

University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2009

Encoding ultrasonic signals to improve the signal to noise ratio for the inline inspection of spot welds

Lawrence Barsanti
University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Barsanti, Lawrence, "Encoding ultrasonic signals to improve the signal to noise ratio for the inline inspection of spot welds" (2009). *Electronic Theses and Dissertations*. 7935.
<https://scholar.uwindsor.ca/etd/7935>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Encoding Ultrasonic Signals to Improve the Signal to Noise Ratio for the Inline Inspection of Spot Welds

by
Lawrence Barsanti

A Thesis
Submitted to the Faculty of Graduate Studies
through Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
University of Windsor

Windsor, Ontario, Canada
2009
© 2009 Lawrence Barsanti



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-57583-3
Our file *Notre référence*
ISBN: 978-0-494-57583-3

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Inline spot weld analysis (ISWA) is a relatively new application of ultrasonic technology that is capable of assessing spot welds as they are made. In previous studies, the fundamental concepts of ISWA have been uncovered and strong correlations with other test methods have been produced. Unfortunately, the current apparatus for ISWA has a low signal to noise ratio (SNR) in some installations which can lead to unusable weld signatures and incorrect interpretations. This work applies coded ultrasonic signals to ISWA in an attempt to improve the SNR of the acquired weld signatures and ultimately improve the reliability of ISWA.

In this work, the apparatus for ISWA is modified to allow for the transmission of chirp encoded ultrasonic signals. The chirp signal that provides the best results with the modified apparatus is found and its scans are compared to scans obtained with the original apparatus. Overall, the scans taken with the original apparatus tend to have better contrast and less noise but the scans taken using the modified apparatus were similar in quality. However, the available equipment severely limited the amplitude of the chirp pulse so the fact that scans of similar quality were generated is impressive.

Table of Contents

Author's Declaration of Originality	iii
Abstract	iv
I. Introduction	1
II. Resistance Spot Welding	2
Spot Weld Quality	4
III. Acoustic Waves	5
IV. Applied Ultrasound	8
Common Ultrasonic Components	8
Pulse-Echo Method	9
V. Signal Processing	12
Noise and Equipment Sensitivity	12
Averaging Filter	13
Bandpass Filter	14
Envelope Detector	15
Pulse Detection	15
VI. Pulse Compression	18
Practical Considerations for Ultrasonic Imaging	20
VII. ISWA Apparatus	25
Data Acquisition	25
Acoustic Wave Propagation	27
Interpreting Data	30
Signal to Noise Ratio	32
VIII. Experimental ISWA Apparatus	35
Data Acquisition	35
Interpreting Data	36
Additional Software	37
IX. Experimental Results	39
Amplified Signals	39
Chirp Optimization	41
Comparison with Pulse Mode	42
X. Conclusion	44
References	45
Appendix	46
Convolution	46
Matched Filtering	47
Template Signal	49
Greyscale Conversion	49
Bandpass Filtering	49
Acquire b-scan	50
Vita Auctoris	54

I. Introduction

Inline spot weld analysis (ISWA) is a relatively new application of ultrasonic technology that is capable of assessing spot welds as they are made. In previous studies [2, 3, 4] the fundamental concepts of ISWA have been uncovered and strong correlations with other test methods have been produced. Furthermore, work with industry partners has lead to robust hardware that can survive in the manufacturing environment.

Industry partners, such as Chrysler, Magna, and Mercedes, have shown great interest in this technology because of it potential to reduce the cost of quality assurance. In order to provide quality assurance, industry currently utilizes an assortment of post-production test methods to perform a variety of ad hoc sampling procedures. This approach is quite expensive because it is labour intensive and causes valuable product to be scrapped. Additionally, when a sample contains defects another labour intensive process, referred to as the containment process, must be performed in order to find and repair all of the bad products.

If implemented successfully, ISWA could greatly reduce the cost associated with quality assurance because it would be integrated into the assembly line and ultimately reduce labour requirements and defective product. In other words, ISWA is capable of detecting problems as they occur so it will eliminate the need for product sampling and containment. Unfortunately, the current apparatus for ISWA has a low signal to noise ratio (SNR) in some installations which can lead to unusable weld signatures and incorrect interpretations. The industry partners require this technology to work reliably in an automated environment before it can be integrated into their assembly lines. This work will apply coded ultrasonic signals to ISWA in order to improve the SNR of the acquired weld signatures and ultimately improve the reliability of ISWA.

II. Resistance Spot Welding

Resistance spot welding is a procedure that uses electrical current to fuse sheets of steel or aluminum alloys. On the surface resistance spot welding looks like a straight forward procedure but there are several electrical and thermal properties that must work in harmony in order to produce a successful weld.

$Q = I^2 R t$	Q is heat in (J)
	I is current (A)
	R is Resistance (Ω)
	t is time (s)

Equation 1: Heat Equation [5]

During welding heat is generated according to the heat equation (Equation 1). Producing successful welds requires that both the location and the amount of heat generated be controlled. If the amount of heat is not properly controlled quality issues like expulsion or undersized and cold welds can occur. If the location of heat is not controlled, then the weld nugget can form in an undesirable location; like in the middle of a thick plate or between a weld cap and a plate.

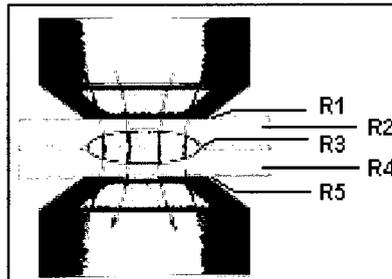


Figure 1: Sources of resistance [5]

Figure 1, shows five different sources of resistance that cause localized heat generation during the welding procedure. The sheets (R2 & R4) have bulk resistance which depends on the sheet's material, thickness, and temperature. Bulk resistance can vary greatly between different metals and alloys; however it always increases with temperature. The graph below shows the bulk resistance of mild steel, pure aluminum, and copper. Weld caps are made from copper which has a low bulk resistance that

limits the heat generated in the weld cap during welding¹. This is important because weld caps tend to be much thicker than the plates that are being welded. Furthermore, the bulk resistance of mild steel makes it ideal for welding because as mild steel heats up its resistance greatly increases which results in more heat. Conversely, the bulk resistance of aluminum is very similar to copper which makes it almost impossible to weld pure aluminum; however aluminum alloys are routinely welded.

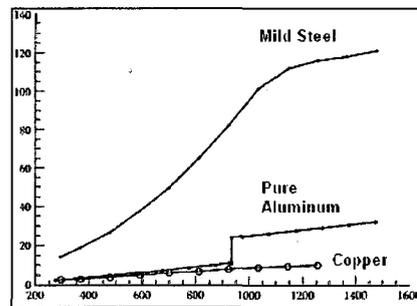


Figure 2: Bulk resistance vs. temperature [5]

The interfaces between the different surfaces (R1, R3, and R5) are sources of contact resistance. Contact resistance is a result of small deformities, bumps and pits, on the surfaces and substances such as oil and dirt on the surfaces all of which prevents full contact between the two surfaces. However, the high heat and pressure during the welding process tends to burn away most foreign substances and eliminate most surface deformities. Thus contact resistance does not play a big role in the welding procedure.

In order to have successful fusion, high temperatures must be generated at the interface between the two plates so they both melt and mix together. In addition, much lower temperatures must be maintained at the interfaces between the electrode and plates in order to prevent them from becoming fused together. For this reason, weld caps are cooled by water running through the weld gun which removes heat from the electrode-plate interfaces.

1. Another reason copper is used is its ability to rapidly diffuse heat.

Spot Weld Quality

Over the past few decades several technologies have been developed that aim to optimize the spot weld process by increasing throughput and reducing cost.

Automation of the assembly line has greatly increased throughput and essentially removed humans from the spot welding process. Furthermore, technologies like current steppers and tip dressers have extended the life of weld caps reducing expenses even more. All these technologies are great but they introduce several variables into the spot welding process and reduce the potential for a human to detect flaws in the process. For these reasons the quality of spot welds must constantly be checked.

The quality of a spot weld is typically based on the strength of the joint, the size and position of the weld nugget, or the appearance of the weld. There is a wide range of techniques and tools for measuring these different characteristics all of which require varying degrees of skill and time. Precise quantitative methods, like tensile tests, can be used to measure the strength of the weld joint and methods like cross sectioning and acoustical imaging can be used to measure the size and position of the weld nugget which is strongly correlated to the strength of the joint. These precise methods make sense in a research environment but they are not practical in a manufacturing environment where the time and skill are not available.

Automotive manufacturers make hundreds of thousands of welds a day so they have to base weld quality on qualitative methods and low precision quantitative methods. Visual inspections allow weld auditors to quickly detect missing welds and welds with characteristics that are common in low quality welds such as excessive indentation, burn through, expulsion markings, braising, and discoloration. A simple chisel check provides a quick way for weld auditors to find very weak welds. Quantitative methods like ultrasound and various teardowns can be used to measure the diameter of the weld nugget but they are more time consuming and often have up to ± 1 mm of error.

III. Acoustic Waves

Acoustical waves transfer energy, pressure, through a medium by vibrating the particles in the medium. When an acoustic wave travels through a medium only the energy propagates; particles in the medium will vibrate and then return to their initial positions². The rate at which the acoustical wave vibrates the particles is its frequency. Ultrasound consists of acoustical waves that have frequencies greater than 20 kHz; the upper limit of human hearing.

Acoustic waves are often classified by their mode which is defined as the motion of particles in relation to the direction of the wave's energy. In unbound, infinite media, acoustic waves can be longitudinal, particles oscillate parallel to direction of energy, or transverse, particles oscillate perpendicular to direction of energy. Longitudinal waves can propagate in solid, liquid, or gaseous media but transverse waves will only propagate in solids.

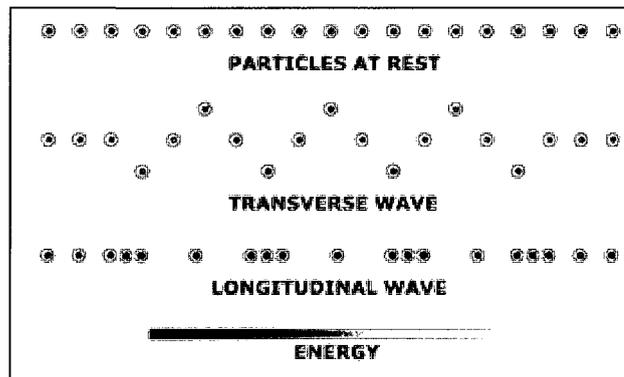


Figure 3: Modes of wave propagation

The velocity of an acoustic wave depends on both the mode and medium in which it is traveling. The velocity of longitudinal waves in a medium is directly proportional to its compressive strength; however the velocity of transverse waves in a medium is directly proportional its sheer strength. Since the sheer strength of a medium is always less

2. If the vibrations exceed the medium's elastic limit particles will not return to their initial position and the medium will be permanently deformed. The remainder of this work assumes that the acoustic waves do not have enough energy to exceed the medium's elastic limits.

than its compressive strength, transverse waves will always travel slower through a medium than longitudinal waves. For both transverse and longitudinal waves the density of a medium is inversely proportional to the velocity of sound in that medium.

$V_{Liquid} = \sqrt{\frac{K}{\rho}}$	K – Bulk Modulus (GPa) of the liquid medium; its resistance to compression
$V_{Longitudinal} = \sqrt{\frac{E(1-\sigma)}{\rho(1+\sigma)(1-2\sigma)}}$	E – Young’s Modulus (GPa) of the solid medium; its stiffness
$V_{Transverse} = \sqrt{\frac{E}{2\rho(1+\sigma)}}$	ρ – Density of the medium (kg/m ³)
	σ – Poisson’s Ratio; a materials tendency to contract in directions perpendicular to a force that is stretching it

Equation 2: Acoustic wave velocity [8]

As an acoustic wave propagates through a medium it loses energy due to scattering and absorption; this loss of energy is referred to as attenuation. The attenuation coefficient of a material depends heavily on the grain structure of a material and is quite difficult to calculate due to the large number of parameters involved. Attenuation also depends on the distance traveled in the media and the frequency of the acoustic wave.

$A(d, f) = A_0 e^{-\alpha(f)d}$	A_0 – original amplitude of the wave
	d – distance traveled (cm)
	f – frequency of the wave (MHz)
	$\alpha(f)$ – frequency dependent attenuation coefficient (dB/(MHz·cm))

Equation 3: Amplitude attenuation [8]

When an acoustic wave encounters a boundary between two mediums some of the wave’s energy is reflected into the original medium and the remainder is transmitted into the new medium. The reflection coefficient is directly proportional to the impedance mismatch of the materials involved. If the mismatch is too large, all the energy will be reflected and it will be impossible for the acoustic wave to transverse the boundary into

the new medium. Furthermore, if the acoustic impedance of the new medium is greater than the current medium, then the reflected wave will experience a phase shift of 180 degrees. The following equations show how to calculate the reflection and transmission coefficient for waves travelling normal to an ideal interface between two mediums.

$Z = \rho \times V_l$	Z – acoustical impedance of a material ($\text{kg}/(\text{m}^2 \cdot \text{s})$); the subscripts 1 and 2 denote the source and destination mediums
$R = \left(\frac{Z_1 - Z_2}{Z_1 + Z_2} \right)^2$	ρ – density of the medium (kg/m^3)
	V_l – longitudinal velocity of waves in the medium (m/s)
	R – reflection coefficient
$T = 1 - R$	T – transmission coefficient

Equation 4: Transmission and reflection coefficients [9]

When acoustic waves cross the boundary from one isotropic³ medium to another they are subject to refraction. As with light, Snell's Law can be used to describe the relationship between velocities in different materials and the angles of incidence and refraction. However, refraction is a bit more complicated with acoustics than with light because waves can propagate in different modes with different velocities. As a result, it is possible for one incident wave to excite multiple waves in different modes and at different angles when crossing the boundary between two mediums [9]. The following example illustrates how a longitudinal wave traveling from one solid to another can generate four separate waves.

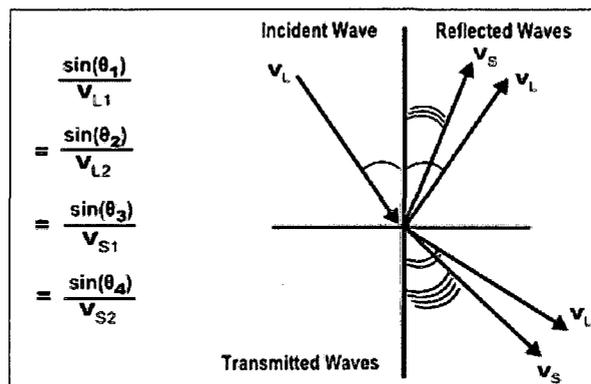


Figure 4: Snell's law (acoustic wave)

3. Isotropic materials have the same elastic properties in all directions

IV. Applied Ultrasound

Ultrasound is used in a variety of imaging, measurement, flaw detection, and material characterization techniques. Ultrasound is a good choice for these techniques because it can look at the internal structure of opaque specimens and because it can be used non-destructively. The apparatus of a typical ultrasonic application consists of a transducer, a way to position or focus the transducer, a pulser and receiver, and a computer or some other display. In addition to the hardware, there are various techniques for collecting ultrasonic data which are based on various characteristics of the travelling wave (time of flight, attenuation, scattering, etc...).

Common Ultrasonic Components

Ultrasonic transducers exploit the piezoelectric effect in order to generate and detect ultrasonic waves. Materials that exhibit the piezoelectric effect produce an electric charge when a subjected to a mechanical stress and, conversely, undergo a mechanical stress when an electrical charge is applied. The piezoelectric effect exists in many naturally occurring crystals such as quartz and cane sugar because the electrons move more easily along some axes than others. However, modern applications that make use of the piezoelectric effect typically employ man-made ceramics because they are non-volatile and customizable. The properties a transducer exhibits are a direct result of both the materials it is made from and its geometry. For example, the resonance frequency, the frequency that most easily excites the transducer, is the determined by the transducers thickness and the beam divergence, angle from the central axis to the outer edge of the sound beam, is determined by the diameter. On the other hand, properties such as efficiency, heat tolerance, and maximum V/mm are determined by the materials that make up the transducer [10]. Most important to this research is the

frequency response; the range of frequencies that can excite the transducer. The frequency response of a typical transducer is bell shaped (Gaussian) and centered at the resonance frequency [7].

Controlling where the ultrasonic waves go is very important and several methods for positioning and focusing transducers have been developed. Acoustic lenses that focus ultrasound and mechanical scanning systems that move transducers are very common. Other methods may use an array or matrix of independent transducers to examine a target area without physically moving the transducer. Phased array transducers use interference from several independent elements to change the transducer's focus and are also capable of scanning a target area without moving.

Pulsers are responsible for providing the electrical pulses needed to excite the transducer. Both the shape and magnitude of the pulses are important for proper excitation. Receivers are responsible for digitizing the voltage at the transducer so that it can be processed by a computer. The digitization process depends on timing so it is very important that the pulser and receiver be synchronized. In some cases it is also necessary to synchronize pulser and receiver with the positioning equipment or specimen.

Lastly, a computer, oscilloscope, or some custom equipment is used to process and display the ultrasonic data in a meaningful way. The most basic ultrasonic measurement is the A-scan which shows the transducer output over a short period of time. More advanced techniques collect multiple A-scans over time and space to build multi-dimensional datasets.

Pulse-Echo Method

The pulse-echo method is just one of the various techniques for collecting ultrasonic data. The pulse-echo method is the preferred method for ISWA because it provides

valuable information and the transducer is simple enough to be integrated into resistance spot welding hardware [4]. The principle idea of the pulse-echo method is to emit an ultrasonic pulse from a transducer and to listen to the reflections from the various interfaces using the same transducer. A simple example of the pulse-echo method is illustrated in the following sequence of images.

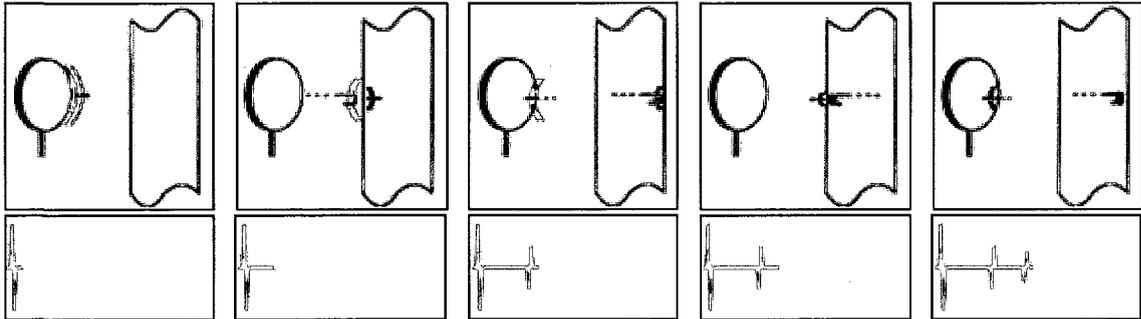


Figure 5: Pulse echo method

Two important observations can be made from these illustrations. First, an ultrasonic wave is required to travel twice the distance from the transducer to an interface before it is detected. This round trip time is called time of flight and can be used to calculate material thickness or distance between interfaces if the speed of sound in a sample is known; $d = (v * t)/2$ [13]. Second, internal reflections are generated when an echo must pass through one or more interfaces before reaching the transducer (fourth image in the series) making interpretation more complex.

When A-scans are collected there is usually a small region of interest that must be analyzed in order to extract some information. It is common practice to limit the A-scan to this region of interest through a process known as gating. Gates are simply two points in time, relative to the initial pulse, that specify when to begin and end the A-scan. Gating helps to reduce the amount of memory needed to acquire the data and process it. It also eliminates some reflections that are not necessary for analysis and that could confuse signal processing software. In the A-scan shown below, the gates would eliminate the

initial pulse and the internal reflections making it easier for software to calculate the thickness of the sample.

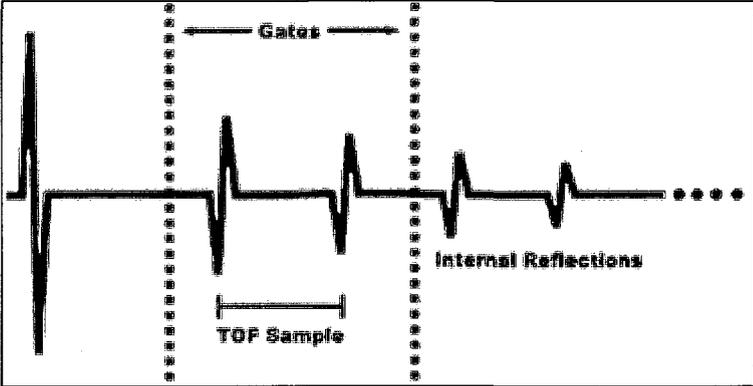


Figure 6: A-Scan

V. Signal Processing

Given the vast amounts of data that can be produced by sensors such as ultrasound it is very useful to have automated ways for dealing with the data. Signal processing is a very broad field that covers both analog and digital signals as well as a wide variety of techniques used for tasks like filtering, compressing, and interpreting data. This section will cover topics in digital signal processing that are relevant to this work.

Noise and Equipment Sensitivity

Measurements from all types of sensors are subject to variety of errors known as noise. Obviously, noise is application dependent but it is typically considered to be additive; meaning it added to the actual signal by wave superpositioning.

$X[i] = S[i] + H[i]$	X – acquired signal S – actual signal H – noise
----------------------	---

Equation 5: Additive Noise [6]

Two ways to classify noise are by its frequency spectrum and by the distribution of its magnitude. Random noise is typically modeled as Additive White Gaussian Noise which means it is evenly distributed in the frequency domain (white) and has a magnitude with a Gaussian distribution. The central limit theorem explains why a Gaussian distribution is a good choice for modeling the magnitude. The theorem states that when several random variables with arbitrary distributions are combined the resulting distribution is Gaussian [6].

The sources of noise for a given system can be lumped into two categories. The noise generated by the equipment used for signal acquisition and the noise found in the environment in which the equipment is used. For the most part, noise in the environment, cannot be controlled and must be managed with various filtering

techniques. However, noise generated by the equipment comes from the various filters and amplifiers, analog-to-digital conversion, sensors, and even the electrical components. This means that noise generated by the equipment is result of its design and determines the sensitivity of the equipment. Both types of noise contribute to the SNR of the measured signals, however when working with signals that are too weak for a given piece of equipment, the equipment can actually introduce a significant amount of noise.

Averaging Filter

Averaging filters are commonly used to mitigate the effects of random noise on experimental data. The filter simply takes several measurements of a variable and then returns the average value. The noise reduction is directly proportional to the number of samples taken (n) because the sum of the noise will tend to zero as n increases. A digital signal can be averaged by acquiring the data several times under the same conditions and then averaging each sample independently. If the same conditions are not maintained averaging will introduce noise instead of reduce it. Thus in ultrasound, care must be taken to insure that internal reflections have dissipated.

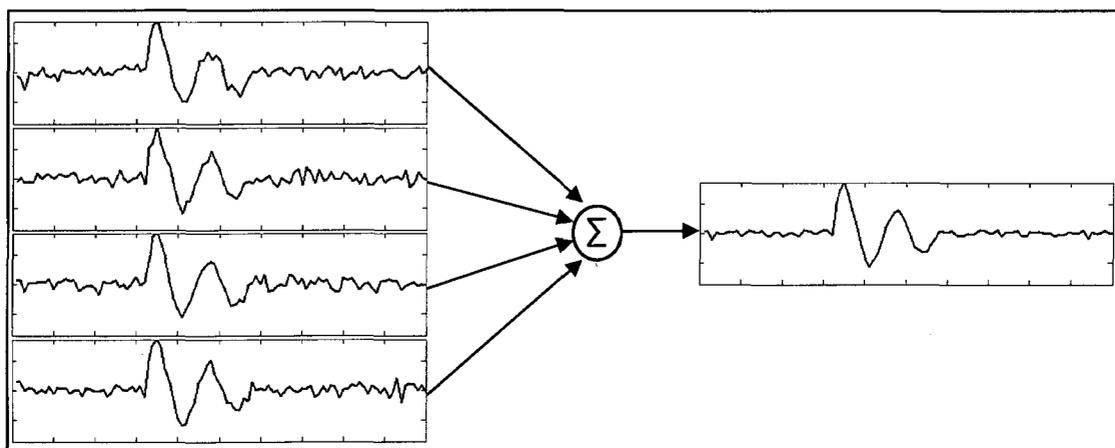


Figure 7: Averaging filter

Bandpass Filter

A bandpass filter is used to restrict the frequency components of a signal to a specific range of frequencies. This type of filter is advantageous when using the pulse-echo method because noise can be reduced by eliminating frequencies that do not exist in the initial pulse. When working with ultrasound, the frequency response of a transducer determines the range of frequencies that are contained in the initial pulse.

The bandpass filter is relatively easy to implement in software because there are several software libraries for computing both the Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT). A simple algorithm for a bandpass filter is:

1. Use FFT to convert the time domain signal to a signal in the frequency domain
2. Zero all the frequency values outside of the specified range
3. Use IFFT to convert the frequency domain signal back to the time domain

The following set of images illustrates how performing a bandpass filter on individual a-scans can significantly reduce the noise in a b-scan⁴.

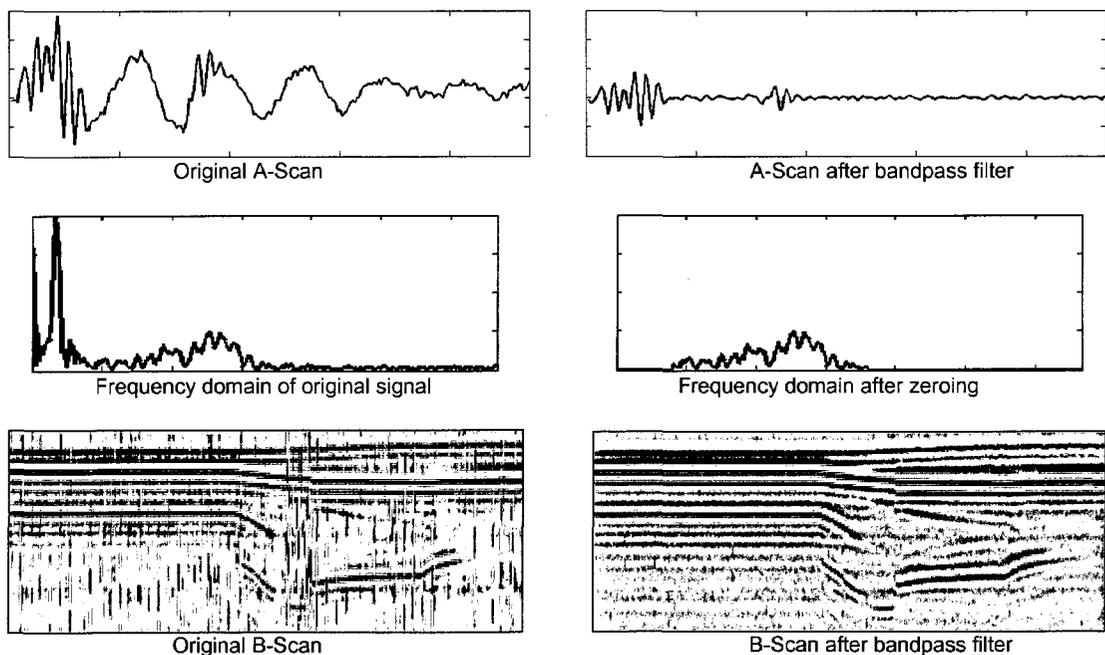


Figure 8: Bandpass filter

4. a collection of a-scans shown side by side; defined on page 25.

Envelope Detector

Some signal processing techniques work best on signals that only contain positive or negative values. An oscillating signal can be confined to either positive or negative values by calculating the envelope of the signal. The envelope of a signal, $s(t)$, is defined as, $E(t) = |s(t) + j * H(s(t))|$, where $H(\cdot)$ denotes the Hilbert transform [15]. The envelope of a signal represents the total energy of a signal over time and can be used to pinpoint the times where pulses are present [13].

Calculating the envelope of a signal is a time consuming operation so there are several analog and digital filters that can be used to approximate the envelope of a signal when performance is a concern. However, performance is not a major concern in this work so the FFT and IFFT are used to calculate, not approximate, the envelope. The algorithm, as explained in [13], is as follows:

1. Use FFT to convert the signal to the frequency domain
2. Zero out all the positive frequencies
3. Multiply the negative frequencies by two
4. Calculate the complex magnitude of the IFFT

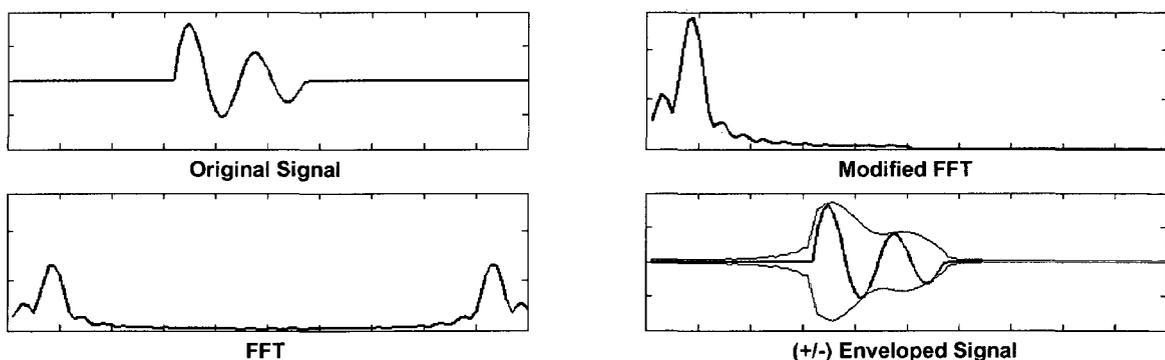


Figure 9: Envelope detector

Pulse Detection

Pulse detection is a common task in signal processing that involves finding a known signal in a longer unknown signal. Matched filtering is a linear filtering technique that obtains the optimal signal to noise ratio for pulse detection regardless of any encoding

scheme [6]. Matched filtering is based on the concept of autocorrelation which is quite simple to understand and to implement. Simply put, autocorrelation is a measure of how well a signal matches a time-shifted version of itself. Formally, it is defined by the following equation for a discrete periodic signal in the time domain.

$$R_{xx}[t] = \sum_{i=0}^{N-1} x_i x_{\text{mod}(i+t, N)}$$

Equation 6: Autocorrelation Function [6]

In general, a matched filter can utilize the concept of autocorrelation by convolving the received signal with a time-reversed copy of the transmitted signal; called the target function. The result of matched filtering is a signal with optimal signal to noise ratio for pulse detection, however, further signal processing is still needed in order to determine if there are pulses and the location of the pulses if they exist.

The pulse detection method just described is optimal for SNR but the methods resolution in the time domain is greatly influenced by the transmitted pulse. The examples in Figure 10 illustrate how the transmitted pulse influences matched filtering. The first row shows an ideal signal containing one or two reflections of a target signal. In the first column there is a short pulse, in the second there is a longer pulse, and in the third column there are two long pulses that are overlapping. The second row shows the result of convolving the ideal signal with the transmitted signal and the last row contains the envelope of the convolved signal. The enveloped signal is quite simple which makes it good choice for further analysis.

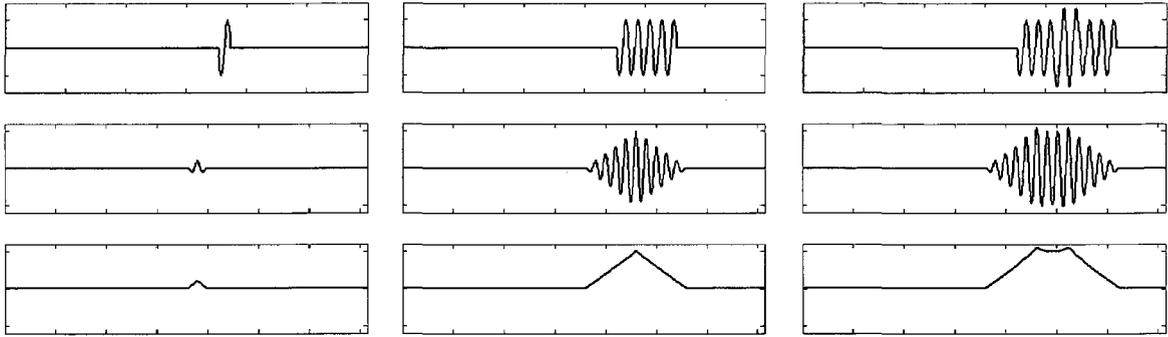


Figure 10: Matched filtering example (ideal)

From this example it clear that after signal processing the longer pulse is much more pronounced and as a result would be easier to detect in the presence of noise. Unfortunately, the longer pulse reduces the resolution in the time domain as can be seen in the third column. It would be hard for an algorithm or even a human to correctly identify the number of pulses and their location when they are close together; especially if they are out of phase. The following example shows the same three transmitted signals in the presence of significant noise.

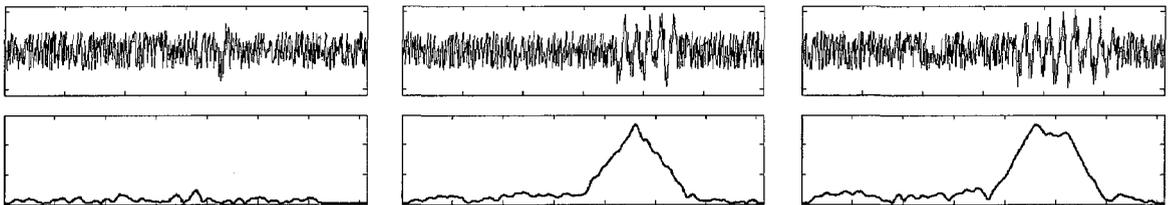


Figure 11: Matched filtering example (noise)

Once the noise is added it is nearly impossible to correctly identify the location of the short pulse; however, the longer pulse is still very easy to identify. The longer pulse remains easy to detect for two reasons. First of all it has more energy; five times as much as the short pulse. More importantly the long pulse has information encoded into it, namely its frequency, which helps to distinguish it from random noise.

VI. Pulse Compression

In Figure 10 the result of the autocorrelation function is an oscillating function containing several peaks and valleys. The peak with the greatest amplitude is referred to as mainlobe and all other peaks are referred to as sidelobes. These sidelobes are a direct result of the autocorrelation function, sometimes referred to as self noise, and are responsible for the degraded axial-resolution. To understand the nature of these sidelobes consider the autocorrelation of the long pulse. There are four positive sidelobes and five negative sidelobes on either side of the mainlobe. The positive sidelobes correspond to time-shifts where one, two, three, and four of the peaks are lined up. Similarly, the negative sidelobes correspond time-shifts where the peaks are lined up but out of phase (negative correlation). The goal of pulse compression is to craft the transmitted pulse so that it reduces the sidelobes produced by the matched filter. In other words, pulse compression encodes information in either the frequency or phase of a pulse so that slight time-shifts result in autocorrelation values that have a small magnitude. To better illustrate this point an example of pulse compression using chirp pulses is provided. The following example uses the same scales and processing methods as the example in the previous section; the first row is a single chirp pulse and the second row contains two overlapping chirp pulses.

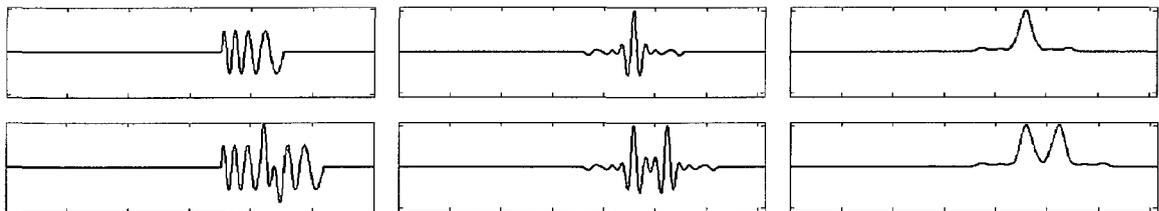


Figure 12: Matched filtering example (chirp)

A chirp pulse is simply an oscillating function whose frequency changes with time. The rate of change is typically a simple linear, quadratic, or exponential function; in the above example the rate of change is linear. The function that describes the rate of

change can be thought of as information that is embedded in the transmitted signal. The embedded information helps the autocorrelation function to distinguish between similar time-shifts and the actual time-shift; a fact that can be observed by the drastic reduction in sidelobes. The reduction greatly improves resolution in the time domain when compared to an uncompressed signal of similar length. Furthermore, the amplitude of the mainlobes is comparable to the amplitude of the mainlobe of the long pulse in the previous example. Thus, the chirp signal still provides significant improvements to the SNR as can be seen in the follow examples of the chirp signals in the presence of noise.

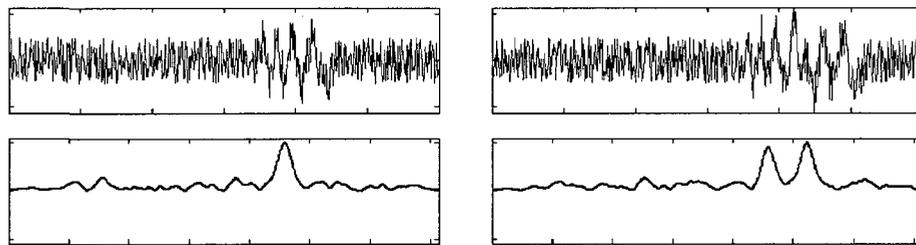


Figure 13: Matched filtering example (noisy chirp)

Other approaches to pulse compression such as Barker and Golay Codes encode binary information in the phase of the signals. Again, the binary codes are designed so that matched filtering results in pulse compression. In fact, Golay Codes can achieve perfect compression but they require multiple transmissions so they are affected by motion.

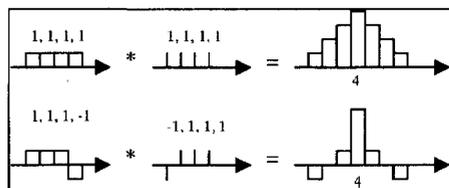


Figure 14: Barker encoding [7]

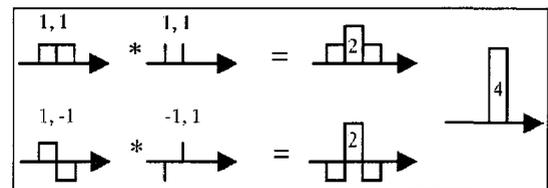


Figure 15: Golay encoding [7]

Figure 14 shows matched filtering for a signal that is not encoded (top) and a signal that uses Barker coding (bottom); the star is the convolution operator. Convolution of

the unencoded signal results in a signal with large sidelobe that cause poor axial resolution. On the other hand, convolution of the Barker coded signal results in a signal that has much smaller sidelobes and a mainlobe that has the same strength as the mainlobe in the unencoded signal. Figure 15 shows an example of a Golay code which requires multiple transmissions. Each pulse is matched filtered independently, then the results of filtering are combined using the principles of super positioning which results in perfect compression.

Practical Considerations for Ultrasonic Imaging

The theory behind pulse compression looks promising and it has been applied successfully in other disciplines such as radar and communications. However, there are some practical considerations unique to ultrasound that must be overcome. Issues such as frequency dependent attenuation reduce the quality of matched filtering, while hardware limits the generation of encoded pulses. This section will look at how the frequency spectrum of ultrasonic transducers affects the generation of coded pulses and how frequency dependent attenuation affects matched filtering.

The first problem stems from the frequency response of ultrasonic transducers. The frequency response of a typical transducer is bell shaped (Gaussian) and centered at some desired frequency. This has the adverse affect of limiting bandwidth and weighting chirp signals. It has even been argued that because of the high frequencies involved, transducers severely degrade phase modulated signals, like Barker and Golay codes, making them impractical for ultrasonic imaging [7]. On the other hand, the weighting actually helps to further reduce the sidelobes generated by linear chirps and is artificially introduced into some radar systems [12]. The weighting phenomenon is illustrated in Figure 16 and Figure 17 where the first pulse is used to excite a transducer and the second pulse is the pulse that would actually be emitted from the transducer.

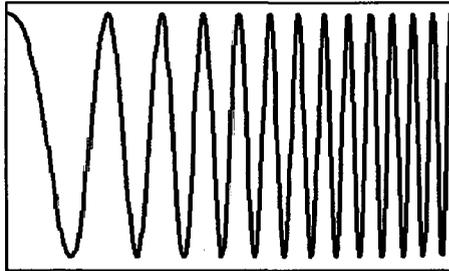


Figure 16: Chirp before transducer

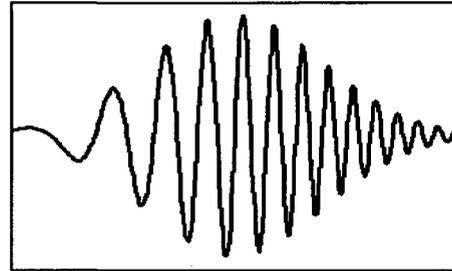


Figure 17: Chirp emitted from transducer

Since the pulse used to excite the transducer is quite different from pulse emitted from the transducer, and ultimately the pulse received by the transducer, it cannot be used for matched filtering. In order to understand the solution to this problem, first recall that convolution in the time domain is equivalent to multiplication in the frequency domain. Thus, matched filtering can be performed in the frequency domain by multiplying the frequency spectrum of the received signal with the frequency spectrum of the transmitted signal⁵. Thus, if the chirp pulse used excites the transducer's full range of frequencies, then the matched filter is simply the frequency response of the transducer [7].

Frequency dependent attenuation has also been shown to affect matched filtering [7]. Since higher frequencies are attenuated more than lower frequencies there is a downshift in the mean frequency of the received signal with respect to the transmitted signal.

5. The conjugate of a signal in the frequency domain is the equivalent to time-reversal in the time domain. Most papers talk about multiplying with the conjugate of the transmitted signal. This is left out because it has already been stated that the pulses in the received signal are time reversed from the transmitted pulse.

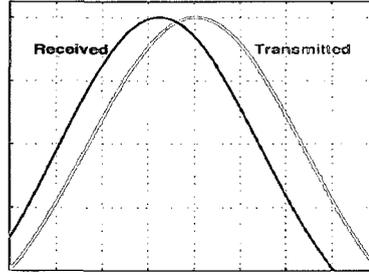


Figure 18: Mean frequencies [7]

The mean frequency of the propagating pulse decreases linearly with depth according to following function where f_0 is the original mean, B is the bandwidth, β is the frequency dependent attenuation factor, and z is depth [7].

$$f_{mean} = f_0 - (\beta B^2 f_0^2) z$$

Equation 7: Mean frequency shift

It has been suggested that this model could be used to create an array of depth dependent matched filters that could be used to compensate for the shift in mean frequency at different depths. However, this approach would be based on estimations and empirical data and would not help in situation where the coded signal is extremely sensitive to shifts in the mean frequency [13]. A more robust solution is to design a pulse that provides good pulse compression and is resilient to frequency dependent attenuation. To do this the ambiguity function, used in radar to combat the Doppler Effect, is used to determine the response of an uncompensated (with respect to shifts in the mean frequency) matched filter. The ambiguity function [7] is defined for complex signals in Equation 3, where τ time-delay, f_D is the shift in the mean frequency, $\mu(t)$ is the analytical representation of the signals (i.e. the Hilbert transform), and $*$ represent the complex conjugate (i.e. time-reversal).

$$X(\tau, f_D) = \int_{-\infty}^{\infty} \mu(t) \cdot \mu^*(t - \tau) \cdot \exp(j2\pi f_D t) dt$$

Equation 8: Ambiguity function [7]

This paper is only concerned with the use of the ambiguity function, for a detailed derivation of the ambiguity function and an in-depth discussion about its properties see [7]. For a specific fixed f_D , the ambiguity function returns the response of an uncompensated matched filter as a function of delay. To better understand this concept take a look at the graphs, mean shift vs. time delay, in Figure 19. The graphs on the left and right show the ambiguity function for a short unmodulated pulse and long unmodulated pulse respectively; similar to the pulses in Figure 10. First notice that when f_D is zero the results correspond with the results show in Figure 10. Also, notice that even the slightest shift in mean frequency drastically reduces the quality of matched filtering for the long pulse and that the short pulse can tolerate minor variations.

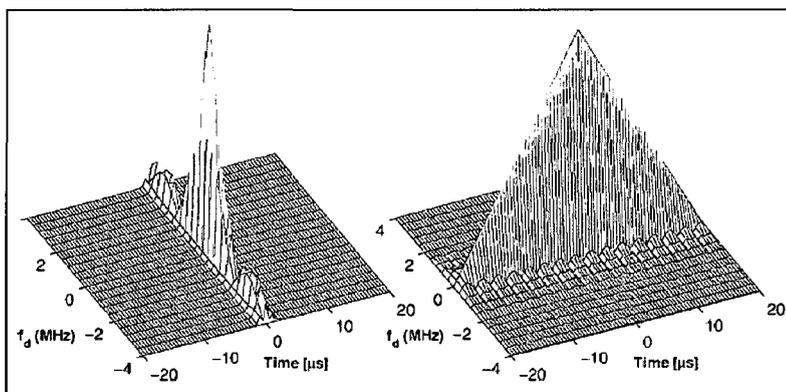


Figure 19: Graph of ambiguity function for short (left) and long (right) pulses [7]

An ideal pulse for ultrasonic imaging would be compressed along the time axis and wide along the frequency shift axis; much like the pulse on the right in Figure 19 if the axes were switched. A pulse with those characteristics would provide good pulse compression and would withstand the effects of frequency dependent attenuation. With this in mind, the graph of the ambiguity function for a chirp pulse is shown in Figure 20.

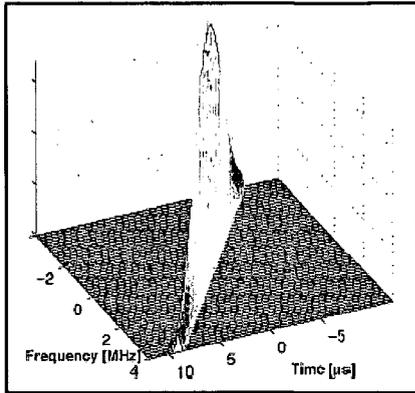


Figure 20: Ambiguity function of chirp [7]

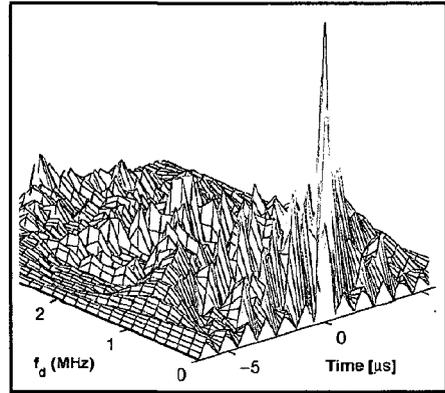


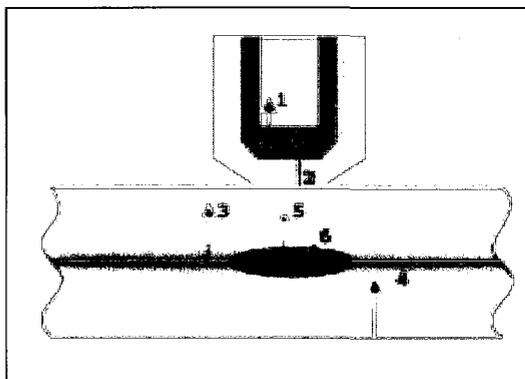
Figure 21: Ambiguity function of Baker code [7]

The chirp pulse is both compressed along the time axis and wide along the frequency shift axis but there is one problem; the function is not parallel to the time axis. As a result, the accuracy (not axial-resolution) of an uncompensated matched filter will degrade as the mean frequency drifts. Fortunately, the equations for mean frequency shift and the ambiguity function provide the tools necessary to compensate for this problem. Conversely, Figure 21 shows the graph of the ambiguity function for a 13-bit Barker code. When f_D is zero the graph shows a strong mainlobe with small sidelobes as expected. More importantly the graph shows that Barker codes are extremely sensitive to shifts in the mean frequency; yet another reason why phase encoded signals are not practical for ultrasonic imaging.

VII. ISWA Apparatus

Data Acquisition

A single element transducer is incorporated into the shank of one of the electrodes on a resistance spot welder. Water acts as a couplant by continuously flowing between the transducer and the weld cap. The transducer emits an unfocused plane wave that travels perpendicular to the weld cap. The wave is able to travel from the transducer into the copper weld cap and then into the plates that are being welded. Through the welding procedure up to six different interfaces may appear and cause part of the wave to be reflected.



- | |
|--|
| <ol style="list-style-type: none">1. water – copper2. copper – steel3. steel – steel4. steel – copper5. steel – liquid6. liquid – steel |
|--|

List 1: Interface Names

Figure 22: Close up

During the welding process the position, intensity, and existence of the interfaces will change. To monitor these changes, a-scans are collected at fixed time intervals during the welding process, typically every 2-4 ms. Once all the a-scans are collected they are converted to a two dimensional image called a b-scan. To create the b-scan, the a-scans are lined up vertically from left to right in the order that they were obtained. The first sample of each a-scan at the top of the image and the last sample of each a-scan is at the bottom. As a result, both the horizontal and vertical axes represent time. The vertical axis represents the time of flight (in μs) of the ultrasonic wave and the horizontal axis represents welding time (in ms). Lastly, each sample is converted to a

grey scale value based on its amplitude. It is common to use a non-linear conversion process in order to increase the resulting images contrast. Figure 23 is a schematic diagram showing how the interfaces change when a good weld is made. Figure 24 is an actual b-scan that was obtained during the formation of a good weld.

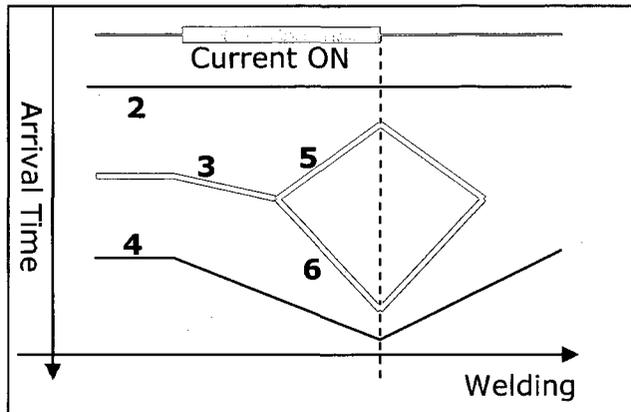


Figure 23: Schematic diagram [4]

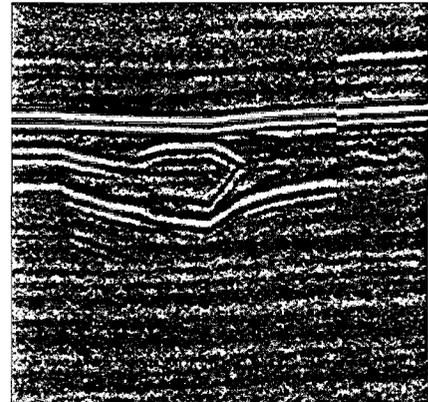


Figure 24: Actual b-scan

The transducer in the shank is connected to custom electronics that perform the functions of both the pulser and receiver. The pulser/receiver connects to a windows based computer via the parallel port interface and is controlled from software by calling functions in the provided dll. The board only returns a-scan data so the acquisition software is responsible for building the b-scans. The pulser/receiver also has a discrete input that can be used to receive a trigger signal. In order to achieve proper timing, the resistance spot weld controller must be connected to the input of the pulser/receiver and programmed to send a signal prior to welding. Again, it is the job of the acquisition software to listen for this signal and determine when to start acquiring the b-scan. It is also very common to provide the acquisition software with the time, relative to the start signal, when the current will be turned on and off. The information is very useful to the signal processing software and is embedded into b-scans when it is available.

Acoustic Wave Propagation

From an acoustical standpoint, the welding procedure produces a complex layered structure with properties that are constantly changing.

The layers formed by the water, copper, and steel severely limit how much acoustic energy makes it back to the transducer from the different steel interfaces. This can be seen in Figure 25 where the energy of an acoustic wave is traced as it propagates through the water, copper, and steel layers. By the time they make it back to the transducer, the reflections from the copper-steel and steel-steel interfaces contain less than 0.1 percent of the energy in the transmitted wave. This huge loss of energy is caused by the reflection and transmission coefficients of the mediums and would be even worse if other sources of attenuation were accounted for.

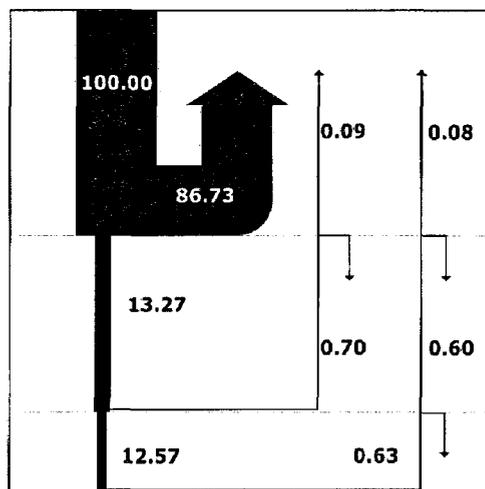


Figure 25: Reflections from layered water, copper, and steel

Figure 25 was generated according to the acoustic impedances of water, copper, steel, molten steel found in [9, 14]. The acoustic impedances are 14.80, 41.61, 46.00, and 36.92 kg/(m²-s) respectively. Using Equation 4 the reflection coefficients of water-copper, copper-steel, steel-steel, steel-liquid are found to be 86.73, 0.25, 0.00, and 1.19 percent respectively. These reflection coefficients are for ideal interfaces but the interfaces between the solids in this apparatus are not ideal; if they were ideal, the steel-

steel interface would not be visible in the b-scans. In order to account for this fact, 5.00 percent was added to the reflection coefficient of the copper-steel and steel-steel interfaces making them 5.25, and 5.00 respectively. This value was selected because it makes the reflection coefficients of these interfaces larger than the reflection coefficient of the steel-liquid interfaces but still much smaller than the reflection coefficient of the water-copper interface. However, the value used is not all that important; doubling it or halving it produces similar results. What is important is the fact that the water-copper interface is highly reflective, meaning the interface makes it really hard for acoustic energy to enter or exit the system, and the other interfaces are highly transmissive, meaning they reflect very little energy back towards the transducer.

The reflection and transmission coefficients of the different interfaces also change during welding because of the localized heat that is generated during welding. This is because acoustic impedance is a function of the sound velocity in a medium and the density of a medium both of which are temperature dependent [4]. Furthermore, the acoustic impedances of copper and steel change at different rates, so, during welding, it is possible for their acoustic impedances to become equal or for the inequality relationship between them to change. Since the transmission and reflection coefficients of an interface dictate how much energy is reflected back towards the transducer, the intensity of the different interfaces tends to change during welding; this can be observed in Figure 24. The copper-steel interfaces can even disappear or become inverted the acoustic impedances of the copper and steel become similar or the inequality relationship between them changes [4].

The localized heating and cooling also creates a temperature gradient along the axis in which the acoustic waves travel. The temperature is coolest near the water-copper interface and increases as the depth increases; i.e. distance from the water-copper interface. The gradient is symmetric, or follows a similar trend, about the steel-

steel interface and the maximum temperature is found near the center of the liquid nugget. Since the speed of sound is temperature dependent the velocity of the acoustic wave changes as it propagates through the different mediums. This phenomena was studied in [3] which calculated the change in acoustic velocity in a 4.5mm thick copper weld cap and 1.5mm thick steel plate during welding. Figure 26 shows the localized acoustic velocity, at the end of welding, as a function of depth in both the copper and steel. The sudden change in acoustic velocity in steel corresponds to the transition from solid to liquid steel.

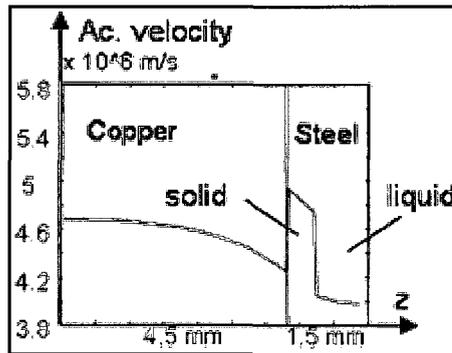


Figure 26: Localized acoustic velocity [3]

Lastly, the geometry of the liquid nugget is also vital to the propagation of acoustic waves. When the liquid nugget grows it will expand rapidly along the faying interface and then start to expand into the welded plates. Because of this, the liquid nugget is considered to be parallel to the other interfaces [3] and can be treated as another layer. This means that the liquid nugget will not introduce refraction into the system so reflections from its interfaces and the interfaces below it should be detectable. However, if the diameter of liquid nugget does not grow to be bigger than the diameter of the acoustic plane waves, then the liquid nugget will not form a simple layer. Instead the liquid nugget will form a layer that only reflects part of the plane wave. The area that is not reflected nugget continues to travel through the solid steel until it is reflected by the steel-steel interface a few microseconds later. As a result both the both the liquid

nugget and the steel-steel interfaces will be visible in the resulting b-scan; this can be seen in Figure 27. Notice that the intensity of the steel-steel interface has decreased, this happens because a smaller percentage of the acoustic wave is reflecting off it once the liquid nugget forms.

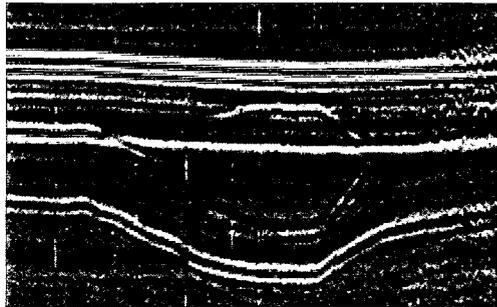


Figure 27: steel-steel interface throughout b-scan

Interpreting Data

The signal processing software has two jobs when analysing a b-scan. The first job is to detect the pattern that is produced by all the interfaces. The goal is to reduce the b-scan to a simple geometric structure, similar to the schematic diagram shown in Figure 23, which can be used to estimate weld quality. However, the SNR of a b-scan limits the ability of signal processing software to correctly detect and extract this pattern. The red lines in Figure 28 and Figure 29 show the patterns that signal processing software should be able to extract from the b-scan.

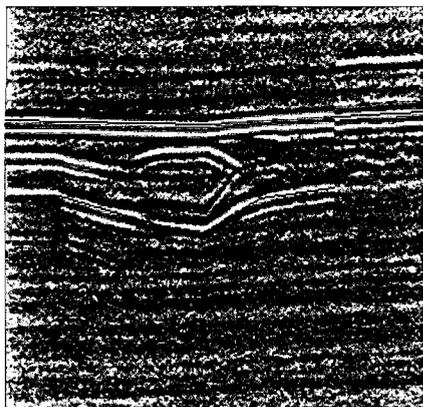


Figure 28: complete information

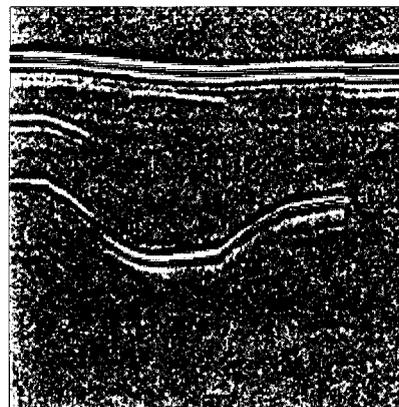


Figure 29: partial information

The second job is to examine the extracted pattern and to state the quality of the weld. The time of flight, penetration of liquid steel, presence of steel-steel interface, and the slope of steel-copper interface are all strong indicators of weld quality.

The heat created during welding decreases the speed of sound in the steel plates and causes them to expand. These two factors increase the time it takes for the ultrasonic waves to travel from the copper-steel interface to the steel-copper interface. Measuring the time of flight between these two interfaces when the current is turned off provides information about the combined effect of those two factors. Furthermore, this time of flight measurement has shown a strong correlation with weld diameter [4].

The reflection from the interfaces between the solid and liquid steel can be used to calculate the penetration of the liquid steel into the plates. The maximum penetration of the liquid pool into the steel plates occurs at the final moment of heating; right before the current is turned off. Thus, the time of flight between the steel-liquid and liquid-steel interface and the time of flight between the copper-steel and steel-copper interfaces are measured when the current is turned off. The percentage of liquid steel is a strong indicator of weld quality; good welds tend to have a value above 70 percent [4].

The presence of a steel-steel interface at or near the end of welding indicates that very little fusion occurred during welding. When a good weld is formed the two plates melt and mix together ultimately eliminating the steel-steel interface. Thus, the presence of a steel-steel interface near the end of welding indicates that the weld is a cold/stick weld, meaning no nugget was formed and the two plates should come apart with very little force.

Because of the increasing delay in time of flight during welding, the steel-copper interface moves away from the copper-steel interface while current is applied. However, as the pool of molten steel grows the force of the electrodes compresses it and allows the electrodes to move closer together. As the electrodes move closer together the

steel-copper interface starts to move towards copper-steel interface (i.e. its slope becomes negative). The presence of a negative slope before the current is turned off is a strong indicator of good weld quality. Figure 29 and Figure 28 show b-scans with and without negative slopes prior to turning the current off.

Since there are quite a few indicators of weld quality, it is not necessary to detect and extract all the interfaces from a b-scan. It is still possible to evaluate weld quality even if some of the interfaces can not be detected; like in Figure 29. However, it is very important that interfaces are not falsely detected; i.e. detected in the wrong spot or detected when they are not present. Doing so will cause the software to interpret a pattern that does not really exist and ultimately produce inaccurate evaluations. Thus, correctly detecting all the interfaces is of the utmost importance.

Signal to Noise Ratio

The section on acoustic wave propagation explained how the layered structure of the apparatus prevents the vast majority of the transmitted signal from making it back to the transducer. Since the received signals are so weak, the equipment introduces a lot of noise during the amplification and digitization processes. Thus, the combined effect of the layered structure in which the acoustic waves travel and sensitivity of the equipment used for transmitting and receiving the waves is believed to be the main source of noise.

The condition of the weld caps and the alignment of the electrodes also contribute to the SNR of a-scans. If the electrodes are not perfectly normal to the steel plates, a common occurrence in automotive plants, then the acoustic waves will experience some minor refraction at the different interfaces. This refraction changes the direction of the wave and can prevent a portion of it from being reflected back towards the transducer. Since, less acoustic energy makes it back to the transducer, the interfaces in the b-scan will be less intense.

During the welding process the tips of the electrodes are subjected to high heat and pressure that slightly deform the weld caps. The deformations accumulate over time and after several hundred welds are made the weld caps need to be dressed (restored to their original shape) or replaced. The deformations in degraded weld caps cause them to have higher attenuation coefficients which ultimately leads to decreased SNR; this can be seen in Figure 30 and Figure 31.

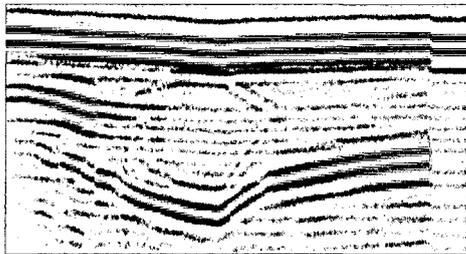


Figure 30: b-scan taken with new caps



Figure 31: b-scan taken after 150 welds

The low SNR of the weld signatures greatly impacts how the data can be processed. Because of the low SNR, the reflections from the different interfaces cannot be detected reliably by signal processing algorithms that operate on individual a-scans; the one exception is the copper-steel interface. This point is illustrated by examining the a-scans shown in Figure 33 and Figure 34 both of which come from the typical b-scan found in Figure 32. When looking at the b-scan, it is clear that the a-scan in Figure 33 contains reflections from four different interfaces but only two, maybe three, of the reflections can actually be seen in the a-scan. In addition, the peak in the center of the a-scan is noise but its amplitude is significant when compared to the other reflections. In the a-scan shown in Figure 34 the reflection from the steel-copper interface is only slightly larger than the random peaks formed by noise.

Fortunately, when a b-scan is formed all the weak reflections from the different interfaces are aligned so they become much more visible. This fact is exploited in [4] by the use two dimensional image processing algorithms that look at the b-scan as a whole in order to extract the different interfaces. However, this approach can not be used until welding is complete which means it cannot be used for any applications that require real-time feedback.

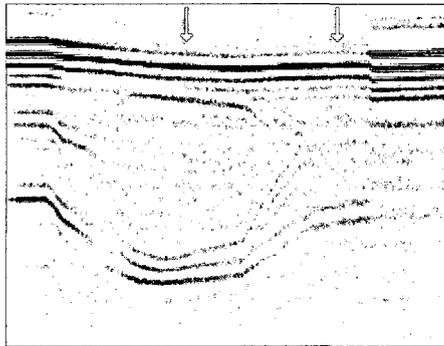


Figure 32: typical b-scan

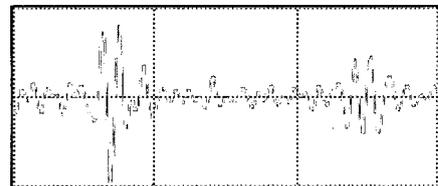


Figure 33: typical a-scan (1st arrow)

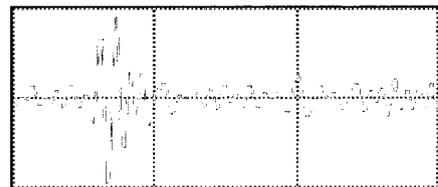


Figure 34: typical a-scan (2nd arrow)

VIII. Experimental ISWA Apparatus

Data Acquisition

In order to excite the transducer with a chirp signal, the pulser/receiver that is used for ISWA had to be modified. The modifications, shown in Figure 35, simply bypass the pulser by disconnecting the trigger signal and pulser output. All other electronics remain the same making the modifications completely transparent to the software.

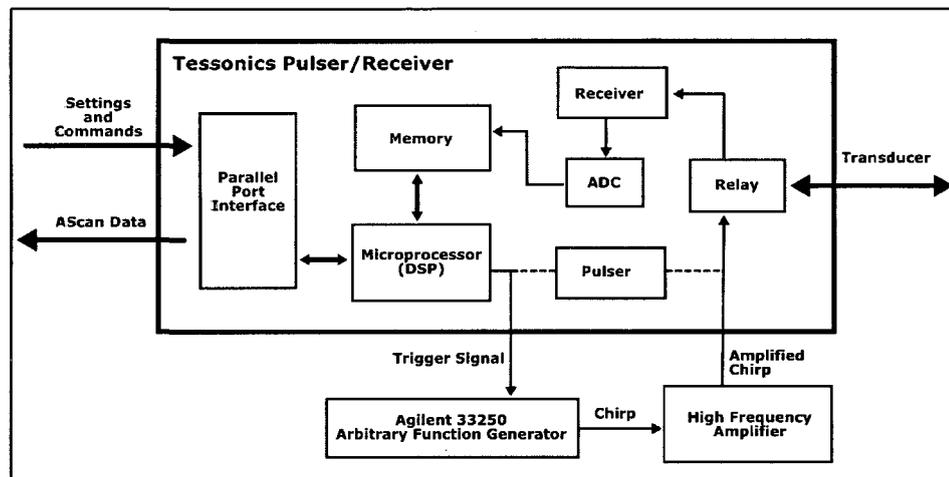


Figure 35: Modifications to pulser/receiver

An Agilent 33250 Arbitrary Function Generator (AFG) was used as the external pulser. The AFG allows waveforms to be loaded on to the device through a serial port interface. A simple Matlab script was used to calculate the points and the period of the chirp signals used in the experiments. The script took as parameters the upper and lower frequency limits and the number of cycles that would be present in the resulting signal.

The one pitfall of the AFG is that its maximum output was 10Vpp and ultrasonic transducers are typically excited with much higher voltages. In an attempt to compensate for the low potential two different high frequency amplifiers were used to amplify the output from the AFG. Unfortunately, the amplifiers introduced more problems than they solved, so in the end, they were left out the experimental apparatus.

Interpreting Data

An additional stage must be added to the processing algorithm that performs matched filtering on the acquired b-scan prior to searching for the different interfaces. However, in order to perform matched filtering a template signal is needed. The reflection from the copper-air interface, reflection 2 when the weld gun is open, is used as the template signal. This reflection is used because it contains the transfer function for the majority of the system; namely the transducer and the water and copper layers. To measure this reflection accurately a b-scan is taken when the weld gun is open. A bandpass filter is then applied to each a-scan to remove the unwanted frequencies. After this is done, the a-scans are averaged together and the samples containing the chirp signal are saved for later use. At this time, a human must specify which samples contain the chirp signal. An example of the input to and output from this process is provided in Figure 36 and Figure 37 respective. From the shape of the measured template signal, it is clear that the experimental setup is consistent with the practical considerations discussed in the section VI.

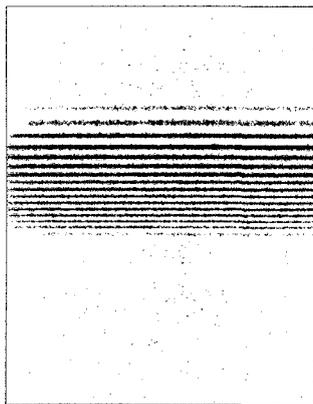


Figure 36: b-scan of open weld gun

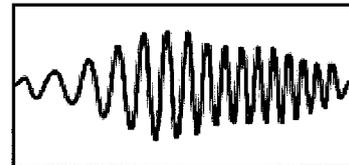


Figure 37: Measured template signal

Additional Software

Software that could measure the template signal, perform matched filtering, control the hardware, manage b-scans, and help with experimentation in general had to be developed; the software is shown in Figure 38 and some interesting code segments can be found in the appendix.

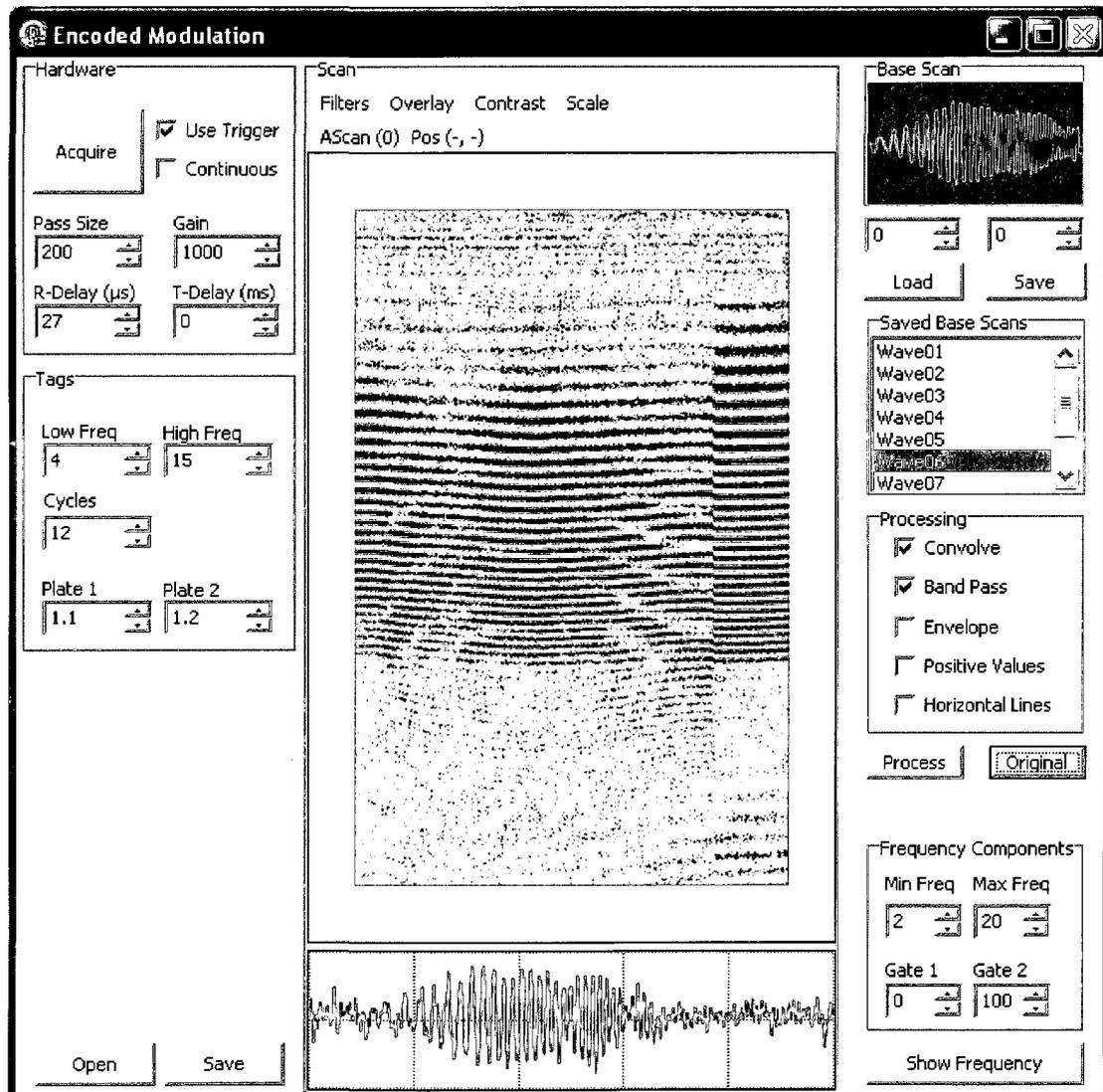


Figure 38: Software used for experimentation

The 'Hardware' section is used to control the parameters that are modified on a regular basis. Parameters that are not modified on a regular basis are hard coded into the software.

Values in the 'Tags' section are embedded into the b-scan files which allows additional information, such as the thickness of the steel plates, to be saved with each b-scan. When a b-scan is opened from disk, the tag values are displayed in the 'Tags' section so the conditions in which the b-scan was obtained are readily available.

The 'Base Scan' and 'Saved Base Scans' sections are used to measure, view, store, and retrieve template signals. The checkboxes in the 'Processing' section are used to specify which filters are applied when the 'Process' button is clicked. If convolution is selected, matched filtering will be performed with the template signal shown in the 'Base Scan' section.

Pressing the 'Show Frequency' button will open a new window that displays the frequency components present in the b-scan. The 'Frequency Components' section provides some parameters that control how the frequency components are calculated.

Lastly, the 'Scan' section is responsible for displaying and manipulating b-scans. The dropdown menus at the top provide some useful options that control how a b-scan is shown. The individual a-scans that make up the b-scan are displayed in the oscilloscope at the bottom; the visible a-scan is selected by clicking on the b-scan.

IX. Experimental Results

Figure 39 and Figure 40 show a b-scan, before and after matched filtering, that was acquired using a chirp signal. Before matched filtering, the reflections from the different interfaces overlap and create patterns of interference that change with weld time. After matched filtering, the different interfaces are easily distinguishable and the resulting image looks just like a b-scan obtained using a single pulse. This result is quite significant because it proves that effective pulse compression can be obtained with the experimental apparatus.

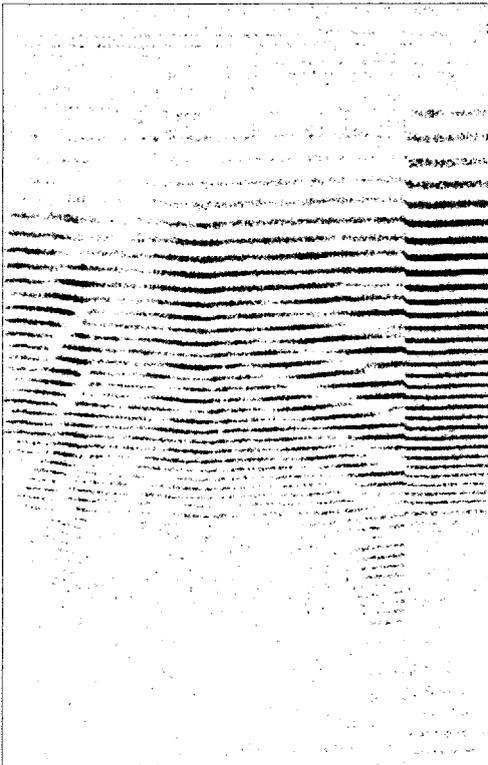


Figure 39: b-scan acquired with chirp

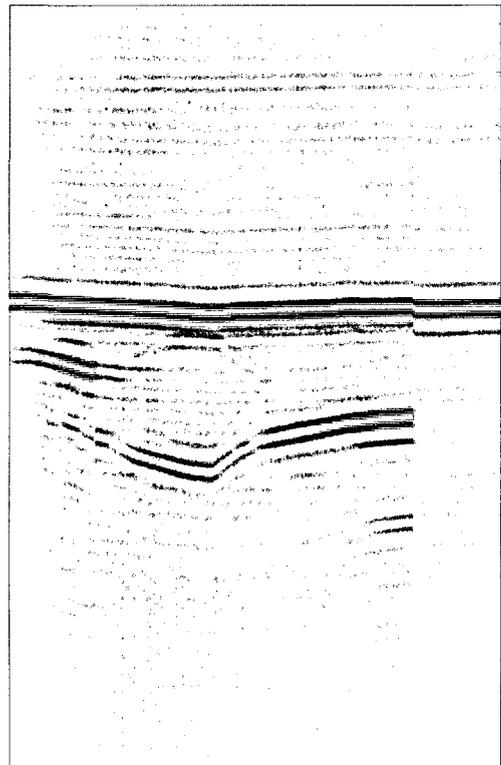


Figure 40: b-scan after matched filtering

Amplified Signals

The better of the two amplifiers was made by Electronic Navigation Industries (ENI 320L) and was capable of amplifying an input signal 50 dB with a maximum output of 20

watts but the input signal had to be less than 1 V_{PP} and could only contain frequencies from 250 kHz to 110 MHz.

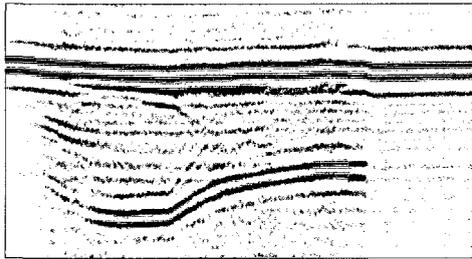


Figure 41: amplifier not used

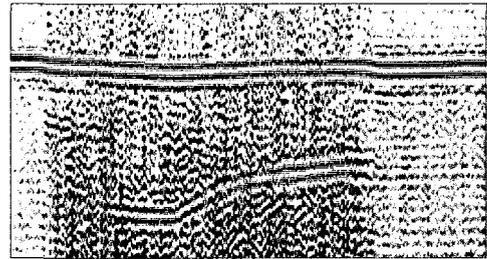


Figure 42: amplifier used

Figure 41 and Figure 42 show b-scans obtained with and without using the amplifier. The b-scan obtained without the amplifier is clearly better than the one obtained with the amplifier. Figure 43 and Figure 44 show the last a-scan in each of the b-scans before matched filtering is performed; both a-scans uses the same scale. Note that by the time the last b-scan is obtained, the jaws on the welder are already open so only the reflection from the copper-air interface is visible. In Figure 43 the reflected chirp signal is easily distinguishable from the noise. However, in Figure 44, the reflected chirp signal is not easily distinguishable from the noise and there is also a lot of noise at the beginning and end of the signal where no reflections would exist. The reasons why the attempts to amplify the output from the signal generator have failed are still unclear. However, it's likely that the amplifiers are picking up a weak output from the signal generator and electromagnetic fields from the welding process.

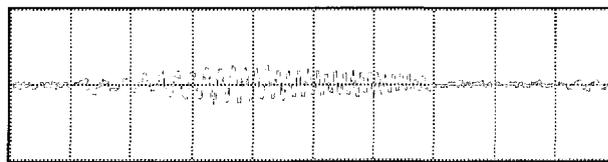


Figure 43: last a-scan in unamplified b-scan

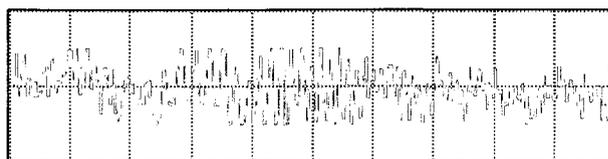


Figure 44: last a-scan in amplified b-scan

Chirp Optimization

The range of frequencies and the period of the chirp signal that would provide the best SNR and resolution for the given apparatus had to be determined before a comparison could be performed. Since the frequency response of the transducer dictates the range of frequencies that can be used, the upper and lower limits of the chirp signal were simply matched to the transducer; the optimal range was from 4 to 15 MHz. Finding the optimal length of the chirp pulse was not as obvious because a good balance between SNR and resolution (compression) needed to be achieved. To determine the best length for the chirp pulse several scans were taken with different pulse lengths; some examples are shown in the following figures.

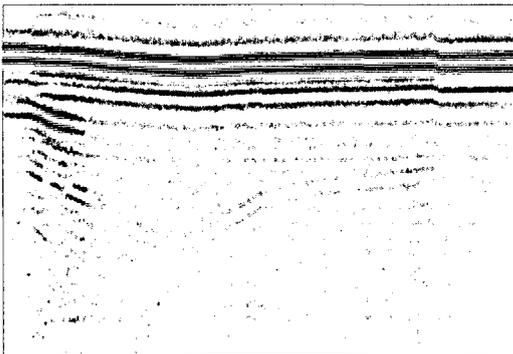


Figure 45: 5 complete cycles (526ns)

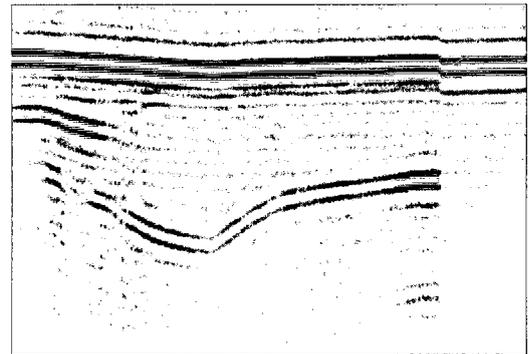


Figure 47: 15 complete cycles (1578ns)

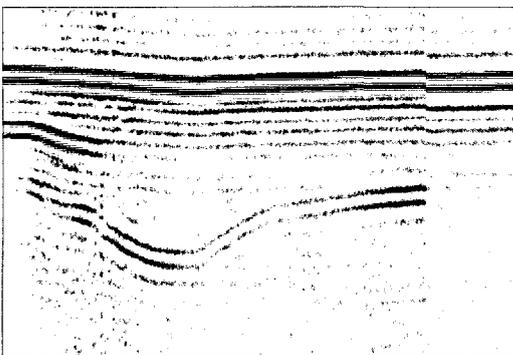


Figure 46: 10 complete cycles (1053ns)

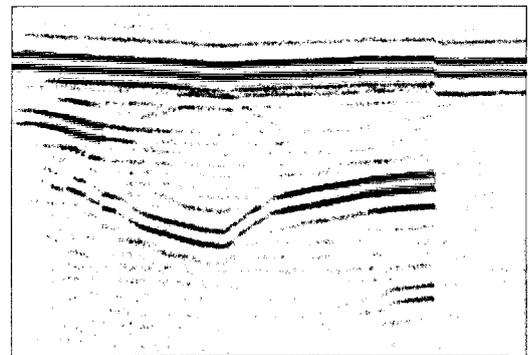


Figure 48: 30 complete cycles (3158ns)

As expected, the SNR of the b-scan increased as the length of the chirp pulse, i.e. its energy, increased. In addition, the compression worked so well that there was no noticeable decrease in resolution as the length of the pulse increased. Because the

compression worked so well, the resolution did not even have to be considered when choosing the optimal length. In the end, a chirp pulse containing 30 complete cycles and having a period of 3158ns was deemed best. This pulse length provided better SNR than the shorter pulses and it was still short enough to be practical. It was practical because the reflection from the water-copper interface did not overlap with the reflection from the copper-air interface making it possible to measure the template signal using the method described in section VIII and because it did not cause the length of the b-scans to grow too much; look at unused space near the top and bottom of Figure 40.

Comparison with Pulse Mode

Several controlled experiments were performed in order to compare the quality of b-scans obtained using the chirp pulse to those obtained using single pulse. This was done by establishing a set of welding conditions, then making one weld using the experimental apparatus, and then making another weld while using the existing apparatus. Several b-scans were collected for each set of welding conditions so any external factors that may have influenced b-scan quality could be weeded out. The different sets of weld conditions were based off of scenarios that are commonly encountered in automotive plants. They consisted of various weld schedules, plate thickness combinations, and weld cap qualities. The schedules were chosen so that good, undersized, and welds with expulsion were produced on various plate combinations. Weld caps with varying degrees of usage were tested because degraded weld caps tend to decrease the SNR of b-scans. Lastly, no attempts were made to compensate for the single pulse's large amplitude because the goal of this work is to improve the existing apparatus.

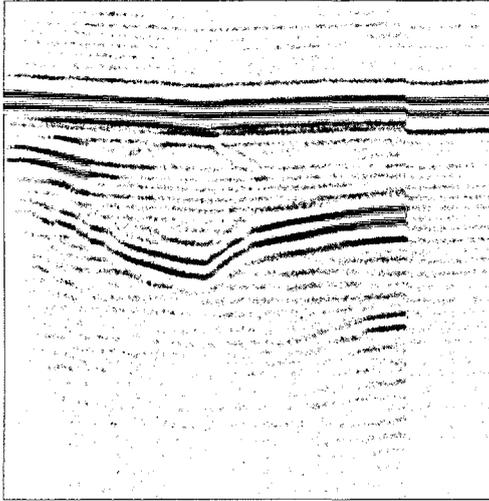


Figure 49: chirp (30 cycles, 4 to 15 MHz)

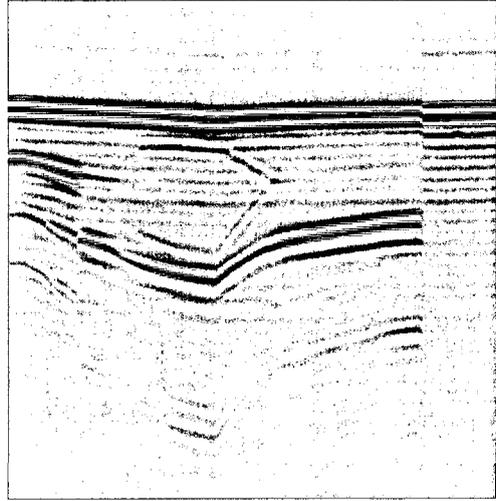


Figure 50: Single Pulse

The b-scans obtained with the single pulse were slightly better than those obtained with the chirp pulse. In most cases the images obtained with the single pulse had better contrast and less horizontal lines running through them; these difference can be seen in Figure 49 and Figure 50. However, the fact that comparable b-scans could be obtained with the chirp pulse, even though its amplitude was about 5 percent of the single pulse's amplitude, is quite impressive. Moreover, these results suggest that the SNR of b-scans can be improved by chirp encoding the ultrasonic signals. However, the amplitude limitations need to be addressed before this can be confirmed.

X. Conclusion

The goal of this research was to determine if the SNR of b-scans obtained during spot welding could be improved by encoding the transmitted ultrasonic signals. Reviews of completed works lead to the conclusion that a chirp signal was the best method for encoding ultrasonic signals. The apparatus for ISWA was modified to allow the transmission of encoded ultrasonic signals. Unfortunately, the equipment used to generate the encoded signals was not as powerful as the original pulser so the amplitude of the single pulses was approximately 20 times greater than the amplitude of the encoded signals. There were two attempts to create a fair comparison by amplifying the encoded signals but these attempts failed so the encoded signals are at a disadvantage in this work.

The range of frequencies and the period of the chirp signal used for encoding were optimized for the transducer used in the experiments. The range of frequencies used, 4-15 MHz, was selected to match the frequency response of the transducer. The period used, 3158ns, was selected because it provided significant improvements in SNR when compared to shorter periods and because it was more practical than longer periods.

Scans taken using the optimal chirp pulse were compared to scans taken using a single pulse. Overall, the scans taken with a single pulse tend to have better contrast and less noise. However, the scans obtained with the chirp pulse were similar in quality despite their limited amplitude. These results suggest that the SNR of b-scans can be improved by encoding the transmitted signal but the apparatus must be improved before this can be confirmed. Future work in this area should concentrate on improving the apparatus so that a fair comparison can be performed.

References

1. L. Barsanti, Tessonics Automotive NDT Technologies, *Associação Brasileira de Ensaio não Destrutivos E Inspeção (ABENDE) Gathering*, Sao Paulo, Brazil, 2008.
2. R. Maev, A. Chertov, L. Barsanti, G. Shu, Real-Time Quality Monitoring of Resistance Spot Weld Using Integrated Ultrasonic Weld Analyzer (RIWA), *ANST topical conference – Automotive Industry Advancements with NDT*, Dearborn, Michigan, 2007.
3. A. Chertov, R. Maev, A One-Dimensional Numerical Model of Acoustic Wave Propagation in a Multilayered Structure of a Resistance Spot Weld, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2005.
4. A. Chertov, Development of the New Physical Method for Real Time Spot Weld Quality Evaluation Using Ultrasound, *PhD Dissertation*, University of Windsor, 2007.
5. H. Zhang, Resistance welding: fundamentals and applications, *CRC Press*, 2006.
6. S. Smith, The scientist and engineers guide to digital signal processing 2nd Edition, *California Technical Publishing*, 1999.
7. T. Misaridis, J. Jensen, Use of Modulated Excitation Signals in Medical Ultrasound. Part I: Basic Concepts and Expected Benefits, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2005.
8. D. Ensminger, F. Stulen, Ultrasonics: data, equations, and their practical uses, *CRC Press*, 2008.
9. NDT Resource Center, Basic Principles of Ultrasonic Testing, <http://www.ndt-ed.org>.
10. Physik Instrumente, Fundamentals of Piezoelectricity, http://www.physikinstrumente.com/en/products/piezo_tutorial.php, 2008.
11. M. Ducan, Real-time analytical signal processor for ultrasonic non-destructive testing, *IEEE Instrumentation and Measurement Technology Conference*, 1990.
12. C. Wolff, Radar Basics, www.radartutorial.eu, 2008.
13. T. Misaridis, M. Pedersen, and J. Jensen, Clinical use and evaluation of coded excitation in B-Mode images, *IEEE Ultrasonic Symposium*, 2000.
14. N. Mountford, G. Ducan, Testing of Liquid Melts and Probes for use in Such Testing, *World Intellectual Property Organization*, Patent Filing WO/1985/005449, 1985.
15. S. Narasimhan, S Veena, Signal Processing: Principles and Implementation, *National Aerospace Laboratory*, Bangalore, India, 2005.

Appendix

Some of the following code uses the FFTW library (www.fftw.org) to calculate the both the Fourier Transform and Inverse Fourier Transform. Functions provided by the FFTW library are prefixed with `fftw`. In addition, most of code is belongs to larger objects that are not included here. Variables belonging to an object are not defined in the visible code but can be easily spotted as they are prefixed with an `F`.

Convolution

The following code is from a c++ dll I wrote to emulate the `conv` function from Matlab. To calculate the convolution, this code multiplies the inputs in the frequency domain. It does this because it is faster, $O(n \log n)$, than actually computing convolution in the time domain $O(n^2)$.

```
void __export __stdcall conv(ConvParams* params) {
    // indicies for complex numbers
    const int RE = 0;
    const int IM = 1;

    // Multiplication in frequency domain is equivalent to convolution in
    // time domain.
    // Converting to frequency domain and back takes  $O(n \log n)$  time.
    // Convolution in time domain takes  $O(n^2)$  time.

    int newLength = params->inputSize + params->convSize - 1;
    fftw_complex *conv, *input, *output;
    fftw_plan planForward, planBackward;

    conv = (fftw_complex*)fftw_malloc(sizeof(fftw_complex) * newLength);
    input = (fftw_complex*)fftw_malloc(sizeof(fftw_complex) * newLength);
    output = (fftw_complex*)fftw_malloc(sizeof(fftw_complex) * newLength);
    planForward = fftw_plan_dft_1d(newLength, input, output, FFTW_FORWARD,
    FFTW_ESTIMATE);
    planBackward = fftw_plan_dft_1d(newLength, input, output,
    FFTW_BACKWARD, FFTW_ESTIMATE);

    try {
        // Convert the convolution array to frequency domain
        // only do this once
        for (int i = 0; i < newLength; i++) {
            if (i < params->convSize) {
                input[i][RE] = params->conv[i];
                input[i][IM] = 0;
            } else {
                input[i][RE] = 0;
                input[i][IM] = 0;
            }
        }
        fftw_execute(planForward);
        MoveMemory(conv, output, newLength * sizeof(fftw_complex));

        // Conolve each input with convolution array
        for (int n = 0; n < params->inputCount; n++) {
```

```

// get frequency domain representation of input
for (int i = 0; i < newLength; i++) {
    if (i < params->inputSize) {
        input[i][RE] = params->inputs[n * params->inputSize + i];
        input[i][IM] = 0;
    } else {
        input[i][RE] = 0;
        input[i][IM] = 0;
    }
}
fftw_execute(planForward);

// multiply with frequency domain of convolution array
for (int i = 0; i < newLength; i++) {
    input[i][RE] = output[i][RE] * conv[i][RE]
        - output[i][IM] * conv[i][IM]
        ;
    input[i][IM] = output[i][RE] * conv[i][IM]
        + output[i][IM] * conv[i][RE]
        ;
}

// get time domain representation of convolution
fftw_execute(planBackward);

// copy to output
for (int i = 0; i < newLength; i++) {
    params->outputs[n * newLength + i] = output[i][RE] / newLength;
}
}
} catch (...) {
    // ensure clean up happens
}

// Clean up
fftw_destroy_plan(planForward);
fftw_destroy_plan(planBackward);
fftw_free(conv);
fftw_free(input);
fftw_free(output);
}

```

Matched Filtering

The following code takes a waveform as a parameter and uses it as the template signal for matched filtering. The conv function, shown above, is used to convolve the template signal with each individual a-scan.

```

procedure TBScanProcessor.Convolve(AWaveform: TSingleDynArray);
var
    Params: TConvParams;
    Input, Output: TSingleDynArray;
    I, N: Integer;
    Mx: Single;
begin
    N := Length(AWaveForm);

```

```

SetLength(Input, AScanLength * AinB);
SetLength(Output, (AScanLength + N - 1) * AinB);
for I := 0 to AinB - 1 do
  Move(FScanData[I, 0], Input[I * AScanLength], SizeOf(Single)*AScanLength);

Params.inputs := @Input[0];
Params.conv := @AWaveform[0];
Params.outputs := @Output[0];
Params.inputSize := AScanLength;
Params.inputCount := AinB;
Params.convSize := N;

Conv(@Params);

for I := 0 to AinB - 1 do
  Move(Output[(AScanLength + N - 1) * I] + N div 2], FScanData[I, 0],
SizeOf(Single) * AScanLength);

// normalize result
for I := 0 to AinB - 1 do
  begin
    Mx := 0;
    for N := 0 to AScanLength - 1 do
      if Abs(FScanData[I, N]) > Mx then
        Mx := Abs(FScanData[I, N]);
      for N := 0 to AScanLength - 1 do
        FScanData[I, N] := FScanData[I, N] / Mx;
      end;
    end;
end;

```

Template Signal

This is the code for extracting a template signal from a b-scan. It takes the range of a-scan samples that contain the template signal as a parameter.

```

procedure TForm1.MeasureTemplate(AStart, AEnd: Integer);
var
  BSP: TBScanProcessor;
  I, J: Integer;
  Tmp: TScanData;
begin
  BSP := TBScanProcessor.Create(FBScan);
  BSP.BandPass(FBScan.LowFreq, FBScan.HighFreq);
  try
    SetLength(FBaseWave, AEnd - AStart + 1);
    FillChar(FBaseWave[0], Length(FBaseWave) * SizeOf(Single), 0);

    // add relevant part of a-scans together
    for I := 0 to BSP.AinB - 1 do
      begin
        Tmp := BSP.GetAScan(I);
        for J := 0 to Length(FBaseWave) - 1 do
          FBaseWave[J] := FBaseWave[J] + Tmp[AStart + J];
        end;

        // average result
        for I := 0 to Length(FBaseWave) - 1 do
          FBaseWave[I] := FBaseWave[I] / BSP.AinB;
      finally
        BSP.Free;
    end;

```

```

end;
end;

```

Greyscale Conversion

The following code is used to enhance contrast when creating greyscale b-scans. The only parameter is an expansion factor which controls the density of the grey zone. The larger the expansion the less dense the grey zone becomes meaning the samples map to values close to black or white. Expansion factors less than or equal to one are ignored and a standard linear conversion is used.

```

procedure TBScanProcessor.Expand(AFactor: Single);
var
  I, J: Integer;
  Normalizer: Single;
begin
  if AFactor <= 1 then
    Exit;

    Normalizer := 1/arctan(AFactor);
    for I := 0 to Length(FScanData) - 1 do
      for J := 0 to Length(FScanData[I]) - 1 do
        FScanData[I,J] := Max(-1, arctan(AFactor*FScanData[I, J]) * Normalizer);
    end;

```

Bandpass Filtering

The following code takes as parameters the low and high frequency limits to be used by the bandpass filter. It then, eliminates frequencies outside of those limits from each individual a-scan.

```

procedure TBScanProcessor.BandPass(ALowFreq, AHighFreq: Single);
var
  FreqResolution: Single;
  LowIndex, HighIndex, Padding: Integer;
  FFTData: TComplexDynArray;
  PlanForward, PlanBackward: Pointer;
  I, J, N: Integer;
  Mean: Single;
begin
  Padding := FOriginal.AScanLength * 2;
  FreqResolution := SAMPLE_FREQ / (FOriginal.AScanLength + Padding);
  LowIndex := Floor(ALowFreq / FreqResolution);
  HighIndex := Ceil(AHighFreq / FreqResolution);

  SetLength(FFTData, FOriginal.AScanLength + Padding);
  PlanForward := fftwf_plan_dft_1d(Length(FFTData), @FFTData[0], @FFTData[0],
  FFTW_FORWARD, FFTW_MEASURE);
  PlanBackward := fftwf_plan_dft_1d(Length(FFTData), @FFTData[0], @FFTData[0],
  FFTW_BACKWARD, FFTW_MEASURE);

  try
    for I := 0 to FOriginal.AInB - 1 do
      begin
        // convert to frequency domain
        Mean := 0;
        for J := 0 to FOriginal.AScanLength + Padding - 1 do

```

```

begin
  if J < FOriginal.AScanLength then
    begin
      FFTData[J].Re := FScanData[I][J];
      FFTData[J].Im := 0;
      Mean := Mean + FScanData[I][J];
    end
  else
    begin
      FFTData[J].Re := Mean / FOriginal.AScanLength;
      FFTData[J].Im := 0;
    end;
  end;
  fftwf_execute(planForward);

  // eliminate unwanted frequencies
  for J := 0 to LowIndex do
    begin
      FFTData[J].Re := 0;
      FFTData[J].Im := 0;
    end;

  for J := HighIndex to Length(FFTData) - 1 do
    begin
      FFTData[J].Re := 0;
      FFTData[J].Im := 0;
    end;

  // convert back to time domain
  fftwf_execute(planBackward);
  N := Length(FFTData);
  for J := 0 to FOriginal.AScanLength - 1 do
    FScanData[I][J] := FFTData[J].Re / N; // must normalize result
  end
finally
  // Clean up
  fftwf_destroy_plan(planForward);
  fftwf_destroy_plan(planBackward);
end;
end;

```

Acquire b-scan

The following code controls the pulser/receiver (FController) during the acquisition of b-scans. It uses the computers high precision counter and the input on the pulser/receiver (FController.CheckButtonDown) to control timing.

```

// FStartDelay > 0: bscan is acquired ADelay ms after the trigger signal
// FStartDelay = 0: bscan is acquired when trigger signal is received
// FStartDelay < 0: bscan is acquired immediately
//
// FStopDelay > 0: stop AStopDelay ms after the trigger signal is turned off
// FStopDelay = 0: stop immediately after the trigger signal is turned off
// FStopDelay < 0: acquire exactly AinB scans
function TBScanGrabber.Acquire(out AErrors: Integer): TBScan;
const
  TIME_OUT = 5; // in seconds
var
  CurrentTime, TargetTime: Int64;
  I, Count, StopScans: Integer;

```

```

Stopping: Boolean;
begin
  // Lock Controller
  FController.Lock;
  try
    if not FController.Initialized then
      begin
        AErrors := ceUninitialized;
      end;

      // Configure the controller
      FController.AScanSettings.PassSize :=
        FBScanSettings.AScanSettings.PassSize;
      FController.AScanSettings.RegDelay :=
        FBScanSettings.AScanSettings.RegDelay;
      FController.AScanSettings.SuperSampling :=
        FBScanSettings.AScanSettings.SuperSampling;
      FController.AScanSettings.SumCount :=
        FBScanSettings.AScanSettings.SumCount;
      FController.AScanSettings.Gain :=
        FBScanSettings.AScanSettings.Gain;
      FController.AScanSettings.PassDelay :=
        FBScanSettings.AScanSettings.PassDelay;
      FController.AScanSettings.ByteSample :=
        FBScanSettings.AScanSettings.ByteSample;
      FController.SetAScan;

      // Figure out how many ticks we need to wait between scans
      Count := Round(FCounterFreq / 1000 * FBScanSettings.Interval);

      // initialize errors
      AErrors := 0;

      // How many scans are needed for a triggered stop; used if FStopDelay >= 0
      StopScans := Round(FStopDelay / FBScanSettings.Interval);
      Stopping := False;

      // wait for signal
      QueryPerformanceCounter(CurrentTime);
      TargetTime := CurrentTime + FCounterFreq * TIME_OUT;
      if FStartDelay >= 0 then
        begin
          while not FController.CheckButtonDown and (CurrentTime < TargetTime) do
            QueryPerformanceCounter(CurrentTime);

            if CurrentTime > TargetTime then
              begin
                Result := nil;
                AErrors := beTimedOut;
                Exit;
              end;

              // continue to wait (note if FStartDelay = 0 this will exit)
              TargetTime := CurrentTime + Round(FCounterFreq / 1000 * FStartDelay);
              while CurrentTime < TargetTime do
                QueryPerformanceCounter(CurrentTime);
              end;

          // Acquire the B-Scan
          QueryPerformanceCounter(TargetTime);
          for I := 0 to FBScanSettings.AinB - 1 do
            begin
              // get the AScan

```

```

    try
        FController.GetData(FScans[I], False);
    except on Exception do
        AErrors := AErrors or beMissed;
    end;
    FTriggerStates[I] := FController.CheckButtonDown; // read board state

    // Logic for stopping early
    if Stopping then
        Dec(StopScans)
    else if not FTriggerStates[I] and (FStopDelay >= 0) then
        Stopping := True;

    if Stopping and (StopScans = 0) then
        Break;

    // Figure out the start time of the next iteration
    Inc(TargetTime, count);

    // Check if the iteration took to long
    QueryPerformanceCounter(CurrentTime);
    if (CurrentTime > TargetTime) then
        AErrors := AErrors or beAScanDelay
    else // wait to start the next iteration
        while (CurrentTime < TargetTime) do
            QueryPerformanceCounter(CurrentTime);
        end;

    // check if we stopped early
    if I < FBScanSettings.AinB then
        begin
            SetLength(FScans, I + 1);
            Result := TBScan.Create(FScans);
            AllocateScanMemory; // need enough space for next BScan
        end
    else
        Result := TBScan.Create(FScans);
    finally
        FController.Unlock;
    end;
end;

function TBScanGrabber.SetGates(AManual: Boolean): Boolean;
var
    NewDetails: TDetectionDetails;
    Delay: Integer;
begin
    if AManual then
        begin
            Result := True;
            FillChar(FAutoGateDetectionDetails, SizeOf(TDetectionDetails), 0);
            if FManualDelay = 0 then
                FBScanSettings.AScanSettings.RegDelay := 0
            else
                FBScanSettings.AScanSettings.RegDelay := Round((FManualDelay *
BASE_FREQ) / 4) + 8;
            FBScanSettings.AScanSettings.Gain := FManualGain;
            if FBScanSettings.AScanSettings.RegDelay < MIN_DELAY then
                FBScanSettings.AScanSettings.RegDelay := MIN_DELAY;
        end
    else
        begin

```

```
Result := DetectInterfaces(FController, FPeakThresh, NewDetails) =
drFound;
  if Result then
    begin
      FAutoGateDetectionDetails := NewDetails;
      FBScanSettings.AScanSettings.Gain := NewDetails.GainUsed;
      // delay time is calculated differently
      Delay := Round(((NewDetails.CopperAir * BASE_FREQ) -
FInterfaceOffset) / 4 + 8);
      if Delay < MIN_DELAY then
        FBScanSettings.AScanSettings.RegDelay := MIN_DELAY
      else
        FBScanSettings.AScanSettings.RegDelay := Delay;
      end;
    end
  end;
end;
```

VITA AUCTORIS

NAME	Lawrence Barsanti
PLACE OF BIRTH	Sault Ste. Marie, Ontario, Canada
YEAR OF BIRTH	1982
EDUCATION	St. Anne High School, Tecumseh, Ontario 1996 – 2001
	University of Windsor, Windsor, Ontario Bachelor of Computer Science – Honours [Co-op] 2002 – 2006
	University of Windsor, Windsor, Ontario Master of Applied Science – Electrical Engineering 2006 – 2009