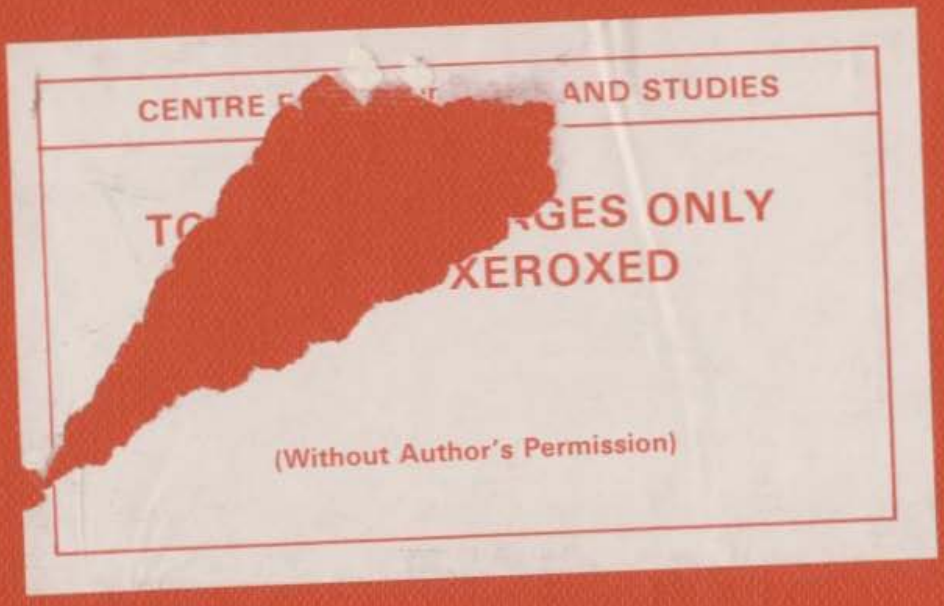
A DIGITAL SIGNAL PROCESSING FISH TRACKER, INVESTIGATION AND DESIGN



MICHAEL SULLIVAN

A Digital Signal Processing Fish Tracker, Investigation and Design

by

© Michael Sullivan, B. Eng

A thesis submitted to the School of Graduate Studies in partial fulfilment of the requirements for the degree of Master of Engineering

Faculty of Engineering and Applied Science,
Memorial University of Newfoundland,
St. John's, Newfoundland

August 1994

ISBN 0-315-96069-8

ABSTRACT

This thesis describes an investigation into the design of a digital signal processing ultrasonic fish tracker. This work was initiated to overcome problems with an analog system presently used at Memorial University to track individual cod. Consisting of a local oscillator and several superheterodyne receivers, the analog system now works intermittently and gives unreliable results. A study into a more compact receiver design based on recent technological advancements was then commenced. The new receiver will connect to the existing four hydrophones, interface through a PC ISA bus, fit on a single PC proto-board, and track three frequencies simultaneously from each of the four hydrophones. Position resolution under 1 m is achievable, over a 500 m range.

For this study only passive fish tracking systems are investigated where a transmitter is attached to the fish and a grid of detecting hydrophones are placed around the area of interest. These systems are reviewed from beat frequency receivers used in the 1950's, to computer controlled superheterodyne receivers in use today. Analog designs are prevalent throughout all these systems.

To understand the received signal, a review of sonar fundamentals is presented. This includes the transmitter, the transmission medium (water) and

the receiving hydrophone. The sonar equation is introduced as a means of calculating the detection level at the receiver. Background noise and problems caused by working in shallow water are also considered.

An undersampled technique is used to reduce the data rate and the computational load on the Digital Signal Processing (DSP) chip. Signals from each of the four hydrophone channels are multiplexed together and then digitized at 41 kHz. Digital data from each hydrophone is transferred to the DSP chip and grouped into blocks of 64 points. A fast Fourier transform (FFT) is used to convert each block into the frequency domain. Transmitter frequency is deemed detected if the magnitude response at one of the transmitter frequencies is greater than the threshold.

Undersampling equations are developed to give upper and lower bounds on the sampling frequency for each of the frequency folds. In the frequency domain, the effect of smearing on adjacent transmitter frequencies is investigated as a function of sample rate and bandwidth. Bit growth problems when using fix-point FFT calculations are also discussed.

The system is designed wherever possible with existing components, in order to keep the cost of a prototype low. Finally, simulations of the DSP source code confirm that the system performs as designed.

ACKNOWLEDGEMENT

In my pursuit of the Master's degree in engineering, I have received valuable advice and assistance from many people and organizations; these are very much appreciated. In particular, I would like to thank:

- Professor Michael Bruce-Lockhart, my supervisor, for his guidance and support throughout the program, and for his advice and explanation of various problems associated with the design;
- Dr. John Green of the Biology Department, for the problem;
- Institute for Marine Dynamics for their financial support;
- Memorial University of Newfoundland for their financial support;
- Judi Mellor, my wife, Julia and Peter my children, without whose support, encouragement and patience this would not have been done.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SYMBOLS, ABBREVIATIONS, AND ACRONYMS

α	Absorption Coefficient	dB/meter
ε	Error	
ε_d	Distance Error	
µPa	Unit of Pressure	Micro Pascal
v	Wind Speed	km/hr
π	3.14159...	
σ_e	Quantization RMS Error	
σ_s	Signal Magnitude RMS	
a	Aperture Width	
b	Bit Spacing	Voltage
ADC	Analog to Digital Converter	
AGC	Automatic Gain Control	
AM	Amplitude Modulation	
BW	Frequency Bandwidth	Hz
c	Sound Velocity in Water	m/sec
d	Depth	m
dB	Unit of Intensity	Decibel
DAC	Digital to Analog Converter	
DR	Dynamic Range	
DSP	Digital Signal Processing	
DT	Detection Threshold	dB
e	2.7182818...	
EPROM	Electrically Programable Read Only Memory	
f	Frequency	
F	Nyquist Folding Frequency	
f_c	Centre Frequency	Hz
FFT	Fast Fourier Transform	

FIFO	First-In, First-Out Buffer	
f_s	Sampling Frequency	
$h(i)$	Impulse Response Function (Continuous)	
H_{OBP}	Resonant Gain	
HS	Hydrophone Sensitivity, dB re 1v/1μPascal	
Hz	Unit of Frequency - One Cycle per Second	Hz
i	Time Index	
IBM	International Business Machines	
IEEE	Institute of Electrical and Electronic Engineers	
IF	Intermediate Frequency	
ISA	Internal Standards Association	
j	$\sqrt{-1}$	
J_{max}	Maximum Acceptable Jitter	s
k	Index	
kHz	Unit of Frequency - One Thousand Cycles per Second	kHz
L	Sample Length ($=NT$)	
LSB	Least Significant Bit	
mA	Unit of Current, Milli-Amp	mA
MSB	Most Significant Bit	
M_T	Magnitude Response Threshold	
MUN	Memorial University of Newfoundland	
n	Number of Bits	
n_g	Maximum Possible Bit Growth	
N	Number of Samples	
NL	Noise Level	dB
NSL_T	Thermal Noise Spectrum Level	dB
NSL_W	Wind Noise Spectrum Level	dB
N_z	Number of Zeros	
p	Index	

PC	Personal Computer	
PLL	Phase Lock Loop	
PROM	Programable Read Only Memory	
q	Index	
r	Range	m
RAM	Random Access Memory	
S/H	Sample and Hold	
SL	Source Level, dB re 1 µPascal @ 1 meter	
SNR	Signal to Noise Ratio	dB
SONAR	SOund NAvigation and Ranging	
s_{ppt}	Salinity, Parts per Thousand	
t	Time (Continuous)	s
T	Sample Interval	s
t_c	Temperature, Degrees C	
TL	Transmission Loss	dB
VCO	Voltage Controlled Oscillator	
V_{dc}	Voltage, Direct Current	V
V_{RMS}	Voltage, RMS	V
w_0	Centre Frequency, Radians per Second	rad/s
$X(f)$	Frequency Domain Function (Continuous)	
$x(t)$	Time Domain Function (Continuous)	
x_i	Time Series (Sampled)	
X_k	Fourier Transform of x_i (Sampled)	

1 INTRODUCTION

This study began as an investigation into problems with a fish tracking system used at Memorial University of Newfoundland (MUN). Dr. John Green of the Biology Department has been using a Liebnitz-Lann Acoustic Receiver, described in Section 2.4.4, for many years to study the movements of individual cod. This system uses a superheterodyne receiver for each hydrophone and tracks one pinger frequency at a time. Each receiver outputs a digital pulse whenever a pinger frequency is detected. A Personal Computer (PC) measures time differences between the digital pulses and then calculates the location of the pinger. Over the past two years, the system has worked intermittently and provided unreliable output. The company that built the receivers is no longer in business. A Canadian company markets similar receivers using the same design but they are expensive and require a receiver for each hydrophone. To track more than one pinger, the system scans each frequency for a fixed period and then switches frequency. For simultaneous tracking, a receiver is required for each hydrophone/frequency pair.

The past decade has seen the development of low cost special purpose, single chip, digital signal processors (DSP). Many applications have benefited from DSP chips including digital filtering, fast Fourier transforms (FFT), image processing, linear predictive coding, and sonar systems. Although not on the

market yet, it is possible to build a digital ultrasonic fish tracker on an IBM-PC compatible proto-board. Unlike earlier analog receivers, it is possible to process more than one hydrophone/frequency per board. This would greatly decrease the cost of equipment required for this type of fishery research. The hardware that was previously external to the computer can now be easily incorporated inside. A general purpose receiver can be designed and built, with tailoring for specific applications being handled in software. The DSP board would replace the receiver, oscillator and logic circuitry, making the system less prone to noise, easier to upgrade, and expandable with new software or improvements in DSP chips. New 486 PC's have adequate power to handle the computations required. Multiple pingers can be simultaneously tracked without additional hardware. With a generalized system it would be possible to use the computer for other applications, including tracking underwater animals by their sounds and tracking the position of underwater vehicles in scale model tests.

This thesis describes a design investigation of a digital signal processing, ultrasonic fish tracking system. The digital receiver is designed to fit on a single IBM-PC proto-board, process signals from four hydrophones and track three pinger frequencies. A literature review of passive fish tracking systems sets the foundation for this study. Following this are the fundamentals of sonar design that include the sonar equation. System requirements are identified and the

design equations that govern the system level design are examined. A detail design of the digital receiver is included with simulations to prove the concept.

2 LITERATURE REVIEW

This review begins with a brief history of underwater acoustic positioning, from the end of the last century to its initial use in fishery research in the mid 1950's. Differences between active and passive systems are explained. Evolution of fish tracking receivers is reviewed, from beat frequency receivers, to the current computer controlled superheterodyne receivers. The chapter finishes with a review of several sonar systems that use digital signal processing techniques.

2.1 History of Underwater Acoustic Positioning

Early in this century man turned his attention to sound waves in water. Submerged bells, attached to lightships, were detected by means of microphones mounted on the hull of a passing ship. (Kinsler et al, 1982) The distance to the lightship was calculated by timing the interval between the bell and a synchronized foghorn installed on the lightship. (Urick, 1983) By attaching two microphones to opposite sides of the ship, and feeding the signals separately to the right and left ear of the listener, the approximate bearing of the lightship could be found.

With the invention of submarines, and their use during the First World War, came the need to detect their presence. In echo-ranging systems, a sound pulse was transmitted into the water and the reflection from the submarine was received by a hydrophone. The signal was then amplified and headphones were used to detect the received pulse. Distance to the submarine was calculated by taking half the time delay from the transmitted pulse to the received pulse and multiplying the result by the speed of sound in water. (Tucker and Gazey, 1966)

By aiming the sound pulse directly down, the received pulse could find the depth of water beneath the ship. This system, known as echo-sounding, is now used mainly as a navigational instrument.

During World War II, much effort was placed on improving underwater acoustic systems for detecting, tracking and classifying submarines. The acronym SONAR (SOund NAvigation and Ranging) was coined during this period. Since the war, these systems have found uses in non-military applications like underwater and mineral exploration, hydrographic surveying, oceanography, fishery research and fish detection. Ranges for these systems vary from meters to kilometres.

The use of underwater acoustic systems in fishery research is relatively new and was first reported in the mid 1950's. (Trefethen, 1956) Applications in

this area include tracking movements of animals, measuring environmental conditions surrounding the animal, and the physiological and behavioral factors of the animals (biotelemetry).

Many fisheries research applications require knowledge of the location of underwater animals such as fish and crustaceans. These animals are usually located with respect to a local reference system or a surface vessel. Radio positioning systems used on land cannot be used underwater due to the high attenuation of electromagnetic waves in salt water. Acoustic positioning systems have a much more favourable propagation characteristic. It was with the development of the transistor, and the reduction in the size of components, that underwater acoustic positioning systems became useful. Acoustic positioning systems are classified into two types: active and passive.

2.2 Active Systems

Active sonar systems are the most commonly used underwater system. Targets are located by detecting, at a receiver, the reflections of an acoustic pulse originating from a transmitting transducer. (Figure 1) In fishery research, in order to track individual fish that have very weak echo pulses, transponders are sometimes attached to the fish. When the transponder detects an acoustic pulse, it generates another pulse that is much stronger and therefore easier to

detect. Directional transmitter/receiver transducers find the bearing and/or inclination angle to the target. Knowing the time delay from the transmitted pulse to the received pulse, the distance to the target can be computed by multiplying the time delay by the speed of sound in water. This pinpoints the location of the target in 3-dimensional space. Accurate measurement of the time delay is quite easy, but systems that can also accurately measure bearing angles are expensive.

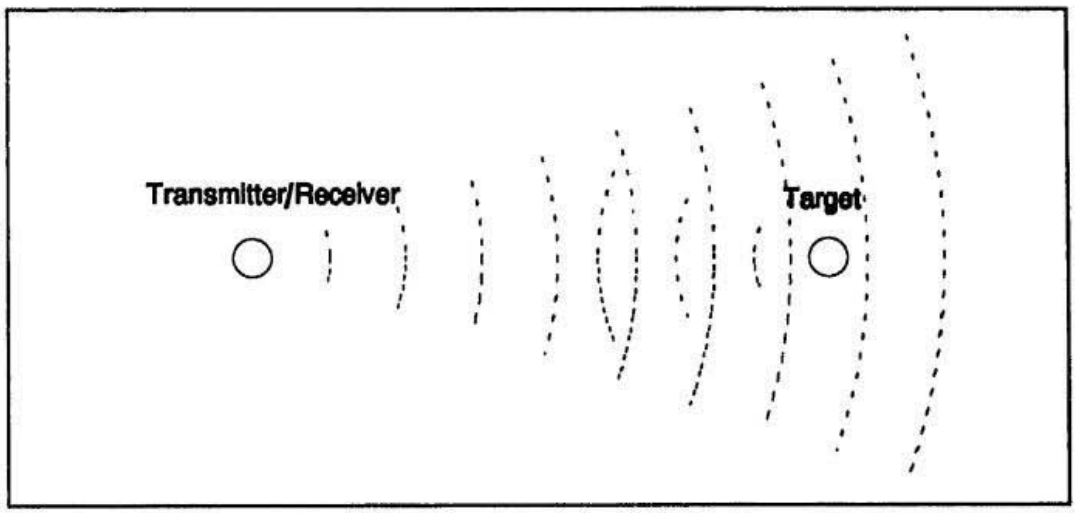


Figure 1 Active Sonar System

2.3 Passive Systems

In a passive system, acoustic signals are generated by the object being tracked. (Figure 2) These might be generated by the object itself, sounds made by the animal, or by an acoustic source (i.e. pinger) attached to the fish. Two

or more directional receivers that measure bearing angles can be used to locate the sound source. Each hydrophone is rotated until the direction of the maximum signal is found. The position of the target is then calculated by triangulation.

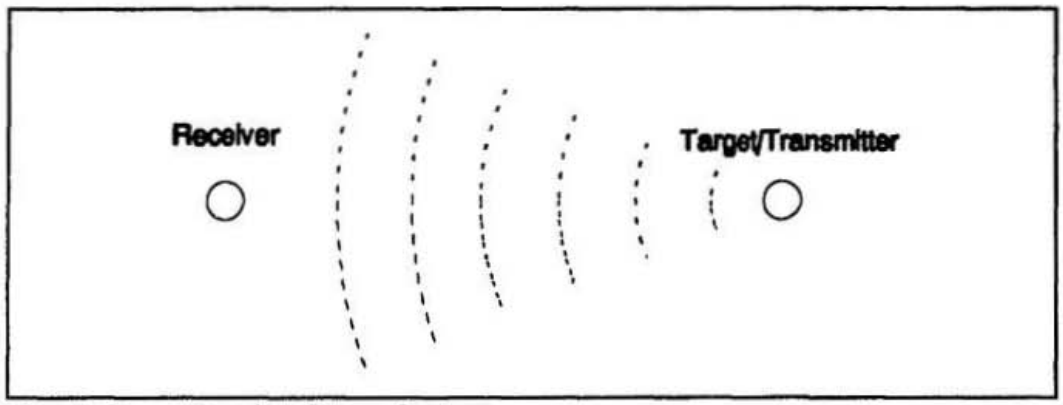


Figure 2 Passive Sonar System

By using three or more omnidirectional hydrophones, the hyperbolic coordinates of the target can be calculated from measured time delays from each hydrophone. Knowing the speed of sound in water, the location of the hydrophones and the delay in the pulse's arrival time between two hydrophones, a hyperbola on which the target lies can be calculated. A second hyperbola can be computed using a third hydrophone and the intersection of these hyperbolas locates the target in two dimensions. A fourth hydrophone is needed to locate the target in three-dimensional space. (Bellingham et al, 1992) When the hydrophones are spaced close together and the target is outside the hydrophone array, this is known as a short baseline array and is used where mobile or

portable equipment is necessary. A long baseline array is one where the target being tracked is within the array and the hydrophones are placed far apart with respect to one another. The long baseline array gives better accuracy within the array, but is less mobile and is generally set up as a permanent site.

2.4 Evolution of Fish Tracking Receivers

Sound pulses detected by a hydrophone and converted into electrical signals need to be processed through a receiver to be useful in fish tracking. Receivers have evolved from the simple beat frequency oscillators, used in the mid 1950's, to computer controlled superheterodyne receivers in use today. The evolution of receivers used in fish tracking system will now be discussed.

2.4.1 Beat Frequency Receivers

The first reported use of a sonic pinger to track an individual fish was Trefethen (1956). A transducer weighing only two grams in water and measuring less than two and one half inches long by one inch in diameter, was attached to the dorsal fin of a salmon. The pinger operated at 132 kHz with a repetition rate of 450 Hz. Audible signals were produced on a loudspeaker and a signal-level indicator showed the strength of the pinger pulse, thus giving an estimate of the distance to the fish. An automatic-tracking device was designed to home in on

the pinger's frequency. Another transducer transmitted narrow beam ultrasonic pulses at a different frequency. Return echoes, off the fish, from these pulses were displayed on a cathode-ray tube display showing range and bearing. A tilt-angle meter displayed the inclination angle at which the beam of sound pulses was aimed at the fish. This system was a combination of passive and active types. In confined areas, a passive portable receiver was used. An omni-directional hydrophone connected to an amplifier and loudspeaker detected the presence of the fish within a restricted area.

The above systems used a simple type of receiver known as a beat frequency receiver. Signals from the hydrophone are amplified and mixed with a signal from an oscillator that differs from the signal's frequency by only a few kilo-Hertz. Sum and difference frequencies are produced by the mixer. A low pass filter is used to remove the sum or higher components. The remaining difference or beat frequency is in the audio range and is amplified for a loudspeaker or headphones. By varying the frequency of the oscillator, tuning can be achieved. The sensitivity of these simple receivers is quite good. (Hawking and Urquhart, 1983)

2.4.2 Oscillograph

Standora et al (1972) and Ferrel et al (1974) described a system operating at 40 kHz used to monitor the open-ocean behaviour of a shark. Seven channels of information, including depth, swimming speed, compass heading, ambient light, and temperature, were encoded as pulse widths and time multiplexed into a one-pulse-interval-per-channel system. An eighth reference channel produced a constant inter-pulse width. Tracking was conducted by using a hand-held directional hydrophone with a receiver tuned to the frequency of the transmitter. Data was collected in the field by recording directly to magnetic tape. In the laboratory the tape was played back onto an oscillograph. Inter-pulse widths were measured manually with the aid of a magnification reticle. As this technique was very time consuming, only a small fraction of the data would be analyzed.

The first reported use of several widely spaced omni-directional hydrophones, a long baseline system, used to track a pinger, was by Hawking et al (1973), (1974). Weighing thirty grams in water and measuring 4.6 cm by 2.6 cm in diameter, the transmitter was inserted into the stomach of a fish. The ultrasonic transducer resonated at 41 kHz with a repeat interval of one second. An array of five omnidirectional hydrophones, placed around a sea loch on the west coast of Scotland, was used to receive the pulses. The distances between

the hydrophones were accurately measured by placing a powerful ultrasonic pinger at each hydrophone in turn. On shore the hydrophone signals were amplified and recorded on a direct record multi-track magnetic tape recorder. In the field, the hydrophone nearest the fish was used to trigger an oscilloscope timebase, and the time delays to the next two channels were measured. A more detailed analysis was done in the laboratory by replaying the tape recorder into a bank of tuned filters, pulse discriminators and then timer/counters attached to printers that recorded the time delay between channels. Movements of the fish were then plotted on a transparent bathometric chart superimposed with a grid of equal-delay hyperbolas for each hydrophone pair. Two fish were tracked using pingers operating at different pulse-lengths and repetition rates, but separation of the pulses proved quite difficult.

2.4.3 Phase Locked Loop Receivers

Pincock and Luke (1975) described several systems used for ultrasonic fish tracking including a description of their own receiver. The signal from each hydrophone was amplified through an automatic gain control system. The signal is passed through a tone decoder or phase lock loop (PLL), tuned to the transmitter's frequency. Signal detection logic examined the lock indication of the tone decoder, and outputs a *signal received* pulse. This system was used to identify and record the passage of fish carrying transmitters.

A phase locked loop receiver consists of a phase detector, a low pass filter and a voltage controlled oscillator (VCO). The free running frequency of the VCO is set to the centre of the expected input frequency and is fed into the phase detector with the input signal from the hydrophone. The detector generates an error voltage proportional to the phase and frequency difference between the two signals. This error voltage is then filtered, amplified and used to control the frequency of the VCO to reduce the frequency difference. The VCO is thus forced to synchronize or lock with the incoming signal and will automatically track any frequency changes in the input signal. Phase locked loop receivers output the step change in the control voltage when the signal pulse arrives at the receiver and the VCO locks in. Phase locked loop receivers have very high noise immunity and provide automatic frequency control. The loop has very good inherent frequency selectivity governed by the characteristics of the low pass filter.

2.4.4 Superheterodyne Receivers

Hawking et al (1979) and Johnstone et al (1991) described the Liebnitz-Lann Acoustic Receiver system that communicates with a computer to time the arrival pulses and display them on a terminal. A transmitter weighing ten grams in water and measuring 5.8 cm by 1.5 cm in diameter was inserted into the stomach of the fish. The ultrasonic transmitter produces a 1.5 ms pulse

at 75 kHz every 1.5 s. Up to seven hydrophones can be plugged into a seven channel amplitude modulated, superheterodyne receiver with a common local oscillator and digital outputs. This system is presently being used to conduct research on fish by Dr. John Green of the Biology Department at Memorial University of Newfoundland.

In a superheterodyne receiver, the first stage is similar to a beat frequency receiver, but the oscillator is known as a local oscillator and its frequency is much higher than the signal frequency. The mixer produces sum and differences well above the audio frequency range and an intermediate frequency (IF), usually the difference frequency, is separated by band-pass filtering. This frequency is often chosen to be about 455 kHz, the standard IF for AM radio receivers. A pinger operating at 45 kHz for example, would require a receiver with the local oscillator tuned to 500 kHz to produce an IF of 455 kHz. After the IF filter, which sets the bandwidth of the receiver, the signal is further amplified in one or more tuned amplifier stages. This produces a signal to noise ratio that is superior to that of a comparable beat frequency receiver. (Shrader, 1985) The IF signal can be mixed with a beat frequency oscillator to produce an audio frequency output.

The signal is demodulated to operate a voltage comparator where signals exceeding a preset threshold voltage are converted to digital pulses. Pulses need to have a minimum duration of 0.5 ms and triggering is inhibited for a

further 15 ms after each accepted pulse so that the circuit responds only to the leading edge of the direct path pulse. These pulses are then fed to the computer. With the fish tracking program running, the computer times the incoming signals to the nearest 0.1 ms and displays a continuous list of pulse timings at the active hydrophone. The computer checks the data and then computes the position of the pinger and stores it to disk. Time and translational x, y positions are also printed out.

A superheterodyne receiver, decoder and tape recorder are also described by Pincock (1979). A commercially available receiver was used with additional processing of the audio detector output to obtain logical signal indication. For a 10 ms transmitter pulse duration, signal detection occurs if the output exceeds some multiple of the average level for a few milliseconds. Multipath and echoes are rejected by disabling the signal detection circuit for an appropriate period, typically 150 ms. The decoder in a separate box, measures the period between received signals, checks for validity and then applies the linear calibration specified by the front panel switches and displays the data. A cassette recorder is used to store the data.

2.4.5 Computer Controlled Superheterodyne Receivers

Urguhart and Smith (1992) described improvements to the previous system. (Hawking et al, 1979) Hydrophones can both transmit and receive, so that their relative positions can be easily calculated. Through an IEEE interface the computer can control the frequency tuning. Software was updated to include on-line graphics, fish tracks and current fish position. A controller was added between the receiver and the computer to perform initial processing of the received signals. The controller monitors all receiver output lines and identifies the order of arrival of the pulses, calculates the time delay from the first pulse to all the others, and passes this information to the computer. The program identifies the nearest hydrophone to the pinger and then sets a time gate of a half second to each new set of pulses. As the transmitter moves, and the nearest hydrophone changes, the software requires three successive patterns before changing its order. Up to eight different frequency transmitters can be tracked by scanning each selected frequency for a fixed period, for example 20 seconds. To reduce the amount of data handled, fish positions were calculated at intervals of one minute, giving x and y positions.

2.5 Digital Sonar Systems

Kraeutner and Bird (1993) described a coherent sonar workstation built at the Underwater Research Laboratory at Simon Fraser University. The workstation is PC based and is used in a four meter by four meter by two meter test tank. The system is designed around an off-the-shelf PC DSP board that includes: two analog to digital converters (ADCs), two digital to analog converters (DACs) and a TMS320C30 digital signal processor. Three other modules make up the system; a frequency reference, a transmit/receive module and a timing/control module. Only the DSP board and the timing/control module reside inside the PC.

A subsampling technique is used to convert the bandpass signal down to the in-phase and quadrature baseband components. Operating range for the system is from 8 kHz to 1.2 MHz and is limited by the characteristic bandpass response of the receive transducer. Although there are two 16 bit ADCs, only one is used in this design. This system is not intended for real time analysis.

Analog Devices (1992) described a digital sonar beamformer designed around the ADSP-2100 digital signal processor. Sonar beamforming is the process of combining the outputs from many hydrophones in a way that detects signals from a specific direction. In practice, the signals from the hydrophones

are weighed, delayed and summed together to enhance signals from a particular direction. Analog systems that beamform in real time are designed using analog tapped delay times, are generally inflexible and very large. The digital beamformer is smaller, more accurate and has flexible hardware and software.

The system is made up of an analog acquisition front end, a number of processing slaves and a master controller that communicates with the slaves and the PC. Hydrophone outputs, connected to the acquisition card, are low pass filtered and converted to a 12-bit digital signal at a sample rate of 10 kHz. Each slave can form several beams and is designed using the ADSP-2100 chip. The master controller also uses an ADSP-2100 chip.

3 SONAR FUNDAMENTALS

To design a digital sonar receiver for shallow water applications, it is helpful to understand the individual components of an underwater acoustic tracking system. Design of transmitters will be reviewed, followed by the effects the medium has on the acoustic pulse. After that the design of hydrophones, which detect the pulse and convert it to an electrical signal, is discussed. Finally the theory behind data acquisition and digital signal processing is reviewed.

3.1 Ultrasonic Transmitter

A simple underwater acoustic pulsed transmitter can be divided into four functional blocks as shown in Figure 3. Some circuits combine two or three of these blocks into one. (Ireland and Lawson, 1979a) As most systems rely on batteries for power, pulsed transmission is used to extend the operation of the unit. An astable multivibrator is used to control the repetition rate of the transmitter circuit while a monostable is used to set the pulse duration. An oscillator produces a carrier frequency, at or near the transducer's resonant frequency. The output of the power amplifier is then applied across the transducer which vibrates at its natural frequency, producing an acoustic pulse or ping. (Woodward, 1988)

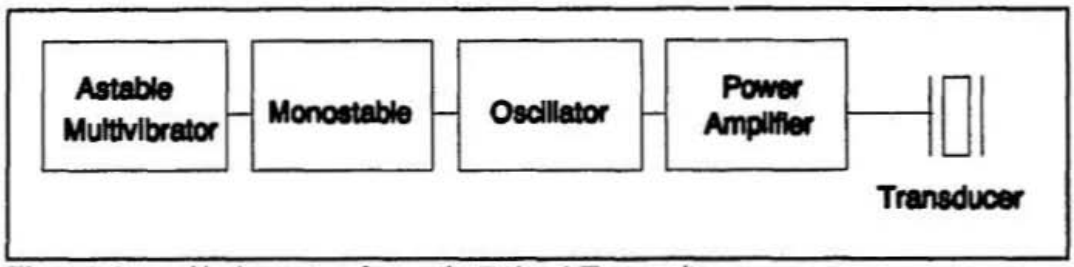


Figure 3 Underwater Acoustic Pulsed Transmitter

Ultrasonic frequencies are usually generated by driving a ceramic transducer at its resonant frequency. Spherical transducers, which produce omnidirectional signals, are desirable but expensive and difficult to mount. Cylindrical transducers are universally used because they give approximately omnidirectional radiation and are cheaper to produce and easier to mount.

To assure omnidirectionality, a cylindrical transducer should be shorter in length than diameter. Under these conditions, the transducer's resonant frequency can be considered proportional to its diameter. For a 20 kHz resonant frequency, a typical cylindrical transducer is 40 mm in diameter, and a 300 kHz transducer is normally less than 3 mm in diameter. (Mitson and Storeton-West, 1971) At high frequencies, the battery is the component limiting the transmitter size. As absorption losses increase with frequency, transmitter design is a tradeoff between size and detection distance. A popular frequency for monitoring coastal and freshwater fish is 70 kHz. At this frequency, the

transmitter package is approximately 50 mm by 10 mm and has a working range up to 1 km.

Source Level (*SL*), measured in decibels (dB), is the acoustic output power from a transmitter referenced to an acoustic pressure of 1 µPa. *SL* is obtained by extrapolating the radiated pressure from far field measurements to a distance one meter from the acoustic centre of the transmitter.

3.2 The Water Medium

Sound Velocity - From the transmitter to the receiving hydrophone, the acoustic pulse travels through water. To understand the detection level at the hydrophone it is necessary to look at the effect water has on the transmitted acoustic pulse. Sound velocity in the water can be calculated from the following empirical equation (Tucker and Gazey, 1966):

$$c = 1410 + 4.21\, t_c - 0.037\, t_c^2 + 1.14\, s_{ppt} + 0.018\, d \qquad (1)$$

Where c is velocity in meters per second, t_c is temperature in degrees C, s_{ppt} is salinity in parts per thousand and d is depth in meters. Under normal conditions sound velocity in water is approximately 1500 m/s. The above equation shows that the variations in sound velocity are mainly due to temperature. In shallow coastal waters, or regions with fresh water runoff from rivers, salinity variations

with depth have a greater influence on sound velocity than does pressure changes due to depth.

Transmission Loss - In the open ocean, if a signal is produced from a point source transmitter, spherical waves are produced. Cylindrical waves are produced in shallow water where the medium is bounded by two parallel boundaries: the ocean surface and the sea floor. For cylindrical spreading, sound intensity is proportional to the reciprocal of the range, therefore there is a loss of 3 dB for every two fold increase in range. Besides loss due to spreading, there is an additional loss due to absorption. Absorption loss, α, is expressed as dB/m and is approximately proportional to the square of the transmission frequency, f_c, in kHz. At 5 degrees C and one atmosphere, the absorption coefficient can be written as follows (Tucker and Gazey, 1966):

$$\alpha = f_c^2 \cdot \left(\frac{8 \cdot 10^{-5}}{0.7 + f_c^2} + \frac{0.04}{6000 + f_c^2} + 4 \cdot 10^{-7} \right) \qquad (2)$$

At 70 kHz, the absorption coefficient is approximately 0.02 dB/m. Transmission loss, *TL*, for cylindrical spreading can be expressed as follows:

$$TL = 10 \log r + \alpha r \qquad (3)$$

Where *TL* is the transmission loss in dB, r is the range in meters from the transmitter to the receiver and α is absorption coefficient. Figure 4 shows transmission loss in dB as a function of range for several operating frequencies.

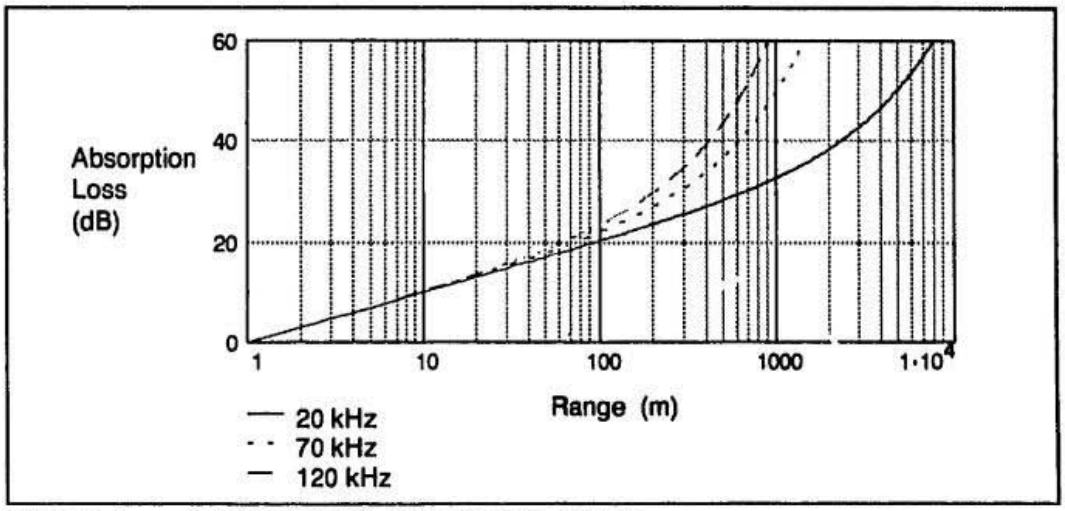


Figure 4 Transmission Loss versus Range

Transmission losses in coastal regions are generally quite different then those calculated from equation (3). A significant amount of scattering materials, including bubbles, particulate matter and fish, are found in sea water, making it less than an ideal sound transmission medium. Scattering of the sound beam as it passes through turbulent regions also affects the signal strength. If the acoustic impedance of the sea bed is similar to that of the water, as in a soft muddy bottom, it will complicate the propagation of the acoustic waves. Factors that influence these losses are: refraction, divergence or convergence of the acoustic pulse; interference caused by multipath from the surface and ocean bottom; diffraction and scattering caused by inhomogeneities in the water. Urick (1983) states that in shallow water, variations in sound levels by 10 to 20 dB have been observed to occur during tidal cycles due to multipath

transmissions. It is theoretically possible to compute the effects of these factors but as the characteristics of the ocean are constantly changing, it is not very practical.

Noise Spectrum - Acoustic noise affects the performance of underwater acoustic systems. (Figure 5) Between 500 Hz and 25 kHz, the primary source of acoustic noise is agitation of the sea surface. This can be calculated from the wind speed. For a given wind speed, the noise spectrum level falls at a rate of 17 dB per decade. (Kinsler et al, 1982) For frequencies ranging from 2 kHz to 100 kHz and for wind speeds between 1 km/hr and 50 km/hr, the noise spectrum level, NSL_w, due to wind is approximately:

$$NSL_w = 40 - 17 \log\left(\frac{f_c}{1{,}000}\right) + 6 \cdot \log_2\left(\frac{v}{1.8}\right) \quad (4)$$

Where v is the wind speed in kilometres per hour. (Kinsler et al, 1982) Above 50 kHz, the principal source of noise is thermal agitation of the water molecules. This further reduces acoustic power reaching the receiver, and the noise spectrum level due to thermal agitation increases at a rate of 20 dB per decade. The thermal noise spectrum level, NSL_T is approximately:

$$NSL_T = -15 + 20 \log\left(\frac{f_c}{1{,}000}\right) \quad (5)$$

At 70 kHz, the thermal noise spectrum level is 22 dB referenced to (re) 1 μPa per 1 Hz. At this frequency and for wind speeds above 5 km/hr, the sea surface noise can still be a contributing factor. For example, at a wind speed of 55 km/hr, the sea surface noise spectrum level is 38 dB re 1 μPa per 1 Hz at 70 kHz. Given that the wind is usually blowing, noise levels calculated from thermal noise alone are very optimistic.

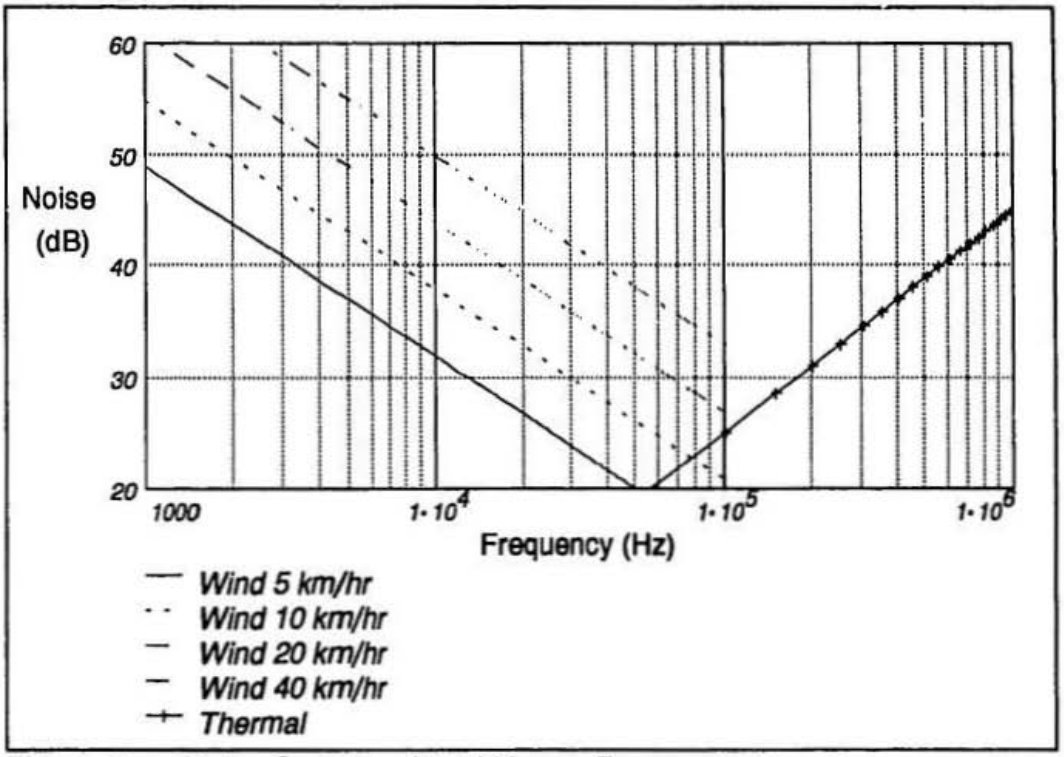


Figure 5 Noise Spectrum Level Versus Frequency

The noise spectrum level, *NSL*, refers to the level of noise in a square root of one hertz wide frequency band. For narrow bandwidth, the thermal

noise level, *NL*, can be calculated from the *NSL* and the bandwidth, *BW*:

$$NL = NSL + 10 \log BW\ (Hz) \qquad (6)$$

Further complicating noise level predictions in coastal regions are breaking waves, movement of ships, and the banging of rocks in the surf zone. These phenomena cause the noise levels to be considerably higher and highly variable in both time and place. (Kinsler et al, 1982)

3.3 Hydrophone

To respond to a wide band of frequencies, hydrophones are designed to operate below their resonant frequency. Above 50 kHz where thermal noise is dominant, hydrophones are designed with a narrow frequency band, typically 10 - 20 % of the resonant frequency. This improves the effective signal range. Cylindrical transducers similar to those found in transmitters, are often used. Elements are stacked in an array, which is omnidirectional in the horizontal plane and has restricted beamwidth in the vertical plane. Thus the hydrophone is more sensitive horizontally and less sensitive to surface noise and reflections. Due to the weak signal and the high source impedance of the transducer, a low-noise preamplifier is mounted in the hydrophone to overcome losses in the cables to

the receiver. In most applications, the receiver is noise limited rather than gain limited. (Stasko and Pincock, 1977) (Ireland and Lawson, 1979b)

Hydrophone sensitivity (*HS*) is generally reported in dB and is referenced to a one volt output produced by a sound pressure of one μPa. Sensitivity is generally obtained from an open-circuit response.

3.4 Sonar Equation

In all applications of underwater acoustics, the objective is to detect the desired signal in the presence of noise. In a passive system, the sound radiated by the transmitter at a source level, *SL*, decreases due to transmission loss, *TL*, on its way to the receiver. This is sometimes called the echo level. The sonar equation shows that the echo level has to be greater than or equal to the detection threshold, *DT*, plus the noise level, *NL*, to detect the signal at the receiver. (Urick, 1983)

$$SL - TL \geq DT + NL \tag{7}$$

Multipath, refraction, and scattering increase transmission loss. These factors vary widely with time and location and will not be included in these calculations.

In Appendix A, the sonar equation is used to calculate signal strength at the hydrophone's terminals, for a transmitter operating at ranges from one meter to three kilometres. Calculations are for the transmitter and hydrophone used at MUN and the results are compared to theoretical thermal noise levels. (Figure 6) These show, that under ideal conditions, the maximum signal to noise ratio for the given hardware is 81 dB. The operating range is limited by transmission loss and receiver noise to less than two kilometres.

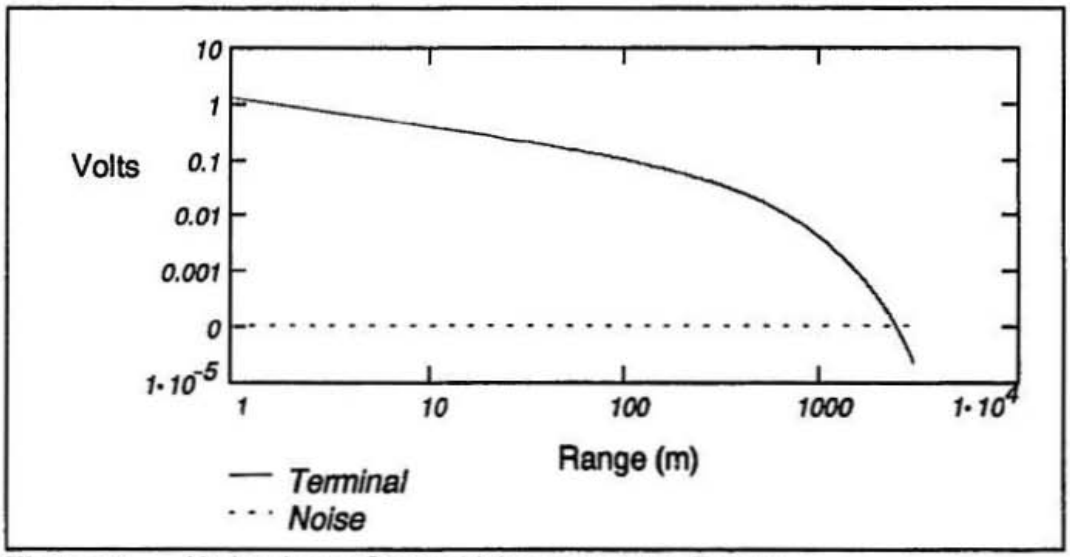


Figure 6 Hydrophone Output Voltage versus Range

3.5 Sources of Errors

The accuracy of an acoustic positioning system depends on several things. The modulation frequency of the transmitter is very important. Source

location resolution increases with higher frequency, but work area decreases due to higher attenuation. There is a tradeoff between working area and accuracy.

Refraction greatly affects sound levels. Variations in sound velocity in water are related to depth, temperature and salinity. As both the temperature and salinity vary with depth, the sound velocity will also vary with water depth. In coastal regions, near the surface, variations in salinity can be great due to river runoff, rain and ice melt. Water temperature near the surface will also fluctuate as sun and wind heat and cool the surface layer. Although these variations are small compared to the sound velocity, they significantly affect the propagation of sound. These variations cause the path of the pulse to be refracted up and down, off the surface and the ocean bottom. This causes reinforcement or cancellation of the pulse energy at different locations.

Hydrophone location must be found by surveying. Errors in hydrophone position will propagate into positional errors in the location of the pinger. In a hyperbolic system, the variation of error with position is very complex. Figure 7, show the lines of constant time differences (hyperbolic) between three hydrophones. The three hydrophones are marked by an overlapping circle and cross, and are at x, y positions (0,200), (-100,0) and (100,0) respectively. The lines of constant time difference between hydrophones 1 and 2 are dotted. The solid lines represent those between hydrophones 2 and 3, while the dashed lines

represent those between hydrophones 1 and 3. Accurate location of the pinger can be obtained within the array, but outside the array it is sometimes difficult to locate the pinger accurately. (MacLennan and Hawking, 1977)

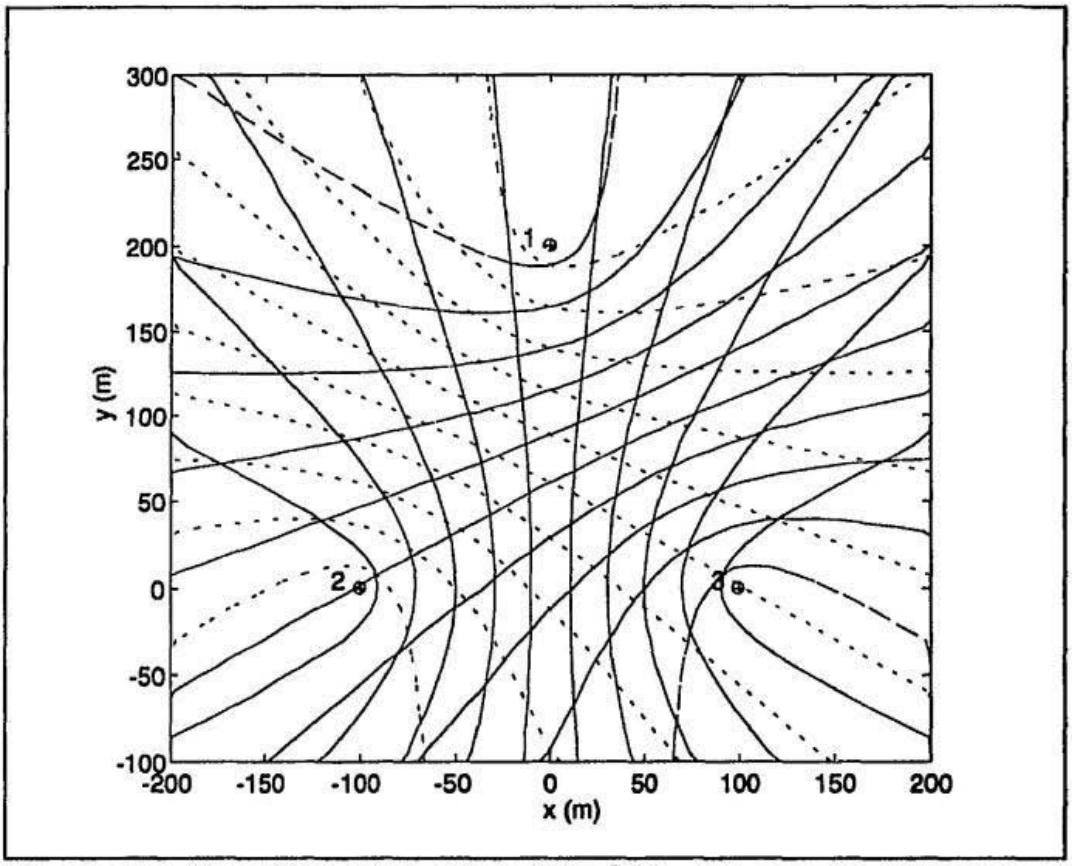


Figure 7 Three Hydrophone Hyperbolic Grid

In two dimensional space, hyperboloids can intersect in more than one location. (Anderson, 1979) (O'Dor et al, 1989) As a result of using the positive and negative square root, two solutions are always calculated for the pinger position. Sometimes the first solution is correct and other times the second is

correct. It is not always obvious which is the correct solution. Within the grid of hydrophones, the accuracy of the solution is very good but near the hydrophone and outside the grid the solution accuracy can be very poor.

Figure 8 is a contour plot of maximum positional errors within one meter of hydrophone 1 shown in Figure 7. The hydrophone is indicated by a delta symbol and is located in the centre of the plot at (0, 200). This plot is calculated from a plus/minus delay measurement error. Exact time delay measurements between the hydrophones are computed over a grid of pinger positions. At each grid point, a time error equivalent to one meter is added and subtracted from the two time delay measurements. Using these four conditions the pinger's position is calculated and compared to its actual position. The maximum variation from the actual position is then plotted on the contour plot. Even within one meter of the hydrophone, the positional error can be four times larger than the error found between the three hydrophones. (Figure 8) Detailed analysis of how the hyperbolic error is calculated and additional error graphs are presented in Appendix B.

The acoustic transducers and signal processing equipment may degrade accuracy if the transmitter output power, signal to noise ratio, receiver sensitivity or time-resolving capabilities are inadequate. Cost and size are usually the limiting factors.

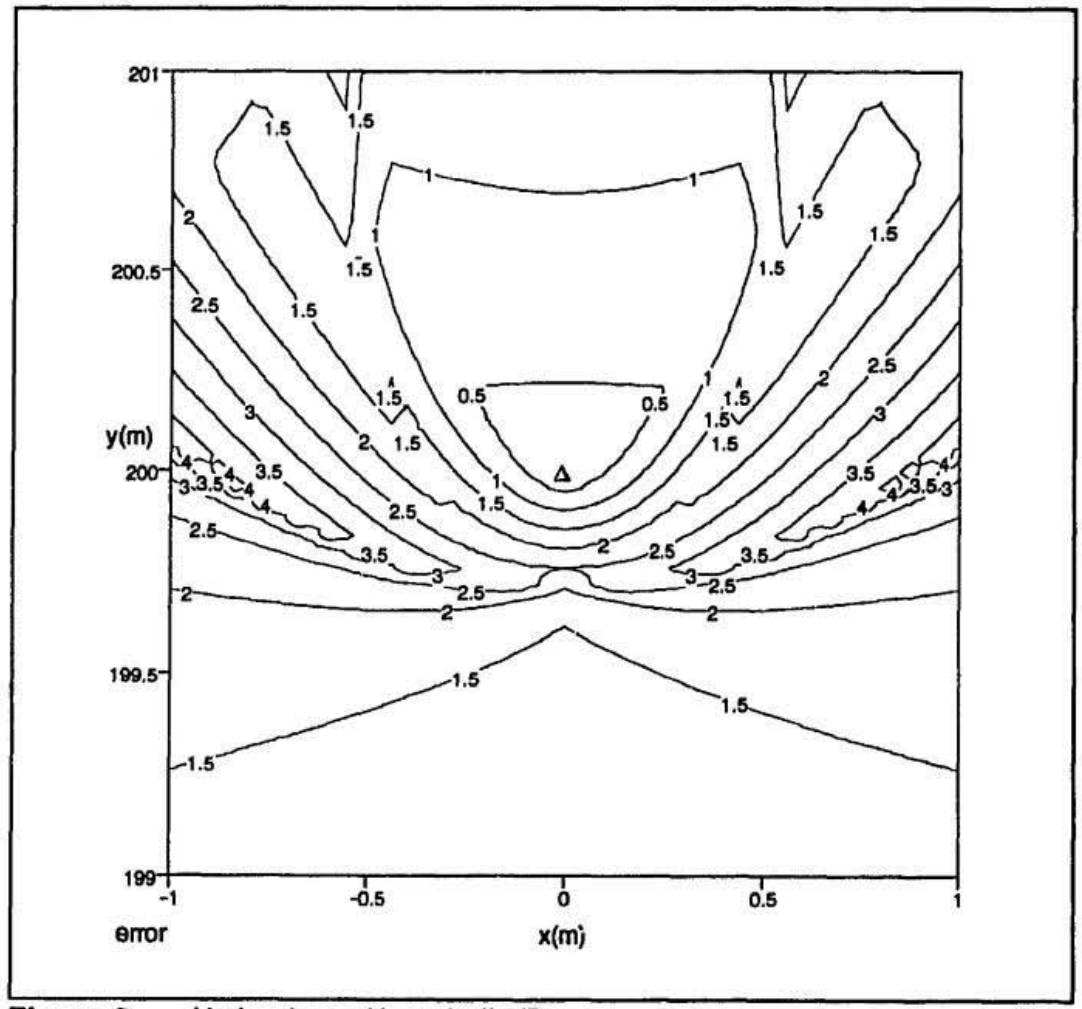


Figure 8 Hydrophone Hyperbolic Error

3.6 Data Acquisition

As this design is for a digital sonar receiver, analog signals from the hydrophones are converted into digital form by a data acquisition system. (Figure 9) A typical digital data acquisition system consists of many transducers,

each connected to a signal conditioning circuit. A multiplexer is used to select one transducer at a time. This signal is then sampled and held so that it can be digitized by the analog to digital converter and stored to memory. The timing of the entire operation is handled by a controller. (Garrett, 1981)

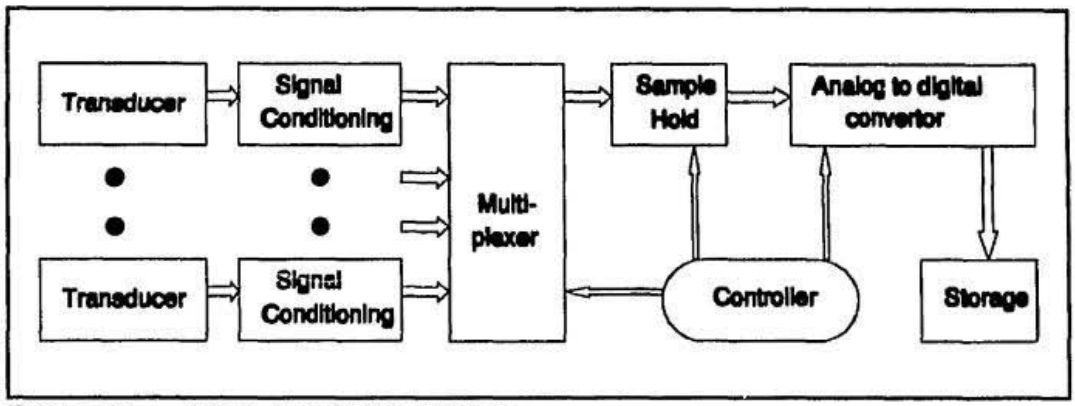


Figure 9 Data Acquisition System

Analog signals from the transducers generally need to be amplified and filtered (signal conditioned), before being processed. Signals are amplified so that they are within the range of the analog to digital converter. Analog signals are also passed through anti-aliasing filters to prevent aliasing of higher-frequency information. Aliasing is an error in the sampled data due to downward frequency translation of undersampled frequency components above half the sampling frequency by heterodyning with the sampling frequency. This error can be reduced by either increasing the sampling rate or by adding an analog filter to remove undesired frequencies. Generally, a lowpass filter with a cut-off frequency of half the sampling frequency is used.

The analog voltage value from each transducer is multiplexed one at a time through a sample and hold (S/H) circuit. A multiplexer is a switch that allows only one transducer at a time to be digitized. The sample and hold circuit samples the analog signal for a finite period. The sampled analog signal is then held at that constant level for the analog to digital converter, *ADC*.

The ADC converts analog voltage from the transducer into digital form. Output from the ADC is in a binary format. A measure of the ADC's output performance is its signal-to-noise ratio, *SNR*. *SNR* is dependent on the number of quantization levels. A more thorough discussion of *SNR* is presented in Section 4.3. Digital output data from the ADC is then stored on disk or in memory awaiting further processing.

The controller ensures that each operation occurs when it should. Multiplexer output needs to be available to the sample and hold circuit long enough for it to acquire and sample the data. Sample and hold output has to be held constant while the ADC performs digitization.

3.7 Digital Signal Processing Concepts

Several different approaches can be taken in the design of a digital receiver. Processing digitized signals could take place in the time domain, the

frequency domain or some combination of the two. The digital receiver described in this paper processes the hydrophone output in the frequency domain. As a result, this section will review some principles and techniques of frequency domain analysis.

Fourier transformation is a tool used in digital signal processing. The Fourier transformation, transforms a time domain function *x(t)* into a frequency function *X(f)* through the following relationship:

$$X(f) = \int_{-\infty}^{\infty} x(t)\, e^{(-j2\pi ft)}\, dt \tag{8}$$

The inverse Fourier transformation, transform a frequency domain function *X(f)* back into a time domain function:

$$x(t) = \int_{-\infty}^{\infty} X(f)\, e^{(j2\pi ft)}\, df \tag{9}$$

The magnitude response of the Fourier transform of a sinusoidal signal evaluated at f_c, is the amplitude of the time series divided by two (Brigham, 1974):

$$X(f_c) = \int_{-\infty}^{\infty} A \sin(2\pi f_c t)\, e^{(-j2\pi f_c t)}\, dt = A/2 \tag{10}$$

In practice the Fourier transformation is performed over a finite duration, L, defined by a data window, $w(t)$, where $-L/2 \le t \le L/2$:

$$w(t) = \begin{cases} 1, & |t| \le \frac{L}{2} \\ 0, & |t| > \frac{L}{2} \end{cases} \tag{11}$$

The Fourier transform of this rectangular pulse is:

$$W(f) = \int_{-\infty}^{\infty} w(t)\, e^{(-j2\pi ft)}\, dt = \int_{-L/2}^{L/2} 1 \cdot e^{(-j2\pi ft)}\, dt \tag{12}$$

$$= \frac{1}{-j2\pi f}\, e^{(-j2\pi ft)}\Big|_{-L/2}^{L/2} = e^{-j2\pi f\frac{L}{2}} - e^{j2\pi f\frac{L}{2}} \left(\frac{1}{-j2\pi f}\right) \tag{13}$$

$$= \frac{\sin \pi fL}{\pi f} \tag{14}$$

This is known as the sinc function (sinx/x) and is shown in Figure 10. L'Hospital's rule is used to obtain the maximum value of the sinc function, by taking the limit of equation (14) as f approaches 0. The maximum value is L, while the first zero crossings occur at:

$$f = \pm\frac{1}{L} \tag{15}$$

Width of the centre lobe in the frequency domain is $2/L$, which is twice the inverse of the width of the rectangular pulse in the time domain.

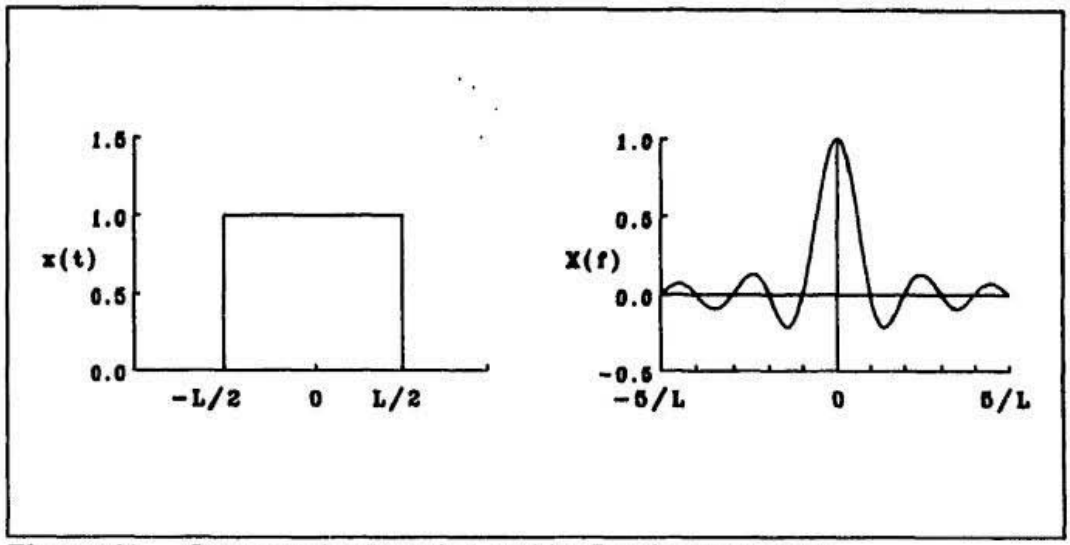


Figure 10 Rectangular Function and Its Fourier Transform

The product of two functions in the time domain is equivalent to the convolution of their Fourier transforms in the frequency domain.

$$y(t) = x(t)\, w(t) \quad \rightleftharpoons \quad Y(f) = \int_{-\infty}^{\infty} X(\tau)\, W(f-\tau)\, d\tau \tag{16}$$

Therefore, multiplying an infinite duration signal by a rectangular function is equal to convolving the transform of the original infinite signal with the sinc function. This has the effect of smearing the infinite signal's transform in the frequency domain. The resolution or bandwidth in the frequency domain using a finite sample interval of $-L/2$ to $L/2$ is $1/L$. In practice only a finite length of the sinusoidal time series is sampled. As a result, the magnitude response of an

infinite sinusoidal signal is the convolution of the sampled signal's magnitude response and the magnitude response of the sinc function:

$$X(f_c) = \int_{-L/2}^{L/2} A \sin(2\pi f_c t) \cdot e^{(-j2\pi f_c t)} dt = \frac{AL}{2} \tag{17}$$

The Fourier transform of a series of impulse functions at equidistant spacing of T and unit height, is another sequence of impulse functions at a spacing of $1/T$ and height $1/T$:

$$X(f) = \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(t - kT)\, e^{(-j2\pi ft)} dt = \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{T}\right) \tag{18}$$

Therefore a sampled finite sinusoidal signal can be obtained by multiplying the finite signal by the series of impulse functions in the time domain which is a convolution of the two transformations in the frequency domain. As a result, the magnitude response of a finite sampled sinusoidal signal at a frequency of f_c is:

$$|X(f_c)| = \frac{AL}{2} \cdot \frac{1}{T} = \frac{AL}{2T} \tag{19}$$

4 SYSTEM DESIGN

The complete design of the digital ultrasonic receiver is presented in this chapter. System requirements are identified from the existing hardware. This is followed by a discussion of the constraints that forced the evolution of the design. Finally, a functional description of the hardware and software components is presented.

4.1 System Requirements

The receiver presented in this paper is designed to replace an existing, Liebnitz-Lann Acoustic Receiver System, described in Section 2.4.4. In this system the outputs of the four hydrophone preamplifiers connect to their own analog superheterodyne receiver. All receivers are tuned to a single pinger frequency by a common local oscillator. Receivers output a 25 ms pulse whenever a pinger at that frequency is detected. Timing delays among the four receivers output pulses, are measured by a PC computer. Pinger position is found by the intersection of two or more hyperboloids calculated by averaging ten time delay measurements. From a selection of three pinger frequencies, one is tracked over a range of 500 meters to a resolution of one meter.

The existing VEMCO VH-65 Omnidirectional hydrophone requires a +12 Vdc, 30 mA supply to power its internal 50 dB low noise amplifier. Output from the preamplifier is band-limited between 50 kHz and 80 kHz by the mechanical design of the hydrophone. VEMCO series V3 16 mm crystal controlled transmitters with operating frequencies of 65.5 kHz, 69.0 kHz, and 76.8 kHz are presently used. These pingers transmit a 15 ms pulse every three seconds.

A digital design approach was chosen over an analog one because of the many advantages. To detect a single frequency pinger, one analog receiver is required for each hydrophone and takes up the same space as a PC board. Twelve analog receivers are needed to continuously detect three pinger frequencies from four hydrophones. A digital receiver can be designed to track multiple pinger frequencies from several hydrophones and fit on a single PC proto-board. Only one digital receiver is needed compared to twelve analog receivers, reducing cost and equipment required. With the digital system, the detection is performed in software making it easy to modify and upgrade by installing new PROM's or uploading improved software. For the analog system, improvements would require shipping equipment back to the factory for hardware changes.

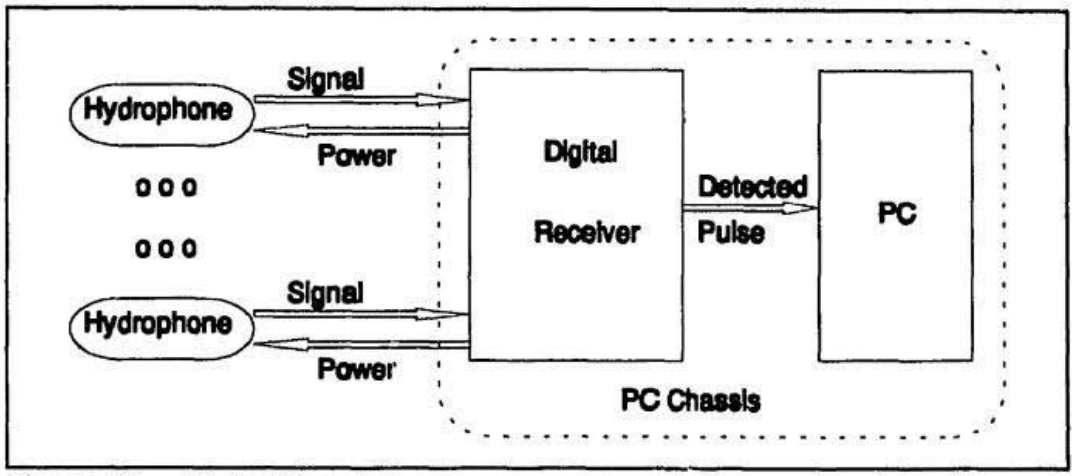


Figure 11 Digital Receiver

The new digital receiver is designed to fit on a single PC proto-board, interface with the four existing VH-65 hydrophones and communicate with the host computer through the standard ISA bus. (Figure 11) Although only three hydrophones are required to locate a pinger in two dimensional space, the fourth adds redundancy, and is helpful in discarding one of the two solutions found by using only three hydrophones. (Section 3.5) The digital system is designed to track simultaneously three crystal controlled pinger frequencies: 65.5 kHz, 69.0 kHz, and 76.8 kHz, over a range of 500 m with a resolution of one meter. Each pinger has a pulse duration of 15 ms and a three second repetition rate. After detecting a pinger pulse, information on the frequency and hydrophone is sent to the computer by the receiver. (Table I)

Table I System Requirements

Hydrophone		Computer
Signal From	Power To	Data To
50 dB Amplifier	12 V dc	Time
65.5,69.0,76.8 kHz	30 mA	Frequency
15 ms Pulse		Hydrophone

The design of the system was limited to components available in-house or easily purchased from suppliers. Although a more efficient and cheaper design is possible for mass production, the system, as designed, can be realized at low cost.

4.2 The System

Hardware for the digital ultrasonic receiver consists of two distinct sections, analog (data acquisition) and digital (digital signal processing). No matter what approach is taken in signal processing, the acquisition design would look similar to the one shown in Figure 9. (Section 3.6) Voltage signals from the four hydrophones would be filtered, digitized and then stored to memory.

Signal processing is designed around a Digital Signal Processing (DSP) chip. With this approach, the digital data from the acquisition system is segmented into blocks and converted via FFT into the frequency domain. If the

magnitude response at one of the pinger frequencies exceeds a threshold level then that pinger pulse is deemed to have been detected in that block. The hydrophone, pinger frequency and time of detection are then transferred to the PC.

Before reviewing the hardware and software design several issues that affect design constraints are discussed. Output signal to noise ratio, *SNR*, from the hydrophone's preamplifier is addressed first as this limits the maximum attainable receiver performance. An introduction to the technique of undersampling is then presented as undersampling is used in this design to reduce sample rates and processing requirements. Next, dynamic range of the DSP chip will be addressed. Finally, the problem of smearing in the frequency domain is investigated.

4.3 Hydrophone's SNR

Ultimately, performance of the receiver depends on the *SNR* of the hydrophone's pre-amplifier. Field data collected using a VR-60 hydrophone and a VEMCO V3-16 transmitter shows at distances of 2 and 55 meters the voltage output from the hydrophone was 0.7 V_{RMS} and 0.07 V_{RMS} respectively. (Evely, 1988) Noise levels of 10.0 mV_{RMS} were measured over a 10 kHz band centred at 45 kHz, giving a *SNR* of 37 dB. Using the sonar equation

(Section 3.4), the theoretical RMS output for the above hydrophone is calculated over a range of 3 km in Appendix A. (Figure 6) For distances of 2 and 55 meters, the sonar equation predicts output voltages of 0.9 V_{RMS} and 0.14 V_{RMS} respectively with an expected noise level of 0.016 mV_{RMS}. Data measured from the field are lower than those calculated. Near shore operating conditions, attenuation in the cables to shore, and the transmitter not operating at peak battery power could account for the differences. Field noise levels are much higher than those predicted from the noise spectrum. Even with a 150 km/hr wind the predicted noise would be only 1.29 mV_{RMS}. This higher noise level agrees with Kinsler et al (1982) who said that noise levels in shallow waters are considerably higher than those calculated from the noise spectrum.

SNR for the pinger/hydrophone pair is calculated from the one meter reference specification. It is possible for the pinger to operate within one meter of the hydrophone, but the probability that the pinger is inside this range is quite small when compared to the time spent over the 500 m operating range.

$$P\{<1\,m\} = \frac{\pi\, 1^2}{\pi\, 500^2} = 4 \cdot 10^{-6} \qquad (20)$$

Because of problems in the hyperbolic solution space discussed in Section 3.5, the data from a pinger operating within the one meter range is not very useful but it is possible for the pulse to saturate the acquisition system. During saturation, any other pinger transmission would be masked from that

hydrophone. With a pulse length of 15 ms and a repeat period of three seconds, any one pinger is transmitting half a percent of the time. Adding another 15 ms for reverberation then the probability of any one pinger being on is 1 percent of the time. Assuming the three pingers are not synchronized, then the probability that any two of the three pulses will be detected simultaneously is:

$$P\{2,3\} = \binom{3}{2} \cdot 0.01 = 0.03 \qquad (21)$$

Thus the joint probability that a pinger is saturating the acquisition system and masking another pinger is the product of the above two probabilities, $0.12 \cdot 10^{-6}$. With such a low probability, it is reasonable to use the one meter reference specification in calculating the *SNR*. At one meter, the pinger's output power is 150 dB re 1 μPascal and the theoretical background thermal noise over the pinger range of frequencies is 69 dB re 1 μPascal giving a *SNR* of 81 dB. The *SNR* measured from the field data is only 37 dB. Somewhere between these two numbers is the operational *SNR* of the hydrophones.

An important specification of the ADC is its output *SNR*. For a bipolar input signal with zero mean, the RMS magnitude, σ_s, is defined as the RMS

value of the peak to peak voltage:

$$\sigma_s = 2^n \cdot \frac{b}{2\sqrt{2}} \tag{22}$$

Where n is the resolution limit for the ADC in bits and b is the spacing between each bit level in volts. Quantization RMS error, σ_e, is randomly distributed within the bit level b in volts and can be calculated from the following formula:

$$\sigma_e = \left(\frac{1}{b} \int_{-b/2}^{b/2} \varepsilon^2 \cdot d\varepsilon \right)^{1/2} = \frac{b}{2\sqrt{3}} \tag{23}$$

The equation for the theoretical signal-to-noise ratio for a sine wave input is the log of the RMS input signal divided by the RMS error:

$$SNR_{dB} = 20 \log \left(\frac{\sigma_s}{\sigma_e} \right) = \left(\frac{2^n \cdot b/2\sqrt{2}}{b/2\sqrt{3}} \right) \tag{24}$$

$$= (6.02n + 1.76)\, dB \tag{25}$$

Where n is the number of bits used by the ADC.

To capture the full theoretical *SNR* would require a 13-bit ADC while the field measurement would require only a 6-bit ADC. (Equation (25)) One approach would be to use an automatic gain control (AGC) and an 8-bit ADC. The danger here is that while the AGC is tracking a strong frequency channel a weaker frequency channel will be lost in the noise. Although a 13-bit converter

is required for the hydrophone's theoretical *SNR*, the field measured *SNR* is much smaller and it was decided to design the receiver with a 12-bit analog to digital converter that has a *SNR* of 74 dB.

4.4 Undersampling

When data is sampled at discrete times, a well-known phenomenon called aliasing or frequency folding occurs. Aliasing is an error in the sampled data due to downward frequency translation of undersampled frequency components above half the sampling frequency by heterodyning with the sampling frequency. In this receiver design, aliasing is taken as an opportunity to reduce data rates, reduce storage and bring the signal down closer to the baseband. A familiar example of this may be seen in the movies when a stagecoach starts to accelerate from a stopped position. The wheels first appear to be revolving slowly. As the stagecoach increases in speed the wheels appear to slow down, stop, and then revolve in the reverse direction. This is aliasing, and is caused by the slower frame rate of the movie projector compared to the much higher angular rate of the spokes. It is possible to calculate the rotation speed of the wagon wheel, by knowing the projector film speed and the number of frequency folds. Take for example a frame rate of 24 per second. When the wheel starts to slow down it is rotating at 12 rev/sec, and when it appears to have stopped it is rotating at 24 rev/sec.

Specifically, if a sinusoid signal is sampled at an interval T, but has a frequency greater than $1/(2T)$, the signal will appear as a frequency between 0 and $1/(2T)$. Instead of appearing at its correct frequency the signal appears at an aliased or lower frequency. To illustrate frequency folding, a sinusoidal signal is first defined:

$$x(t) = \sin 2\pi ft \tag{26}$$

where $x(t)$ is sampled at interval i, with an interval spacing of T.

$$x_i = \sin 2\pi f(iT) \tag{27}$$

Next, divide the sinusoid frequency range into intervals or folds from:

$$0 \rightarrow \frac{1}{2T}, \quad \frac{1}{2T} \rightarrow \frac{2}{2T}, \quad \frac{2}{2T} \rightarrow \frac{3}{2T}, \quad \ldots . \tag{28}$$

Where p is an integer defining the fold and q is the fractional part of that fold.

$$f = \frac{p+q}{2T} \tag{29}$$

Combining Equations (27) and (29) gives:

$$x_i = \sin 2\pi f(iT) = \sin 2\pi \left(\frac{p+q}{2T}\right)(iT) = \sin \pi (p+q)i \tag{30}$$

Using the sine addition formula, sin(a+b) = sin(a)cos(b) + cos(a)sin(b):

$$= \sin \pi (p)i \cos \pi (q)i + \cos \pi (p)i \sin \pi (q)i \tag{31}$$

As p is an integer, $\sin\pi(p)i = 0$, for all i:

$$= 0 + \cos\pi(p)\, i\, \sin\pi(q)\, i \tag{32}$$

This equation can be further simplified depending on whether p is even or odd.

First for even p, $\cos\pi(p)i = 1$, for all i

$$x_i = 1 \cdot \sin\pi(q)\, i = \sin\pi(q)\, i \left(\frac{2T}{2T}\right) \tag{33}$$

$$= \sin 2\pi \left(\frac{q}{2T}\right) i\, T \tag{34}$$

Therefore for even fold p, the effective frequency is $q/(2T)$.

Now for odd p, $\cos\pi(p)i = \cos\pi i$, for all i. Rewriting Equation (32):

$$x_i = \cos\pi i\, \sin\pi(q)\, i \tag{35}$$

The previous sine addition formula can be written as

cos(a)sin(b) = sin(a)cos(b) - sin(a-b). Therefore:

$$x_i = \sin\pi i\, \cos\pi(q)\, i - \sin(\pi i - \pi q i) = 0 - \sin(\pi i - \pi q i) \tag{36}$$

$$= -\sin\pi(1-q)\, i = -\sin\pi(1-q)\, i \left(\frac{2T}{2T}\right) \tag{37}$$

$$= -\sin 2\pi \left(\frac{1}{2T} - \frac{q}{2T}\right) i\, T \tag{38}$$

The Nyquist folding frequency, F, is defined as:

$$F = \frac{1}{2T} \tag{39}$$

For odd fold, p, the effective frequency is $(F - q/(2T))$. For even folds, it can be seen that as the actual frequency increases, so does the effective frequency but for odd folds the effective frequency decreases.

Figure 12 is a graph of frequency folding resulting from sampling data at a frequency f_s. The x-axis is the actual sampled frequency while the y-axis is the effective or aliased frequency. Both axes have been normalized by dividing by one-half the sampling frequency, f_s. For example if two frequencies, 25 Hz and 175 Hz were sampled at 100 Hz then normalizing both by one half the sampling frequency, gives 0.5 and 3.5. From the graph it can be seen that the two sampled signals cannot be distinguished from one another because both have the same effective frequency of 25 Hz.

Undersampling makes use of the aliasing phenomenon, but precautions need to be taken in choosing a sampling frequency. For example choosing a sampling frequency in the middle of the band of interest would fold all frequencies above the sampling frequency down on top of the frequencies below the sampling frequency. Sampling frequency upper and lower bounds can be found for each frequency fold in terms of the centre frequency of the band of

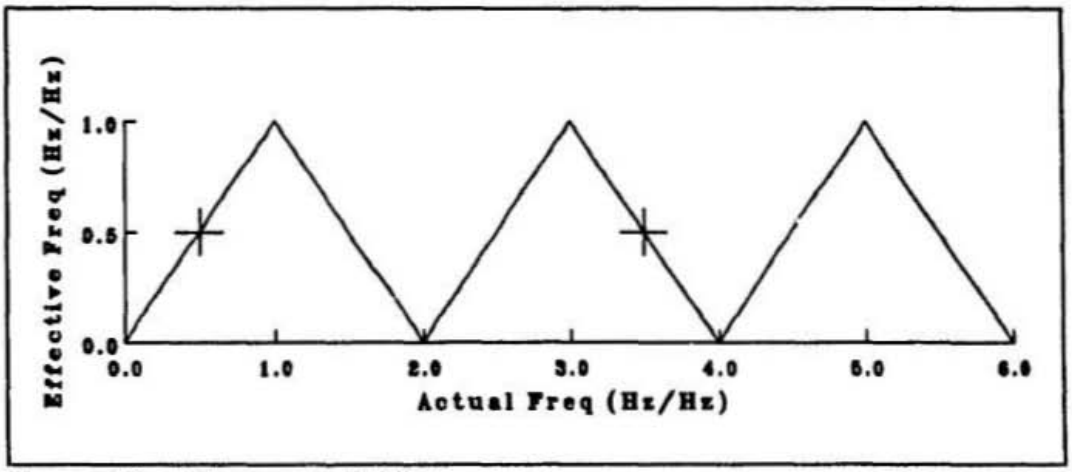


Figure 12 Frequency Folding

interest and its bandwidth. For the first fold, the lower bound for the sampling frequency is the Nyquist frequency while the upper bound is limited by the equipment used and can be considered infinite:

$$\textit{first fold} \quad : \quad 2\cdot(f_c+1/2\,BW) \le f_s \le \infty \tag{40}$$

Sampling frequency for the second fold is bounded between half of the Nyquist frequency and twice the lowest frequency of interest:

$$\textit{second fold} \quad : \quad \frac{2\cdot(f_c+1/2\,BW)}{2} \le f_s \le \frac{2\cdot(f_c-1/2\,BW)}{1} \tag{41}$$

The sampling frequency of the third fold is bounded between one third of the Nyquist frequency and the lowest frequency of interest:

$$\textit{third fold} \quad : \quad \frac{2\cdot(f_c+1/2\,BW)}{3} \le f_s \le \frac{2\cdot(f_c-1/2\,BW)}{2} \tag{42}$$

Sample frequency range for the k^{th} fold can be written as:

$$\frac{2\cdot(f_c+1/2\,BW)}{k} \le f_s \le \frac{2\cdot(f_c-1/2\,BW)}{(k-1)} \tag{43}$$

The Nyquist criterion states that the sample frequency has to be at least twice the bandwidth to record the signal of interest:

$$f_s \ge 2\,BW \tag{44}$$

Combining Equations (43) and (44) the maximum usable *k th* fold is:

$$\frac{2\cdot(f_c+1/2\,BW)}{k} \ge 2\,BW \tag{45}$$

$$n \le \frac{f_c}{BW}+\frac{1}{2} \tag{46}$$

Where n defines the range of acceptable folds for a frequency band with a bandwidth BW centred at f_c.

Figure 13 is used to find the sampling frequency. The x-axis is twice the centre frequency, normalized by the sampling frequency. From zero to one is the first fold, one to two is the second fold, two to three is the third fold and so on. The available bandwidth normalized by the centre frequency is plotted on the y-axis. The value of the y-axis is calculated by taking the absolute value of the available bandwidth divided by the centre frequency. The available bandwidth is the fold number multiplied by the sampling frequency and subtracted from twice the centre frequency. To use the graph, the bandwidth of

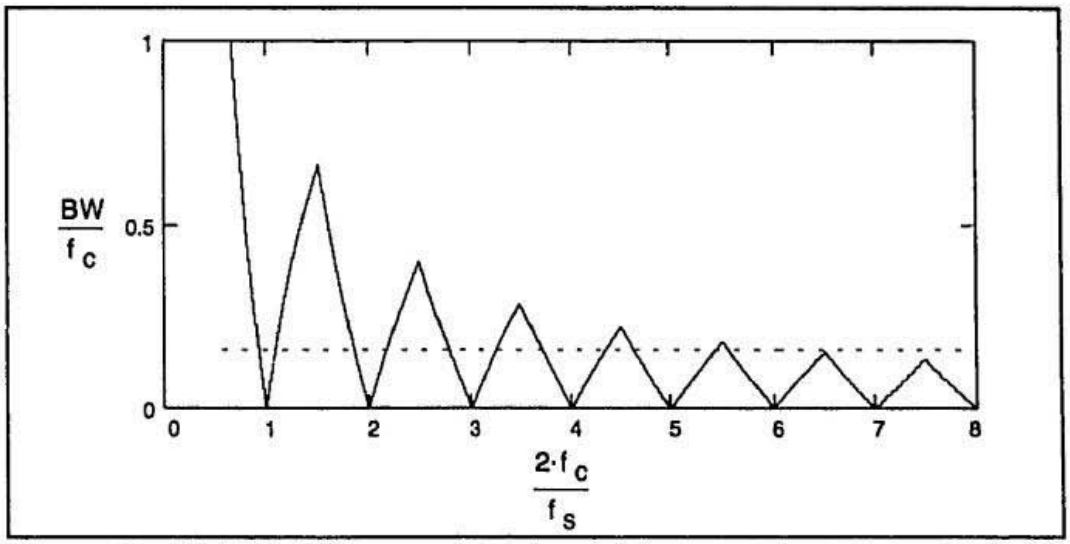


Figure 13 Folding Frequency Bounds

interest is divided by the centre frequency and a line is drawn across from the y-axis at that value. For example, with a bandwidth of 11.3 kHz and a centre frequency of 71 kHz, a line would be drawn across from the y-axis at the value of 0.159. Acceptable sampling frequencies are those where the folding frequency bound curve is above that line. For this example, only the first six folds can be used, because above this, the bandwidth is wider than half the sample frequency. The relationship between the sample frequency and the centre frequency of the *k th* fold is developed in Appendix C.

Design pinger frequencies for the digital receiver are 65.5 kHz, 69.0 kHz and 76.8 kHz respectively. Thus, the operating range has a centre frequency of 71 kHz and a bandwidth of 11.3 kHz. From equation (44), the minimum sampling frequency is 22.6 kHz and equation (46) defines the maximum fold to

be six. Upper and lower bounds on the sample frequency are calculated for each of the six folds. (Table II)

Table II Upper and Lower Bounds on Sampling Frequency

k^{th} (fold)	1	2	3	4	5	6
f_s (Upper Bounds)	∞	131	65	43	33	26
f_s (Lower Bounds)	153	77	51	38	31	26

Using a sampling frequency from the sixth fold, the upper and lower pinger frequencies are placed right at the folding frequency and DC. To ensure that there is enough room between the pinger frequencies and the ends of the frequency range, only the first four folds will be considered as sampling frequency ranges. Thus, the lowest reasonable sampling rate is 41 kHz, which is in the middle of the fourth fold.

4.5 Frequency Dynamic Range

Dynamic range in the frequency domain is the magnitude response peak of a full resolution sinusoidal wave divided by the RMS value of the quantization noise. (Otnes and Enochson, 1978) To find the peak value in the frequency

domain a sampled sinusoidal signal is first defined as:

$$x_i = A \sin 2\pi f(i\,T) \tag{47}$$

Where A is the amplitude, f is the frequency, i is the sample number, and T is the sampling interval. The magnitude of the peak in the frequency domain depends on the number of samples and the amplitude of the sinusoidal signal (Section 3.7):

$$|X_k| = \frac{AL}{2T} = \frac{AN}{2} \tag{48}$$

Where L is the period and is equal to the number of samples, N, multiplied by the sampling interval, T. If b is the spacing between each bit level in volts and n is the number of bits in the digitized data, then the maximum amplitude, A, of the sinusoidal signal is only half the range. Rewriting Equation (48) in terms of the bit resolution and number of samples:

$$|X_k| = \frac{\left(\frac{b\,2^n}{2}\right)N}{2} = \frac{b\,2^n N}{4} \tag{49}$$

Parseval's relation will be used to define the quantization error in the frequency domain. For sampled data, power in the time domain is equal to power in the frequency domain divided by N (Brigham, 1974):

$$\sum_{i=0}^{N-1} x_i^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X_k|^2 \tag{50}$$

Which is equivalent to:

$$\sigma_f^2 = N\sigma_t^2 \tag{51}$$

Quantization error in the time domain as defined from Equation (23) is $b/2\sqrt{3}$. The quantization error in the frequency domain is:

$$\sigma_e = \sqrt{N}\,\frac{b}{2\sqrt{3}} \tag{52}$$

The dynamic range in the frequency domain is calculated by combining Equations (49) and (52):

$$DR_{dB} = 20\log\left(\frac{|X_k|}{\sigma_e}\right) = 20\log\left(\frac{\sqrt{3}}{2}\sqrt{N}\,2^n\right) \tag{53}$$

Dynamic range in the frequency domain is dependent on both the number of samples, N, and the number of bits, n, in the digitized data. For each additional bit in the digitized data, the dynamic range increases by 6 dB. By doubling the number of samples, the range goes up by 3 dB. From Figure 14, the 12-bit digital receiver will have a frequency domain dynamic range greater than 80 dB, for block sizes larger than 8.

4.6 Frequency Smearing

The transform of a finite length sinusoidal signal is the convolution of its Fourier transform and the Fourier transform of the rectangular pulse.

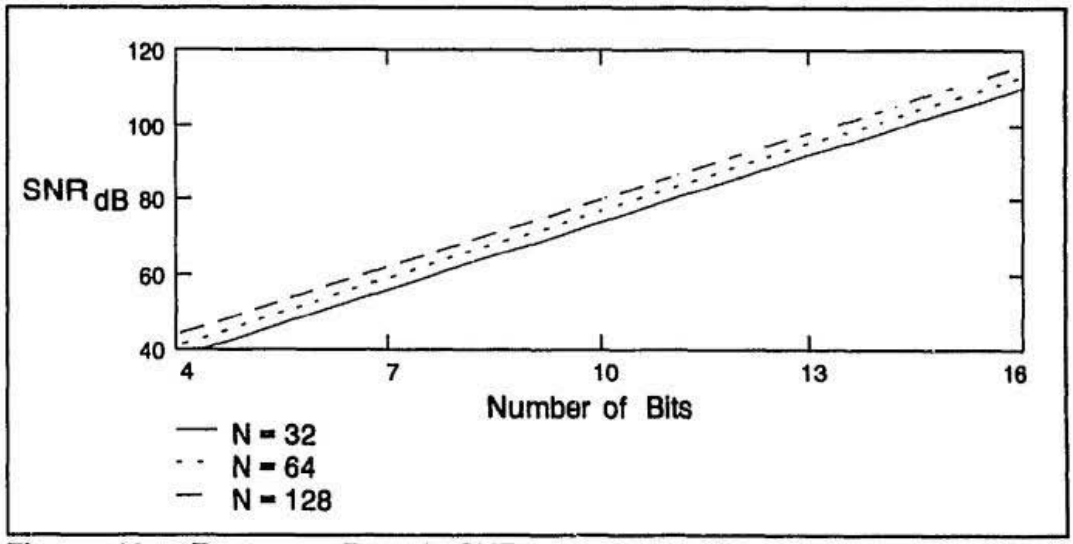


Figure 14 Frequency Domain SNR

(Equation (16)) This results in smearing of energy in the frequency domain. Smearing is a problem when the smeared magnitude response of a strong signal masks a nearby weaker frequency response. To understand the masking problem, an investigation of the sinc function follows.

If a sine wave's frequency is an integer multiple of the FFT's frequency resolution, then only one peak shows up in the frequency domain plot with no smearing. This is because in the convolution, the centre peak of the sinc function overlaps that of the sinusoidal peak while at all other values it overlaps the sinc function at its zero crossings, which are at the frequency resolution intervals, $1/L$. Maximum smearing occurs when the sine wave's frequency is an integer multiple plus a half of the FFT's frequency resolution. This results in the maximum and minimum points of the sinc function overlapping the sinusoidal

peak at the frequency resolution intervals (Figure 15). Maximum smearing, S_m can be calculated by evaluating the sinc function (Equation (14)) at odd intervals of f equals $1/(2L)$:

$$|S_m| = \left|\frac{\sin(\pi f L)}{\pi f}\right| = \frac{1}{\pi\left(\frac{(2k-1)}{2L}\right)} = \frac{L}{\pi(k-1/2)} \tag{54}$$

Where L is the total time and k is the (k^{th}-½) frequency resolution interval from the peak of the sinc function.

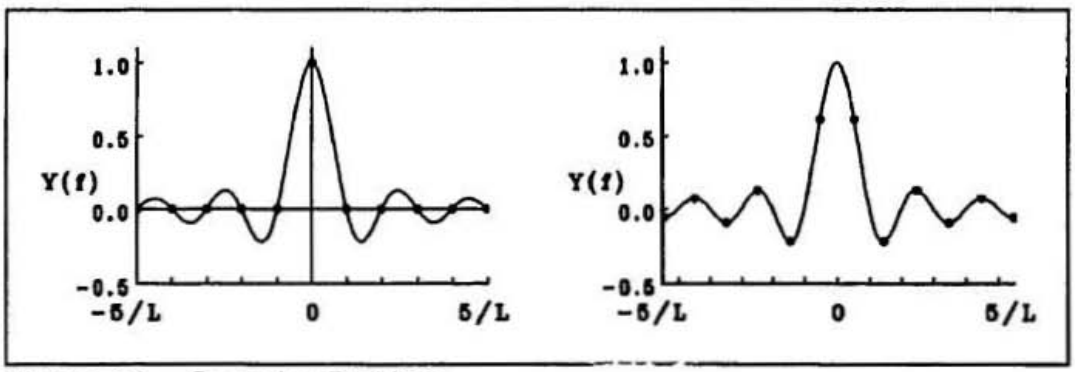


Figure 15 Smearing Problem

In this application the two frequencies 65.6 kHz and 69.0 kHz are only 3.5 kHz apart and have the greatest potential of smearing one another. Smearing depends on the number of frequency resolution intervals between the two frequencies. For a sample rate of 41 kHz, the frequency resolution in terms of block size is equal to 41,000/N. The number of frequency resolution intervals

between the two signals is:

$$k = 3{,}500 \cdot \left(\frac{N}{41{,}000}\right) + 1 = 0.085 \cdot N + 1 \qquad (55)$$

Now the maximum magnitude response in dB at one frequency caused by smearing from the other can be written in terms of the block size:

$$|S_m| = \left|\frac{N/41{,}000}{\pi \cdot (0.085 \cdot N + 1 - 1/2)}\right| = \left|\frac{N/41{,}1000}{0.268 \cdot N + \pi/2}\right| \qquad (56)$$

From Section 3.7, the peak value of the sinc function is L or $N/41{,}000$. Table III shows the peak magnitude response of a 65.5 kHz sinusoidal signal divided by its smeared magnitude response at 69.0 kHz, in dB, as a function of block size. To reduce the smear magnitude to background noise levels would require a block size of over 4096.

Table III Block Size versus Masking Response

Block Size	32	64	128	256
Smear Magnitude (dB)	-20	-25	-31	-37

A common technique used to reduce smearing in the frequency domain is to window the time series data. In the time domain, windowing is used to reduce the effects of potential discontinuities at both ends of the time segment. These windows have the effect of suppressing the side lobes of the sinc function. Several windows have been proposed in the literature including Bartlett,

Hamming, Hanning (Ahmed and Natarajan, 1983). Their effects include widening and reducing the heights of the main lobe and decreasing the remaining side lobes. Figure 16 shows the effect in the frequency domain when the windows are applied in the time domain. A Raised Cosine or Hanning window is defined in the time domain as :

$$w_c(t) = \begin{cases} \frac{1}{2}\left(1 + \cos\frac{\pi t}{L}\right), & |t| \le \frac{L}{2} \\ 0, & |t| > \frac{L}{2} \end{cases} \tag{57}$$

The Fourier transform is (Ahmed, 1983):

$$W_c(f) = \sin\left(\frac{\pi f L}{\pi f}\right)\left(\frac{1}{1 - f^2 L^2}\right) \tag{58}$$

Table IV shows the effect of a Hanning window on the smeared magnitude responses of Table III. The Hanning window reduces the effects of masking to an acceptable level for block sizes greater than 64.

Table IV Block Size versus Hanning's Masking Response

Block Size	32	64	128	256
Smear Magnitude (dB)	-49	-60	-72	-84

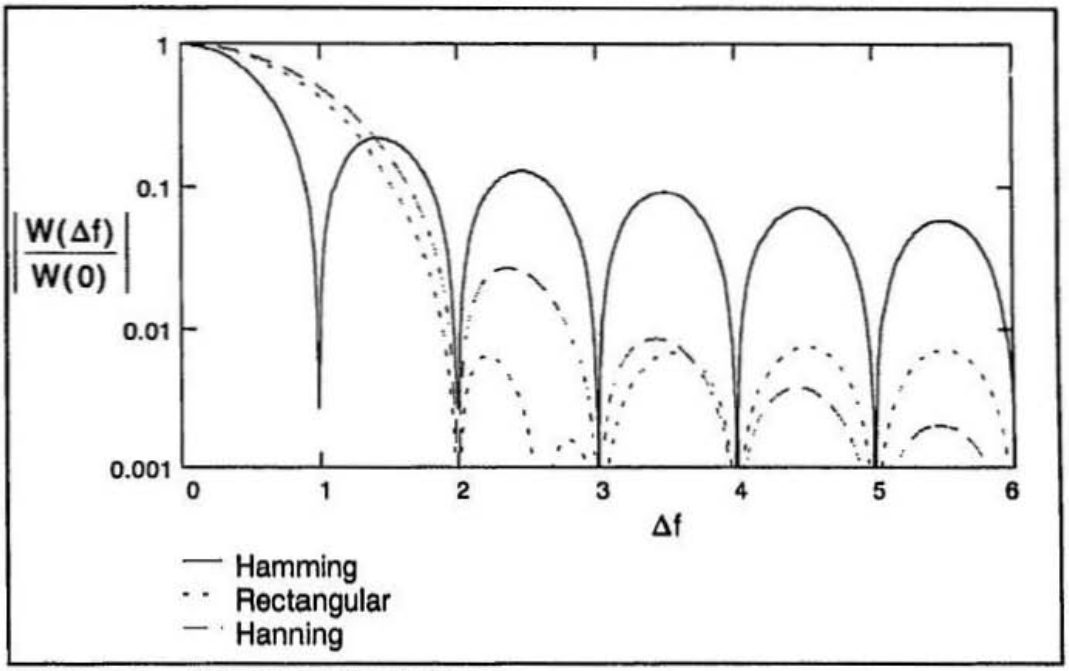


Figure 16 Windowing Techniques

4.7 Cycle and FFT Times

The digitized sampled data is grouped into blocks of *N* points. Cycle time is the time to sample a block of *N* data points. For the sample frequency of 41 kHz, the interval time between samples is 24.4 μs, yielding a cycle time of *N* times 24.4 μs. Positional resolution is also affected by block size as it is calculated from the speed of sound in water multiplied by the cycle time. Cycle time and resolution as a function of block size are shown in Table V.

Table V Block Size versus Cycle Time

Block Size (samples)	32	64	128	256
Cycle Time (ms)	0.78	1.56	3.12	6.24
Resolution (m)	1.17	2.34	4.48	9.36

The speed of FFT computations depends on the block size and the clock speed of the chip. Running at 16.66 MHz, the ADSP-2101 chip will perform a 64 point Radix-4 FFT in 85 μs.(Analog Devices, 1992) Approximate times to process the four hydrophones and overhead (collection of data and computation of results) for different block sizes is displayed in Table VI. For block sizes of 256 or larger the data from the four hydrophones will not fit in the DSP's internal memory and therefore is not capable of fast FFT processing. As a result only block sizes of 128 and smaller were considered for the design.

Table VI Block Size versus FFT Time

Block Size (samples)	32	64	128	256
FFT Time (ms)	0.26	0.57	1.31	3.10

Table V and Table VI shows that for any of the listed block sizes, there is enough time to collect data and perform the FFTs in real time. Pinger position is calculated in the preset analog system by averaging 10 delay times. Averaging the delays decreases the positional resolution by a factor of three. Therefore using block sizes of 128 and 256 will not achieve the design resolution

of one meter. From Table IV, the smear magnitude on the adjacent frequency, with a Hanning window, is down 49 dB for a block size of 32 and 60 dB for a block size of 64. A block size of 64 was chosen as the best compromise because at 60 dB down it is only 14 dB above the ADC SNR. Recall that masking is only a problem when pulses from different pingers arrive at the same time and that this is a rare event.

4.8 Fast Fourier Transform

Discrete Fourier transforms (DFTs) are used to compute the Fourier transform of a sampled signal at equally spaced frequency intervals. (Oppenheim and Schafer, 1975)

$$X_k = \sum_{l=0}^{N-1} x_l W_N^{lk} \qquad k = 0, 1, \ldots, N-1 \tag{59}$$

Where W_N are the exponential phase rotations $e^{-j2\pi/N}$. Evaluation of a DFT can be quite time consuming as there are $4N^2$ real multiplications and $N(4N-2)$ real additions, thus the computation time is approximately proportional to N^2.

As early as the beginning of this century, C. Runge used the symmetry and periodicity of W_N^{lk} to reduce the computational requirement of the DFT to $N\log_2 N$, but its significance was lost because of the small values of N used in hand calculations. It was not until 1965 when Cooley and Tukey published a

paper describing an efficient algorithm that the DFT become widely applied. This paper sparked a surge of activity into the development of other computational efficient algorithms which are now collectively known as the fast Fourier transform, FFT. (Oppenheim and Schafer, 1975) A full development of the fast Fourier transform is handled in many books on signal processing and as a result only a quick review will be presented here. (Ahmed and Natarajan, 1983) (Brigham, 1974) (Oppenheim and Schafer, 1975)

To achieve the dramatic increase in efficiency, it is necessary to decompose the DFT into successively smaller DFTs. For this application, as N is an integer power of four, the radix-4 FFT is used. The radix-4 FFT first divides a 64-point DFT into four 16-point DFTs, then into 16 4-point DFTs. Reduction of the computational effort of the DFT is performed by first dividing it into four summations which are then split into four separate equations. Each of these equations computes every fourth output sample.

$$X_k = \sum_{i=0}^{N/4-1} x_i W_N^{ik} + \sum_{i=N/4}^{N/2-1} x_i W_N^{ik} + \sum_{i=N/2}^{3N/4-1} x_i W_N^{ik} + \sum_{i=3N/4}^{N-1} x_i W_N^{ik} \qquad (60)$$

By changing the limits on each summation, the equation can be written as:

$$X_k = \sum_{i=0}^{N/4-1} x_i W_N^{ik} + \sum_{i=0}^{N/4-1} x_{i+N/4} W_N^{(i+N/4)k} + \sum_{i=0}^{N/4-1} x_{i+N/2} W_N^{(i+N/2)k} + \sum_{i=0}^{N/4-1} x_{i+3N/4} W_N^{(i+3N/4)k} \qquad (61)$$

As each of the four summations is now over the same interval, all the terms can

be all moved under the one summation:

$$X_k = \sum_{i=0}^{N/4-1} \left[x_i + x_{i+N/4} W_N^{(N/4)k} + x_{i+N/2} W_N^{(N/2)k} + x_{i+3N/4} W_N^{(3N/4)k} \right] W_N^{ik} \qquad (62)$$

The twiddle (twirl) factor coefficients, W_N, are periodic and can be written as:

$$W_N^{(0)k} = 1 \quad , \quad W_N^{(N/4)k} = -j^k \quad , \quad W_N^{(N/2)k} = -1^k \quad , \quad W_N^{(3N/2)k} = j^k \qquad (63)$$

As a result, equation (62) simplifies to:

$$X_k = \sum_{i=0}^{N/4-1} \left[x_i + (-j)^k x_{i+N/4} + (-1)^k x_{i+N/2} + (j)^k x_{i+3N/4} \right] W_N^{ik} \qquad (64)$$

Four equations can be created from this equation such that $k = 4r + m$ for $m = 0$ to 4 and $r = 0$ to $(N/4 - 1)$:

$$X_{4r+m} = \sum_{i=0}^{N/4-1} \left[\left(x_i + (-j)^m x_{i+N/4} + (-1)^m x_{i+N/2} + (j)^m x_{i+3N/4} \right) W_N^{mi} \right] W_{N/4}^{ik} \qquad (65)$$

These are known as the four $N/4$-point DFT equations of the radix-4 FFT. Each of these equations is divided into four $N/16$-point DFTs which in turn are divided by four again. This division continues until there are only four-point DFTs. They are shown graphically in Figure 17 and it is from this figure that they get the name butterfly calculations.

The butterfly calculations are performed on complex numbers. The twiddle factors are divided into real and imaginary parts, $W_N = \cos(2\pi/N) - j \sin(2\pi/N)$. Representing the twiddle factors as $C - jS$, the real and imaginary output values

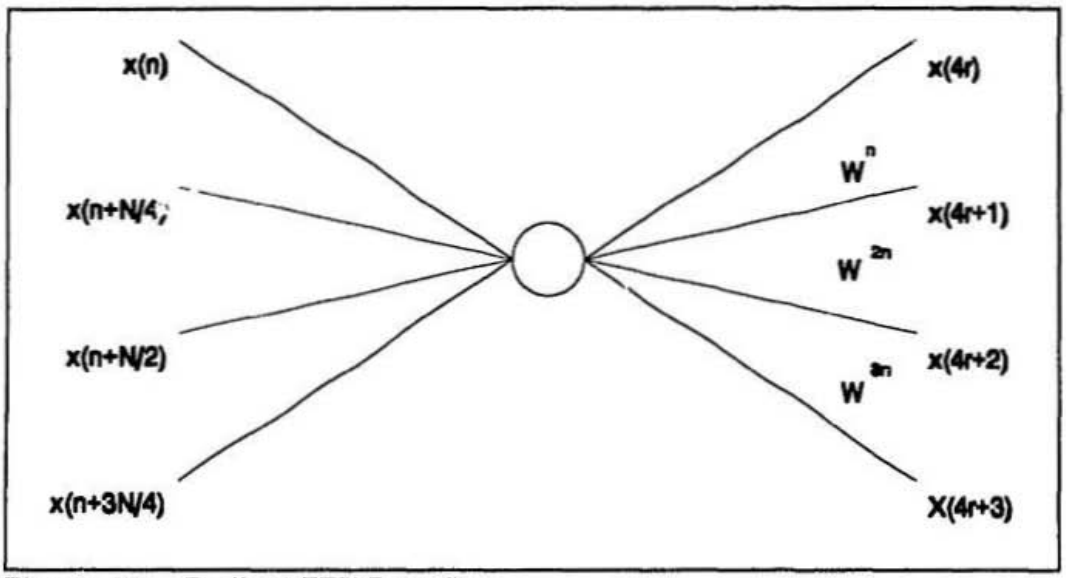


Figure 17 Radix-4 FFT Butterfly

are calculated by:

$$\begin{aligned} x_a' &= x_a + x_b + x_c + x_d \\ y_a' &= y_a + y_b + y_c + y_d \end{aligned} \tag{66}$$

$$\begin{aligned} x_b' &= (x_a + y_b - x_c - y_d)C_b - (y_a - x_b - y_c + x_d)S_b \\ y_b' &= (y_a - x_b - y_c + x_d)C_b + (x_a + y_b - x_c - y_d)S_b \end{aligned} \tag{67}$$

$$\begin{aligned} x_c' &= (x_a - x_b + x_c - x_d)C_c - (y_a - y_b + y_c - y_d)S_c \\ y_c' &= (y_a - y_b + y_c - y_d)C_c + (x_a - x_b + x_c - x_d)S_c \end{aligned} \tag{68}$$

$$\begin{aligned} x_d' &= (x_a - y_b - x_c + y_d)C_d - (y_a + x_b - y_c - x_d)S_d \\ y_d' &= (y_a + x_b - y_c - x_d)C_d + (x_a - y_b - x_c + y_d)S_d \end{aligned} \tag{69}$$

This algorithm belongs to a class of FFT's that perform their computation in place. The results from a given butterfly becomes the inputs into the next stage, as a result the outputs are written over the inputs and only $2N$ memory locations for the data are required. Twiddle factors can be calculated on the fly, but this uses valuable processor cycles. Since data memory is available, the real and imaginary parts of the twiddle factors are pre-calculated and stored separately in two array structures, each of length N.

As the DSP chip uses fixed point arithmetic, the butterfly has the potential to overflow by three bits at its output. Summation of the four inputs is the cause of this bit growth, not the multiplication, as this is always by a number less than or equal to one. Although three bits are required for one butterfly, data does not grow by this maximum amount over two consecutive stages. The maximum possible Fourier transform can be obtained from Equation (59) where x_l is the maximum bit value and W_N is always a value between plus and minus one, thus:

$$X_k = \sum_{l=0}^{N-1} 2^n \cdot 1 = N \cdot 2^n \tag{70}$$

As a result the maximum possible bit growth, n_g, across an N-point DFT is the

$\log_2$ of the difference of the maximum output and the maximum input or:

$$n_g = \log_2 N \tag{71}$$

For this application, the n_g is 6 bits and it is therefore important that precautions be taken to ensure that the 16-bit word does not overflow. Three methods are available to avoid this situation: input data scaling, unconditional block floating-point scaling, and conditional block floating-point scaling. Input data scaling requires that the input data have enough extra bits so overflow does not occur. In this application, the input data would be shifted down to 10 bits, leaving 6 bits for bit growth. This is not an acceptable solution as low level signals may be shifted out before being processed. Scaling down the outputs by a factor of three after each stage is called unconditional block floating-point scaling. With this approach, the initial input data would include only three guard bits. This is necessary for every stage except the output of the last stage. For conditional block floating-point scaling, the output data is scaled only if there has been bit growth in one or more of the outputs. Although computationally more expensive, this technique only shifts the data when bit growth has occurred and only by the amount of growth. The number of shifts would be saved so that the scaling of the Fourier transform would be known.

For a 64-point radix-4 FFT, there is half as many stages (3 instead of 6) and half as many butterflies in each stage (16 instead of 32) as for a 64-point

radix-2 FFT. Each butterfly has 12 multiplies and 22 additions for the radix-4 FFT compared to 4 multiplies and 6 additions for the radix-2 FFT. As a result the 64-point radix-4 FFT is computed in 85 % of the time of the radix-2 FFT computation.

5 DETAILED DESIGN

5.1 System Interface

Four VEMCO VH-65 Omnidirectional hydrophones are connected via a three-wire cable to the receiver. One wire supplies +12 V_{dc} at 30 mA, another the common ground and the third wire is the signal output from the hydrophone's preamplifier. These connections are made on the PC proto-board, with the power and ground being supplied by the PC computer.

The receiver is designed to fit on a JDR Microdevices PR-2 proto-board. This board includes buffering and decoding circuitry to the ISA 62-pin bus. The definitions for the pin numbers of the ISA bus are found in Uffenbeck (1987). Eight bit data transfer to and from the PC is handled using a three-state bi-directional bus transceiver. Access to the I/O ports between Hex 300-31F is also provided.

5.2 System Level Hardware

The system level hardware design is shown in Figure 18. The analog signals from each of the four hydrophones are bandpass filtered around the pinger frequencies. A multiplexer selects one hydrophone channel at a time to

be sampled. During the digitizer's conversion cycle, the multiplexer's output is held constant by a sample/hold and the previously sampled data is transferred into the ADC FIFO. Timing of the acquisition circuit is performed by the ADC controller, as well as supplying an address tag to the ADC FIFO. The DSP Decoder selects between reading data from the ADC FIFO or writing data to the DSP FIFO. The digital signal processing is performed by the DSP and is discussed in Section 5.3 on software design. After processing, the detected pinger's information is transferred to the PC via the DSP FIFO buffer.

5.2.1 Signal Conditioning

The maximum peak-to-peak output from the hydrophone is calculated from the pinger output power, 150'dB re 1 µPascal, and the hydrophone's sensitivity, -148 dB re 1 V/µPascal:

$$150\,dB + -148\,dB = 2\,dB = 10^{2/20}\,V_{rms} = 3.5\,V_{pp} \qquad (72)$$

In the field, the maximum peak-to-peak output voltage was measured at 2.0 V. Since the operating range for the ADC selected is ±2.5 volts, the hydrophone's output is not amplified.

Physically the hydrophone acts as a bandpass filter with roll-offs at 50 kHz and 80 kHz. Lower and upper pinger frequencies are 65.5 kHz and

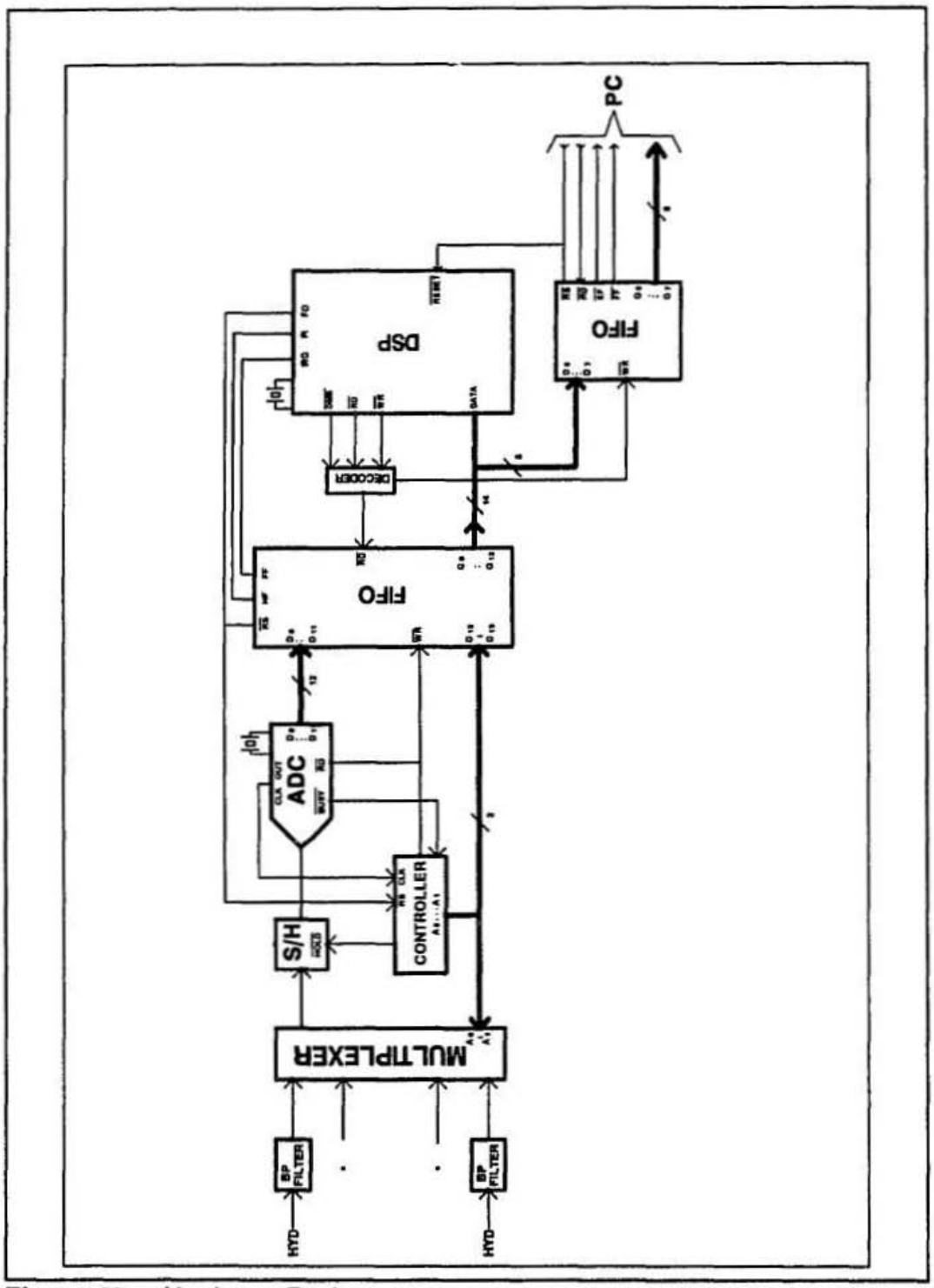


Figure 18 Hardware Design

76.8 kHz. To help further reduce wide band noise, a bandpass filter with a centre frequency of 71 kHz and a bandwidth of 11.3 kHz is used. Any phase shift introduced by the filter will not affect the time delay measurements between the hydrophones because phase shift is constant for any single frequency pinger. The quality factor, Q, of the filter is a measure of its selectivity and is defined as the ratio f_o/BW. For this filter, the Q is equal to 6.3. As mentioned earlier the attenuation for this filter is one.

A multiple-feedback bandpass filter with an input attenuator is used in the design. (Franco, 1988) The bandpass circuit and its frequency responses are shown in Figure 19 and Figure 20 respectively. The resonance gain (H_{OBP}) for this circuit is approximately:

$$H_{OBP} = 2 \cdot Q^2 = 38_{dB} \tag{73}$$

Initially, a 741 op-amp, with a unit-gain bandwidth of 1 MHz, was used in the design. With only a gain of 12 dB at 71 kHz, this op-amp was discarded. The 318 op-amp with a unity-gain bandwidth of 50 MHz and a gain of 55 dB at 71 kHz was more acceptable.

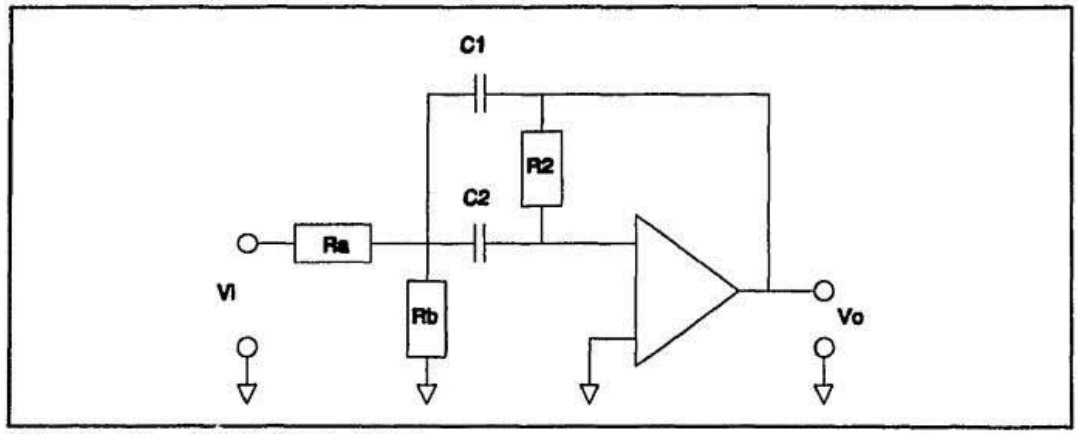


Figure 19 Band-Pass Filter

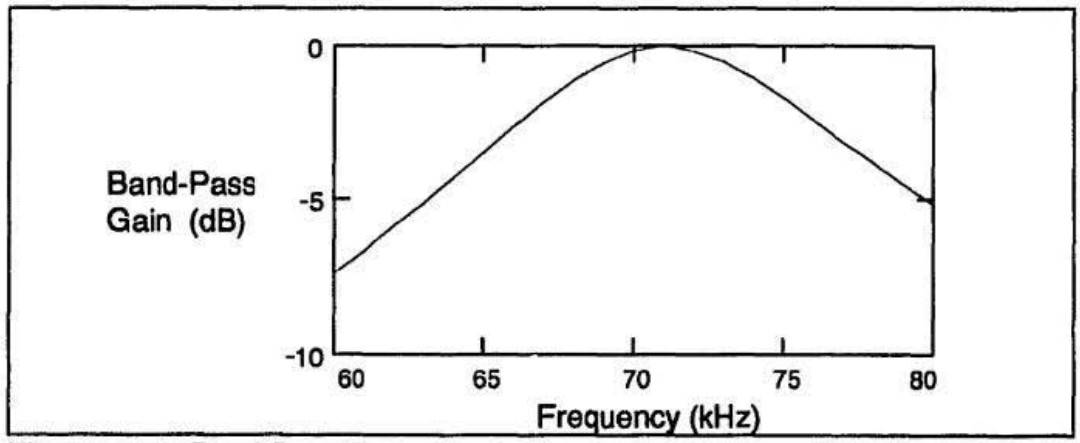


Figure 20 Band-Pass Response

5.2.2 Multiplexer

Outputs from each of the four hydrophones are selected one at a time by the multiplexer. As the hydrophones are not sampled simultaneously, there is an inter-channel time error. This error is the time difference between sampling the first and fourth hydrophone channels, and propagates into a distance

measurement error. Inter-channel error, IC_e, is calculated by dividing the speed of sound in water (1500 m/s) by the time interval between the first and fourth channels:

$$IC_e = \frac{1{,}500\ m/s}{3 \cdot 41{,}000\ samples/s} = 0.01\ m/sample \tag{74}$$

This error should be less than half the distance resolution of the system. For this application, the inter-channel error is much less that the distance resolution, one meter. The multiplexer's input should be switched to the output during the hold phase of the sample and hold circuit.

An ADG526A 16-channel analog multiplexer was used simply because the chip was available in-house. This chip runs off a dual supply from ±11 to ±16 volts and typically draws 20 mA of current. Switches are guaranteed break-before-make so that the input signals are protected from shorting. The latch capability was not used as the Write line is tied low and the Enable line is tied high. In this way, a change in the address lines results in a change in the outputs within 200 ns.

5.2.3 Sample and Hold

A sample and hold circuit is used to freeze the input signal during the conversion cycle of the ADC. An important part of the sample-hold operation is

associated with the sample-to-hold transition. There are three time definitions associated with this transition that define the performance of the circuit: acquisition time, aperture delay, and aperture jitter. (Figure 21) Acquisition time is the maximum time needed by the sample and hold circuit to lock on and track the signal after the beginning of the sample command and can also be defined as the minimum sampling time before a hold command. This time depends on the slew rate of the amplifier and is the time it takes for the output to change from a maximum to minimum voltage after an instantaneous change at the input. Aperture delay is the elapse time between the leading edge of the hold command and when the instant hold is achieved. This specification is important when simultaneous sampling is to take place with more than one sample and hold chip. Aperture jitter is the variation in aperture delay time from sample to sample. The error from the jitter relates to an uncertainty in the conversion time and consequently sets a limit on the maximum sampling frequency. Maximum acceptable jitter, J_{max} is calculated from 1/2 LSB error divided by a peak-to-peak sinusoidal input at the maximum frequency of interest, f, evaluated at zero degree phase:

$$J_{max} = \frac{2^{-(n+1)}}{1/2 \sin 2\pi ft\big|_{t=0}} = \frac{2^{-(n+1)}}{\pi f} \tag{75}$$

It should be noted that when using undersampling, the maximum acceptable jitter refers to the actual maximum frequency and not the sampling frequency. For

this application with 12-bit ADC and a maximum frequency of 76 kHz, aperture jitter should be less than 0.5 ns.

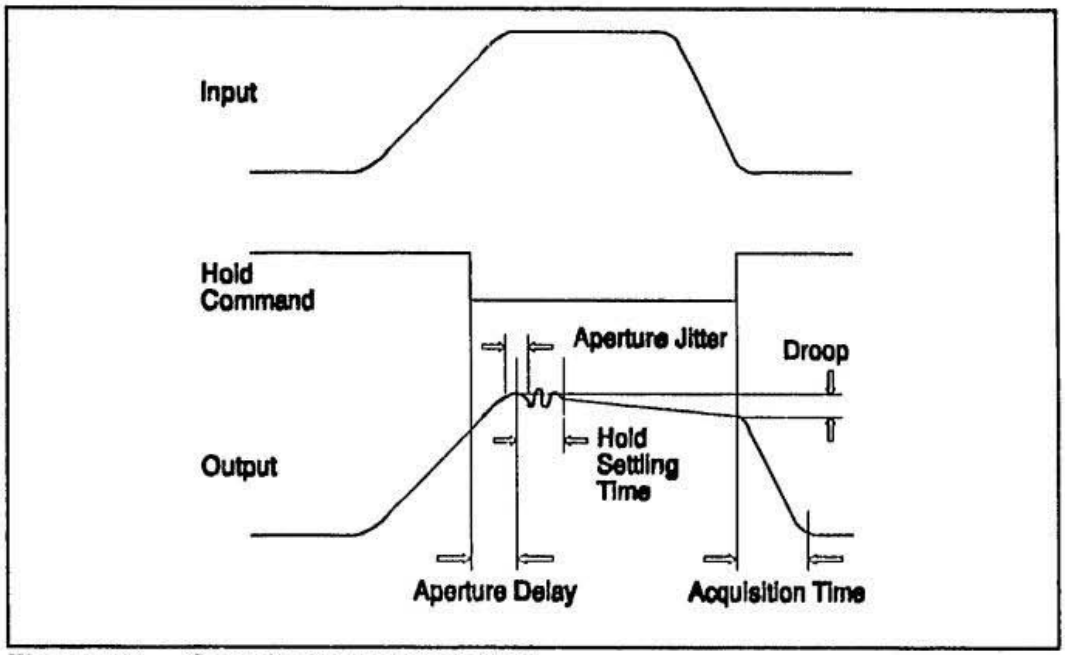


Figure 21 Sample-to-Hold Transition

During the hold cycle leakage from the hold capacitor, known as droop rate, *DR*, will affect the performance of the ADC. Total droop, *TD*, which is calculated by the droop rate times the ADC conversion time, has to be less that 1/2 LSB.

$$TD = DR \cdot T_{conv} < V_{VS} \cdot 2^{-(N+1)} \qquad (76)$$

An HAS-5330-5 Sample and Hold chip is used. This chip has a slew rate of 90 V/μs, operates off a dual power supply over a range of ±10 to ±20 volts and has its own internal hold capacitor. Acquisition time to within 0.01 % of signal is 650 ns. With a droop rate of 0.01 μV/μs, the total droop during the conversion cycle is 0.05 μV, well under the 1/2 LSB error of the ADC, 1 mV. Aperture jitter is specified at 0.1 ns, and as a result is not a concern. Output resistance in the hold mode is 0.2 ohms and output noise is 190 μV_{RMS}, lower than the quantization error of the ADC. Power requirement for the chip is 40 mA.

5.2.4 The Analog to Digital Converter

A 12-bit ADC was specified in section 4.3 on the discussion of the hydrophone *SNR*. Four hydrophone channels are digitized at 41 kHz each, resulting in an aggregate sample rate of 164 kHz. An ADC with a conversion time better than 6 μs is needed.

The 7572 High Speed 12-Bit ADC with an conversion time of 5 μs was available in-house. ADC is clocked using a 2.4576 MHz crystal. This chip includes a five volt on-chip reference. Conversion starts on a Read operation and the Busy line remains low during conversion. While the Read line is low, the

previous sampled data is available on the output Data lines. Typical power dissipation is 135 mW.

Digital output is written into a First-In/First-Out buffer, FIFO, buffering the data between the ADC and DSP. The data is written in the FIFO buffer after each data conversion cycle of the ADC, then asynchronously the DSP transfers a block of data into its internal memory. It this way, the DSP is not continuously interrupted after each conversion. The FIFO should be long enough to hold 4 blocks of 64 samples or 256 data words. A 14-bit wide data word is required, which includes the 12-bit digitized word and 2 bits of address tag. The device should perform asynchronous and simultaneous read and write operations. Cycle times should be quick enough to write on one DSP instruction cycle.

An AM-7202-A FIFO with a 25 ns access time was available. This FIFO is 9 bits wide by 1024 deep. Two in parallel are required for the 14-bit data word. Overflow, an error, is indicated by a Full flag which is connected to the Interrupt Request Line of the DSP. Half full flag is used to control data flow to the DSP. In this way handshaking is not needed; when the FIFO is half full the DSP will empty four blocks of 64 samples of data. Data is toggled in and out of the device using the Read and Write lines. The maximum power dissipation is 1 W and the chips operate from a +5 volt supply.

5.2.5 The Data Acquisition Controller

The data acquisition controller regulates all analog operations from multiplexing, to digitizing and then filling of the ADC FIFO. Timing of the controller circuit is from the ADC_CLOCK output, running at 2.46 MHz. First, the ADC_CLOCK signal is feed into a 74193, Binary Up/Down Counter. This counter is set up to divide the ADC_CLOCK by 15. Output Q4 from the counter is then a 164 kHz clock. This clock is asymmetric with a high/low ratio of 7/8. READ input on the ADC and the WRITE input from the ADC FIFO are clocked from this 164_CLOCK. A 74193, 4-Bit Up/Down Binary Counter, is clocked from the 164_CLOCK, counts to 4. Its outputs, Q0 and Q1, are the address control lines for the multiplexer and the address tags for the ADC FIFO. The ADC BUSY output is connected to the HOLD of the Sample and Hold circuit and controls the sampling of the selected hydrophone channel. (Figure 22)

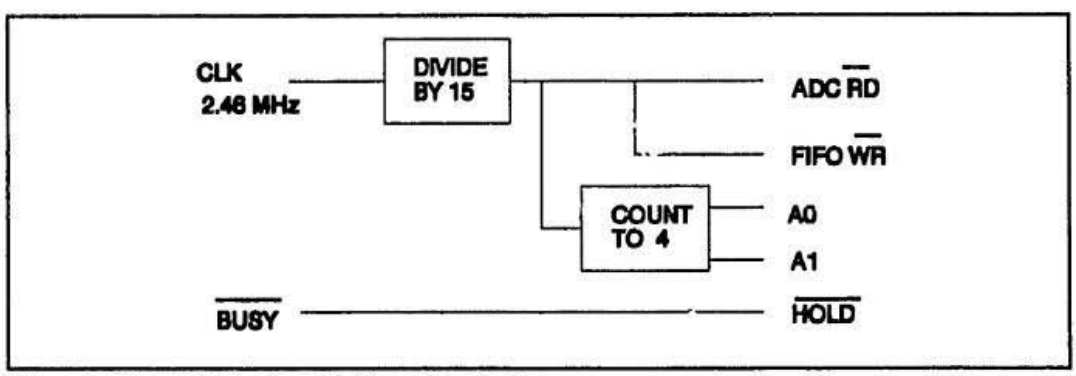


Figure 22 Data Acquisition Controller

Timing considerations for the data acquisition controller are now discussed in more detail. Data conversion is initiated by a falling edge of the 164_CLOCK on the READ input of the ADC. Within 90 ns, the data from the last conversion is valid at the ADC output pins and remains valid until 20 ns after the rising edge of the 164_CLOCK. As the data has been valid for 18 ns before a rising edge of the 164_CLOCK, the data will be transferred to the FIFO. Outputs from the count-to-four chip change on the rising edge of the 164_CLOCK, ensuring that the multiplexer switches hydrophone channels during the hold cycle of the Sample/Hold circuit. These address lines are used as tags for the ADC output but are out of sync by one and will be taken care of in software. The BUSY output from the ADC is used as the HOLD input for the Sample/Hold circuit. The BUSY pulse is low for 13 of 15 clock cycles and as a result the S/H will be in the sample mode for at least 800 ns. (Figure 23)

5.2.6 The Digital Signal Processor

Digital signal processors (DSPs) are programmable single-chip processors optimized for digital signal processing. These processors use a modified Harvard architecture with separate buses for data and instructions. They also incorporate computational units, data address generators and a program sequencer in one device. DSPs are used to perform the following operations:

digital filtering, adaptive filtering, fast Fourier transformations, imaging processing, and linear predictive speech coding.

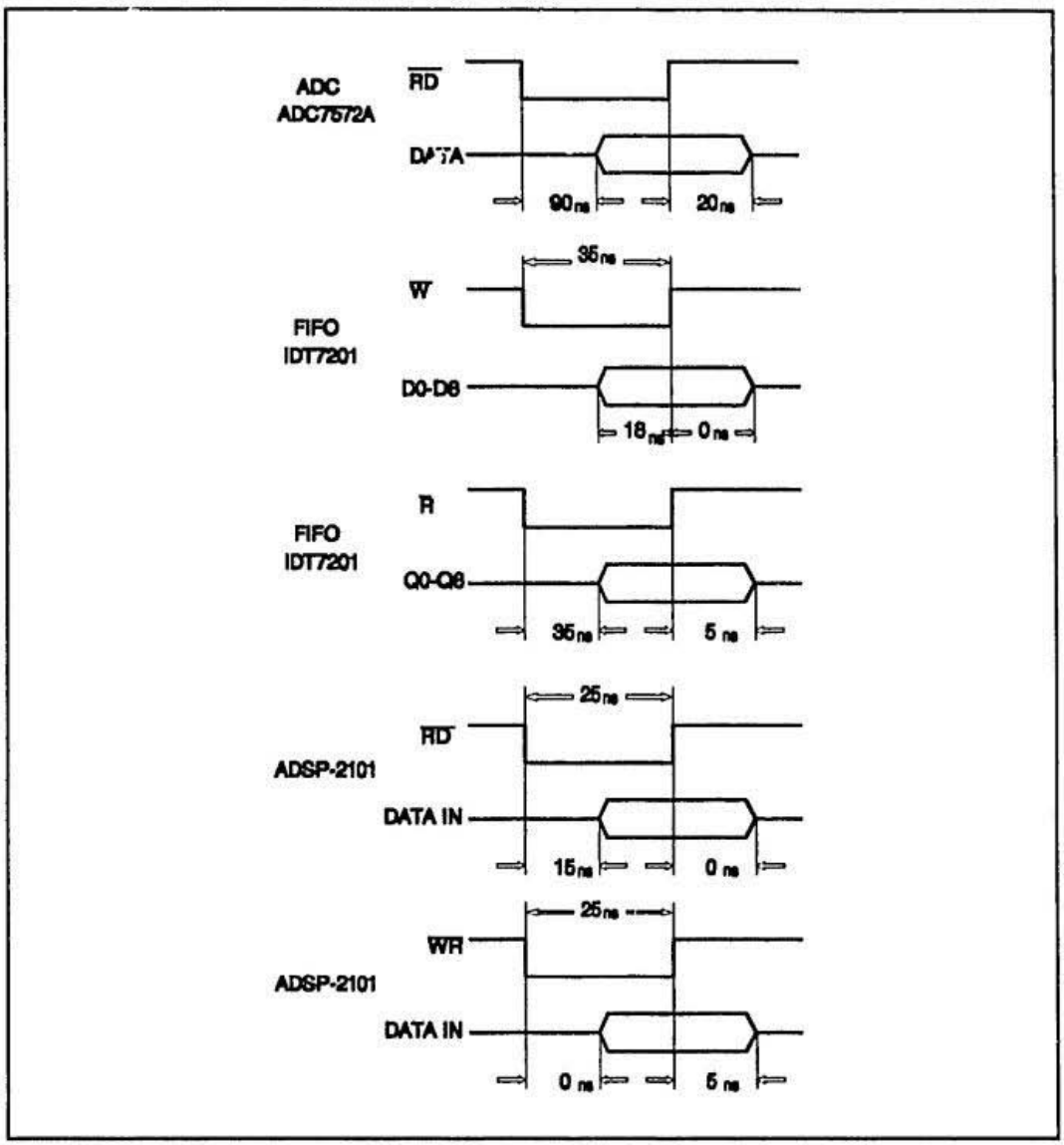


Figure 23 Analog Controller Timing

An Analog Devices ADSP-2101 Microcomputer DSP is used in this design. This chip is a single-chip microprocessor with 1 K words of 16 bit data memory RAM and 2 K words of 24 bit program memory RAM on chip. The internal bus structure is extended off-chip via a single external memory address bus and data bus which supports up to 15 K words of data memory and 16 K words of program memory. It contains three full function, independent computation units: an arithmetic/logic unit, a multiplier/accumulator and a barrel shifter. Also provided on chip are two data address generators and a program sequencer, two serial ports and a timer. The chip uses a 16.66 MHz external crystal and operates with a 60 ns instruction cycle time. The DSP chip operates off a single supply of +5 volts and draws 135 mA of current.

Data is transferred into the DSP from the ADC FIFO over the external data bus. Handshaking capabilities are not available on this bus and, as a result, blocks of data from the ADC FIFO will be transferred only after the FIFO is half full. The full flag from the ADC FIFO will initiate an interrupt routine that informs the PC of an error and resets the system. Detected pulse information will be transferred to a similar DSP FIFO. The AM-7202-A FIFO is used to buffer data between the DSP and PC.

A decoder is used to control reading from the ADC FIFO or writing to the DSP FIFO over the external data bus. These two FIFOs are mapped to the

MSB of Data Address Space and a 7432 Quad And Chip is used to select either chip based on a read or write pulse. (Figure 24)

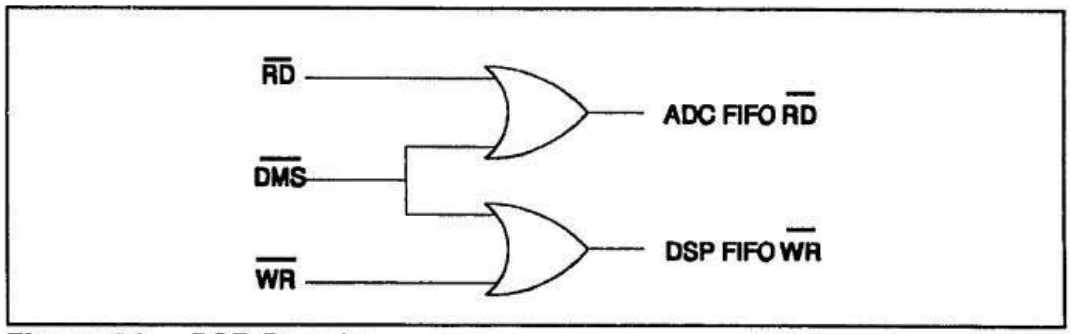


Figure 24 DSP Decoder

5.3 System Level Software

The system level software is introduced at this time. (Figure 25) Digital data sampled at 41 kHz by the 12-bit data acquisition system is stored in the DSP internal memory in blocks of 64 samples, for each hydrophone. Each block of data is transformed into the frequency domain using an FFT. Frequency domain data is then converted from real and imaginary to magnitude response. A pinger frequency will be considered present when the magnitude response at

each transmitter frequency exceeds a threshold, T_P:

$$|X(f)| \geq T_P \tag{77}$$

After all the hydrophones are processed the procedure begins again. The DSP receiver's source code listing is included in Appendix D. Each of these functional blocks will now be examined in more detail.

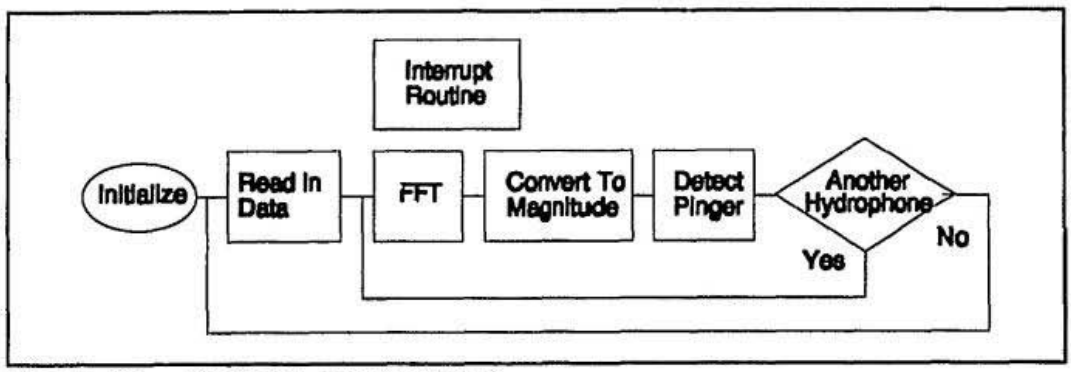


Figure 25 System Level Software

Initialize - The DSP boots up after receiving a low level on its RESET pin from the PC. All counters, variables and arrays are initialized. An ERROR register is initialized to zero. The FLAG_OUT pin is used to reset the ADC controller and the ADC FIFOs. An interrupt routine is created so that when the IRQ1 line is triggered by the ADC FIFO FULL_FLAG, the ERROR register is updated.

Read In Data - This is the re-entry point of the software and as a result all appropriate counters and registers are initialized. The ERROR register is

checked and if an error has occurred, the FLAG_OUT pin is used to reset the acquisition system. Then the ERROR register is set back to zero. Status information is sent to the PC, confirming that the system has been reset and defining the error. The FLAG_IN pin is tied to the HALF_FULL output of the ADC FIFO buffer. Data from the ADC FIFO can be read into the DSP with zero wait states. The data is read in one word at a time. As each data point is read in, it is multiplied by the Hanning window coefficients. Address tags are checked and then the data is demultiplexed and stored in separate blocks for each hydrophone. If the address tags are out of sync, the ERROR register is set. After four blocks of 64 points have been read in, an FFT is performed. (Figure 26)

FFT - A full explanation of the FFT routine is contained in Analog Devices (1992) and is not included here. The FFT's will be performed one block at a time using a Radix-4 routine that does the computation in place.

Magnitude - Transformed data is then converted from real and imaginary part into magnitude response. Phase is not used and is therefore not calculated.

Detector - A counter is used by each hydrophone/frequency pair to inhibit detection for half a second after a pinger has been detected. This reduces the

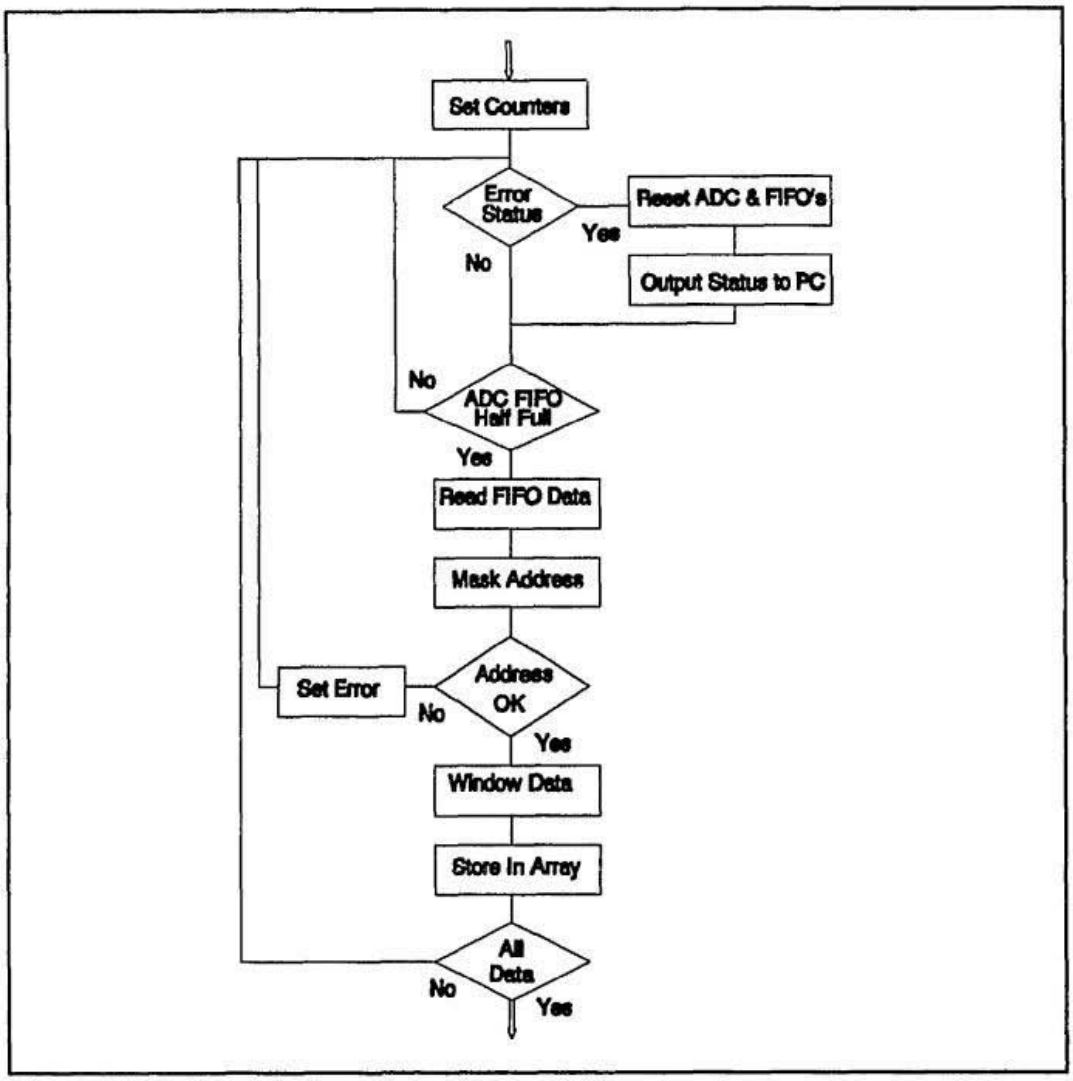


Figure 26 Read Data

number of erroneous detections caused by multipath. Initially, the counters are set to zero. Upon detection of a pinger, the counter is set to one. For each additional pass through the data, if the counter is not zero, then that frequency is not checked and the counter is incremented by one. When the counter reaches 20,000 it is reset to zero and that frequency is again checked for pinger

frequency. Each of the three frequency magnitude responses will be checked against the threshold if the counter is set to zero for that hydrophone/frequency pair. If it exceeds the threshold, then the time, hydrophone number and frequency will be output to the PC and the hydrophone/frequency counter will be set to one. In this way, only the onset of the pinger pulse will be detected. The software will be inhibited from detecting false pinger pulses for 0.5 seconds. For each hydrophone, noise measured at 73 kHz is used to set the threshold level. Eight times the running average of the noise level is used as the threshold level for detecting a pinger pulse. This will continue until all four hydrophones have been processed, at which point the whole process repeats again. (Figure 27)

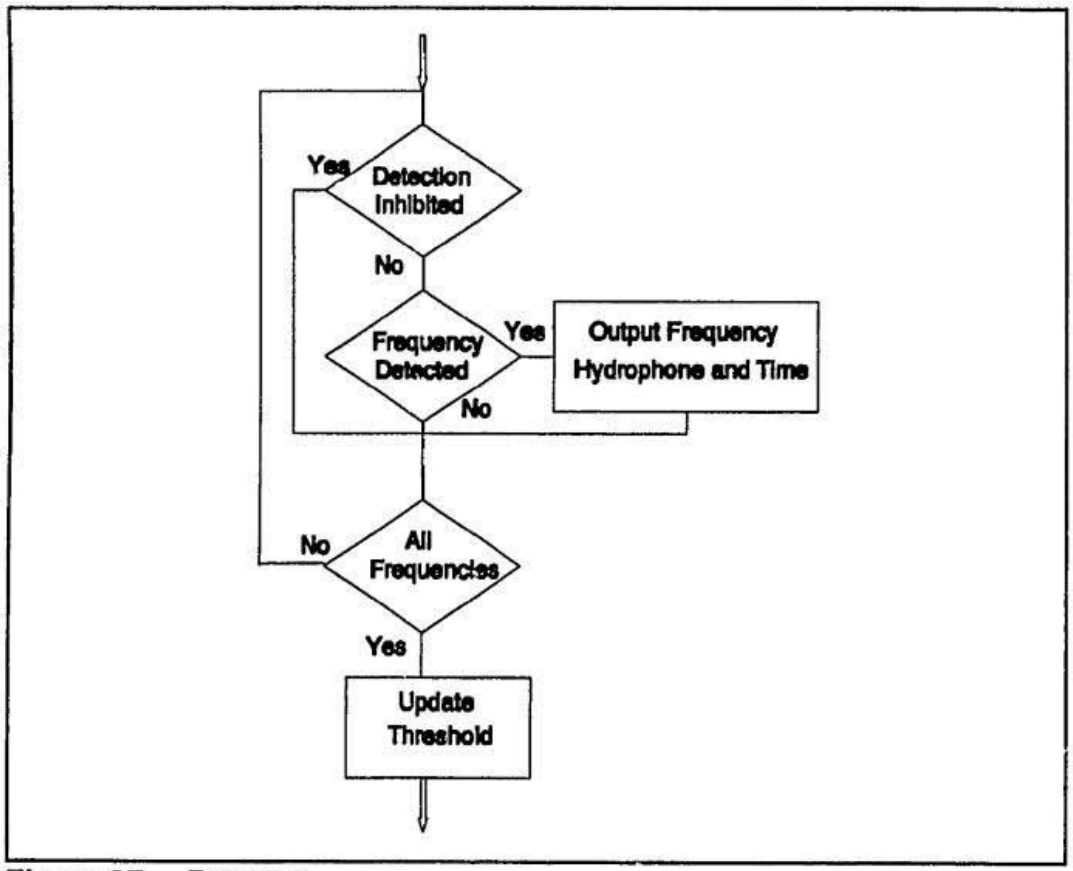


Figure 27 Detector

6 SIMULATION

6.1 MathCad Results

Simulation of the masking problem was modelled using Mathcad 4.0, from MathSoft, Inc. Using the two closest pinger frequencies, 65.5 kHz and 69.0 kHz, the effect of noise on the masking of the response of 69.0 kHz by the adjacent 65.5 kHz was investigated. Noise levels from 1 mV to 200 mV were run for 500 blocks each. For each block, both frequencies were defined with an amplitude of 5 volts and random phase. Three different signals were defined, one with 65.5 kHz and noise, another with both frequencies and noise, and the last with only noise. Signals were digitized at a sampling rate of 41 kHz and stored in blocks of 64 samples. All three signals were multiplied by a Hanning window and digitized at 12 bit resolution. Fast Fourier Transforms were performed on the signals. Real and imaginary response was then converted into a magnitude response. For a noise level of 10 mV, Figure 28 shows the magnitude response of 65.5 kHz, both frequencies and, noise. Appendix E contains an example of one run. For each of the noise levels, the mean difference betweer the response of 69.0 kHz and the masked response of the 65.5 kHz at 69.0 kHz is calculated from 500 blocks. The standard deviation is also computed. Table VII shows the mean difference and standard deviation for different noise levels.

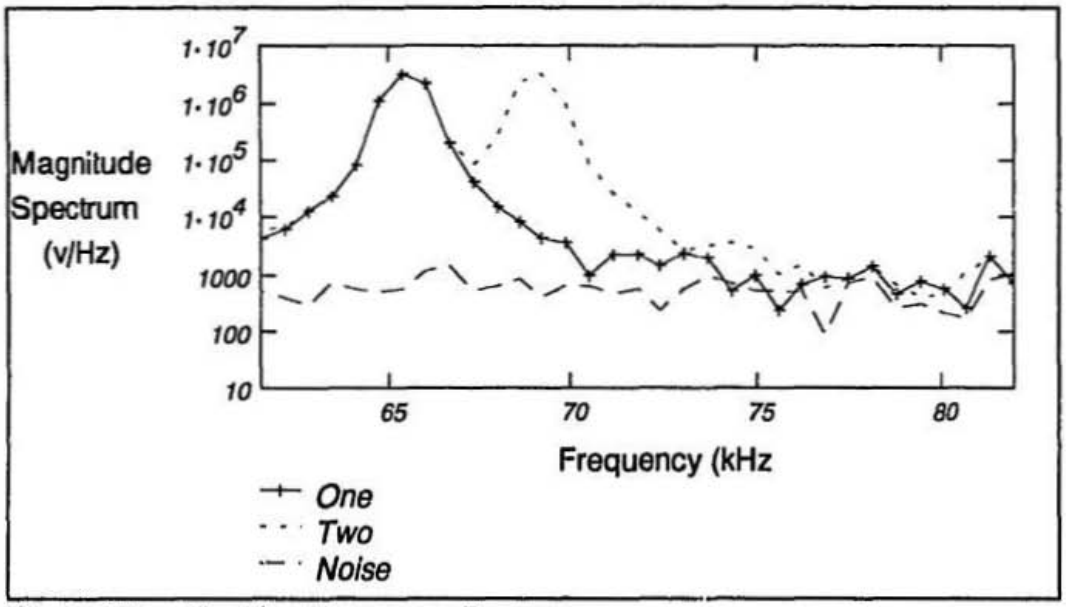


Figure 28 Masking Frequency Response

Table VII MathCad Simulation Results

RMS Noise (mV)	1	10	25	50	100	200
Mean Difference (dB)	57.1	57.0	56.9	56.7	54.7	50.2
Standard Deviation (dB)	0.1	0.8	2.1	4.0	5.2	5.3

The detection level threshold was also simulated with MathCad. The adaptive detection level threshold is computed from the following formula:

$$DT = m \cdot \left(\frac{|X(f)_n|}{2} + \frac{|X(f)_{n-1}|}{4} + \frac{|X(f)_{n-2}|}{8} + \frac{|X(f)_{n-3}|}{16} + \cdots \right) \quad (78)$$

Where is DT is the detection threshold level, m is a multiplication factor, and $|X(f)_n|$ is the magnitude level at 73 kHz for the n^{th} block. This is a modified form of a running average, where the present magnitude response is added to the

previous average and then the sum is divided by two. The 73 kHz frequency is far enough from the pinger frequencies that it measures the instantaneous background noise. Selection of the right value for the multiplication factor is a balancing act. A multiplication factor set too high will only detect strong pinger frequencies, while one set too low will falsely detect pinger frequencies in the background noise.

For a range of multiplication factors, a program similar to the one found in Appendix E was used to investigate the effect of different noise levels on false detection of pinger pulses. Table VIII shows numbers of false detects resulting for different multiplication factors versus noise levels, for 1000 blocks. From these results a multiplication factor of 8 was chosen.

Table VIII Multiplication Factor

Noise Level (mV)	Multiplication Factor							
	1	2	3	4	5	6	7	8
5	460	99	20	3	2	1	0	0
10	461	107	15	5	1	0	0	0
50	466	111	28	6	3	1	1	0
100	466	112	20	2	1	0	0	0

The effect of detecting weak signals in noise was also investigated. The detection threshold was calculated using a multiplication factor of 8. Over a

range of signals and noise levels, 1000 blocks were run. For each block, if the magnitude response at the pinger's frequency was greater then the detection threshold then the frequency was deemed to be detected. Table IX shows results from these simulations. If the signal level is greater then the noise then the signal is always detected. When the signal and noise levels are equal, the signal is detected most of the time. For this system the quantization level is 0.6 mV, which explains why at 1 mV signal level and 1 mV noise, only a small number of pinger frequencies were detected.

Table IX Detected Signal in Noise

Signal Amplitude (mV)	Noise Level (mV)					
	1	10	25	50	100	200
1	154	0	0	0	0	0
10	1000	942	136	2	0	0
25	1000	1000	957	221	14	0
50	1000	1000	1000	940	238	9
100	1000	1000	1000	1000	941	287
200	1000	1000	1000	1000	1000	931

6.2 DSP Simulator

Included with the development software for the ADSP-2100 family is a processor simulator. This simulator runs on the PC and provides interactive instruction-level simulation and debugging. Code developed for the simulator can be compiled directly for the target chip. The software described in Section 5.3 was coded (listings are in Appendix D). Simulation runs of the code were performed to ensure timing of the complete program. It requires 0.6 ms to process 64 samples from four hyrdophones and detect up to three frequencies from each of the hydrophones. The program code fits within the 2K bytes of program memory available and the data arrays fit in the 1K bytes of data memory. Digitized time series data from MathCad were used in the simulations. MathCad's FFT routines are in floating point math while the DSP chip uses fixed point math. Figure 29 shows a comparison of the magnitude response from MathCad and the DSP simulator while Figure 30 shows the difference between the two results at the bit level. This difference equates to about 0.3 mV which is well below the noise level. Simulations using the DSP processor simulator ran 2000 times slower than similar runs in MathCad. For this reason only enough DSP runs were perform to give confidence in the MathCad simulations.

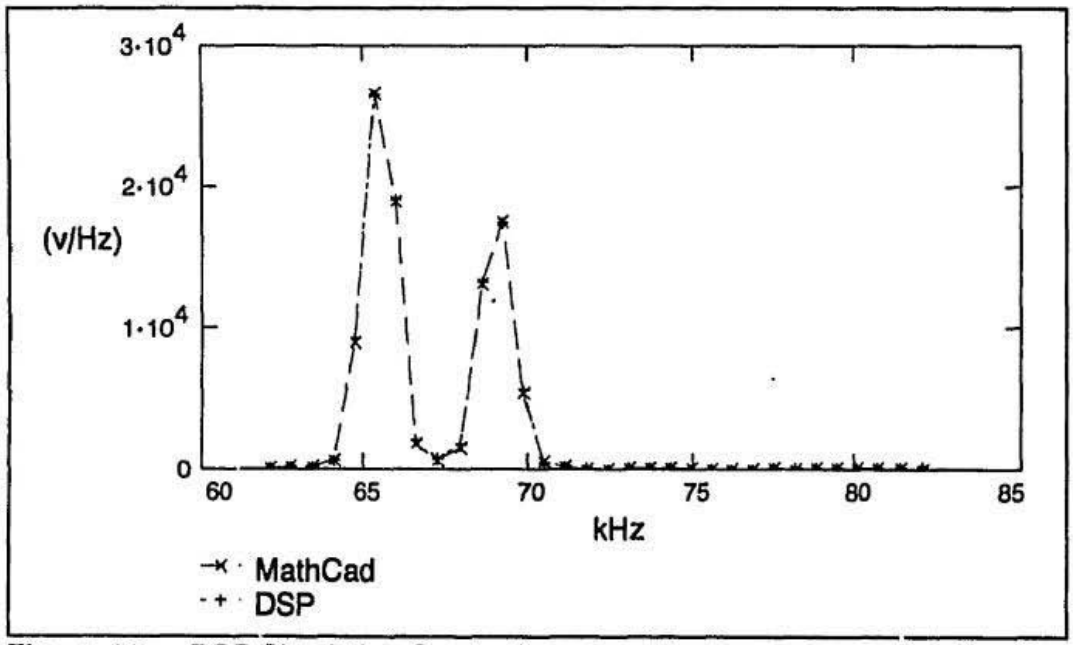


Figure 29 DSP Simulation Comparison

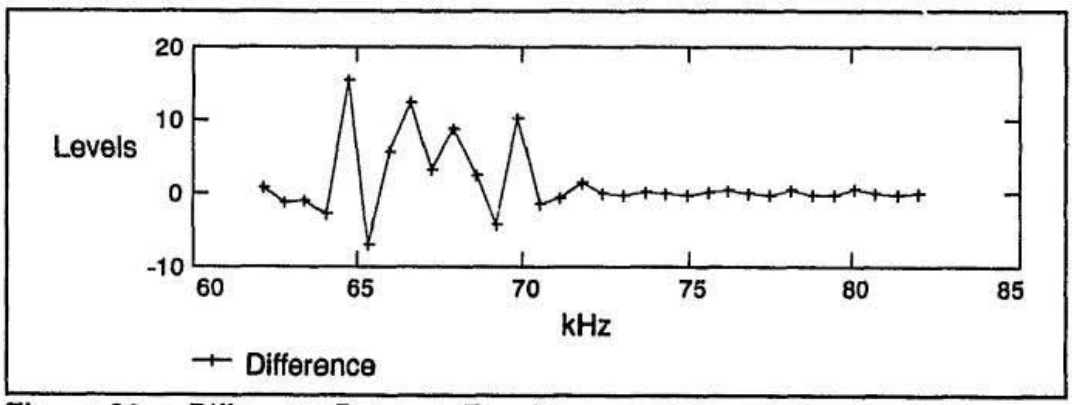


Figure 30 Difference Between Results

7 CONCLUSIONS

An investigation of a digital signal processing technique for fish tracking shows that a digital receiver using a frequency domain technique can be built. This digital receiver will fit on a single PC proto-board and connect to four existing hydrophones. It is possible to track three pinger frequencies simultaneously from each of the four hydrophones.

Analog signals from the hydrophones are bandpass filtered with a centre frequency of 71 kHz and bandwidth of 11.3 kHz. In order to reduce the data rate, reduce data storage and bring the signals down closer to baseband an undersampled technique is used. Aperture jitter was a concern as the sample and hold chip had to be specified for the actual Nyquist frequency of 153 kHz and not for the undersampled Nyquist frequency of 82 kHz. Undersampling equations help set the minimum sampling rate to 41 kHz for each channel. The signal to noise ratio at the hydrophone sets the bit resolution of the ADC to 12 bits.

Problems caused by frequency smear on adjacent frequencies were investigated. A minimum block size of 64 samples, and the use of a Hanning window keeps the smeared magnitude down 50 dB at adjacent pinger

frequencies. Although not significantly faster, a radix-4 FFT routine was chosen over the smaller size radix-2 FFT.

Mathcad was used throughout the investigation and design phases. The software was coded for the DSP processor simulator and with minimal modifications the code can be ported to the target DSP chip. The processor simulator confirms that the process requires only 0.6 ms to run and fits in internal memory.

From this design a digital receiver can now be built. A total of seventeen chips are required, most are available in-house. The remain chips can be purchased, in stock, from suppliers for under a hundred dollars. The complete circuit has been designed with the interconnections shown in Figure 18. Due to the high Q of the bandpass filter the resistor values will have to be hand tuned. Timing issues have been simplified as the data acquisition section runs asynchronously to the DSP chip. The DSP software has been coded for the processor simulator and with minor modifications will run on the target chip. With minimum effort, the prototype can now be built and tested.

8 REFERENCES

Ahmed, N. and Natarajan, T., (1983). *Discrete-Time Signals and Systems*, Reston-Publishing Company, Inc. Reston, Virginia, 398p.

Analog Devices, (1992). *Digital Signal Processing Applications Using the ADSP-2100 Family, Volume 1*, ed. Amy Mar, Prentice Hall, Englewood Cliffs, NJ, 591p.

Anderson, L.A. (1979). "Using Error Propagation Methods to Study Various Tracking Algorithms", *Proceedings, Oceans 79*, pp. 519-524.

Bellingham, J.G., Consi, T.R., Tedrow, U. and Di Massa, D. (1992). "Hyperbolic Acoustic Navigation for Underwater Vehicles: Implementation and Demonstration", *Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology*, Washington, DC, pp. 304-309.

Brigham, E. O., (1974). *The Fast Fourier Transform*, Prentice-Hall, Inc., New Jersey, 252p.

Evely, B. R. (1988). "Sonar Receiver for the IBM Personal Computer, Final Report", *Internal Report*, Memorial University of Newfoundland, St. John's, Newfoundland, 36 p.

Ferrel, D. W., Nelson, D. R., Standora, E.A, and Carter, H.C. (1974). "A Multichannel Ultrasonic Biotelemetry System for Monitoring Marine Animal Behavior at Sea", *ISA Transactions*, Vol 13, no. 2, pp. 120-131.

Franco, S., (1988). *Design with Operational Amplifiers and Analog Integrated Circuits*, McGraw-Hill Company, New York,636 p.

Garrett, P. H. (1981). *Analog I/O Design, Acquisition: Conversion: Recovery*, Reston Publishing Company, Inc., Reston, Virginia, 265 p.

Hawkings, A.D., Maclennan, D.N. and Urguhart, G.G.(1973). "Tracking Cod in a Scottish Sea Loch", *Underwater Telemetry Newsletter, Tracking Aquatic Animals*, Vol. 3 (1), June, pp. 1-6.

Hawkings, A.D., Maclennan, D.N., Urguhart, G.G., and Robb, C. (1974). "Tracking Cod Gadus morhua L. in a Scottish Sea Loch", *J. Fish Biol.*, Vol. 6, pp. 225-236.

Hawkins, A.D., Urquhart, G.G., and Smith, G.W. (1979). "Ultrasonic Tracking of Juvenile Cod by Means of a Large Spaced Hydrophone Array", *A Handbook of Biotelemetry and Radio Tracking*, eds. C.J. Amlaner and D.W. MacDonald,, Porgamon, pp. 461-470.

Hawkins, A.D., and Urquhart, G.G. (1983). "Tracking Fish at Sea", *Experimental Biology at Sea*, eds. A.G. MacDonald and I.G. Priede, I.G., Academic Press, pp 103-166.

Ireland, L.C., and Lawson, K.D. Jr. (1979a). "Assembly of a Small Ultrasonic Transmitter", *A Handbook on Biotelemetry and Radio Tracking : Proceedings of an International Conference on Telemetry and Radio Tracking in Biology and Medicine*, eds. C.J. Amlaner and D.W. MacDonald, Pergamon, pp. 205-208.

Ireland, L.C., and Lawson, K.D. Jr. (1979b). "Construction of an Acoustic Receiver and Hydrophone", *A Handbook on Biotelemetry and Radio Tracking : Proceedings of an International Conference on Telemetry and Radio Tracking in Biology and Medicine*, eds. C.J. Amlaner and D.W. MacDonald, Pergamon, pp. 461-470.

Johnstone, A.D.F., Glass, C.W., Mojsiewicz, W.R., and Smith, G.W. (1991). "The Movements of Saithe (Pollachius virens (L.)) Revealed by Acoustic Tracking", *Prog. Underwater Sci.* 16:, pp. 61-73.

Kinsler, L. E., Frey, A. R., Coppens, A. B., and Sanders, J. V. (1982). *Fundamentals of Acoustics*, John Wiley and Sons, New York, 480p.

Kraeutner, P. and Bird, J. (1993). "A PC Coherent Sonar Workstation for Experimental Underwater Acoustics", *Ocean 92 Proceedings*, Victoria, Canada, Vol. 3, pp. 359-364.

MacLennan, D.N., and Hawkins, A.D. (1977). "Acoustic Postion Fixing In Fisheries Research", *Rapp. P.-v Réun. Cons. Int. Explor. Mer*, Vol. 170, pp. 88-97.

Mitson, R.B., and Storeton-West, T.J. (1971). "A Transponding Acoustic Fish Tag", *The Radio and Electronic Engineer*, Vol. 41, no. 11, pp. 483-489.

O'Dor, R.K., Webber, D.M., and Voegeli, F.M. (1989). "A Multiple Buoy Acoustic-Radio Telemtetry System for Automated Positioning and Telemetry of Physical and Physiological Data", *Proceedings of the Tenth International Sysposium on Biotelemetry*, ed. C.J. Amlaner, Arkansas Press, pp. 444-452.

Oppenheim, A. V., and Schafer, R. W. (1975). *Digital Signal Processing*, Prentice-Hall, Inc., New Jersey, 585p.

Otnes, R. K., and Enochson, L., (1978). *Applied Time Series Analysis, Volume 1, Basic Techniques*, John Wiley and Sons, New York, 449p.

Pincock, D.G., and Luke, D.M. (1975). "Systems for Telemetry from Free-Swimming Fish", *Proceedings of Joint Conference on Instrumentation in Oceanograph*, Bangor, U.K., pp. 175-186.

Priede, L.G. (1992). "Wildlife telemetry: an introduction", *Wildlife Telemetry : Remote Monitroing and Tracking of Animals*, eds. I.G. Priede and S.M. Swift, Horwood, pp. 3-25.

Shrader, R. L. (1985). *Electronic Communications*, McGraw-Hill, New York, 754p.

Standora, E.A., Sciarrotta, T.C., Carter, H.C., Nelson, D.R., and Ferrel, D.W. (1972). "Development of a Multichannel, Ultrasonic Telemetry System for the Study of Shark Behavior at Sea", *Tech Report No. 5*, California State University, Long Beach Foundation, p. 69.

Stasko, A.B., and Pincock, D.G. (1977). "Review of Underwater Biotelemetry, with Emphasis on Ultrasonic Techniques", *Journal of the Fisheries Research Board of Canada*, Vol. 34, no. 9, pp. 1261-1285.

Trefethen, P.S. (1956). "Sonic Equipment for Tracking Individual Fish", *Special Scientific Reports - US Fish Wildlife Service*, No. 179, p. 11.

Tucker, D.G., and Gazey, B.K., (1966). *Applied Underwater Acoustics*, Pergamon Press, Oxford, England, 244p.

Uffenbeck, J. (1987). *The 8086/8088 Family: Design, Programming, and Interfacing*, Prentice-Hall, Inc, New Jersey, 630p.

Urick, R. J. (1983). *Principles of Underwater Sound*, McGraw-Hill, New York, 416p.

Urquhart, G.G., and Smith, G.W. (1992). "Recent developments of a fixed hydrophone array system for monitoring movements of aquatic animals", *Wildlife Telemetry : Remote Monitroing and Tracking of Animals*, eds. I.G. Priede and S.M. Swift, Horwood, pp. 342-353.

Woodward, B. (1988). "Logic Design and Range Capability of Underwater Acoustic Beacons", *Underwater Technology*, Vol 14, no. 3, pp. 23-29.

9 BIBLIOGRAPHY

Horvat, D. C. M., Bird, J. S. and Goulding, M. M. (1992). "True Time-Dealy Bandpass Beamforming", *IEEE Journal of Oceanic Engineering*, Vol. 17, no. 2, pp. 185-192.

Pincock, D. (1979). "Automation of Data Collection in Ultrasonic Biotelemetry", *A Handbook of Biotelemetry and Radio Tracking*, eds. C.J. Amlaner and D.W. MacDonald,, Pergamon, pp. 471-476.

Pincock, D.M., Luke, D.M., and Voegeli, F.A. (1979). "An automated multisensor underwater telemetry system for biotelemetry", *Ultrasonics International 79, Conference Proceedings*, Graz Austria, pp. 333-338.

Vavik, G. (1987). "UPOS, A Hydroacoustic Positioning System for High Precision and Large Dyamics", *Proceedings of the Institute of Acoustics (Great Britian)*, Vol. 9, no. 4, pp. 106-115.

Watkins, W.A., and Schevill, W.E. (1972). "Sound source location by arrival-times on a non-rigid three-dimensional hydrophone array", *Deep-Sea Research*, Vol 19, pp.691-706.

APPENDIX A

SONAR EQUATION CALCULATIONS

A - SONAR EQUATION CALCULATIONS

Specifications:

VH-65 Omnidirectional hydrophone from
VR-60 Equipment Manual V1.11 88-09-15:

Operating Frequency	50 - 80 kHz, 65 kHz optimum
Preamplifier	50 dB Low noise internal amplifier
Sensitivity	-148 dB re 1 V/uPascal
Noise Floor	27 dB re 1 uPascal/sqrt(Hz) at 65 kHz

VEMCO series V3 16 mm transmitter from
VEMCO Short Form Catalog:

Output Power	150 dB re 1 uPascal @ 1m
Operating Frequency	69 kHz

Define a distance, *d*, in meters between
the transmitter and hydrophone as:

$d := 1, 5 .. 3000$

Source Level (*SL*), is the acoustic output power of the transmitter:

$SL := 150$

Absorption Coeffiecient at 71 kHz is defined as:

$\alpha := 0.02$

Transmission Loss (*TL*), defined for cylindrical spreading as
a function of distance is:

$TL_d := 10 \cdot \log(d) + \alpha \cdot d$

Constants:

$kHz := 10^3 \cdot Hz \qquad F := 75 \cdot kHz \qquad BW := 10 \cdot kHz$

Noise Level (*NL*), above 50 kHz is defined in terms of frequency, *F*, and bandwidth, *BW*:

$$NL := -15 + 20 \cdot \log\left(\frac{F}{kHz}\right) + 10 \cdot \log\left(\frac{BW}{Hz}\right) \qquad NL = 62.501$$

Detection Level (*DL*) or signal to noise ratio (S/N) is defined as:

$$DL := SL - TL - NL$$

Hydrophone Sensitivity (*HS*) is defined from above as:

$$HS := -148$$

Noise Floor (*NF*) for the hydrophone is defined from above as:

$$NF := 27 + 10 \cdot \log\left(\frac{BW}{Hz}\right)$$

Voltage at the hydrophone terminals as a function of distance is $20*\log(T_d)=HS+SL-TL_d$:

$$Terminal_d := 10^{\left(\frac{HS + SL - TL_d}{20}\right)}$$

Convert noise to voltage at the hydrophone terminals:

$$NF_{volt} := 10^{\left(\frac{HS + NF}{20}\right)} \qquad NF_{volt} = 89.125 \cdot 10^{-6}$$

$$NL_{volt} := 10^{\left(\frac{HS + NL}{20}\right)} \qquad NL_{volt} = 53.096 \cdot 10^{-6}$$

Noise at the hydrophone's terminal is a combination of the above:

$$Noise_{volt} := \sqrt{NF_{volt}^2 + NL_{volt}^2} \qquad Noise_{volt} = 103.742 \cdot 10^{-6}$$

$$Noise_{dB} := 20 \cdot \log(Noise_{volt}) - HS \qquad Noise_{dB} = 68.319$$

Maximum Signal to Noise Ratio at the Hydrophone terminals is when the transmitter is one meter away:

$$SN_{dB} := SL - TL_1 - Noise_{dB} \qquad SN_{dB} = 81.7$$

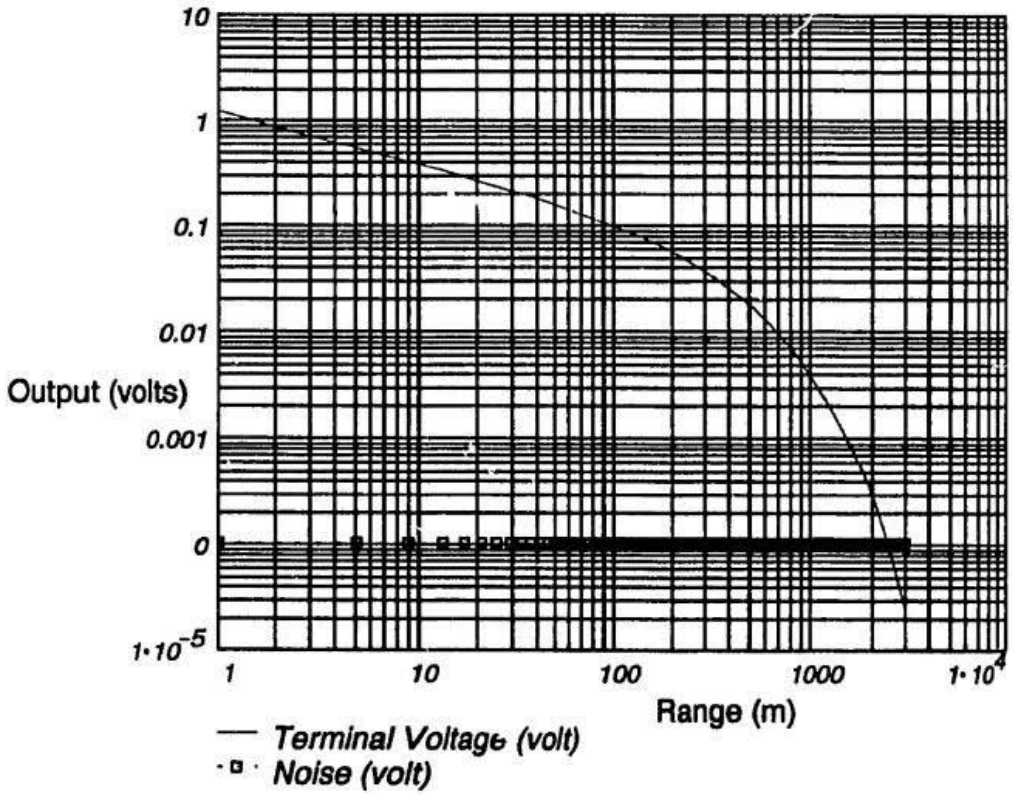


Figure A-1 Terminal Voltage and Noise vs Range

APPENDIX B

HYDROPHONE GRID HYPERBOLIC ERRORS

B - HYDROPHONE GRID HYPERBOLIC ERRORS

Resolution limits in the time delay measurements can cause errors in the pinger's position. In order to calculate these errors it is first necessary to understand the algorithm used to locate the pinger. The technique used to locating the pinger in two dimensions is taken from comments in the subroutine LOCATE.C, version 2.0. FISHTRAK.C calls this subroutine and both were written by Professor Michael Bruce-Lockhart, Faculty of Engineering, MUN.

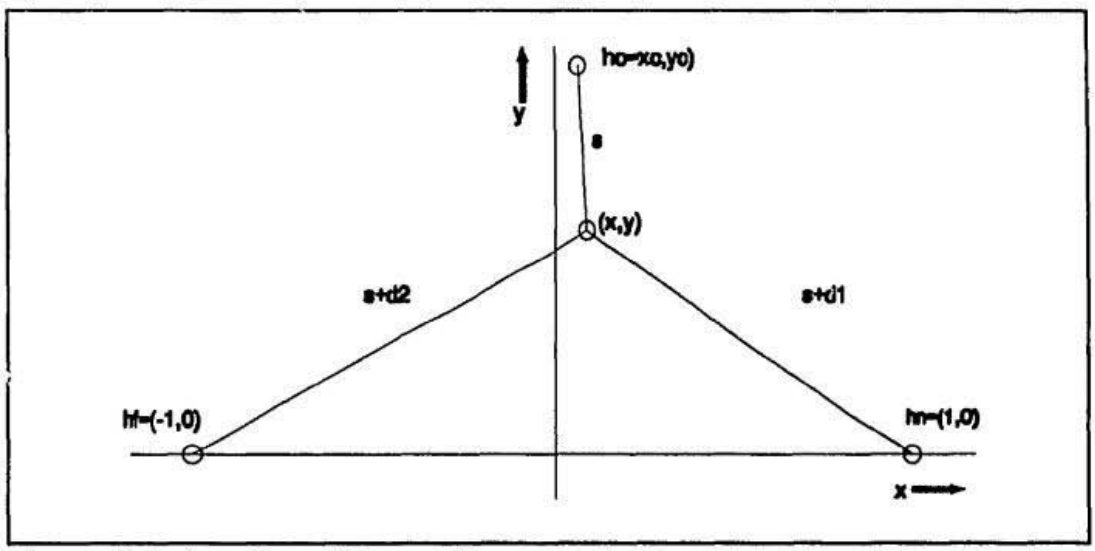


Figure B-1 Location of Transmitter

The subroutine locates the pinger using the echoes from exactly three hydrophones. The three receiving hydrophones are first flipped, rotated and normalized such that the furthest hydrophone from the transmitter (h_f) is located at (-1,0), the next closest hydrophone (h_n) is located at (1,0). (Figure B-1) The

closest hydrophone to the transmitter (h_c) is then at (x_c, y_c), such that $x_c \geq 0$. The distance, s, from h_c to the transmitter is defined as an unknown and the transmitter is located at (x,y). Distance from the transmitter to the n^{th} hydrophone is $s+d_n$.

The location of the transmitter is governed by the following equations:

$$s^2 = (x-x_c)^2 + (y-y_c)^2 \tag{B-1}$$

$$(s+d_1)^2 = (x-1)^2 + y^2 \tag{B-2}$$

$$(s+d_2)^2 = (x+1)^2 + y^2 \tag{B-3}$$

Subtracting equation C-1 from C-2 and C-3, yields two linear equations:

$$x\cdot(x_c-1) + y\cdot y_c = d_1^2\cdot s + \frac{d_1^2+\alpha}{2} \tag{B-4}$$

$$x\cdot(x_c+1) + y\cdot y_c = d_2^2\cdot s + \frac{d_2^2+\alpha}{2} \tag{B-5}$$

where:

$$\alpha = x_c^2 + y_c^2 - 1 \tag{B-6}$$

Equations B-4 and B-5 can be written in matrix form:

$$A \cdot x = B \cdot s + C \tag{B-7}$$

where:

$$A = \begin{vmatrix} x_c - 1, & y_c \\ x_c + 1, & y_c \end{vmatrix}, \quad x = \begin{vmatrix} x \\ y \end{vmatrix}, \quad B = \begin{vmatrix} d_1 \\ d_2 \end{vmatrix}, \quad C = \frac{1}{2}\begin{vmatrix} d_1^2 + \alpha \\ d_2^2 + \alpha \end{vmatrix} \tag{B-8}$$

then solving for:

$$x = A^{-1} \cdot B \cdot s + A^{-1} \cdot C = E \cdot s + F \tag{B-9}$$

$$A^{-1} = \frac{\begin{vmatrix} y_c, & -y_c \\ -(x_c + 1), & x_c - 1 \end{vmatrix}}{det\ A} = \frac{1}{2}\begin{vmatrix} -y_c, & y_c \\ (1 + x_c), & 1 - x_c \end{vmatrix} \tag{B-10}$$

where:

$$det\ A = (x_c - 1)y_c - (x_c + 1)y_c = -2y_c \tag{B-11}$$

and

$$E = \begin{vmatrix} \frac{d_2 - d_1}{1} \\ \frac{x_c(d_1 - d_2) + d_1 + d_2}{2y_c} \end{vmatrix}, \quad F = \begin{vmatrix} \frac{d_2^2 - d_1^2}{4} \\ \frac{x_c(d_1^2 - d_2^2) + d_1^2 + d_2^2 + 2\alpha}{4y_c} \end{vmatrix} \tag{B-12}$$

Substituting into Equation (B-1) yields:

$$s^2 = (E[1] \cdot s + F[1] - x_c)^2 + (E[2] \cdot s + F2 - y_c)^2 \qquad \text{(B-13)}$$

which can be written as:

$$s^2 (E[1]^2 + E[2]^2 - 1) + 2 (E[1] G[1] + e[2] G[2]) s + G[1]^2 + G[2]^2 \qquad \text{(B-14)}$$

where:

$$G[1] = F[1] - x_c \qquad G[2] = F[1] - y_c \qquad \text{(B-15)}$$

Substituting the following:

$$\begin{aligned} a &= (E[1]^2 + E[2]^2 - 1) \\ b &= 2 (E[1] G[1] + E[2] G[2]) \\ c &= G[1]^2 + G[2]^2 \end{aligned} \qquad \text{(B-16)}$$

Now equation (B-14) can be written in the quadratic form:

$$a s^2 + b s + c = 0 \qquad \text{(B-17)}$$

From any basic book on Calculus the solution for s is:

$$s = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad \text{(B-18)}$$

and

$$\begin{aligned} b^2 - 4ac > 0 &\rightarrow \textit{two real solutions for s} \\ b^2 - 4ac = 0 &\rightarrow \textit{one real solution for s} \\ b^2 - 4ac < 0 &\rightarrow \textit{no real solutions for s} \end{aligned} \qquad \text{(B-19)}$$

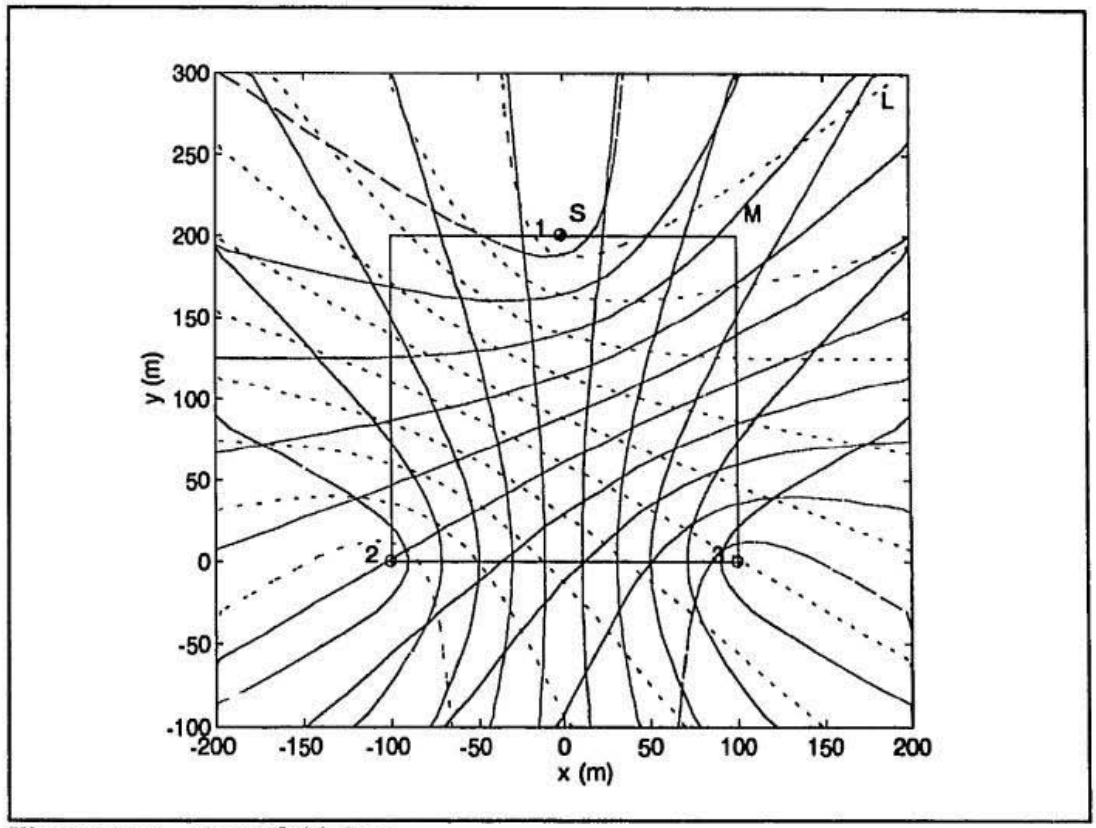


Figure B-2 Error Grid Area

Figure B-2 shows a plan view around three hydrophones and their hyperbolic lines. There are three areas of interest around these hydrophones. The largest area designated by the letter L is four hundred by four hundred meters centred on the hydrophones. A two hundred by two hundred meter area with the hydrophones on the outside edge of the square is designated by the letter M. S designates a small area two meter by two meter centred at hydrophone number 1.

MathCad is used on the following pages to calculate the maximum pinger position error based on errors in the time delay measurements. In software the pinger is first located on a 50 by 50 grid over the area of interest. At each grid point the actual time delays are calculated between the hydrophones. Next an error equal to plus or minus one meter is added to each of the time delay measurements. Using the previously explained algorithm the pinger's position is calculated. The positional error is the difference between the actual position and the calculated position. Figure B-3 shows the positional error if the two time delay measurement have a positive one meter error. Two graphs on the top of the page show the position error along a line intersecting hydrophone 2 and 3, and along a line perpendicular to this one and intersects hydrophone 1. At the bottom of the page is a contour plot of the position error over the large area. Figure B-4 shows the positional error if the one of the time delay measurements has a positive one meter error while the other has a negative one meter error. The following three contour plots, Figures B-5, B-6 and B-7, show the positional error computed from maximum of the four plus/minus time delay measurement combinations errors. Figures B-5 and B-6 show that within the area the maximum position error for a time delay measurement error of one meter is less than one meter of most of the area. Within one meter of the hydrophone Figure B-7 shows that even within the hydrophone grid the positional errors can be between two and four meters.

Pinger Position Errors

Normalized location of the hydrophones are:

h_c $\quad x_c := 0.0 \quad y_c := 2.0$

h_n $\quad x_n := 1 \quad y_n := 0$

h_f $\quad x_f := -1 \quad y_f := 0$

Define x and y limits to calculate over:

$x_{min} := -1.0 \quad x_{max} := 1.0$

$y_{min} := -0.0 \quad y_{max} := 2.0$

Delay measurement error is in meters and divided by full scale:

$delay := 0.5 \quad scale := 100$

$error\_s_{d1} := \frac{delay}{scale} \quad error\_s_{d2} := \frac{delay}{scale}$

Define grid size in points, an index for x and y, and calculate step size:

$grid := 50 \quad i := 0..grid \quad j := 0..grid$

$x_{step} := \frac{x_{max} - x_{min}}{grid} \quad y_{step} := \frac{y_{max} - y_{min}}{grid}$

Grid points are:

$x_{p_i} := x_{min} + (i \cdot x_{step}) \quad y_{p_j} := y_{min} + (j \cdot y_{step})$

Calculate distance s, from pinger to closest hydrophone over the grid:

$$s_{i,j} := \sqrt{\left(x_c - x_{p_i}\right)^2 + \left(y_c - y_{p_j}\right)^2}$$

Delay distances are calculated and errors are added:

$$s_{d1_{i,j}} := \left[\sqrt{\left(1 - x_{p_i}\right)^2 + \left(y_{p_j}\right)^2} - s_{i,j} \right] + error\_s_{d1}$$

$$s_{d2_{i,j}} := \left[\sqrt{\left(-1 - x_{p_i}\right)^2 + \left(y_{p_j}\right)^2} - s_{i,j} \right] + error\_s_{d2}$$

Now solve for s using the technique explained in the preceding section.

Calculate E:

$$E1 := \overrightarrow{\frac{s_{d2} - s_{d1}}{2}} \qquad E2 := \overrightarrow{\frac{x_c \cdot \left(s_{d1} - s_{d2}\right) + s_{d1} + s_{d2}}{2 \cdot y_c}}$$

Calculate F:

$$F1 := \overrightarrow{\frac{s_{d2}^2 - s_{d1}^2}{4}}$$

$$F2 := \overrightarrow{\frac{x_c \cdot \left(s_{d1}^2 - s_{d2}^2\right) + s_{d1}^2 + s_{d2}^2 + 2 \cdot \left(x_c^2 + y_c^2 - 1\right)}{4 \cdot y_c}}$$

Calculate G:

$$G1 := \overrightarrow{\left(F1 - x_c\right)} \qquad G2 := \overrightarrow{\left(F2 - y_c\right)}$$

Now calculate the quadric form: $as^2 + bs + c = 0$

$$a := \overrightarrow{\left(E1^2 + E2^2 - 1\right)}$$

$$b := \overrightarrow{\left(2 \cdot \left(E1 \cdot G1 + E2 \cdot G2\right)\right)}$$

$$c := \overrightarrow{\left(G1^2 + G2^2\right)}$$

Radical equals:

$$r := \overrightarrow{\left(b^2 - 4 \cdot a \cdot c\right)}$$

Find the two solutions, plus and minus:

$$s1 := \overrightarrow{\left(\frac{-b + \sqrt{r}}{2 \cdot a}\right)} \qquad s2 := \overrightarrow{\left(\frac{-b - \sqrt{r}}{2 \cdot a}\right)}$$

The distance closest to the correct one is right solution:

$$sd1 := \overrightarrow{|s1 - s|} \qquad sd2 := \overrightarrow{|s2 - s|}$$

$$s_{i,j} := if\left(sd1_{i,j} < sd2_{i,j}, s1_{i,j}, s2_{i,j}\right)$$

Calculate the measured x and y pinger position:

$$x_{i,j} := \frac{2 \cdot s_{i,j}\left(s\,d2_{i,j} - s\,d1_{i,j}\right) + \left(s\,d2_{i,j}\right)^2 - \left(s\,d1_{i,j}\right)^2}{4}$$

$$y_{i,i} := \sqrt{\left(s_{i,j} + s\,d1_{i,j}\right)^2 - \left(x_{i,j} - 1\right)^2}$$

$$y_{i,j} := if\left(y\,p_i \le 0, -y_{i,j}, y_{i,j}\right)$$

The full scale error is the difference between the actual and calculated multiplied by the scale factor:

$$error_{i,j} := \sqrt{\left(x_{i,j} - x\,p_i\right)^2 + \left(y_{j,i} - y\,p_j\right)^2} \cdot scale$$

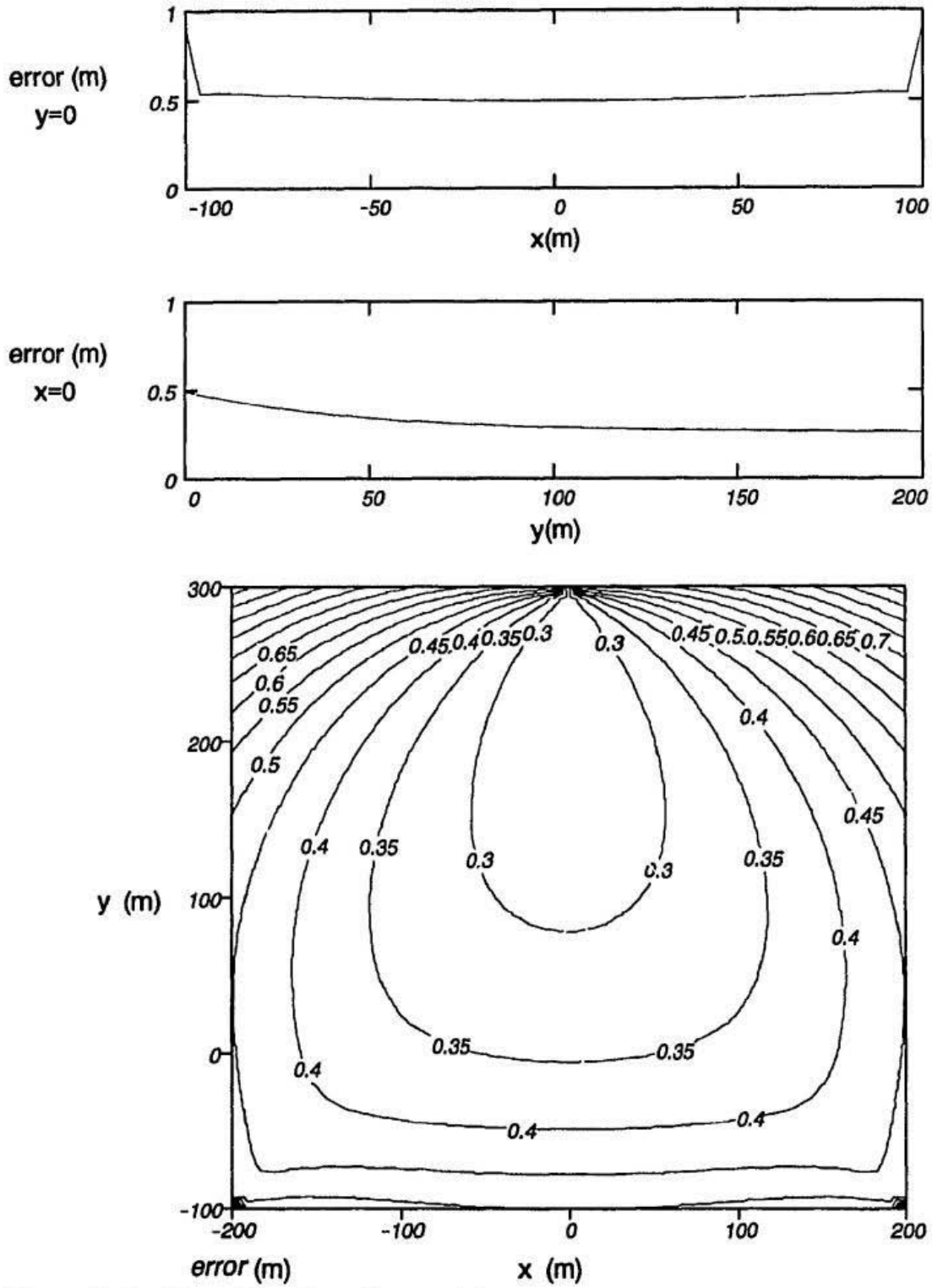


Figure B-3 Plus - Plus Error Contour Plot

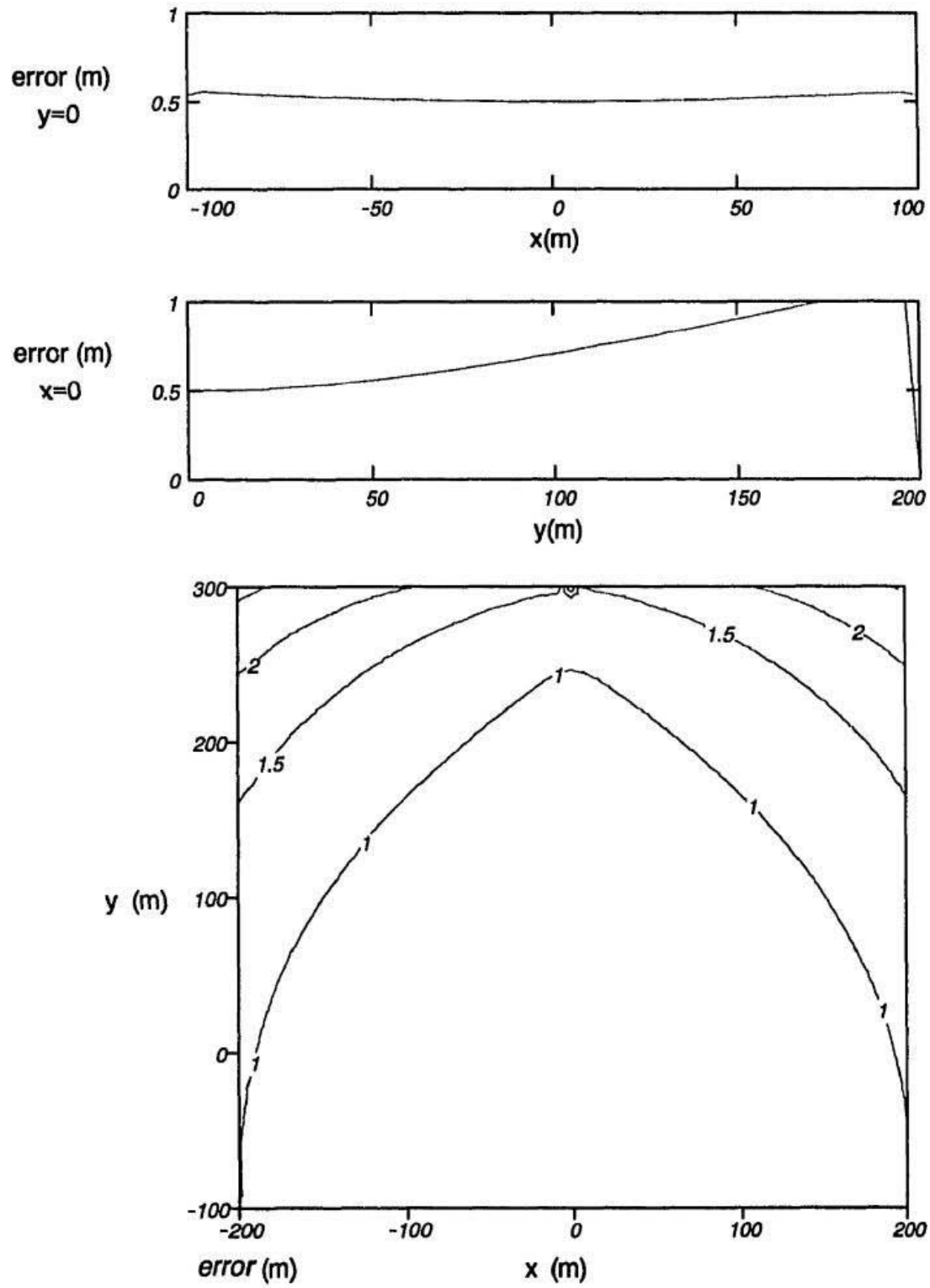


Figure B-4 Plus - Minus Error Contour Plot

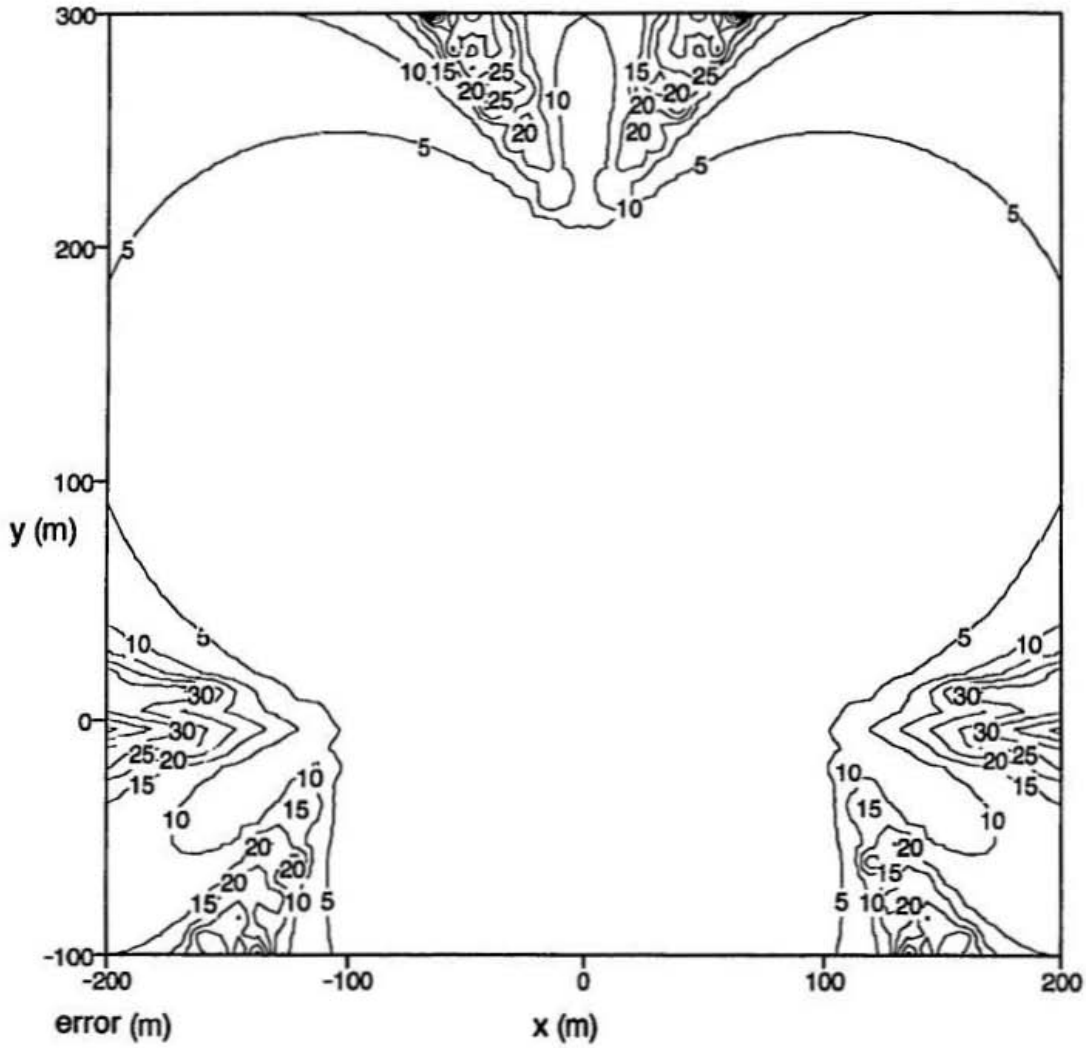


Figure B-5 Large Grid Error Plot

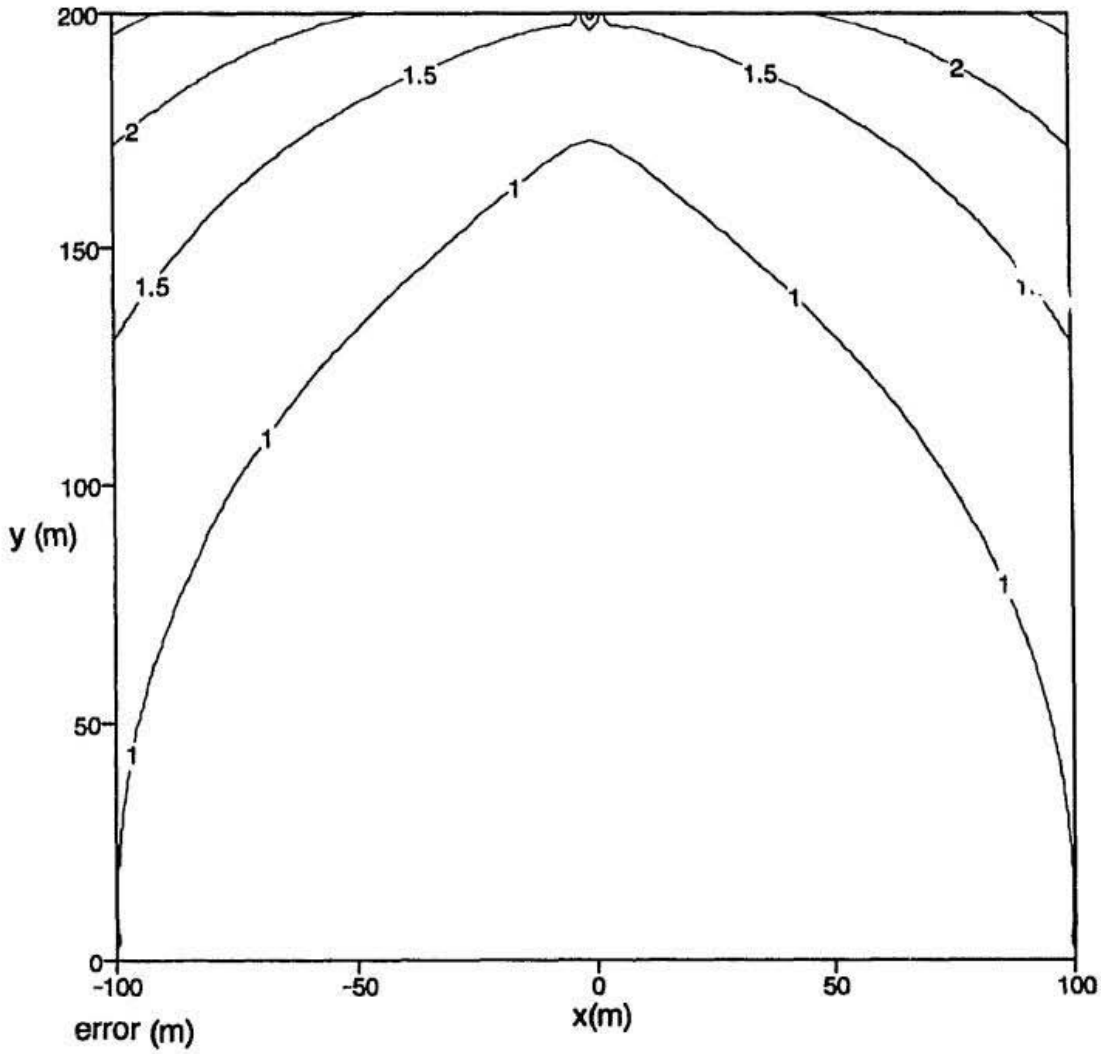


Figure B-6 Medium Grid Error Plot

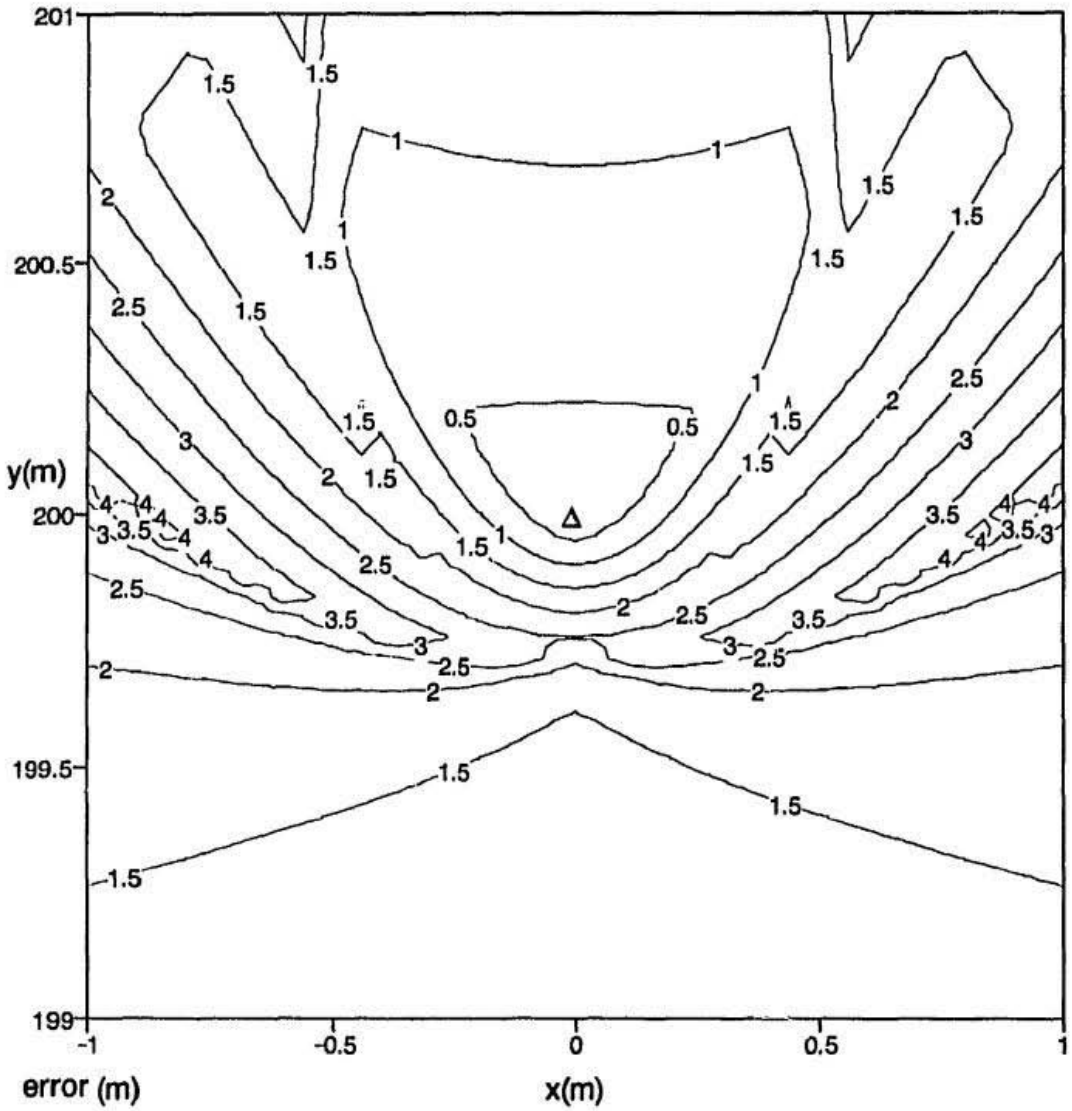


Figure B-7 Small Grid Error Contour Plot

APPENDIX C

FOLDING FREQUENCY ANALYSIS

To find the relationship between the sample frequency and the centre frequency of the k^{th} fold, first define the sample frequency to be $f_s = 1/2T$ and the centre frequency of the k^{th} fold to be:

$$\begin{aligned}
\textit{first fold} &- f_{c_1} = \frac{1}{4} f_s \rightarrow f_s = \frac{4 f_{c_1}}{1} \\
\textit{second fold} &- f_{c_2} = \frac{3}{4} f_s \rightarrow f_s = \frac{4 f_{c_2}}{3} \\
\textit{third fold} &- f_{c_3} = \frac{5}{4} f_s \rightarrow f_s = \frac{4 f_{c_3}}{5} \\
\textit{fourth fold} &- f_{c_4} = \frac{7}{4} f_s \rightarrow f_s = \frac{4 f_{c_4}}{7}
\end{aligned} \qquad \text{(C-1)}$$

Consider the k^{th} fold, as k increases the denominator increases by a factor of two. The denominator can be rewritten in terms of the k^{th} fold:

$$f_s = \frac{4 f_c}{2k - 1} \qquad \text{(C-2)}$$

The bandwidth of interest, BW, has to be less than half the sampling frequency:

$$BW \leq \frac{f_s}{2} \qquad \text{(C-3)}$$

The maximum number of usable folds is found by substituting equation (C-2) into equation (C-3):

$$k \leq \frac{f_c}{BW} + \frac{1}{2} \qquad \text{(C-4)}$$

APPENDIX D

DSP Code Listing

```
{=================================
=================================

        Program:    Main_64.dsp

        Description: This is the main calling shell for the fish tracking software.

        Benchmark:
              read      =    119.7 µs
              misc      =      2.0 µs    =    4(0.5)
              fft       =    371.2 µs    =    4(92.8)
              mag       =     56.4 µs    =    4(14.1)
              detect    =     21.2 µs    =    4(5.3)

              TOTAL     570.5 µs

 =================================
=================================}

.MODULE/RAM/BOOT=0/ABS=0   main;

.CONST      N=64;                   {number of samples to be FFT'd}
.CONST      Nx2=128;
.CONST      Nx4=256;

.VAR/DM     input[Nx4];             {holds original data}
.VAR/DM     inplace[Nx2];           {holds input and intermediate of fft}
.VAR/DM     error;                  {type of error}
.VAR/DM     hydrophone;             {hydrophone}

.VAR/DM/ABS=h#3800  output[Nx2];    {holds results in_order}

.PORT       fifo;                   {input and output port, fifo}

.GLOBAL     input, inplace, output, error, fifo, hydrophone;

.EXTERNAL     read, fft, mag, detect;
```

```
        {load interrupt vector addresses}

        JUMP start;nop;nop;nop;          {restart interrupt}
        M7=H#0001;DM(error)=M7;RTI;nop;   {fifo full, set status and return}
        RTI;nop;nop;nop;
        RTI;nop;nop;nop;
        RTI;nop;nop;nop;
        RTI;nop;nop;nop;
        RTI;nop;nop;nop;


start:  SI=0;                    {these 3 lines reset the system }
                DM(0X3FFF)=SI;          {control register and the data }
                DM(0X3FFE)=SI;          {memory control register to allow}
                                        {zero wait state memory access}

                SET FLAG_OUT;           {reset ADC and FIFO}
                CNTR=2000;
                DO RESET_F UNTIL CE;
RESET_F:              RESET FLAG_OUT;
                 SET FLAG_OUT;

        {loop forever}

FOREVER:  CALL read;             {read in data from 4 hydrophones}
                CNTR=4;
                DO HYDRO UNTIL CE;      {loop through each hydrophone}
                AY0=-1;
                AX0=CNTR;
                AR=AX0-AY0;
                DM(hydrophone)=AR;      {update hydrophone by -1}
                CALL fft;               {perform fft}
                CALL mag;               {calculate magnitude}
HYDRO:          CALL DETECT;            {detect frequencies}

                JUMP FOREVER;           {loop forever}

.ENDMOD;
```

```
{-------------------------------------------------------------------------
-------------------------------------------------------------------------

        Program:    READ.DSP

        Description:  This module reads in multiplexed data from the ADC
                      FIFO's. The data is then demultiplexed, multiplied by a
                      Hanning window coefficient and stored to memory.

        Benchmark:

         setup      =     11
         readin     =   1984    =     64(3 + 4(7))

         TOTAL      1995 * 60 ns      =     119.7 µs

-------------------------------------------------------------------------
-------------------------------------------------------------------------}


.MODULE/BOOT=0   read_sub;

.CONST      N=64;
.CONST      M=-255;

.VAR/PM     WINDOW_COEFFS[N];
.INIT       WINDOW_COEFFS:      <window.dat>;

.EXTERNAL       input, fifo, error;

.ENTRY      READ;


READ:       AR=DM(error);                   {check error status}
            IF EQ JUMP POLL;                {if error output error to PC}
                  DM(FIFO)=AR;              {and reset the adc and fifo}
                  CNTR=2000;

                  DO RESET_F UNTIL CE;
RESET_F:                RESET FLAG_OUT;

            SET FLAG_OUT;

POLL:       IF FLAG_IN JUMP READ;           {wait until half full }
```

```
                I0=^input;                      {where to place fifo input}
                M0=64;                          {demult factor}
                L0=0;
                M1=M;                           {pointer for next cycle}
                I4=^WINDOW_COEFFS;              {pointer for Hanning window}
                M4=1;
                L4=0;
                CNTR=N;

                DO READIN UNTIL CE;             {loop through 64 points}
                      MY0=PM(I4,M4);            {get Hanning coefficient}
                      CNTR=4;

                      DO DEMULT UNTIL CE;       {loop through hydrophones}
                            MX0=DM(fifo);       {read in data}
                            AX0=MX0;
                            AY0=CNTR;
                            AR=AX0-AY0;         {check address}

                            IF EQ JUMP HANN;        {address ok?}
                                  DM(error)=2;      {set error status}
                                  JUMP READ;        {jump to read}

HANN:                       MR=MX0*MY0(RND);        {mult by Hanning}

DEMULT:           DM(I0,M0)=MR1;    {store demult output}

READIN:     MODIFY(I0,M1);                  {reset input point, next cycle}

RTS;                                        {return}

.ENDMOD;
```

Program: FFT_64.DSP

Description: Optimized complex 64-Point Radix-4 DIF FFT Program for the Analog Devices ADSP-210X Digital Signal Processor Family. This routine uses a slightly modified radix_4 algorithm to effectively descramble results as they are computed. The result is in_order output. This section of code is a modified version of a 1024-point program found in Analog Devices (1992).

Complex data stored X0,Y0, X1, Y1,... (real, imag, real, imag)

The following terminology will be used to describe the "legs" of the radix-4 butterfly:

xa = 1st real input leg	x'a = 1st real output leg
xb = 2nd real input leg	x'b = 2nd real output leg
xc = 3rd real input leg	x'c = 3rd real output leg
xd = 4th real input leg	x'd = 4th real output leg
ya = 1st imag input leg	y'a = 1st imag output leg
yb = 2nd imag input leg	y'b = 2nd imag output leg
yc = 3rd imag input leg	y'c = 3rd imag output leg
yd = 4th imag input leg	y'd = 4th imag output leg

The index registers of the 210X will be used as pointers to the following:

I0 --> xa,xc
I1 --> xb,xd
I2 --> ya,yc
I3 --> yb,yd
w0 (= Ca = Sa = 0)
I5 --> w1 (1st Cb, - pi/4 for Sc)
I6 --> w2 (2nd Cc, - pi/4 for Sc)
I7 --> w3 (3rd Cd, - pi/4 for Sd)

Benchmark:

setup	=	150	
stage 1	=	500	= 20+16(30)
stage 2	=	556	= 16+4(15+4(30))
stage 3	=	340	= 20+16(20)

Total 1,546 cycles * 60ns = 92.8 µs

```
.MODULE/BOOT=0    FFT_SUB;

.const        N = 64;
.const         NX2 = 128;
.const         NOV2 = 32;
.const         NOV4 = 16;
.const         N3OV8 = 8;
.CONST        M=-64;

.VAR/DM/CIRC      COS_TABLE[64];

.VAR/DM     M3_SPACE; {memory space used to store M3 values}
                      {when M3  is loaded with its alternate value}

.VAR/DM             BFY_COUNT;
.INIT               COS_TABLE:<cos_64.dat>;       {N cosine values}

.ENTRY              FFT;

.EXTERNAL           INPLACE, INPUT, hydrophone;


FFT:  I0=^INPUT;                        {point to start of input data}
      M1=N;
      CNTR=DM(hydrophone);
      DO POST UNTIL CE;                     {update pointer to correct}
POST:       MODIFY(I0,M1);                  {hydrophone}
      M0=1;
      L0=0;
      I1=^INPLACE;                          {inplace array used by fft}
      M1=1;
```

```
        L1=0;
        CNTR=N;

        DO MOVE UNTIL CE;              {move data from input to inplace}
              AX0=DM(I0,M0);
MOVE:         DM(I1,M1)=AX0;

        M4=-Nov4;                      {-N/4 = -90 degrees for sine}
        L0=0;
        L1=0;
        L2=0;
        L3=0;
        L5=%COS_TABLE;
        L6=%COS_TABLE;
        L7=%COS_TABLE;
        SE=0;

{ ---------------------------------------- Stage 1 ----------------------------------------}

STAGE1: I0=^INPLACE;                   {in      ->Xa,Xc}
        I1=^INPLACE+Nov2;              {in+N/2  ->Xb,Xd}
        I2=^INPLACE+1;                 {in+1    ->Ya,Yc}
        I3=^INPLACE+Nov2+1;    {in+N/2+1 ->Yb,Yd}
        I5=^COS_TABLE;
        I6=^COS_TABLE;
        I7=^COS_TABLE;
        M0=N;                  {N,   skip forward to dual node}
        M1=-N;                 {-N,  skip back to primary node}
        M2=-N+2;               {-N+2, skip to next butterfly}

        M3=-2;          {because we have modified the middle branches}
                        {of bfly, pointers for I0 have to be treated a}
                        {little more carefully. This requires a more}
                        {complex pointer manipulation.  M3 is used in }
                        {doing this}

        M5=Nov4+1;              {N/4 + groups/stage*1, Cb Sb offset}
        M6=Nov4+2;              {N/4 + groups/stage*2, Cc Sc offset}
        M7=Nov4+3;      {N/4 + groups/stage*3, Cd Sc offset}
        AX0=DM(I0,M0);                  {get first Xa}
        AY0=DM(I0,M1);                  {get first Xc}
        AR=AX0-AY0, AX1=DM(I2,M0);  {Xa-Xc,get first Ya}
        SR=LSHIFT AR(LO), AY1=DM(I2,M1);   {SR1=Xa-Xc,get first Yc}
```

```
CNTR=Nov4;                              {Bfly/group, stage one}

{This radix_4 bfly algorithm is a modified one. It has it's middle 2 }
{Branches reversed. This alteration, done to every stage, results in }
{a bit_reversed output instead of a digit reversed output           }

DO STG1BFY  UNTIL CE;
    AR=AX0+AY0, AX0=DM(I1,M0);  {AR=xa+xc, AX0=xb }
    MR0=AR, AR=AX1+AY1;         {MR0=xa+xc, AR=ya+yc}
    MR1=AR, AR=AX1-AY1;         {MR1=ya+yc, AR=ya-yc}
    SR=SR OR LSHIFT AR (HI), AY0=DM(I1,M1);
                                {SR1=ya-yc, AY0=xd}
    AF=AX0+AY0, AX1=DM(I3,M0);      {AF=xb+xd, AX1=yb}
    AR=MR0+AF, AY1=DM(I3,M1);
                                {AR=xa+xb+xc+xd, AY1=yd}
    DM(I0,M0)=AR, AR=MR0-AF;    {output x'a=(xa+xb+xc+xd),
                                    AR=xa+xc-xb-xd}
    AF=AX1+AY1, MX0=AR;  {AF=yb+yd, MX0=xa+xc-xb-xd}
    AR=MR1+AF, MY0=DM(I6,M4);  {AR=ya+yc+yb+yd,
                                    MY0=(Cc)}
    DM(I2,M0)=AR, AR=MR1-AF;    {output y'a, AR=ya+yc-yb-yd}
    MR=MX0*MY0(SS), MY1=DM(I6,M6);  {MR=(xa+xc-xb-xd)(Cc),
                                    MY1=(Sc)}
    MR=MR+AR*MY1(RND),SI=DM(I0,M2);
                    {MR=(xa-xb+xc-xd)(Cc)+(ya-yb+yc-yd)(Sc)}
                    {si here is a dummy to perform a modify(I0,M2)}
    DM(I1,M0)=MR1, MR=AR*MY0(SS);
                        {output x'c to  positionx'b}
                        {MR=(ya+yc-yb-yd)(Cc)}
    MR=MR-MX0*MY1(RND), MY0=DM(I5,M4);
            {MR=(ya+yc-yb-yd)(Cc)-(xa+xc-xb-xd)(Sc), MY0=(Cb)}
    DM(I3,M0)=MR1, AR=AX0-AY0;  {output y'c=to position y'b}
                                {AR=xb-xd}
    AY0=AR, AF=AX1-AY1;         {AY0=xb-xd, AF=yb-yd}
    AR=SR0-AF, MY1=DM(I5,M5);   {AR=xa-xc-(yb-yd),
                                    MY1=(Sb)}
    MX0=AR, AR=SR0+AF;      {MX0=xa-xc-yb+yd,
                                    AR=xa-xc+yb-yd}
    SR0=AR, AR=SR1+AY0;     {SR0=xa-xc+yb-yd,
                                    AR=ya-yc+xb-xd}
    MX1=AR, AR=SR1-AY0;     {MX1=ya-yc+xb-xd,
                                    AR=ya-yc-(xb-xd)}
    MR=SR0*MY0(SS),AX0=DM(I0,M0);
                                {MR=(xa-xc+yb-yd)(Cb),
```

```
                                        AX0= xa of next bfly}
      MR=MR+AR*MY1(RND),AY0=DM(I0,M3);
                                    {MR=(xa-xc+yb-yd)(Cb)+
                                        (ya-yc-xb+xd)(Sb)}
                                        {AY0= xc of next bfly}
      DM(I0,M2)=MR1, MR=AR*MY0(SS);
                                    {output x'b=to position x'c}
                                        {MR=ya-yc-xb+xd)(Cb)}
      MR=MR-SR0*MY1(RND), MY0=DM(I7,M4);
                            {MR=(ya-yc-xb+xd)(Cb)-(xa-xc+yb-yd)(Sb)}
                                    {MY0=(Cd)}
      DM(I2,M2)=MR1, MR=MX0*MY0(SS);
                                    {output y'b= to position y'c}
                                    {MR=(xa-yb-xc+yd)(Cd)}
      MY1=DM(I7,M7), AR=AX0-AY0;  MY1=(Sd), AR=xa-xc }
      MR=MR+MX1*MY1(RND), AX1=DM(I2,M0);
                            {MR=(xa-yb-xc+yd)(Cd)+(ya+xb-yc-xd)(sd)}
                                    {AX1=ya of next bfly}
      DM(I1,M2)=MR1, MR=MX1*MY0(SS);
                        {output x'd=(xa-yb-xc+yd)(Cd)+(ya+xb-yc-xd)(sd)}
                                    {MR=(ya+yb-yc-yd)(Cd)}
      MR=MR-MX0*MY1(RND), AY1=DM(I2,M1);
                                    {MR=(ya+yb-yc-yd)(Cd)-(xa-xc-yb+yd)Sd}
                                    {yc of next bfly}
STG1BFY:   DM(I3,M2)=MR1, SR=LSHIFT AR(LO);
                                    {output y'd=(ya+xb-yc-xd)Cd-(xa-xc-yb+yd)Sd}
                                    {SR0=ya-yc of next bfly}

{ ---------------------------------------- Stage 2 ----------------------------------------}

STAGE2:    I0=^INPLACE;             {in      -> Xa,Xc}
           I1=^INPLACE+8;           {in+N/8  -> Xb,Xd}
           I2=^INPLACE+1;           {in+1    -> Ya,Yc}
           I3=^INPLACE+9;           {in+N/8+1 -> Yb,Yd}
           M0=Nov4;                 {N/4,   skip forward to dual node}
           M1=-Nov4;                {-N/4,  skip back to primary node}
           M2=-Nov4+2;              {-N/4+2, skip to next butterfly}

           M3=24;                   {N*3/8,  skip to next group}
           DM(M3_SPACE)=M3;             {M3_space is  temporary storage}
                                        {space needed because M3 issued}
                                        {in 2 contexts and will alternate}
                                        {in value}
```

```
        M5=Nov4+4;                  {N/4 +groups/stage*1, Cb Sb offset}
        M6=Nov4+8;                  {N/4 +groups/stage*2, Cc Sc offset}
        M7=Nov4+12;                 {N/4 +groups/stage*3, Cd Sd offset}
        SI=4;                   {Bfy/group, save counter for inner loop}
        DM(BFY_COUNT)=SI;   {SI is used as a temporary storage dummy}

        CNTR=4;                     {groups/stage}

DO MIDGRP UNTIL CE;
        I5=^COS_TABLE;
        I6=^COS_TABLE;
        I7=^COS_TABLE;
        AX0=DM(I0,M0);              {get first Xa}
        AY0=DM(I0,M1);              {get first Xc}
        AR=AX0-AY0, AX1=DM(I2,M0);      {Xa-Xc,get first Ya}
        SR=LSHIFT AR (LO), AY1=DM(I2,M1);   {SR1=Xa-Xc,get first Yc}
        CNTR=DM(BFY_COUNT);         {butterflies/group}

         M3=-2;                     {M3 is loaded with the special value}
                                    {required for pointer manipulation }
                                    {as explained previously}

       {This radix_4 bfly algorithm is a modified one. It has it's middle 2  }
       {Branches reversed. This alteration, done to every stage, results in }
       {a bit reversed output instead of a digit reversed output             }

       DO MIDBFY UNTIL CE;
            AR=AX0+AY0, AX0=DM(I1,M0);
                    {AR=xa+xc, AX0=xb }
            MR0=AR, AR=AX1+AY1;
                    {MR0=xa+xc, AR=ya+yc}
            MR1=AR, AR=AX1-AY1;
                    {MR1=ya+yc, AR=ya-yc}
            SR=SR OR LSHIFT AR (HI), AY0=DM(I1,M1);
                    {SR1=ya-yc, AY0=xd}
            AF=AX0+AY0, AX1=DM(I3,M0);
                    {AF=xb+xd, AX1=yb}
            AR=MR0+AF, AY1=DM(I3,M1);
                    {AR=xa+xb+xc+xd, AY1=yd}
            DM(I0,M0)=AR, AR=MR0-AF;
                    {output x'a=(xa+xb+xc+xd), AR=xa+xc-xb-xd}
            AF=AX1+AY1, MX0=AR;
                    {AF=yb+yd, MX0=xa+xc-xb-xd}
```

```
AR=MR1+AF, MY0=DM(I6,M4);
          {AR=ya+yc+yb+yd, MY0=(Cc)}
 DM(I2,M0)=AR, AR=MR1-AF;
          {output y'a, AR=ya+yc-yb-yd}
 MR=MX0*MY0(SS), MY1=DM(I6,M6);
          {MR=(xa+xc-xb-xd)(Cc), MY1=(Sc)}
 MR=MR+AR*MY1(RND), SI=DM(I0,M2);
          {MR=(xa-xb+xc-xd)(Cc)+(ya-yb+yc-yd)(Sc)}
          {SI is a dummy here to cause a modify(I0,M2)}
 DM(I1,M0)=MR1, MR=AR*MY0(SS);
          {output x'c to position x'b}
          {MR=(ya+yc-yb-yd)(Cc)}
 MR=MR-MX0*MY1(RND), MY0=DM(I5,M4);
          {MR=(ya+yc-yb-yd)(Cc)-(xa+xc-xb-xd)(Sc), MY0=(Cb)}
 DM(I3,M0)=MR1, AR=AX0-AY0;
          {output y'c to position y'b}
          {AR=xb-xd}
 AY0=AR, AF=AX1-AY1;
          {AY0=xb-xd, AF=yb-yd}
 AR=SR0-AF, MY1=DM(I5,M5);
          {AR=xa-xc-(yb-yd), MY1=(Sb)}
 MX0=AR, AR=SR0+AF;
          {MX0=xa-xc-yb+yd, AR=xa-xc+yb-yd}
 SR0=AR, AR=SR1+AY0;
          {SR0=xa-xc+yb-yd, AR=ya-yc+xb-xd}
 MX1=AR, AR=SR1-AY0;
          {MX1=ya-yc+xb-xd, AR=ya-yc-(xb-xd)}
 MR=SR0*MY0(SS),  AX0=DM(I0,M0);
         {MR=(xa-xc+yb-yd)(Cb), AX0= xa of next bfly}
 MR=MR+AR*MY1(RND),  AY0=DM(I0,M3);
          {MR=(xa-xc+yb-yd)(Cb)+ (ya-yc-xb+xd)(Sb)}
          {AY0= xc of next bfly}
 DM(I0,M2)=MR1, MR=AR*MY0(SS);
          {ouput x'b to position x'c}
          {MR=ya-yc-xb+xd)(Cb)}
 MR=MR-SR0*MY1(RND), MY0=DM(I7,M4);
          {MR=(ya-yc-xb+xd)(Cb)-(xa-xc+yb-yd)(Sb)}
          {MY0=(Cd)}
 DM(I2,M2)=MR1, MR=MX0*MY0(SS);
          {ouput y'b to position y'c}
          {MR=(xa-yb-xc+yd)(Cd)}
 MY1=DM(I7,M7), AR=AX0-AY0;
           {MY1=(Sd), AR=xa-xc }
```

```
            MR=MR+MX1*MY1(RND), AX1=DM(I2,M0);
                    {MR=(xa-yb-xc+yd)(Cd)+(ya+xb-yc-xd)(sd)}
                    {AX1=ya of next bfly}
            DM(I1,M2)=MR1, MR=MX1*MY0(SS);
                    {output x'd=(xa-yb-xc+yd)(Cd)+(ya+xb-yc-xd)(sd)}
                    {MR=(ya+yb-yc-yd)(Cd)}
            MR=MR-MX0*MY1(RND), AY1=DM(I2,M1);
                    {MR=(ya+yb-yc-yd)(Cd)-(xa-xc-yb+yd)Sd}
                    {yc of next bfly}
MIDBFY:     DM(I3,M2)=MR1, SR=LSHIFT AR(LO);
                    {output y'd=(ya+xb-yc-xd)Cd-(xa-xc-yb+yd)Sd}
                    {SR0=ya-yc of next bfly}

        M3=DM(M3_SPACE);    {modifier M3 is loaded with 'skip to}
                            {next group'_count and is used in the}
                            {next four instructions}

        MODIFY (I0,M3);
        MODIFY (I1,M3);             { point to next group }
        MODIFY (I2,M3);             { of butterflies }
MIDGRP:  MODIFY (I3,M3);

{ ---------------------------------------- Stage 3 ----------------------------------------}

laststage:  I4=^INPLACE;            {in          ->Xa,Xc}
          I5=^INPLACE+2;            {in+N/512    ->Xb,Xd}
          I6=^INPLACE+1;            {in+1        ->Ya,Yc}
          I7=^INPLACE+3;            {in+N/512+1 ->Yb,Yd}
          M4=4;                     {N/256,  skip forward to dual node}

          M0=h#0200;          {this magic modify value is used to }
                            {perform a bit reverse as the final results}
                            { are written out.  Because of interleave}
                            {this is not as simple as explained in }
                            {1410 Handbook.  Derivation of these values}
                            {is in the Apps. note of this code}

          I0=h#0000;        {these  base address values are derived }
          I2=h#2000;        {for an output at address 0000.}
          I1=h#0100;
          I3=h#2100;
```

```
L4=0;              {this last stage has no twiddles and}
L5=0;              {because it is where the DAG#1 bit_}
L6=0;              {reverses the output, the I's M's & L's}
L7=0;              {have been reassigned and re_initialized}

AX0=DM(I4,M4);     {first Xa}
AY0=DM(I4,M4);     {first Xc}
CNTR=Nov4;         {groups/stage}
ENA BIT_REV;       {all data accesses using I0..I3 are}
                   {bit-reversed  }

{radix_4 bfly algorithm is a modified one.  It has it's middle 2 }
{Branches reversed.  This alteration, done to every stage, results in }
{a bit_reversed output instead of a digit reversed output          }


DO LASTSTGBFY UNTIL CE;
        AR=AX0-AY0, AX1=DM(I6,M4);   {AR=xa-xc, AX1=ya}
        SR=LSHIFT AR(LO), AY1=DM(I6,M4);  {SR0=xa-xc, AY1=yc}
        AR=AX0+AY0, AX0=DM(I5,M4);   {AR=xa+xc, AX0=xb}
        MR0=AR, AR=AX1+AY1;   {MR0=xa+xc, AR=ya+yc}
        MR1=AR, AR=AX1-AY1;   {MR1=ya+yc, AR=ya-yc}
        SR=SR OR LSHIFT AR (HI), AY0=DM(I5,M4);
                                          {SR1=ya-yc, AY0 xd}
        AF=AX0+AY0, AX1=DM(I7,M4);   {AF=xb+xd, AX1=yb}
        AR=MR0+AF, AY1=DM(I7,M4);    {AR=xa+xc+xb+xd,
                                          AY1=yd}
        DM(I0,M0)=AR, AR=MR0-AF;     {output x'a=xa+xc+xb+xd}
                                     {AR=xa+xc-(xb+xd)}
        DM(I1,M0)=AR, AF=AX1+AY1;    {output x'c to position x'b}
                                     {AF=yb+yd}
        AR=MR1+AF;                   {AR=ya+yb+yc+yd}
        DM(I2,M0)=AR, AR=MR1-AF;     {output y'a=ya+yc+yb+yd}
                                     {AR=ya+yc-(yb+yd)}
        DM(I3,M0)=AR, AR=AX0-AY0;    {output y'c to position y'b}
                                     {AR=xb-xd}
        AX0=DM(I4,M4);               {AX0=xa of next group}
        AF=AX1-AY1, AY1=AR;          {AF=yb-yd, AY1=xb-xd}
        AR=SR0+AF, AY0=DM(I4,M4);    {AR=xa-xc+yb-yd}
                                     {AY0=xc of next group}
        DM(I0,M0)=AR, AR=SR0-AF;     {output x'b to position x'c}
                                     {AR=xa-xc-(yb-yd)}
        DM(I1,M0)=AR, AR=SR1-AY1;    {output x'd=xa-xc-(yb-yd)}
                                     {AR=ya-yc+(xb-xd)}
```

```
                DM(I2,M0)=AR, AR=SR1+AY1;    {output y'b to position y'c}
                                             {AR=ya-yc-(xb-xd)}
LASTSTGBFY:     DM(I3,M0)=AR;                {output y'd=ya-yc-(xb-xd)}

                DIS  BIT_REV;                {shut-off bit reverse}
                                             {mode, not necessary}
                RTS;                         {end and exit from FFT
                                                   subroutine}

.ENDMOD;

{ ---------- END CODE ----------}
```

{

Program: MAG.DSP

Description: This module multiplies the real and imaginary parts of the fft to get the magnitude. The real and imaginary parts are squared, their double precision sum is placed over the existing values;

Benchmark:

setup = 10
mult = 224 = 7(32)

Total 234 cycles * 60 ns = 14.04 µs

}

```
.MODULE/BOOT=0   mag_sub;

.const           Nov2=32;

.EXTERNAL        OUTPUT, fifo;

.ENTRY           mag;


mag:  I0=^OUTPUT;              {pointer for real}
      I1=^OUTPUT;              {pointer for squared sum}
      M0=1;
      M1=1;
      L0=0;
      L1=0;
      MX1=0;                   {MSB set to zero}
      MY1=0;
      CNTR=Nov2;

      DO MULT UNTIL CE;        {loop through fft}
         MX0=DM(I0,M0);        {get real}
         MY0=MX0;
         MR=MX0*MY0(SS), MX0=DM(I0,M0);  {square real}
```

```
            MY0=MX0;                {set up imaginary}
            MR=MR+MX0*MY0(SS);      {square and sum}

                    {output for debugging
                        DM(fifo)=MR1;
                        DM(fifo)=MR0;   }

            DM(I1,M1)=MR1;          {back fill array}
MULT:        DM(I1,M1)=MR0;

RTS;

.ENDMOD;
```

```
|_____________________________________________________________________
_____________________________________________________________________

      Program:     DETECT.DSP

      Description:  This module  detects the presence of the pinger frequency;

      Benchmark:

       setup      =      23
       freq       =      54    =     3(18)
      thres       =      11

       Total     88 cycles * 60 ns      =      5.3 µs

_____________________________________________________________________
_____________________________________________________________________)


.MODULE/BOOT=0   DETECT_SUB;

.const           N = 64;

.VAR/DM          FREQUENCY[12];
.INIT            FREQUENCY[0]:0,0,0, 0,0,0, 0,0,0, 0,0,0;
.VAR/DM          MAG[4];
.INIT            MAG[0]: 16, 40, 52, 28;
.VAR/DM          THRESHOLD[8];
.INIT            THRESHOLD[0]:500,0, 500,0, 500,0, 500,0;

.ENTRY           DETECT;
.EXTERNAL        OUTPUT, hydrophone, fifo;

DETECT:    I0=^FREQUENCY;        {pointer for reject period}
       M0=3;
       L0=0;
       I1=^THRESHOLD;            {pointer for threshold values}
       M1=1;
       L1=0;
       I2=^MAG;                        {pointer to frequency values}
       M2=1;
       L2=0;
       L3=0;
```

```
        CNTR=DM(hydrophone);          {update for hydrophone}
        DO NEXT UNTIL CE;
              MODIFY(I0,M0);
NEXT:         MODIFY(I1,M1);
        AY1=DM(I1,M1);
        AY0=DM(I1,M1);
        M0=1;
        CNTR=3;
        DO FREQ UNTIL CE;
              AX0=DM(I0,M0);
              IF NE JUMP FREQ;
                    AX0=DM(I2,M2);
                    I3=^OUTPUT;
                    M3=AX0;
                    MODIFY(I3,M3);
                    M3=1;
                    AX1=DM(I3,M3);
                    AX0=DM(I3,M3);
                    AR=AX0-AY0;
                    SR0=AR, AR=AX1-AY1+C-1;
                    SR1=AR;
             IF GE JUMP THRES;
                DM(fifo)=CNTR;
                MX0=DM(hydrophone);
                DM(fifo)=MX0;
                DM(fifo)=AX0;
FREQ:           NOP;
THRES:           AX0=DM(I2,M2);
             I3=^OUTPUT;
             M3=AX0;
             MODIFY(I3,M3);
             AX1=DM(I3,M3);
             AX0=DM(I3,M3);
             AR=AX0+AY0;
             SR0=AR, AR=AX1+AY1+C;
             SR1=AR;
             SR=ASHIFT SR1 BY -1 (HI);
             SR=ASHIFT SR0 BY -1 (LO);

RTS;                          {end and exit}

.ENDMOD;
```

APPENDIX E

MathCad Simulations

E - THE MASKING PROBLEM

First define some derived units:

$ms := 10^{-3} \cdot sec$ $\quad kHz := 10^{3} \cdot Hz$ $\quad \mu s := 10^{-6} \cdot sec$

Let:

$f_s := 41 \cdot kHz$	Sample frequency	
$N := 64$	Block size	
$T := \frac{1}{f_s}$	Time resolution	$T = 24.3902 \cdot \mu s$
$L := N \cdot T$	Block time	$L = 1.561 \cdot ms$
$i := 0 .. N - 1$	Index for block	
$j := 0 .. 500$	Index for repeats	
$k := 0 .. \frac{N}{2}$	Index for frequency range	
$n := 12$	Bit resolution	
$V := 5$	Voltage range	
$q := \frac{2^n}{V}$	Digitize resolution	

Define the two closest frequencies and noise level:

$f_l := 65.5 \cdot kHz$ $\quad A_l := 5$ $\quad phase_{l_j} := rnd(2 \cdot \pi)$

$f_u := 69.0 \cdot kHz$ $\quad A_u := 5$ $\quad phase_{u_j} := rnd(2 \cdot \pi)$

$noise\_level := 10 \cdot 10^{-3}$

Signal equation for both frequencies:

$$c\,l_{i,j} := A_l \sin\left(2\cdot\pi\cdot f_l\cdot i\cdot T + phase_{l_j}\right)$$

$$c\,u_{i,j} := A_u \sin\left(2\cdot\pi\cdot f_u\cdot i\cdot T + phase_{u_j}\right)$$

Noise model:

$$n_{i,j} := rnd(noise\_level) - \frac{noise\_level}{2.0} \qquad n\,f_{i,j} := floor\left(n_{i,j}\cdot q\right)$$

Define two channels, one with one frequency and the other with two, add noise then digitize:

$$s\,1_{i,j} := floor\left[\left(c\,l_{i,j} + n_{i,j}\right)\cdot q\right]$$

$$s\,2_{i,j} := floor\left[\left(c\,l_{i,j} + c\,u_{i,j} + n_{i,j}\right)\cdot q\right]$$

Hanning window

$$h_{i,j} := floor\left[0.5\cdot\left(1 - \cos\left(2\cdot\pi\cdot i\cdot\frac{T}{L}\right)\right)\cdot 409.6\right]$$

Calculate magnitude response of one, both and noise:

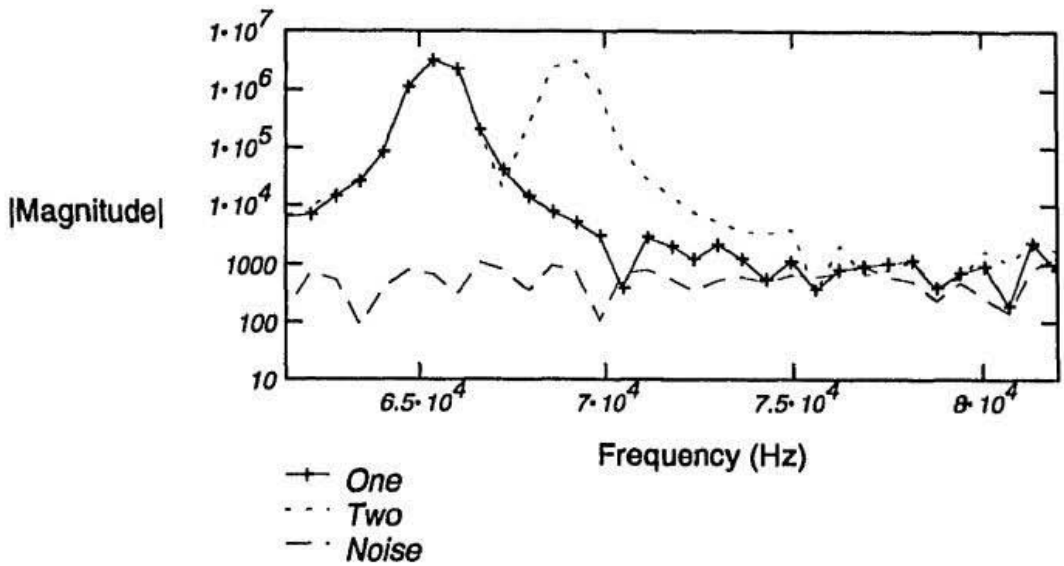
$$One^{<j>} := \overrightarrow{\left|fft\left(s\,1^{<j>}\cdot h^{<j>}\right)\right|}$$

$$Two^{<j>} := \overrightarrow{\left|fft\left(s\,2^{<j>}\cdot h^{<j>}\right)\right|}$$

$$Noise^{<j>} := \overrightarrow{\left|fft\left(n\,f^{<j>}\cdot h^{<j>}\right)\right|}$$

Convert to frequency range for plot

$$f_k := 82 \cdot kHz - k \cdot \frac{f_s}{N}$$



$$D_{dB_j} := 20 \cdot \log\left[\frac{\left(One^{<j>}\right)_{20}}{\left(One^{<j>}\right)_{26}}\right] \qquad mean\left(D_{dB}\right) = -56.9276$$

$$stdev\left(D_{dB}\right) = 0.8254$$