;
        zero_reading = 0;
        word = 'b';
    }
}

//DEAD TIME
//if no previous values and at least one bin
```

```
//equals 0 (or less than 3), choose non-zero bin.
else if (prev_depth < 3 && prev_depth2 < 3 &&
                (max_bin1 < 3 || max_bin2 < 3)) {
    if (max_bin2 + max_bin1 < 6) {
        zero_reading++;
        max_bin = max_bin1;
        word = 'k';
    }
    else if (max_bin2 > max_bin1) {
        max_bin = max_bin2;
        max_value = max_value2;
        zero_reading = 0;
        word = 'c';
    }
    else {
        max_bin = max_bin1;
        max_value = max_value1;
        zero_reading = 0;
        word = 'd';
    }
}

//DEAD TIME
//if values are both null (or less than 3) and
//prev_depth2 is not, then hold previous value
else if (max_bin1 < 3 && max_bin2 < 3 &&
                            prev_depth2 > 2) {
    zero_reading++;
    max_bin = prev_depth;
    max_value = prev_mag;
    word = 'e';
    }
```

```
//if values are much larger than previous depth,
//then use testcond value
else if (max_bin1 >= testcond &&
                          max_bin2 >= testcond) {
    max_bin = testcond;
    max_value = 0;
    word = 'f';
}

//if there are previous values, compare readings to
//previous depth and choose best value
else {
    change1 = (max_bin1 - prev_depth);
    change2 = (max_bin2 - prev_depth);
    if (change1 < 0)
        change1 *= -1;
    if (change2 < 0)
        change2 *= -1;
    if (change1 <= change2) {
        max_bin = max_bin1;
        max_value = max_value1;
        word = 'g';
    }
    else {
        max_bin = max_bin2;
        max_value = max_value2;
        word = 'h';
    }
}


if (zero_reading > 1) {
    max_bin = max_bin/2;
    max_value = 0;
```

```
    if (max_bin < 3)
        max_bin = 0;
        zero_reading = 1;
    word = 'j';
}


// convert to string and output through USART

// format data to output to USART
len = strlen(str);
for (i = 0; i < len; i++)
    str[i] = 0;
itoa(max_bin,str1);
itoa(max_value,str2);
len1 = strlen(str1);
len2 = strlen(str2);

//Display previous depth values
itoa(prev_depth,str3);
len3 = strlen(str3);

itoa(prev_depth2,str4);
len4 = strlen(str4);


for (i = 0; i < len1; i++)
    str[i] = str1[i];
str[len1] = ' ';

for (i = 0; i < len2; i++)
    str[i+len1+1] = str2[i];
```

```
            str[len2+len1+1] = ' ';
            str[len2+len1+2] = word;
            str[len2+len1+3] = ' ';

            for (i = 0; i < len3; i++)
                str[i+len2+len1+4] = str3[i];

            str[len3+len2+len1+4] = ' ';

            for (i = 0; i < len4; i++)
                str[i+len3+len2+len1+5] = str4[i];

            str[len4+len3+len2+len1+5] = '\r';
            str[len4+len3+len2+len1+6] = '\0';


//          str[len2+len1+3] = '\r';
//          //str[len2+len1+2] = '\n';
//          str[len2+len1+4] = '\0';

            putsUSART(str);           // write string to USART

            for (i = 0; i < 128; i++) {     //  clear sum
                data1[i] = 0;
                data2[i] = 0;
                sum1[i] = 0;
                sum2[i] = 0;
            }
            for (i = 0; i < len1; i++)      // clear strings
                str1[i] = 0;
            for (i = 0; i < len2; i++)
                str2[i] = 0;

            prev_depth2 = prev_depth;
```

```
            prev_mag2 = prev_mag;
            prev_depth = max_bin;
            prev_mag = max_value;

            flags = 0;
        }
    }
    return;
}
```

Appendix B

Microprocessor Original program source

```
list    p=18f252, r=dec
#include p18f252.inc

pcnt    equ 0
qcnt    equ 1
tcnt    equ 2
adhold  equ 3
flags   equ 4
resbin  equ 5
resval  equ 6
maxval  equ 7
maxbin  equ 8
temp    equ 9

    org 0x0000
    goto    start
    org 0x0008

isr                                 ; interrupt location
    bcf     PIR1,TMR2IF,0           ; clear interrupt flag
```

```
    btfsc   flags,0,0
    ; get new data if old processing done

    goto    isr_res

    movf    pcnt,F,0
    btfss   STATUS,Z,0
    goto    rxpart

txpart
    bsf     PORTC,0,0               ; blank on
    movlw   1
    movwf   FSR0H,0                 ; load with 101h and 201h
    movwf   FSR0L,0
    movwf   FSR1L,0
    movlw   2
    movwf   FSR1H,0

    btfss   tcnt,4,0
    goto    txp2
    bsf     flags,0,0               ; ok to report findings
    clrf    tcnt,0
    ; resume after reporting findings
    movlw   0
    movwf   CCPR1L,0                ; shut off tx pulses
    goto    isr_res

txp2
    movlw   0x0b                    ; turn on transmit pulses
    movwf   CCPR1L,0
    incf    pcnt,F,0
    goto    isr_res

rxpart
```

```
    movlw   0x00
    movwf   CCPR1L,0                ; shut off transmit pulses
    movf    tcnt,f,0
    btfss   STATUS,Z,0
    goto    rx1
    clrf    INDF0,0
    ; zero out buffer on first pass
    clrf    INDF1,0

rx1
    btfsc   pcnt,2,0
    ; turn blanking off four pulses later
    bcf     PORTC,0,0               ; blanking off
;   movlw   42                      ; testing
;   movwf   ADRESH,0                ; testing
    movf    ADRESH,W,0
    movwf   adhold,0
    bsf     ADCON0,2,0              ; get next AD value

    addwf   POSTINC0,F,0            ; maintain running sum
    movlw   0
    addwfc  POSTINC1,F,0
    incf    pcnt,F,0
    movf    pcnt,F,0
    btfsc   STATUS,Z,0
    incf    tcnt,F,0

isr_res
    retfie  FAST

hex
    rlncf   WREG,F,0
    addwf   PCL,F
    retlw   '0'
```

```
    retlw   '1'
    retlw   '2'
    retlw   '3'
    retlw   '4'
    retlw   '5'
    retlw   '6'
    retlw   '7'
    retlw   '8'
    retlw   '9'
    retlw   'a'
    retlw   'b'
    retlw   'c'
    retlw   'd'
    retlw   'e'
    retlw   'f'

start
    movlw   0x4e
    movwf   ADCON1,0
    movlw   0x01
    movwf   TRISA,0
    movlw   0x00
    movwf   TRISB,0
    movlw   0x00
    movwf   TRISC,0
    movlw   0x7c
    movwf   T2CON,0
    movlw   0x0c
    movwf   CCP1CON,0
    movwf   CCP2CON,0
    movlw   0x15
    movwf   PR2,0
    movlw   0x00
    movwf   CCPR1L,0            ; pulse off on startup
```

```
    movlw   0x0b
    movwf   CCPR2L,0
    movlw   0x02
    movwf   PIE1,0
    movlw   B'01000000'
    movwf   INTCON,0
    bsf     INTCON,GIE,0

    movlw   0x81
    movwf   ADCON0,0

    movlw   129
    movwf   SPBRG,0

    clrf    TXSTA,0
    bsf     TXSTA,5,0
    bsf     TXSTA,2,0

    movlw   0x90
    movwf   RCSTA,0

    clrf    resval,0
    clrf    resbin,0
    clrf    maxval,0
    clrf    maxbin,0
    clrf    tcnt,0
    clrf    pcnt,0
    clrf    flags,0

mainloop
    btfss   flags,0,0
    goto    mainloop
    call    process
    call    depthrep
```

```
    bcf     flags,0,0
    goto    mainloop

process
    movlw   4
    movwf   qcnt,0
    movlw   1
    movwf   FSR0H,0
    movlw   4
    movwf   FSR0L,0
    movwf   FSR1L,0
    movlw   2
    movwf   FSR1H,0

pl1
    swapf   INDF0,W,0
    ;divide acculmulated values by 16
    andlw   0x0f
    movwf   INDF0,0
    swapf   POSTINC1,W,0
    andlw   0xf0
    iorwf   POSTINC0,F,0
    incfsz  qcnt,F,0
    goto    pl1

    movlw   1
    movwf   FSR0H,0
    movlw   4
    movwf   FSR0L,0

    clrf    maxval,0
    clrf    maxbin,0

    movlw   4
```

```
    movwf   qcnt,0

pl2
    movf    POSTINC0,W,0
    movwf   temp,0
    ;movwf  PORTB,0
    subwf   maxval,W,0
    btfsc   STATUS,C
    goto    p3
    btfsc   STATUS,Z
    goto    p3
    ;bsf    PORTC,3,0
    movf    temp,W,0
    addlw   10
    movwf   maxval,0
    movlw   2
    subwf   qcnt,W,0
    movwf   maxbin,0
    ;bcf    PORTC,3,0

p3
    incfsz  qcnt,F,0
    goto    pl2
    movf    maxval,W,0
    addlw   -10
    movwf   resval,0
    movf    maxbin,W,0
    movwf   resbin,0
    return

depthrep
    swapf   resbin,w,0
    andlw   0x0f
    call    hex
```

```
    call    txready
    movwf   TXREG,0
    movf    resbin,w,0
    andlw   0x0f
    call    hex
    call    txready
    movwf   TXREG,0
    movlw   ' '
    call    txready
    movwf   TXREG,0
    swapf   resval,w,0
    andlw   0x0f
    call    hex
    call    txready
    movwf   TXREG,0
    movf    resval,w,0
    andlw   0x0f
    call    hex
    call    txready
    movwf   TXREG,0
    movlw   0x0d
    call    txready
    movwf   TXREG,0
    clrf    resbin,0
    clrf    resval,0
    return

txready
    btfss   TXSTA,1,0
    goto    txready
    return

    end
```

Appendix C

Vita

Jessica Luan was born in South Bend, Indiana. After choosing to attend the University of New Hampshire in 1998, she received her Bachelor of Science degree in electrical engineering in May of 2002 and then moved to Blacksburg to study control systems at Virginia Tech. She is currently working at the Northrop Grumman Corporation in Baltimore, Maryland.