

1992

Study of beamforming techniques for ultrasound imaging in nondestructive testing

Sleiman Riad Ghorayeb
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Acoustics, Dynamics, and Controls Commons](#), [Electrical and Electronics Commons](#), [Materials Science and Engineering Commons](#), and the [Physics Commons](#)

Recommended Citation

Ghorayeb, Sleiman Riad, "Study of beamforming techniques for ultrasound imaging in nondestructive testing" (1992). *Retrospective Theses and Dissertations*. 9798.
<https://lib.dr.iastate.edu/rtd/9798>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

92

20969

U·M·I

MICROFILMED 1992

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9220969

**Study of beamforming techniques for ultrasound imaging in
nondestructive testing**

Ghorayeb, Sleiman Riad, Ph.D.

Iowa State University, 1992

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

**Study of beamforming techniques for ultrasound imaging
in nondestructive testing**

by

Sleiman Riad Ghorayeb

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department: Electrical Engineering and Computer Engineering
Major: Electrical Engineering (Communication Systems and Signal Processing)

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Members of the Committee:

Signature was redacted for privacy.

Iowa State University
Ames, Iowa

1992

Copyright © Sleiman Riad Ghorayeb, 1992. All rights reserved.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iv
ABSTRACT	vi
CHAPTER 1. INTRODUCTION	1
Background and Motivation	1
Brief Literature Review	1
Scope of the Dissertation	6
CHAPTER 2. REVIEW OF ULTRASOUND PRINCIPLES AND IMAGING TECHNIQUES	10
Creation, Propagation and Scattering of Ultrasound Waves	10
Overview of Ultrasound Imaging	20
CHAPTER 3. REVIEW OF THE FINITE ELEMENT METHOD	29
Brief Review of Some Numerical Methods	29
The Building Blocks of FEM	32
Governing Equations	34
Discretization, Energy Functional and Interpolation	37
Minimization and Assembly	40
CHAPTER 4. SAFT FORMULATION AND IMPLEMENTATION	44
Introduction	44
The Concept of SAFT	44

The Formulation of SAFT	45
Experimental Set-up	48
Description of the SAFT Algorithm	49
Analysis of Results	52
Anisotropy Effects	56
Finite Element Simulation	61
CHAPTER 5. BEAMFORMING: A DECONVOLUTION ALGORITHM	69
Description of General Beamforming	69
Data Independent Beamformer	75
Least-Squares Estimation	77
Singular-Value Decomposition	79
CHAPTER 6. RESULTS OF DEFECT PREDICTION	83
Filter Implementation and Procedure	84
Prediction and Reconstruction: Tests and Results	86
Discussion of Results	106
CHAPTER 7. CONCLUSION	128
Milestones and Major Work Accomplishments	128
Future Work	130
BIBLIOGRAPHY	132
APPENDIX A. SAFT PROGRAMS	141
APPENDIX B. BEAMFORMING PROGRAM	160
APPENDIX C. OTHER PROGRAMS AND SUBROUTINES	188

ACKNOWLEDGEMENTS

In his masterpiece *The Prophet*, and speaking to a teacher, Philosopher Khalil Gibran wrote, and I quote:

*No man can reveal to you aught but that which
already lies asleep in the dawning of your knowledge.
The teacher who walks in the shadow of the temple,
among his followers, gives not of his wisdom but
rather of his faith and his lovingness.*

To my teacher and my mentor, Professor William Lord, I would like to extend my very special gratitude for the wisdom, the faith, and the confidence he instilled in me throughout my doctoral work. I would also like to thank him for the financial assistance he provided from the Palmer endowment during the latter part of my program.

To Professor Satish Udpa, I would like to express a thank you note for his devotion and continuous advice and support which led to the selection of this research topic.

To Professor Chester (Chip) Comstock, Jr., I would like to offer my deepest thanks for his uncountable help and constant encouragement during the whole period of my graduate studies.

To Professor David Carlson who patiently listened to me when I regularly shared with him the latest in my research and provided me with his expertise knowledge in ultrasonic systems, thank you.

To Professor David Holger, I am especially thankful for his provision, through the Center for Nondestructive Testing at Iowa State University, of the research assistantship (U.S. Navy, Office of Naval Research, Contract #

N0014-86-K-0799) during the first half of my doctoral program, and his persistent guidance which laid the grounds for the successful implementation and testing of the SAFT algorithm.

Last, but not least, to Professor Loren Zachary, I also wish to express my sincere and profound gratitude for the technical assistance that he supplied during the development of the SAFT system.

Thank you to all of the above people for their enthusiasm and their interest to serve on my graduate committee.

Finally, to the people who really made a difference in my life... my family. My father Riad, my mother Sonia, my sister Ghada, and my brother Samer; I am perpetually grateful for their loving support and never-ending confidence which have influenced me to work hard and strive for excellence.

ABSTRACT

Many of the innovations in modern materials testing technology make use of ultrasound. Therefore, the theory and application of ultrasound have become of extreme importance in nondestructive inspection of complete engineered systems. However, despite the fact that most of these ultrasound inspection techniques are based on well-established phenomena, two key problems pertaining to their application still remain unresolved. These problems can be identified as (1) the material being tested is assumed to be isotropic and homogeneous by nature, and (2) the scanning/data collection process, prior to the reconstruction scheme, is very time consuming. As a result, techniques for fast, accurate testing of anisotropic and non-homogeneous media have been the focus of attention in modern non-destructive testing research.

This dissertation first describes the development and implementation of a time domain synthetic aperture focusing technique (SAFT) to reconstruct flaws imbedded within Plexiglass™ and Graphite/Epoxy samples. A modification to the present SAFT algorithm is then proposed in order to improve the quality of the images produced by SAFT when applied to composites. In addition, since the finite element method (FEM) can be used to solve hyperbolic partial differential equations, which govern wave propagation, FEM solutions are used to mimic a SAFT measurement. That is, the FEM is used to simulate the action of a transducer array. This is done to study the sensitivity of parameters involved in the SAFT algorithm. Using the same FEM model as a test bed, the *data independent* beamformer, in its basic form, is studied to determine its performance in reducing data acquisition time. It is seen that this technique is capable of adjusting the weights of the interpolating filter (beamformer) to predict an incoming signal from a desired direction while discriminating against other signals from different directions.

SAFT results indicate that the FEM model can be used as a *Test Bed* for

SAFT sensitivity studies, and point out the possible use of lateral displacement data for SAFT reconstruction. Extensive beamforming test comparisons reveal that the resulting optimal filters are indeed able to predict not only A-scan signals from a set of data produced by a nonuniformly spaced transducer baseline, based on one material geometry, but also A-scan signals from a data set produced by a nonuniformly spaced transducer baseline based on a different material geometry. This has particular significance for fast testing and imaging of isotropic and anisotropic materials in ultrasonic nondestructive evaluation techniques.

CHAPTER 1. INTRODUCTION

Background and Motivation

It is of crucial importance that the individual components, used to constitute the parts of complete engineered systems and structures, are free from damaging defects, such as porosity and cracks, and other possible causes of premature failure.

During the last few decades, various methods have evolved and new techniques are presently being developed to assess the reliability and integrity of those structural and functional components. Ultrasonic nondestructive testing (NDT) has been one of the prominent technologies used for that purpose. However, ultrasonic testing has been (1) conditional in the sense that the material under test be isotropic and homogeneous in nature, and (2) time consuming as far as the scanning and the data collection processes are concerned prior to the material reconstruction scheme.

The research objectives described in this dissertation focus on the following main three points: (1) to summarize some of the ultrasonic reconstruction methods developed in early years, such as SAFT, (2) to briefly discuss how the finite element method can be looked upon as a *test bed* for a typical NDT test, and finally, (3) to suggest a filtering/deconvolution technique called *Beamforming* to improve the scanning time process.

Brief Literature Review

Historically, ultrasound has been an excellent tool used for detecting flaws contained within solid materials. Characterizing these flaws, though, by their shape, size, and orientation, was a hard task. Image restoration techniques classified as flaw characterization methods were needed to form high quality images.

One technique which attempts to provide much more useful information, as far as the size and location of many material discontinuities than conventional ultrasonic displays, by improving the S/N ratio of the images, adopted the idea of synthetic aperture radar (SAR), since the concepts of wave physics of the microwave radar antenna and the ultrasonic transducer are quite similar.

Early ultrasonic synthetic aperture research development is credited to Flaherty in 1967 [1]. Similar experimental work was applied in the medical field and developed by Burckhardt and his colleagues in 1974 [2]. Here the work was distinguished from the rest by the fact that no reference signal was needed in order to form the synthetic aperture image, which is usually the case in NDT and medical applications.

In 1976, at the University of Michigan, Frederick and his colleagues presented the first version of the synthetic aperture focusing technique (SAFT) [3]. The report included a thorough overview ranging from the data acquisition hardware to the digital processing of the collected ultrasonic signals reflected off two side-drilled holes contained in an aluminum block. During the period of 1977 to 1979, more reports followed [4, 5, 6], under the direction of Frederick and Sydel. Every report presented a more improved version of the SAFT algorithm as applied to NDE of pressure vessels. These studies resulted in a very lengthy and explicit chapter, describing the basic concepts and fundamentals underlying the SAFT system, in addition to the evolution of the ultrasonic implementation along with very well interpreted experimental results. This work was presented by Seydel in 1982 [7].

In order to provide general rules for determining the processing parameters for various experimental situations, a study of a computer model of SAFT, when idealized data from one and two point reflectors are obtained, was presented by Johnson [8].

Up to this point, the development of SAFT focused on the two-dimensional case, but was easily extended to three [9]. By using the finite element method (FEM) to define the front surface geometry of the test specimen, the transducer scan path was arbitrarily varied by tilting the angle of the transducer, allowing the synthetic aperture to take full range of the values over the entire inspection field.

Expressions for the transverse and longitudinal resolutions of SAFT were derived by Thomson in 1984 [10], and experimental results to support the theory were presented. A suitable aperture weighting function for the SAFT algorithm [11], was determined in order to minimize the sidelobe level introduced by SAFT, without significant loss of the lateral resolution.

In 1986, Doctor and his colleagues [12] described the evolution of SAFT, as well as its flexibility, and a new operational angle-beam mode with a 3-D, line (pulse-echo) and tandem options of signal processing. Major improvements in the processing speed using special-purpose processors were also indicated. During that same year, Langenberg and his colleagues [13] presented an alternative SAFT processing scheme in terms of a Fourier domain algorithm. The newly developed twist to the conventional time-domain SAFT was called *Diffraction Tomography*. This technique requires a planar two-dimensional measurement aperture and broadband pulse-echo ultrasonic transmission and reception [13]. The recorded data are Fourier transformed with respect to time and the aperture coordinates, mapped into the Fourier space of the area containing the defect, and finally, processed into object space with an inverse Fourier transform. The output is quantitatively given in terms of either the characteristic function of the defect volume or the singular function of the defect surface. This scheme was determined to have better potential for carrying out the mathematical operations numerically, and served as a means to predict incomplete data produced by limited data apertures and limited frequency bandwidths. A 3-D imaging system based on Fourier transform SAFT was further investigated by Mayer and his colleagues in 1989 [14]. Here the algorithm was implemented to

provide a high resolution graphics to display a 3-D reconstruction volume.

Beamforming is an approach which is classified as a filtering process that can be applied to signals carried by propagating waves. The beamformer, in its simplest form, can be designed to pass signal components from a desired direction while isolating some others emanating from different directions, such as noise from interferers. The beamformer can either be *deterministic* (data independent), or *adaptive* (data dependent) and thus called *statistically optimum*.

Rather than attempting to attribute the widespread number of developments in the many different research projects in beamforming, to their respective researchers, a brief review of the previous work pertaining to the area of interest that this work lies upon, is given.

In a special issue [16], Van Veen and his colleague describe various beamformer processors, and provide the reader with an excellent overview of the signal processing contained in beamforming. A number of beamformers; namely, data independent, statistically optimum and partially adaptive, are discussed in depth with a few examples of the different applications relating to each of the aforementioned classifications, to support the theory. A section of the article is devoted to defining every aspect of the subject ranging from the basic terminologies and notations to the fundamental concepts underlying beamforming.

In general, the output of a beamformer is given by a linear combination of the data collected by an array of sensors. Prior to the coherent summation, the data is conveniently multiplied by some weights which represent the acquired data. It should be mentioned, at this point, that SAFT can be classified as a special case of beamforming. In a data independent beamformer, the weights are designed in such a way that the response of the beamformer can be predicted, via an approximation, in any desired direction, and completely independent of the data statistical structure.

The design for such a beamformer is similar to that for the finite impulse response (FIR) filter design [17-18]. A related algorithm called the *Constrained Least Mean Squares* (CLMS), was derived by Frost [19]. This algorithm adjusts an array of sensors in real time to respond to a desired signal from a chosen direction while discriminating against noise received from other directions. The concept of adaptive systems working on the principle of minimizing the output noise power under the constrained response to specified directions, is also described in various papers [20-24]. One surprising result obtained by Buckley and his colleagues [24], is the observation that employing derivative constraints by linearly-constrained systems is highly dependent upon the location of the phase reference point of the array. Prior to this latter work, derivative constraints were applied to broadband element space antenna array processors [25]. Here the effect of derivative constraints was studied on the beamwidth in the look direction. It was determined that the beamwidth in that direction can be made as broad as desired and the beam separations can be defined without any concern for signal suppression in the event of signal reception between beams. Griffiths and colleagues [26] present an alternative approach to Frost's linearly constrained adaptive beamforming algorithm. The *Generalized Sidelobe Canceller* (GSC), as described in this paper, was used to both analyze the performance of Frost's algorithm and to suggest generalizations of the constrained beamforming method.

Generalized adaptive beamforming is also discussed in length under a broad range of applications. It has been shown [27] that the techniques of adaptive filtering can be applied to processing the output of the individual elements in a receiving antenna array. This process also resulted in reduced sensitivity of the antenna array system to noisy interferers of unknown characteristics. Later, Applebaum and his colleague [28] realized that the initial adaptive array theory ignored the problem of incidental cancellation of the desired signal returns; so they examined a few constraining techniques for the response of the adaptive processor.

A key assumption in all of the above beamforming work, is that the

interfering signals are not coherent with the desired signal. Shan and colleagues [29] introduced a new on-line adaptive array beamformer capable of working well even when the desired signal and interferers are coherent. Widrow and colleagues [30] suggested that coherent interference could be eliminated by movement of the sensor array via what is referred to as the "Duvall" beamformer and the "Spatial Dither" algorithm. An electronic version of that suggested technique was then presented [29], and was especially appropriate for fixed sensor structures.

The effect of errors on adaptive beamforming is discussed by Cox [31]. These errors were found to pass through the beamformer like uncorrelated white noise. In this paper an improved adaptive beamformer was suggested, permitting simultaneous mixed (linear equality and quadratic inequality) constraints on the gain against the white noise.

Recently, Wahlberg [32] compared three different approaches for finding the beamformer weights in single receiver adaptive arrays. The last of these approaches offers a much faster beamforming technique. In their study, the authors show how covariance matrix based methods can be used in adaptive beamforming applications where only output power measurements are available. By applying perturbation theory to the nominal beamformer weights, the covariance matrix of the sensor outputs was estimated. The proposed methods are non-iterative unlike their least mean square (LMS) counterpart. This, then, provides the potential to reduce the computational time required for typical steering applications. Experimental work has confirmed their newly developed techniques and offered similar null steering performance, with a significant reduction in the number of power measurement samples, when compared to adaptive LMS methods.

Scope of the Dissertation

To date, there has not been, to the best of the author's knowledge, any available method used in ultrasound imaging, neither in the medical nor in

the NDT fields, which is capable of "reconstructing" scanning baselines containing irregular coverage of the medium under test. In the interest of learning more about the newly developed beamforming technique and its possible role in NDT, this research work focuses on three main topics. These topics are (1) finite element modeling used as a scheme to simulate the ultrasound behavior in the medium of propagation, (2) synthetic aperture focusing techniques used to enhance the quality of the images produced by the scanning process, and (3) the study of a beamforming technique to reconstruct the missing records in a nonuniformly spaced scanning baseline. The latter part of this work is considered to be a new concept and represents, in the author's opinion, a valuable approach for future ultrasonic tool developments in both the medical and NDT industries.

Following this introductory chapter, a moderately sufficient review of the most relevant background material on ultrasound principles and related imaging techniques, is the main concern of Chapter 2.

Chapter 3 reviews the building blocks of the finite element method (FEM). It introduces the global equations that govern the modeling/simulation process of ultrasound energy propagation and interaction with the medium in question, in addition to the necessary assumptions and boundary conditions imposed upon these equations. The choice of the energy functional, the discretization and interpolation procedures are also discussed in conjunction with the integration schemes that are employed. Special attention is given to the implementation of this very useful analysis method, in terms of transforming the theory into a practical computer tool.

Chapter 4 discusses the formulation and implementation of the synthetic aperture focusing technique (SAFT). This chapter contains material from two papers. The first paper [47] introduces the concept of SAFT and underlying equations, and the assumptions and parameters involved in the processing scheme of this technique. A description of a laboratory system experimental set-up is also shown in addition to a quantitative comparison

made between two SAFT processed images of Plexiglass™ and Graphite/Epoxy samples. The various effects of anisotropy on SAFT, are discussed next in this chapter. Here, a more accurate SAFT implementation on composites is proposed. This alteration to SAFT involves the use of the slowness surface curves and their relationship to the stress-wave velocities present in the medium. The second paper [48] uses the finite element method (FEM) in order to simulate the action of a transducer array, and then use the resulting signals, along with SAFT, to reconstruct the reflectors under study. This work indicates that the resulting FEM model can indeed be valid as a "test bed" for SAFT sensitivity studies and points out to the possibility of using the lateral displacement data in order to improve the SAFT resolution.

The body of this dissertation revolves around Chapters 5 and 6. Chapter 5 introduces the basic concepts of beamforming. It then branches out to give a detailed description of one beamforming technique, namely, the *Data Independent (DI) Beamformer*, which will eventually become the center of attention of this work. By employing the theoretical as well as the practical approaches underlying DI, one can design an "optimal" beamformer. This is done through the determination of a minimum norm solution. The method of Least Squares is used for that purpose.

The actual design and testing of the beamformer is demonstrated in Chapter 6. The experimental development and defect prediction and reconstruction procedures are detailed. The effects of transducer spacings and material geometry on the beamformer are also discussed through a comparison between the predicted and actual A-scan signals. Images of the beamformer filters and their corresponding predicted B-scan images are shown for situations where 50%, 26%, 13% and 3% of the data are kept. In addition, plots of the relative mean-squared error between the reconstructed images and the actual images are shown in order to assess the reliability of the beamforming technique. The sizing and the location of the flaws in the material under test are also estimated and compared with the actual values. FEM-simulated pulse-echoes are used for all of the tests performed by the

beamformer.

Finally, Chapter 7 discusses the results obtained in Chapter 6 and highlights the major accomplishments provided by this work. It also points out possible future extensions and suggestions pertaining to this research topic.

Reprints of the programs and subroutines used in this research are contained in the appendices of this dissertation.

CHAPTER 2. REVIEW OF ULTRASOUND PRINCIPLES AND IMAGING TECHNIQUES

In this chapter, the basic concepts behind ultrasound waves are introduced using simplified definitions. Enough information is given for a reasonable understanding of the most current ultrasonic nondestructive testing imaging techniques.

For years, ultrasound methods have been used to cover all aspects of NDT research ranging from the detection of internal cracks in materials to small sub-surface defects [33]. These methods have played a major role in quality inspection of partially manufactured components as well as fully finished sections of a larger apparatus.

Ultrasound waves propagate through solid media in several ways. The creation, propagation and scattering of these waves are considered next.

Creation, Propagation and Scattering of Ultrasound Waves

Ultrasonic wave generation is mostly noticeable when the piezo-electric (pressure-electric) effect is present [33]. This effect is the result of dilating or constricting certain crystalline materials when subjected to a voltage across the faces of the crystal. Conversely, when mechanical strain is applied to the crystal, an electrical field is created. These piezo-electric materials constitute the basis of electro-mechanical transducers. A large number of piezo-electric materials, including man-made ceramics and polymers [35], have been used in the construction of modern ultrasound transducers. However, the original material used was, and still is, natural quartz [33, 35]. When a disc of piezo-electric material is subjected to an alternating voltage across its thickness, vibration, caused by contraction and expansion of the disc, gives rise to a compressional wave normal to the disc's surface. When the transducer crystal vibrates at its natural frequency, wave generation is most efficient. As mentioned earlier, piezo-electric materials

play the double-role of generating as well as detecting sound waves. As a propagating ultrasonic wave is intercepted by a transducer crystal, vibration of the crystal will again occur, causing an alternating current to be produced across its faces. This is the reason why many ultrasonic NDT experiments use only one transducer to act as both transmitter and receiver, hence bearing the name of *pulse-echo*. Usually, while ultrasound is transmitted as a series of pulses of extremely short duration, the crystal can detect the reflected waves during the time interval between those transmissions.

Ultrasound waves are elastic waves which can be transmitted through both fluids and solids. One of the required criteria of ultrasound is that the medium be continuous for the energy to propagate freely [34]. Any obstruction such as internal voids, delaminations, inclusions or cracks will interfere with the transmission/reflection of ultrasound signals. These ultrasound waves used in the NDI of materials, usually operate in the 0.5 to 20 MHz frequency range.

When these ultrasound waves propagate in a fluid medium, they are of a *compressional* type, in which case the wave traverses in the longitudinal direction of propagation. However, in solids, a *shear* component can arise where part of the wave displacement is normal to that direction of propagation. Elastic surface waves, called *Rayleigh* waves, can also occur. In a particular fluid, the longitudinal velocity of ultrasound waves is given by

$$v_c = \sqrt{\frac{V_a}{\rho}} \quad (2.1)$$

where V_a is the adiabatic volume elasticity and ρ is the density. When the ultrasonic wave enters the solid material, the expression for its longitudinal velocity is not as trivial. If E represents the modulus of elasticity (Young's modulus), and κ is the Poisson's ratio, then v_c becomes

$$v_c = \sqrt{\frac{E(1-\kappa)}{\rho(1+\kappa)(1-2\kappa)}} \quad (2.2)$$

Some representative typical propagation mean longitudinal velocities in various media are given in Table 2.1.

Table 2.1. Ultrasound longitudinal velocities in some materials

Material	Mean Velocity v_c (m/sec)
Air	330
Water	1500
Aluminum	1750
Plexiglass™	2750
Steel	5810
Graphite/Epoxy (0°/90°)	2919

On the other hand, the shear wave velocity detected in the solid portion of propagation, is approximately half the longitudinal wave velocity and is given by

$$v_s = \sqrt{\frac{G}{\rho}} \quad (2.3)$$

where G represents the modulus of rigidity of the material.

In an attempt to distinguish the compressional component (L-wave) from the shear component (S-wave) in a conventional ultrasonic NDT experiment, Ludwig [38] and You [39] presented results of some situations where analytical solutions are not usually available. A numerical technique,

namely, finite element modeling (FEM), was used to visualize the interactions of ultrasonic waves with simple defects. In one example [39], a 20cm × 10cm Aluminum specimen with a rectangular slot of 8mm × 0.4mm located at the center, was considered (Figure 2.1). The block was subjected to a point source input having a raised cosine waveform shape. Displacement plots revealed that compressional, shear, surface and head waves all originated from the same source, and deviated from each other as they propagated further inside the material, due to their dissimilar propagating velocities. Figure 2.2 shows the longitudinal wave as it is intercepted by the defect, forming a strong reflection at the surface. Once this L-wave passes the slot, an incident shear pulse can then be seen. The most prominent shear waves are the ones diffracted by the tips of the crack. These, in fact, result from mode-converted shear waves which arise when the L-waves interact with the tips of the crack. As will be seen later, this concept will help further in determining the crack tip locations, and will ultimately provide us with a better sizing scheme.

As sound propagates through a material, its intensity, I , generally diminishes with the distance of propagation z [36], according to

$$I = I_0 e^{-\alpha z} \quad (2.4)$$

where I_0 is the intensity at $z=0$ (surface of the material). Unlike the velocity of sound, the attenuation coefficient, α , depends highly on the frequency and increases with an increase in frequency. This is especially true during the inspection of polymeric materials where the frequency range 2 to 5 MHz is used. In many common fluids, however, such as water, the attenuation is primarily due to viscous absorption, and in these cases the attenuation is proportional to the square of the frequency. This limitation on frequency has different impacts on equipment performance. These impacts are due to the fact that the frequency, f , along with the velocity of sound, v , in a specific medium, determine the wavelength, λ , of the ultrasonic pulse. Note that the

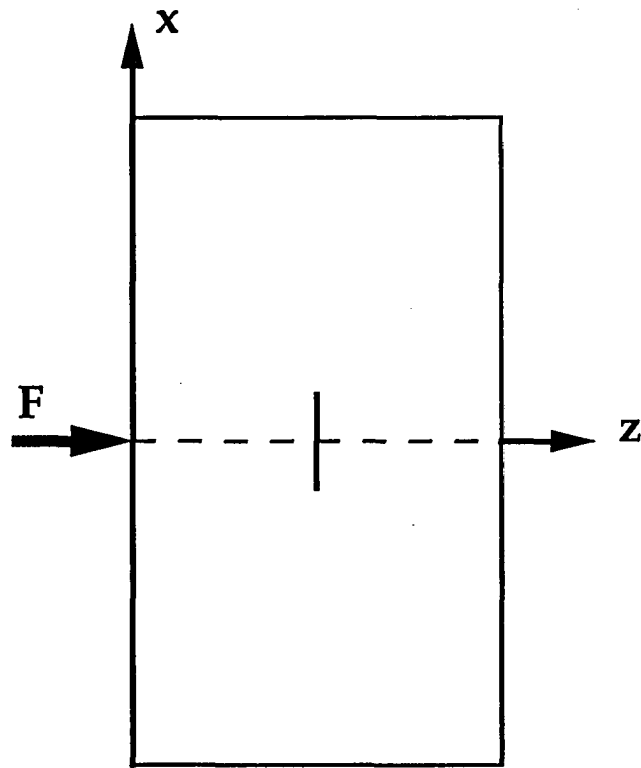


Figure 2.1. Sample geometry used for the study of the L-wave/defect interactions (You, 1991)

TIME = 8.000 (MICROSECONDS)

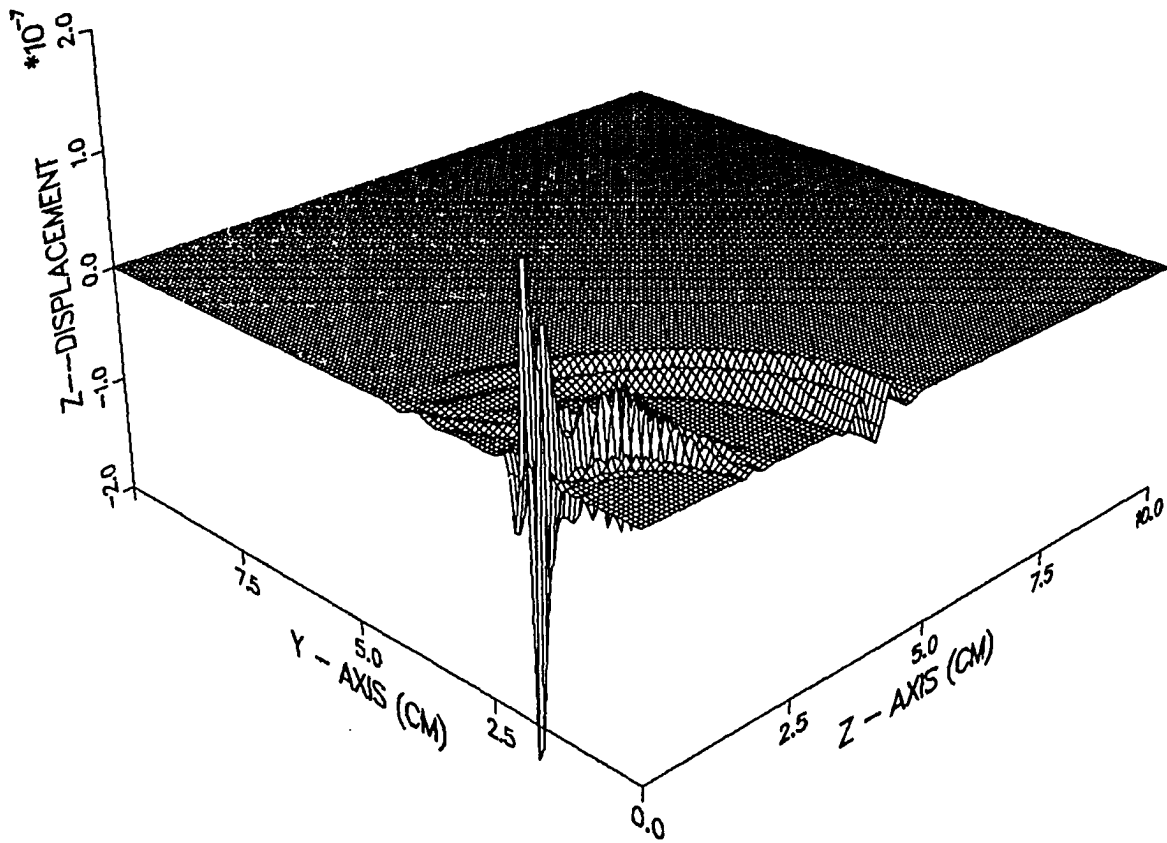


Figure 2.2. (continued)

TIME = 12.000 (MICROSECONDS)

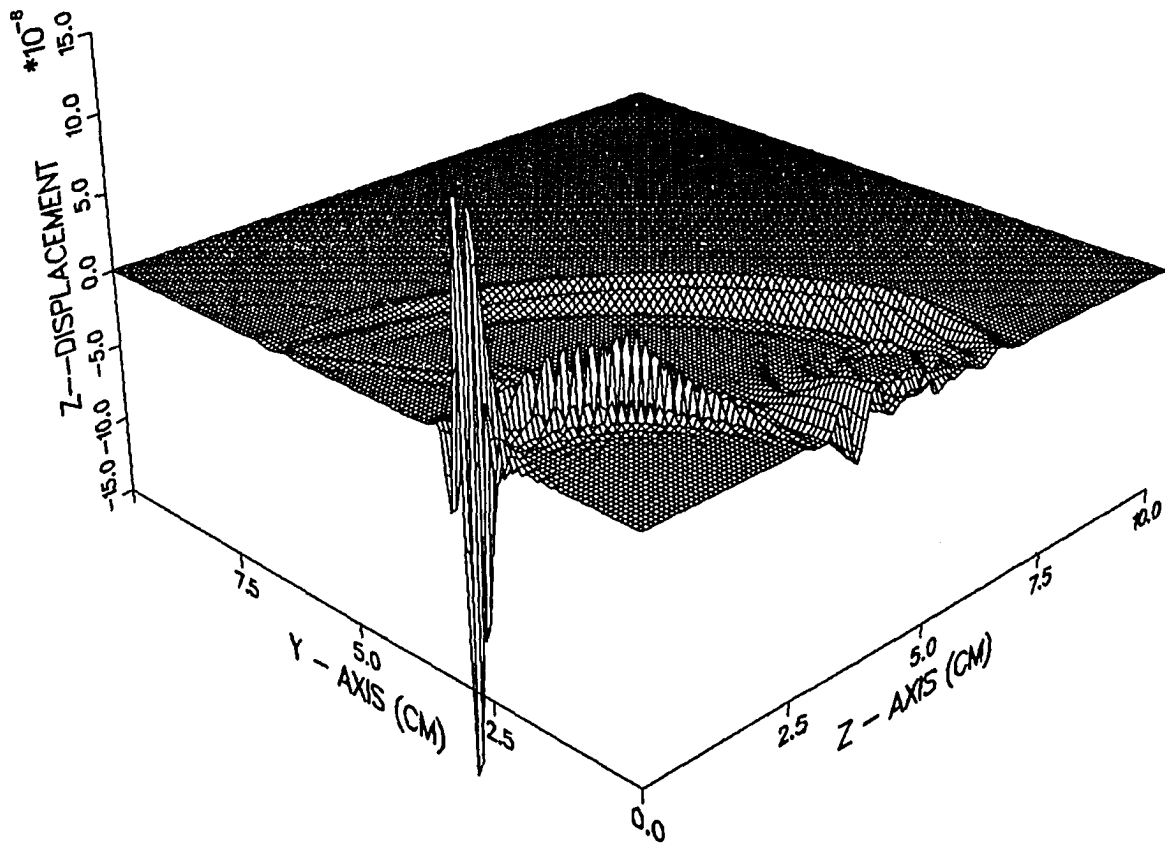


Figure 2.2. (continued)

TIME = 16.000 (MICROSECONDS)

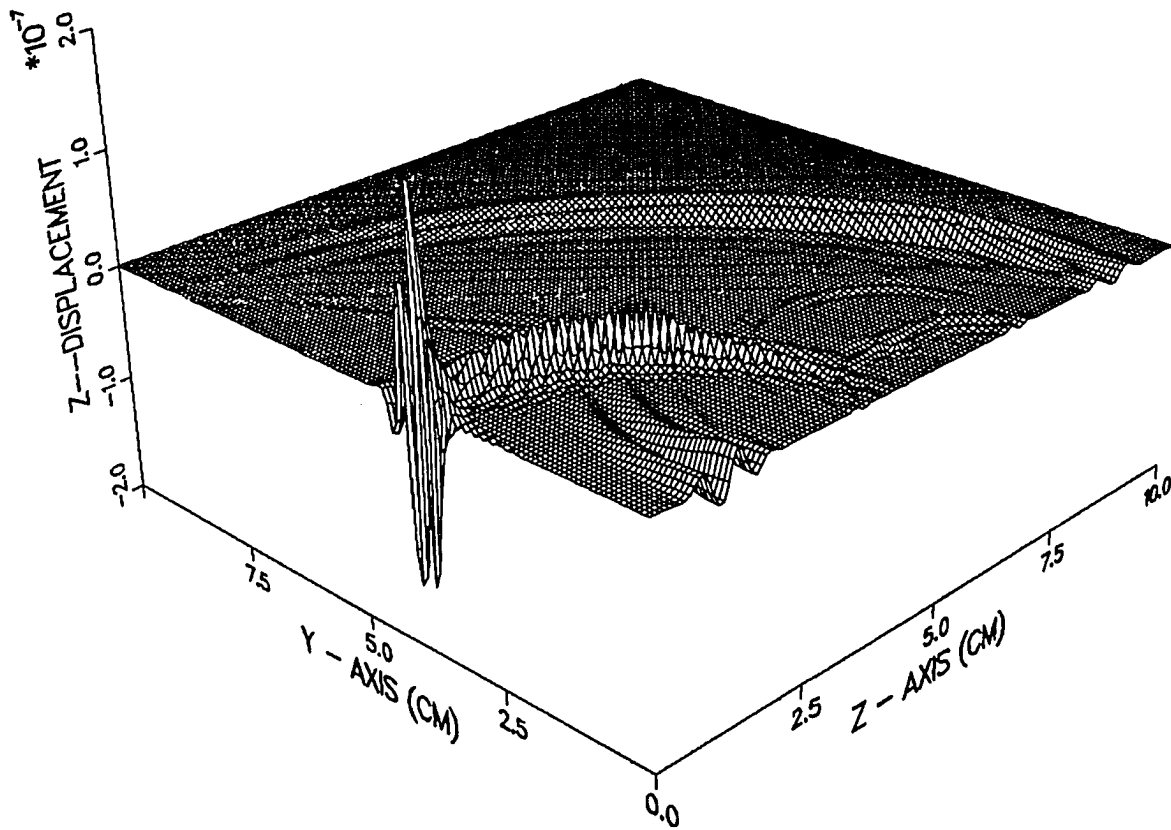


Figure 2.2. (continued)

17a

TIME = 20.000 (MICROSECONDS)

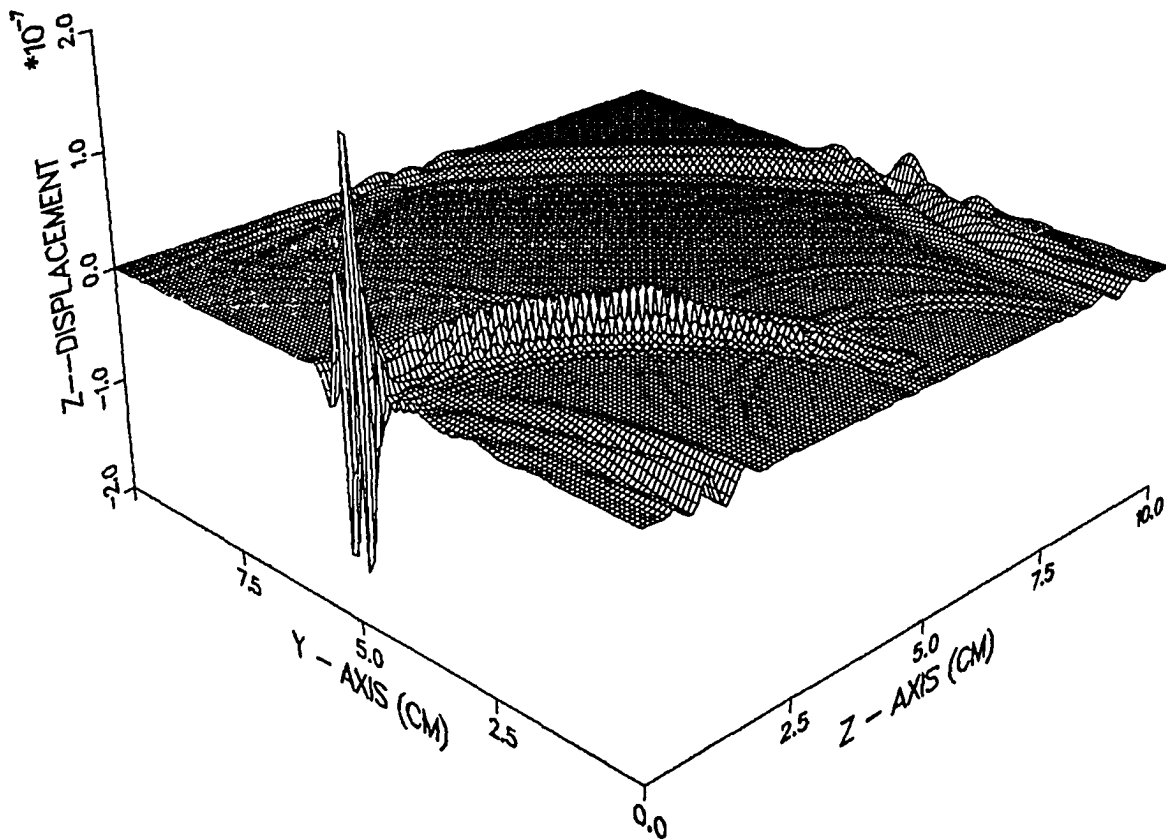


Figure 2.2. (continued)

wavelength is kept as short as possible to improve axial resolution [37]. The equation relating these three variables is given by

$$\lambda = \frac{v}{f} \quad (2.5)$$

Table 2.2 gives the wavelengths of compressional sound waves in some materials at various frequencies.

Table 2.2. Ultrasound longitudinal wavelengths in some materials

Material	λ (mm) for frequencies (MHz) of			
	1.25	2.5	5.0	10.0
Air	0.26	0.13	0.066	0.033
Water	1.20	0.60	0.30	0.15
Aluminum	4.95	2.48	1.24	0.62
Plexiglass™	2.20	1.10	0.55	0.275
Steel	4.65	2.32	1.16	0.58
Graphite/Epoxy (0°/90°)	2.335	1.167	0.583	0.29

From Table 2.2, it is seen that the higher the frequency the better the resolution, which is the trade-off for having high attenuation and for the sound beam not being able to propagate as deeply in the material as the lower frequency. For example, defects in a Plexiglass™ block of sizes greater than 0.275 mm can be detected by an ultrasonic beam with a center frequency of 10 MHz, but if these defects are larger than 2.2 mm, an ultrasound beam with a center frequency of 1.25 MHz has to be used.

Two last, but important aspects of sound wave propagation are the *Reflectivity*, or in more familiar terms, the *Reflection Coefficient*, R ; and the *Transmission Coefficient*, T . The reflection coefficient and the transmis-

The acoustic impedance, Z , of a certain medium, is defined as the product of the compressional sound velocity, v , and the medium density, ρ , as in

$$Z = v_c \rho \quad (2.6)$$

Listed in Table 2.3 are some values of acoustic impedances for a variety of media.

Table 2.3. Acoustic impedances for various media

Medium	Acoustic Impedance ($10^6 \text{ kg} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)
Air	0.0004
Water	1.50
Aluminum	16.77
Plexiglass™	3.245
Steel	45.72
Graphite/Epoxy (0°/90°)	4.641

The reflection coefficient, R , for a normally incident ultrasound beam propagating through two interfacing media with acoustic impedances, Z_1 and Z_2 is given by

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad (2.7)$$

The corresponding transmission coefficient, T , is given by

$$T = \frac{2 Z_1}{Z_2 + Z_1} \quad (2.8)$$

Table 2.4 gives the reflectivity and the transmittivity at normal incidence for a combination of media interfaces.

Table 2.4. Reflectivity and transmittivity of normally incident ultrasonic waves for various media interfaces

Media Interface	Reflectivity	Transmittivity
Air-Water	0.9995	0.0005
Water-Steel	0.937	0.0635
Water-Aluminum	0.836	0.1642
Water-Gr/Ep(0°/90°)	0.512	0.4885
Water-Plexiglass™	0.368	0.6322

It is seen from Tables 2.3 and 2.4, that the greater the difference of the acoustic impedances of two adjoining media, the greater the amount of reflection coefficients at their respective junction. Conversely, the greater the reflectivity at the interface, the smaller the transmission coefficient of the propagating wave. Finally, it is important to note that the amount of sound reflected from and transmitted through an object depends not only on the difference between the acoustic impedances [36] of that object and its immediate vicinity, but also on the size, shape and orientation of the object.

Overview of Ultrasound Imaging

Now that the theory of sound waves has been investigated, the basic concepts underlying the imaging techniques using ultrasound are discussed. The images produced by ultrasonic waves are unique since they represent the internal interaction with the mechanical properties of defects and other features in materials, and hence, serve as a powerful tool in modern NDT research. This section is intended to cover a brief overview of some of the imaging methodologies involved in the application of defect characterization.

A basic reflection imaging arrangement for a typical flaw detector is shown in Figure 2.3. The apparatus consists of a digital sampling oscilloscope, where the signals are usually displayed, a motorized transducer holder mounted on a water tank, and a pulser/receiver unit used to transmit ultrasonic pulses and amplify the return echoes. All of this equipment is controlled by acquisition software running on a portable computer. The idea behind the data collection is simple. When transmission is triggered, the transducer converts the electrical pulse into a mechanical vibration at the chosen frequency, emitting a wavefront. When this wavefront hits a discontinuity, a scattered wave is received by the same probe (or a different probe, depending on the application) and the resultant energy is converted back from mechanical pressure to an electrical signal and then displayed on the CRT screen.

There are several ways in which received signals can be displayed in order to extract the necessary information describing any particular medium during a typical ultrasonic test. Ideally, the displayed signal represents the reflectivity, as a function of time, of the workpiece, where time is a figure of the various depths in the specimen [40]. This display of the reflectivity as a function of time (or depth) is the most commonly used system and is known as the *A-scan* signal. This signal is actually the result of deflecting the beam (of a CRT) that represents the reflectivity versus depth [41]. A typical A-scan of a Plexiglass™ block containing two slots is shown in Figure 2.4. The largest pulse at the left-hand side corresponds to the front wall echo of the sample, and the following "blips" correspond to the echoes emanating from both defects and the back wall, respectively. The amplitude of these echoes is generally proportional to the size of the reflected surface but is usually affected by the attenuation effects introduced by both the travelled distance and the inherent composition of the material itself.

The second type of display is the most popular one. The *B-scan* is used to depict the reflectivity of a two-dimensional slice through a portion of the structure [42]. This system enables the recording of a defect location within a

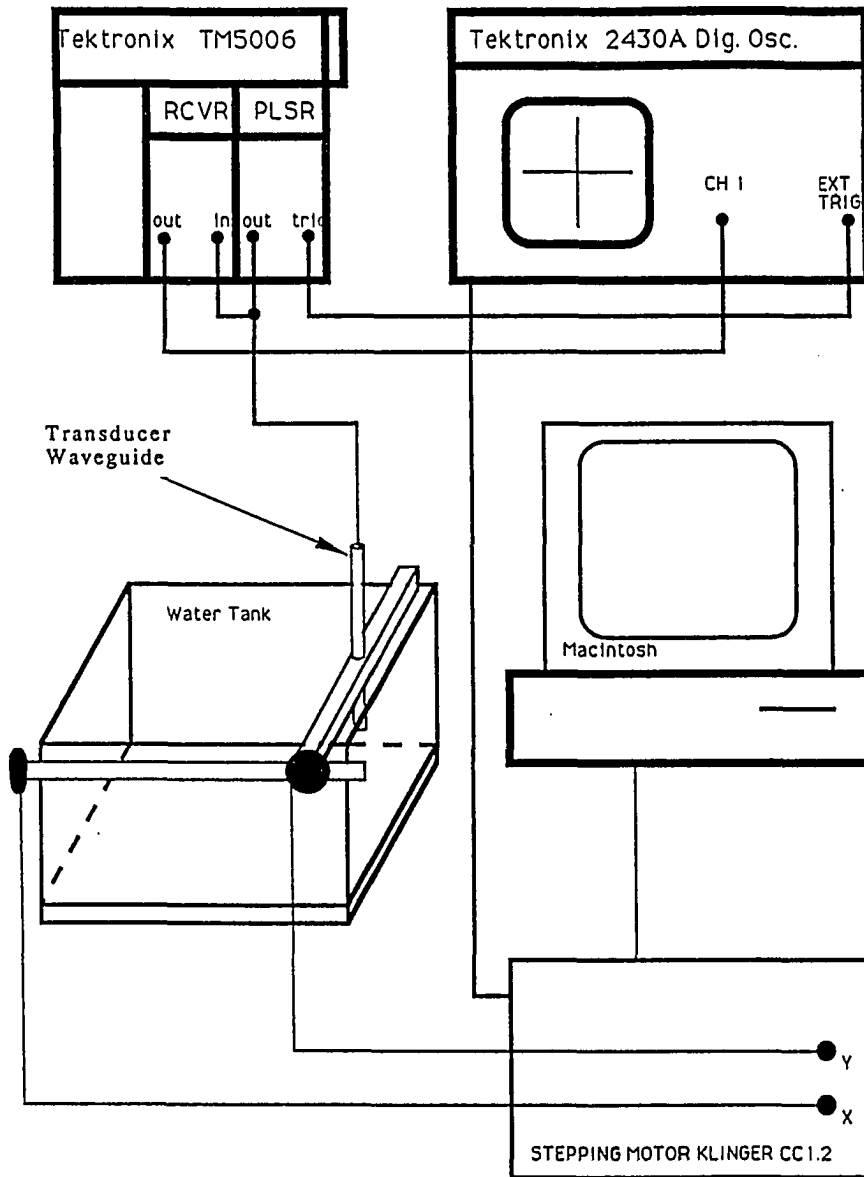


Figure 2.3. Basic reflection imaging apparatus for a typical ultrasound flaw detector (Ghorayeb, 1991)

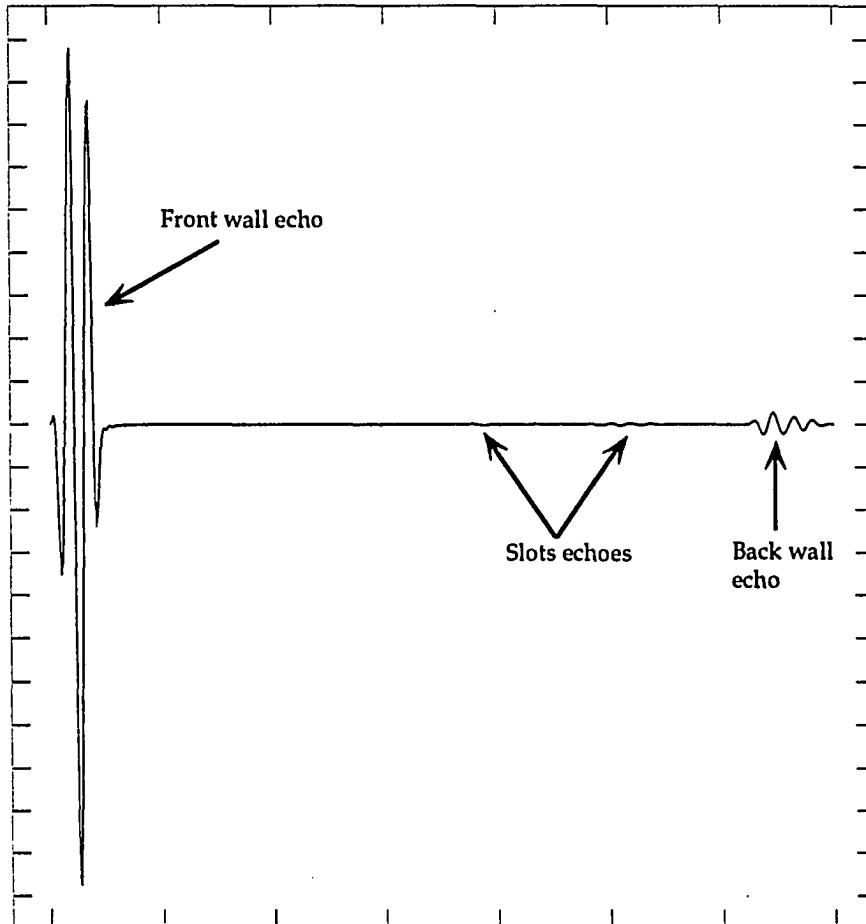


Figure 2.4. Example of a typical A-scan taken from a Plexiglass™ sample containing two slots

material. The data for this system are acquired by a transducer scanning in a linear fashion and at a constant velocity. The system is illustrated in Figure 2.5. Note that a B-scan is nothing more than the result of a family of successive A-scans taken at different transducer positions during which the transducer is assumed to be essentially stationary when the echoes are being acquired. Figure 2.6 illustrates a B-scan for the same Plexiglass™ sample whose A-scan was shown in Figure 2.4.

Processing a set of consecutive B-scans results in a plan view, or a volumetric display, of the medium under test. This method is termed *C-scan*. C-scanning can also be interpreted as the arrangement of the individual A-scans in a raster data collection configuration. Raster scanning is shown in Figure 2.7.

Once the raw data (A-scans, B-scans or C-scans) have been collected, it is necessary to reconstruct the images produced in order to obtain maximum resolution in both the lateral and the longitudinal directions. Various reconstruction schemes have been introduced throughout the years. The synthetic aperture focusing technique (SAFT) is one of them [7]. It is a method used to improve the lateral resolution of unrectified ultrasonic data. The theory and application of this method is thoroughly described in Chapter 4. Although SAFT enhances the lateral resolution, expressions that determine both the lateral as well as the longitudinal resolutions have been derived [10]. However, since intensive computational processing was still a major requirement of SAFT, other reconstruction techniques, involving the idea of deconvolution, were investigated [43]. Wiener filtering and the maximum entropy method (MEM) have been shown to give higher lateral resolution than SAFT processing, as well as a unique improvement in longitudinal resolution. Real-time SAFT systems were also studied on different occasions [44, 45], in order to minimize the time it takes for the conventional SAFT to compute an image. Hall and his colleagues [44] focused their attention on the acceleration of the computationally intensive coherent summation SAFT algorithm. Their work essentially dealt with successfully

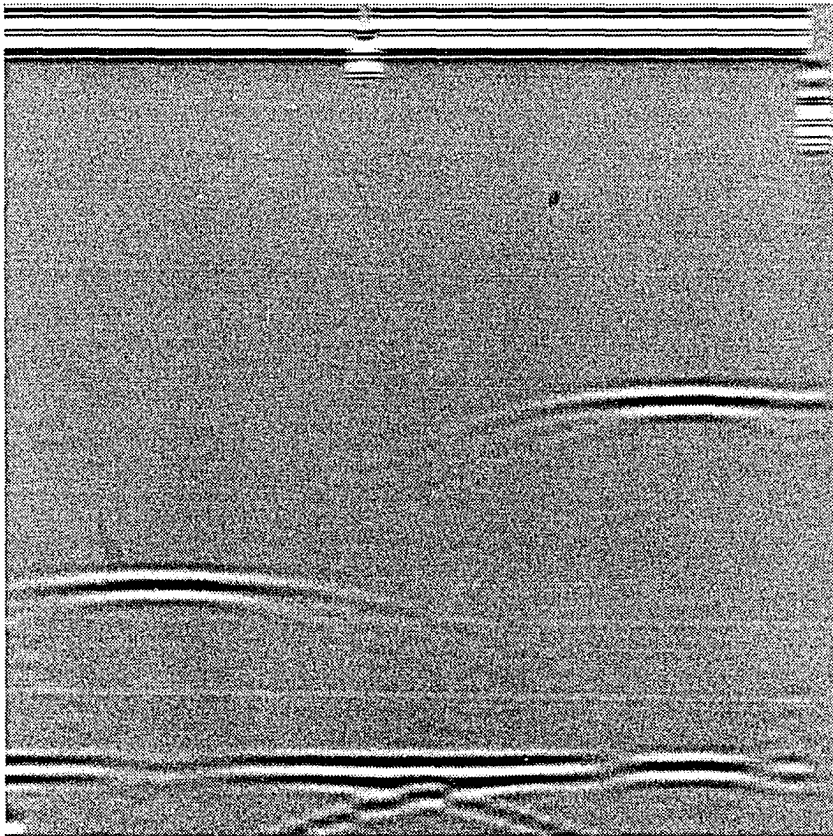


Figure 2.6. B-scan grey level plot for the same Plexiglass™ sample

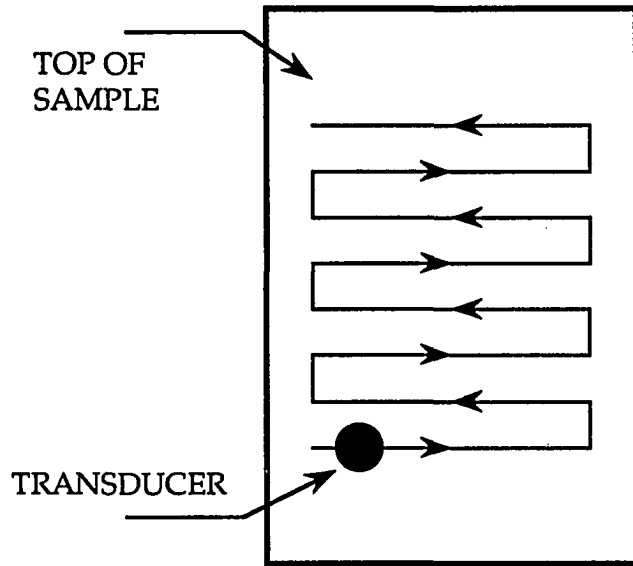


Figure 2.7. Method of raster scanning the surface of a sample

deploying SAFT in a commercial environment to achieve rapid imaging, which allows the operator to make timely judgments with respect to specimen integrity, and eventually reducing the costs involved with reactor in-service inspection. In a similar fashion, Ozaki and his colleagues [45] have devised a new way to implement the SAFT algorithm and have developed a real-time ultrasonic imaging system that provides, for any object size, a cross-sectional image of an object without any interruption. The resulting images were composed of an assembly of A-scans, and displayed as scroll pictures on a cathode ray tube (CRT) with no interruption regardless of the object size.

CHAPTER 3. REVIEW OF THE FINITE ELEMENT METHOD

Today's engineers, scientists and applied mathematicians, are frequently faced with the determination of solutions to complex problems. Computer models have been effectively used to analyze, and thus interpret the many physical and mathematical aspects underlying these intricate systems. Some of the areas of application include [52] fluid and solid mechanics, heat transfer, electrical and magnetic fields and vibratory potentials. During the last two decades, analyses of such processes have rapidly focused on the finite element method (FEM) as the analytical tool that can be used efficaciously to study the various models of interest.

With the attention focused on ultrasonic NDT, it is very important to have a complete insight on the behavior of the elastic waves as they propagate in solid materials and interact with internal features such as porosities, defects, grain structures and layers. In order to study the physics underlying these phenomena, in-depth understanding of forward models, used to accurately simulate both the medium and the energy interactions with the internal features, is essential. These forward models provide much of the needed information relative to inverse characterization of the defects. One of this chapter's objectives is to provide a brief, but yet sufficient, description of some of the numerical methods used to solve for the forward models. A review of the FEM technique and its role in ultrasonic NDT, is then presented in a more elaborate fashion.

Brief Review of Some Numerical Methods

In ultrasonic NDT, the art of obtaining qualitative as well as quantitative information about the physical shape and orientation of unknown defects in materials, from simple ultrasonic measurements, is most challenging [54]. Over the past few decades, numerous hypotheses have been developed to interpret how ultrasound interacts with defects in elastic solids.

In all cases, the aim was to determine an exact solution to the problem; however, most of the methods acquired a certain level of inherent limitations.

In the method of *separation of variables*, the wave equation has to be decomposable (separable) in certain coordinate systems. The method proved to give exact solutions for problems with simple obstructions such as a penny-shaped crack [53], spherically-shaped inclusions [55, 56], or cylindrical discontinuities [57, 58]. Elastic wave propagation in isotropic [59, 60] and transversely isotropic media was also predicted through the use of semi-numerical techniques involving this method. However, in cases where arbitrary shaped defects are present, or when the wave fields encounter some sort of inhomogeneity in the material, these approaches are unable to provide exact solutions to these otherwise more complicated situations. Instead, integral equation based techniques can be considered.

The basis for the *Kirchhoff* approximation is to use the fields generated by the incident and reflected waves that are produced off an infinite plane reflector (half-space) to approximate those found on the finite reflector [61]. It has been shown [62], that the Kirchhoff approximation does a very good job of describing a crack size and fits the flaw to an equivalent flat elliptical crack shape. Note that the Kirchhoff approximation is a method that can be applied to high frequency problems where the defect size is large compared with the wavelength. Also, those results based on this method have been determined for perfect reflectors like flat cracks. So if the scattering response is to be obtained for a volumetric type defect, such as an inclusion, more simplifying assumptions have to be made.

One approximation that has been useful in this respect is the *Born* approximation. Born is a weak scattering approximation. Unlike the Kirchhoff approximation, it is based on the assumption that the scatterer is small and that the scattered field is negligible compared with the incident

field inside the scatterer [39]. It further assumes that the material properties of the inclusion are not very different from those of the host. Although excellent results have been obtained using these two methods either separately [63], or in a unified fashion [54], neither Born nor Kirchhoff approximations would be applicable to situations where the size of arbitrary shaped flaws is comparable with the wavelength.

The *boundary element method* (BEM) [64, 65] is one of the various techniques used for this type of application and requires only the surface of the scatterer to be discretized into a number of boundary elements. A set of equations, with unknowns formed by the nodal displacements, is produced. These equations are usually solved using fundamental solutions that produce singularities at the point under consideration. Lately, this technique has been applied extensively in NDT for problems dealing with three-dimensional elastic waves radiation and scattering from arbitrary shaped and planar scatterers [66, 67]. An inherent advantage of BEM is that it can be applied on scatterers in infinite solids. The one drawback of the method, however, is that it can only be applied on isotropic materials where fundamental solutions are available.

Another technique that can be used when the dimension of the defect is comparable with the wavelength is the *finite difference method* (FDM). This method is not based on solving the initial value problem. It is rather classified as a *direct method* [68]. When used to solve partial differential equations, FDM uses an equal mesh point spacing [69] to approximate the unknowns, representing the field values of the mesh nodes, through a set of difference equations based on the Taylor series expansion. Modifying the series of the field values in the time domain into an "extrapolated" form provides a scheme to carry out the time stepping process. FDM solution approaches were developed in ultrasonic NDT research by Bond [70, 71], Bond *et al.* [72], Harker [73], Harumi [74] and Harumi *et al.* [75]. Although elastic wave propagation in these studies was successfully predicted, a main difficulty in the use of the conventional FDM lies in the incorporation of the boundary conditions [69]. Since the differential equations, representing

the system, are usually approximated directly by the difference scheme, it is necessary to satisfy both the essential and the natural boundary conditions. The fact that FDM requires a uniform mesh topology, poses a restriction on the form in which differencing is carried out, which creates a difficulty in maintaining symmetry properties in the coefficient matrix, and thus applying the boundary conditions to arbitrary boundaries, limits the generality of the approach.

As mentioned earlier, in order to simulate accurately a real ultrasonic testing system, a robust forward model has to be capable of handling material properties, such as anisotropy and attenuation, and awkward geometries. Both the transmitter and the receiver models should also be embodied in the overall prototype. The finite element modeling technique has been developed [38] with these intentions kept in mind. The following section describes the major points that constitute the building blocks of the finite element method (FEM).

The Building Blocks of FEM

When physical problems are analyzed through the application of the finite element method, there are certain *basic ingredients* [76] that constitute the path to be followed regardless of the particular area of application. This section briefly introduces these basic ingredients and points out, in general, the areas where FEM has been successfully applied.

Applying FEM to a mathematical or a physical system involves several, implicit or explicit, distinct steps or *building blocks*. These can be identified as [76]:

- (1) Discretization
- (2) Interpolation
- (3) Elemental formulation
- (4) Assembly

- (5) Constraints
- (6) Solution
- (7) Computation of derived or secondary unknowns

These steps are of great importance when establishing and following the procedure that leads to the solution of the finite element model. In more concise terms, these seven steps can be broken down into two basic ideas (or categories):

- (1) Construct the so-called *weak formulation* of a boundary value problem, and
- (2) Decompose the domain into smaller subdomains, or elements.

There are many important classes of physical problems to which these two ideas have been applied. One of the most important and successful areas is solid mechanics [77, 78]. Here the weak formulation of the boundary value problem is associated with the energy principles in the statics of solids. Furthermore, representing large structures as a number of smaller substructures, that are properly assembled, describes the phenomenon of domain decomposition. In the field of wave propagation, FEM was not applied until 1972 by Lysmer *et al.* [79] in seismology. Thereafter, the FE formulation started to converge more steadily to the study of wave transmission and scattering to and from defects embedded in elastic solids. The application of FEM to ultrasonic NDT was first introduced by Ludwig [38] and Ludwig and Lord [50, 51], where models were studied for isotropic materials in two-dimensional geometries. Further refinement to the latter work was accomplished by You [39] and You *et al.* [80]. In this research, complicated geometries, such as axisymmetric and three-dimensional situations, and general anisotropic materials were incorporated into the development of the elastodynamic finite element code.

The following two sections examine the physics behind ultrasound wave propagation through the derivation of related governing equations and

the energy functional of the system under study.

Governing Equations

The governing equation of motion which describes the propagation of waves in situations where the medium is linear, homogeneous and isotropic [51], is

$$(\lambda + \mu) \nabla \nabla \cdot u - \rho \frac{\partial^2 u}{\partial t^2} + \mu \nabla^2 u = 0 \quad (3.1)$$

where ρ and u represent the material density and the displacement vector, respectively; and λ and μ are the Lamé constants. Figure 3.1 indicates the interior of the domain D in which the above equation is described. In addition, the Figure shows the kinematic and the traction boundaries, Γ_1 and Γ_2 , respectively, where *Dirichlet*- or *Neumann*-type boundary conditions must be specified. A boundary condition that specifies the dependent variable u on Γ_1 will be referred to as *type I* [76] or *Dirichlet* boundary condition. A boundary condition that specifies the normal derivative of the dependent variable on Γ_2 will be referred to as a *type II* [76] or *Neumann* boundary condition. If the entire boundary is of type I, the boundary value problem is known as a *Dirichlet* problem. If the entire boundary is of type II, the boundary value problem is known as a *Neumann* problem. The boundary condition of type II is generally a *local* balance equation [76] which must be satisfied at the boundary. The above problem is classified as a *hyperbolic* initial boundary value problem.

In general, the solution to this boundary value problem is a displacement vector u which possesses second partial derivatives throughout the domain D , and satisfying the partial differential equation and the boundary conditions on Γ_1 and Γ_2 . Usually, a classical or analytical approach

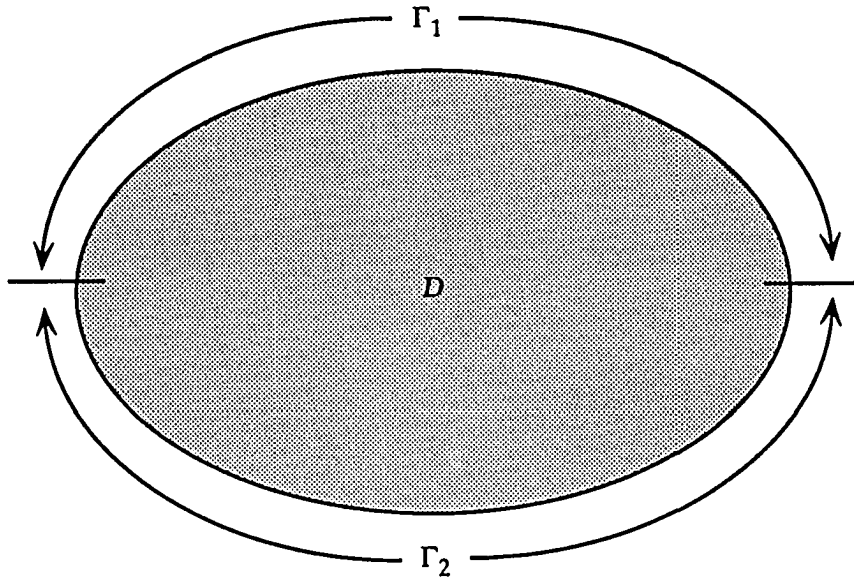


Figure 3.1. Domain and kinematic and traction boundaries

to the solution is not possible if the domain D does not hold a particular regular shape such as a rectangle or a circle. In this situation, a numerical approach, such as, the finite element method, is appropriate.

The combination of the first and the third terms in Equation (3.1) represents a stress tensor, T , in the elastic medium. If Equation (3.1) is written in a different form, we get the following

$$T_{ij,i} = \rho \ddot{u}_j \quad (3.2)$$

For the case where small displacement values are considered, as in most NDT situations, a linear stress-strain relationship can be established by

$$T_{ij} = C_{ijkl} S_{kl} \quad (3.3)$$

This is referred to as Hooke's law. C_{ijkl} represents the fourth rank material tensor while S_{kl} symbolizes the strain-displacement relations and can be given by

$$S_{kl} = \frac{(u_{k,l} + u_{l,k})}{2} \quad (3.4)$$

where the commas denote partial differentiation. Substituting Equation (3.4) into Equation (3.3) leads to the following expression

$$T_{ij} = C_{ijkl} \frac{(u_{k,l} + u_{l,k})}{2} \quad (3.5)$$

The indices i, j, k, l can take the values 1, 2 or 3 to conform with the x, y and z spatial directions convention. It is also assumed that the summation convention (Σ) over the indices will take place throughout.

Discretization, Energy Functional and Interpolation

In order to reach a numerical solution to the problem in question, the basic steps of FEM, as were previously described, are to be executed. The very first step in developing a finite element model is the *discretization* of the domain of interest. Typical discretization schemes using a set of simple straight-sided triangular or rectangular elements, are shown in Figures 3.2 (a) and (b). Nodal points situated along the interelement boundaries are also indicated in the Figures. Note that, whether triangular or rectangular elements are used, there usually exist inherent errors in modeling the curved edges of the domain. These errors are expected to be minimized when the elements are reduced in size. In general, the mesh should be selected to be relatively fine in areas where large gradients or slopes are foreseen. Gradual transition from the relatively fine mesh region to the relatively crude mesh region should take place.

The interior of these discretizing elements is described through *shape functions* given in terms of the displacement values at the nodes connecting them together. This leads to an *energy functional* which represents the total energy involved in the system [39]. The energy functional can be written as the sum of integrals over the areas of the elements in D , and as the sum of integrals over the corresponding line segments only on the Γ_2 portion of the boundary, since it is assumed that body forces are ignored and that the only external force present is the surface traction acting on the traction boundaries.

Once the stored energies, lost energy and the work done by the different external forces are accounted for, an expression for the energy functional can be written as

$$I(u_i, t) = \frac{1}{2} \int_D C_{ijkl} S_{kl} S_{ij} dv - \int_{\Gamma_2} u_i t_i ds + \int_D u_i \rho \ddot{u}_i dv \quad (3.6)$$

The first and third terms on the right-hand side of the equation represent the

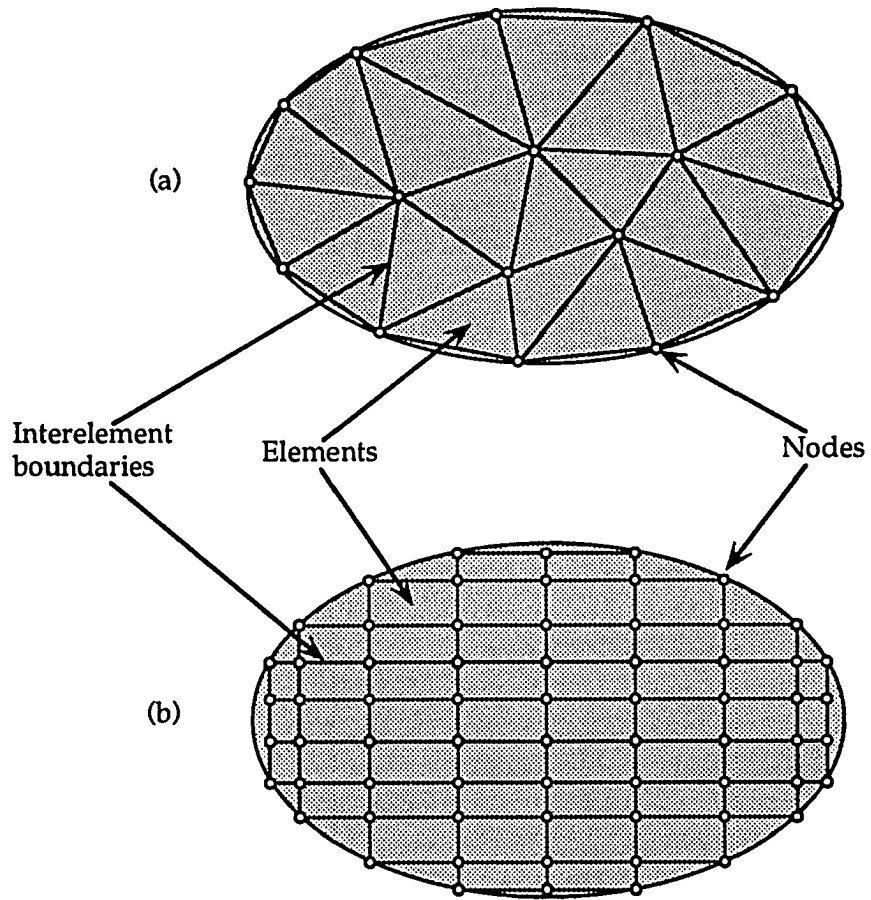


Figure 3.2. Typical domain discretization for the boundary value problem using linearly interpolated (a) triangular elements or (b) rectangular elements

total potential and kinetic energies, respectively, stored in the solid.

With a typical discretization as depicted in Figures 3.2 (a) and (b), the approximate solution $u(x, y, t)$ can be written in terms of a specially selected nodally-based set of *interpolation* or shape functions $n_i(x, y)$. These shape functions are used along with the discretization as in

$$u(x, y, t) = \sum_1^N u_i(t) n_i(x, y) \quad (3.7)$$

where the summation is carried out over the total number of N nodes per element.

The selection of the shape functions depends on the type of elements used (i.e. triangular, rectangular etc.) in the discretization process. When *isoparametric* elements (i.e. equal numbers of parameters are used to represent the geometry and the dependent variables) are used, the same interpolation functions or shape functions represent the dependent variable within the element as well as the shape or the geometry of the element. The most commonly used shape functions are polynomials and collectively referred to as the *quadratic interpolation functions* or, as often termed in numerical analysis, the *quadratic Lagrange interpolating polynomials* [76]. These are easily derived from a local elemental coordinate system.

Once the elemental shape functions are established, the energy functional (Equation (3.6)) can be evaluated for each of the elements in the discretized domain, and thus, the elemental energy functional can be written as

$$I_e(u_i, t) = \frac{1}{2} \int_{D_e} C_{ijkl} S_{kl} S_{ij} dv - \int_{\Gamma_{2e}} u_i t_i ds + \int_{D_e} u_i \rho \ddot{u}_i dv \quad (3.8)$$

where Γ_{2e} represents the portion of the traction boundary where element e is located. It is worth mentioning that if the element under study is not adjacent to the traction boundary or if that element is located in the middle of the domain, then the second term in Equation (3.8), for that specific element, vanishes.

In the following section, the minimization of the elemental energy functional is briefly developed. The assemblage process and the structuring of the FEM code are also considered.

Minimization and Assembly

As just mentioned above, the finite element methodology has converted the functional given by Equation (3.6) into the elemental functional given by Equation (3.8). The stationary value of this function is obtained by requiring the partial derivatives, with respect to each of the u_i , to vanish; that is

$$\frac{\delta I_e}{\delta u_i} = 0 \quad i = 1, \dots, N$$

leading to

$$\int_{D_e} \delta u_{i,j} C_{ijkl} u_{k,l} dv - \int_{\Gamma_{2e}} \delta u_i t_i ds + \int_{D_e} \delta u_i \rho \ddot{u}_i dv = 0 \quad (3.9)$$

Evaluating (3.9) for all elements in the domain and assembling all elemental computations, gives rise to a global matrix differential equation

$$[K] \{u\} + [M] \{\ddot{u}\} = \{F\} \quad (3.10)$$

where $[K]$ and $[M]$ are referred to as the global stiffness and mass matrices, respectively; and where their related elemental forms k_e and m_e are given by

$$k_e = \int_{D_e} \left(\frac{\delta n_I}{\delta j} C_{ijkl} \frac{\delta n_J}{\delta l} \right) d v \quad (3.11)$$

$$m_e = \int_{D_e} \rho n_I n_J d v$$

where $I, J = 1, \dots, N$, where N is again the total number of nodes per element.

In a similar fashion, $\{F\}$, which is a vector containing all loading values at any particular time t , can be represented by its respective elemental form, f_e , as follows

$$f_e = \int_{\Gamma_{2e}} n_I t_I ds \quad (3.12)$$

Note that the coefficients of both $[K]$ and $[M]$ depend on the interpolating shape functions and on both the longitudinal and the shear velocities inside the material [38, 50, 51]. Approximating $\{\ddot{u}\}$ by the explicit central difference formula

$$\ddot{u}_t = \frac{1}{\Delta t^2} (u_{t+\Delta t} - 2u_t + u_{t-\Delta t}) \quad (3.13)$$

yields the iterative form

$$\frac{1}{\Delta t^2} [M] \{u\}_t = \{F\}_{t-\Delta t} - \left([K] - \frac{2}{\Delta t^2} [M] \right) \{u\}_{t-\Delta t} - \frac{1}{\Delta t^2} [M] \{u\}_{t-2\Delta t} \quad (3.14)$$

The above equation constitutes the scheme followed for a direct computer code implementation for the vector $\{u\}$, which contains all the nodal displacement values, to be obtained at each time step Δt . Figure 3.3 shows a block diagram depicting the overall program structure of the finite element code [38, 39], that is capable of predicting quantitatively accurate displacement fields for elastic wave propagation in isotropic as well as composite materials.

As is seen in the following chapter, results of this program are used as a *Test Bed* to study the sensitivity of the synthetic aperture focusing technique (SAFT). The same data set is also used throughout the rest of this research to determine, from a simulation point of view, how effective the newly applied beamforming approach is at predicting A-scan signals in a nonuniform set of measurements.

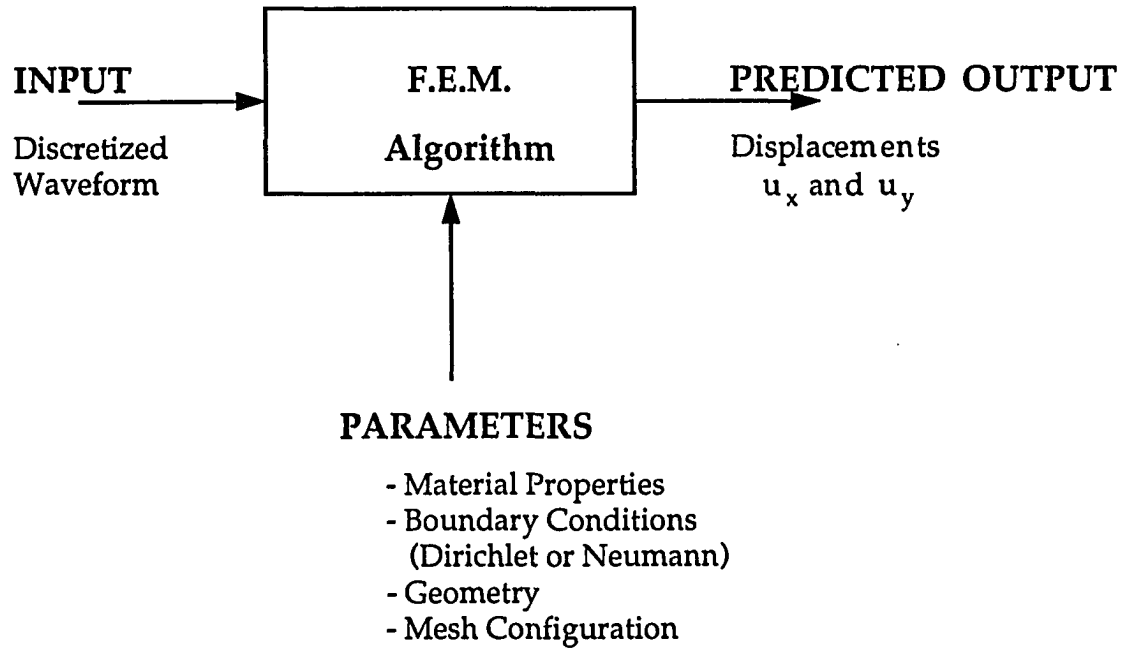


Figure 3.3. Overall schematic representing the finite element modeling process (Ludwig *et. al*, 1986)

CHAPTER 4. SAFT FORMULATION AND IMPLEMENTATION

Introduction

As mentioned in Chapter 2, various ultrasonic imaging techniques have been developed to enhance the performance of conventional ultrasonic nondestructive testing (NDT). The synthetic aperture focusing technique (SAFT) is one of these methods with a primary goal to detect flaws and defects within structural and functional components and to identify them in terms of shape, size, location and orientation.

The Concept of SAFT

SAFT is based on the simple concept of collecting ultrasonic data waveforms, known as elements, from a scanning transducer, and then processing these elements as a unit, known as an aperture, in a fashion that can be described as shifting adjacent elements with respect to the middle element of the aperture, summing all elements point by point across their lengths and then placing the summing vector at the center of the chosen aperture [7]. It is important to note that, in order for the above procedure to show good results, as much information as possible has to be gathered about the discontinuity. Therefore, it is desirable that the material under test be insonified with a very broad beam so that the discontinuity is often seen by the transducer as it moves over the surface of the test piece. A focussed transducer is used for that purpose [7, 8], where the sound beam converges on the material's top surface and then diverges at a very wide angle as it propagates through the body of the specimen, as depicted by Figure 2.5. As the transducer scans over the material, a series of A-scans are collected. One characteristic aspect of these A-scans is that they produce a family of echoes forming the locus of a hyperbola having a concave shape that is proportional to the depth of the point reflector in the material [7, 46]. The apex of that hyperbola occurs when the transducer is located directly above the defect. If the center of the aperture (COA) does not coincide with the occurrence of

a strong reflector, a weak destructive interference will result from the shifting/summing process; otherwise, a strong constructive interference will result, as shown in Figure 4.1.

The Formulation of SAFT

It is of a crucial importance to note that the procedure just described actually takes place at only one specific depth site in the material under test. But what if the flaw location is not known (which is usually the case), and what if we are dealing with a test piece which contains various defects having different shapes and located at different depths? One way to get around this problem is to imagine a "focus grid" made of x_M by y_P pixels and inserted vertically inside the material just below the scanning line (Figure 4.2).

Each of the pixels represents a particular depth. The distance between transducer t_i and pixel (x_j, y_k) can be calculated by

$$L_{t_i x_j y_k} = \sqrt{(x_j \cdot \Delta x - t_i \cdot \Delta s)^2 + (y_k \cdot \Delta y)^2} \quad (4.1)$$

where

$\Delta x = \Delta y =$ pixel spacing

$\Delta s =$ separation between the transducers

Knowing the longitudinal stress wave velocity, v_c , in the material, the time-of-flight, T_{ijk} , can be determined by

$$T_{ijk} = 2 \cdot \frac{L_{t_i x_j y_k}}{v_c} \quad (4.2)$$

where the 2 factor represents the round trip travel of the ultrasound beam to and from each pixel in the grid.

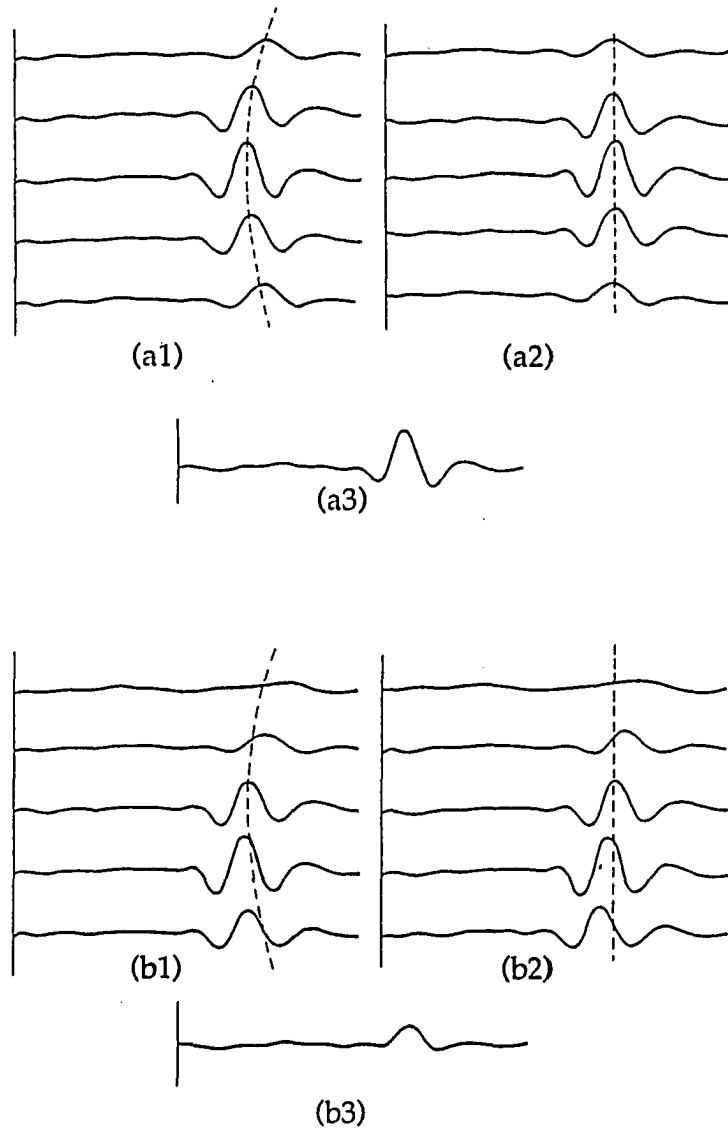


Figure 4.1. Shifting and coherent summing of raw A-scans.
 (a1) raw A-scans with COA over a reflector
 (a2) time-shifted A-scans
 (a3) correlated A-scans (strong interference)
 (b1) raw A-scans with COA not over a reflector
 (b2) time-shifted A-scans
 (b3) correlated A-scans (weak interference)
 (Seydel, 1982)

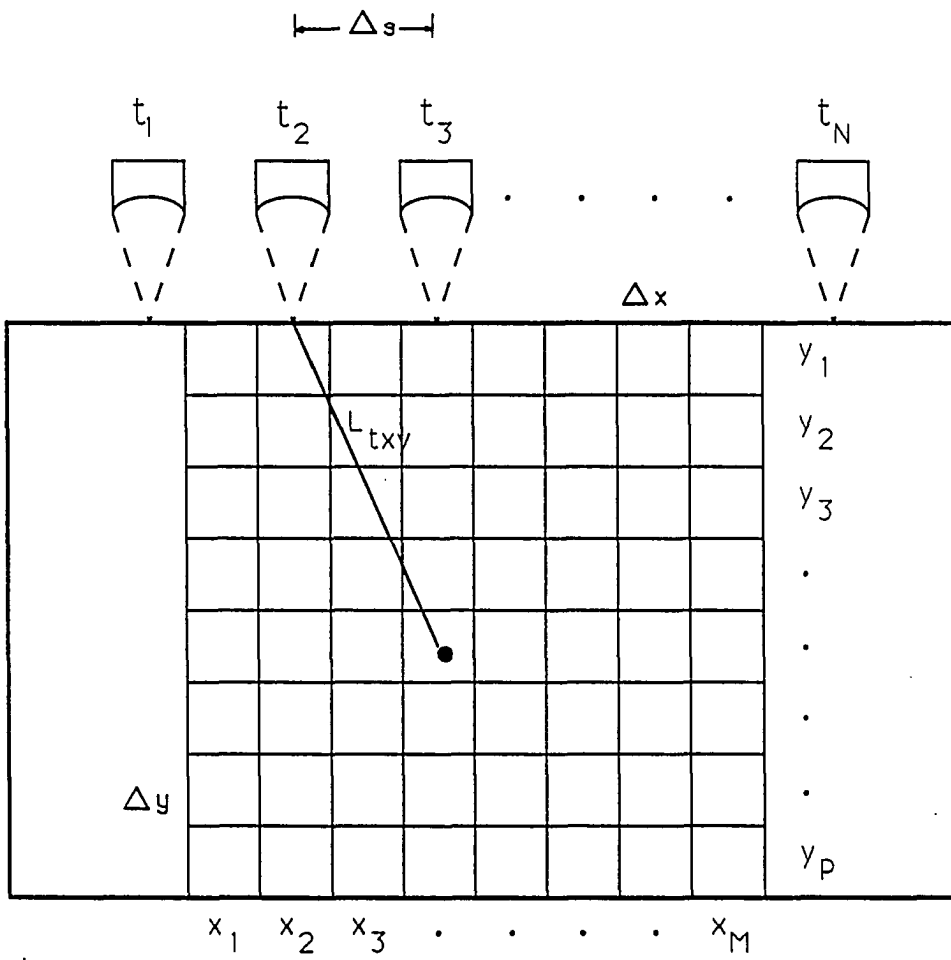


Figure 4.2. Configuration of the "focus grid"

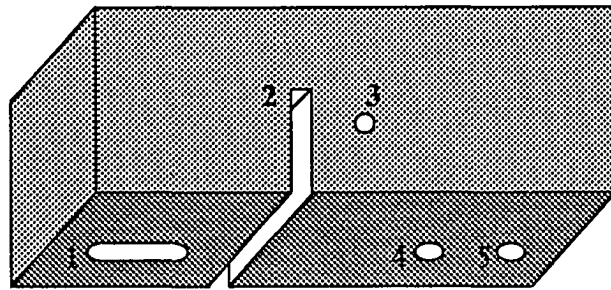
an improvement in digital resolution, by bit dithering, beyond the 8 bits obtainable in single-shot mode. The dynamic range of the final data ranged from 72.25 to 84.29 dB and was obtained by averaging each time domain point into a 16 bit word inside the sampling oscilloscope. These data are then stored onto hard disk before the acquisition of another trace begins. The acquisition program permits these data to be stored in a variety of formats, ASCII, binary, Welch, or Huffman compressed. For the scans, these data are compressed using Huffman coding and then stored in binary format. This compression is absolutely necessary considering the amount of data which is required for the SAFT reconstruction as it is presently done. For the purpose of acquiring a basic understanding of the SAFT algorithm, the time domain version has been developed. In future work, it is planned to accomplish the time shifts required for SAFT by using the Fourier shift theorem; namely, FT-SAFT.

As implied above, the data are stored off-line for later analysis. This analysis is done on a Stellar™ workstation.

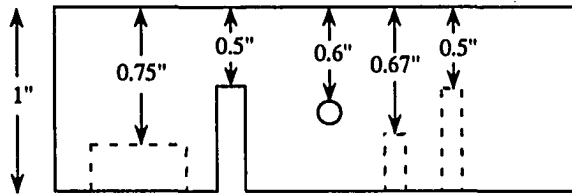
Two test blocks were prepared for the experiment. Since the effect of the SAFT algorithm was to be checked and compared on two different kinds of materials, Plexiglass™ and Graphite/Epoxy (0°/90° layup) specimens were used. The dimensions of both blocks were the same (1 x 1 x 4 inches). A set of five holes were drilled in each block, having the same size, shape and location. An elongated flat-bottom hole, a slot hole, a side-drilled hole and two flat-bottom holes with different depths, were machined in both blocks. The sizes of the holes (width and depth) varied, as shown in Figure 4.3.

Description of the SAFT Algorithm

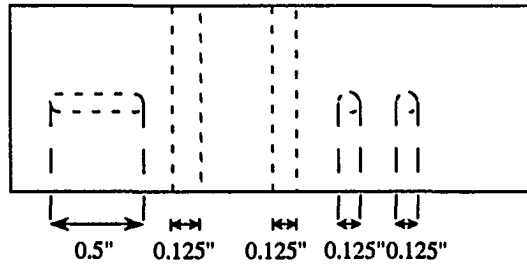
A block diagram (Figure 4.4) was put together in order to best describe the SAFT algorithm. Some SAFT processing parameters are first required. These include parameters such as the focal distance of the transducer, the



1. Elongated Flat-Bottom
2. Slot across material
3. Side-drilled across material
4. Flat-bottom
5. Flat-bottom



Side View



Top View

Figure 4.3. Plexiglass™ and Graphite/Epoxy samples geometry

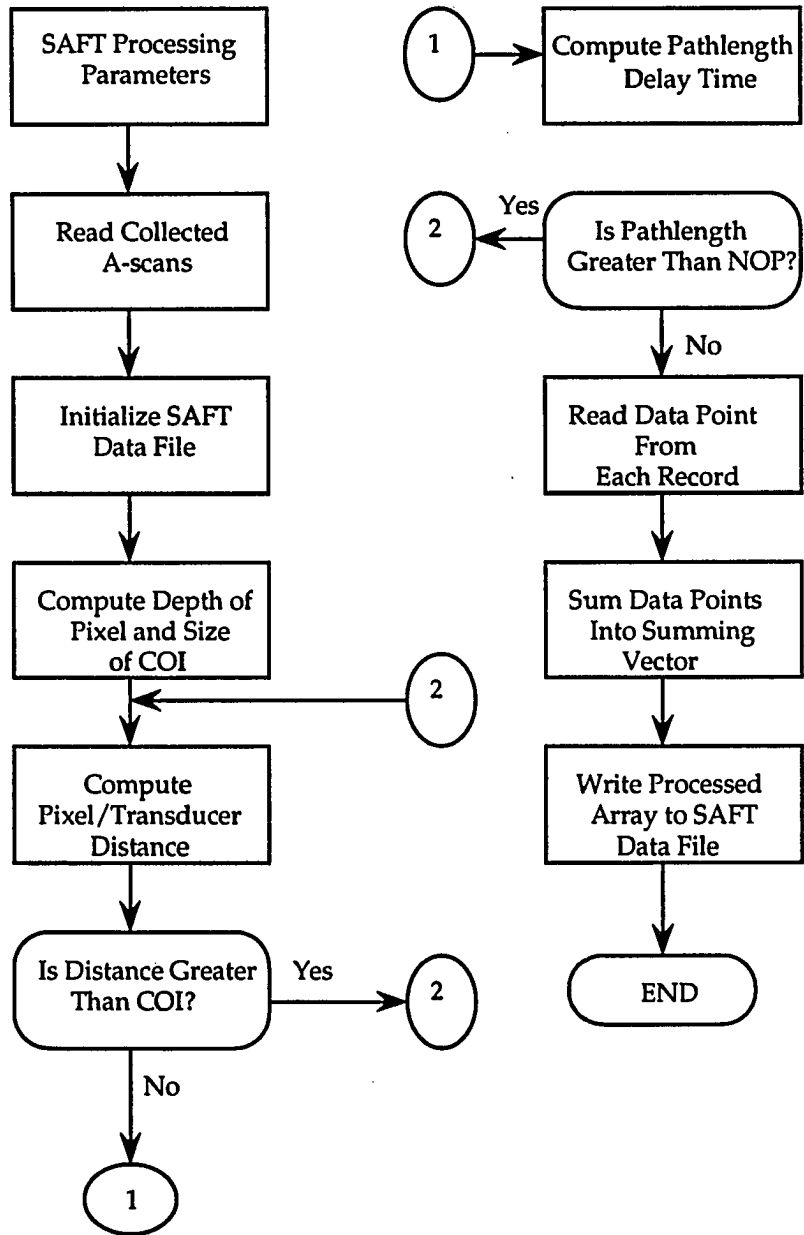


Figure 4.4. SAFT programming block diagram

velocity of sound in the coupling medium, the velocity of sound in the test specimen, the number of data points (NOP) in a pulse-echo, the number of records collected, the incremental distance between the transducer positions, the sampling frequency, and the thickness of the test piece. Once these parameters are entered, the collected A-scans are read. The SAFT summing array is then initialized to zero. The subsequent step is the actual SAFT processing loop. Here the different pathlengths traversed by the ultrasound signal from each transducer position to every pixel in the focus grid and back, are computed. It should be noted that these pathlength values represent the phase delays introduced by the transducer/pixel geometry. Each phase delay is used to read its respective signal value from each of the collected A-scans. All signal values are then summed into the summing vector. If a strong signal is present, a constructive image is produced, indicating the presence of a flaw, crack, or delamination; otherwise, the summing vector will contain a weak destructive interference.

Analysis of Results

Different SAFT tests were performed on the collected A-scans, in order to provide both qualitative as well as quantitative comparisons between the Lucite and the Graphite/Epoxy samples. The resulting SAFT-processed B-scan images, for both the Plexiglass™ and the Graphite/Epoxy samples, are shown in Figures 4.5 (a) and (b), after a -6 dB threshold has been applied. The darkest grey-level in both images represents the zero amplitude. The horizontal direction (left to right) corresponds to the transducer movement over the surface of the test specimen, while the vertical downward direction corresponds to the depth within the sample.

The results clearly indicate the presence of all five targets, except that the elongated flat-bottom hole located at the far left of the images, three quarter of the way down from the top surface, is not too obvious. Two reasons for this discrepancy can be considered. One, the transducer was not positioned exactly over that particular target, during the collection process of the data,

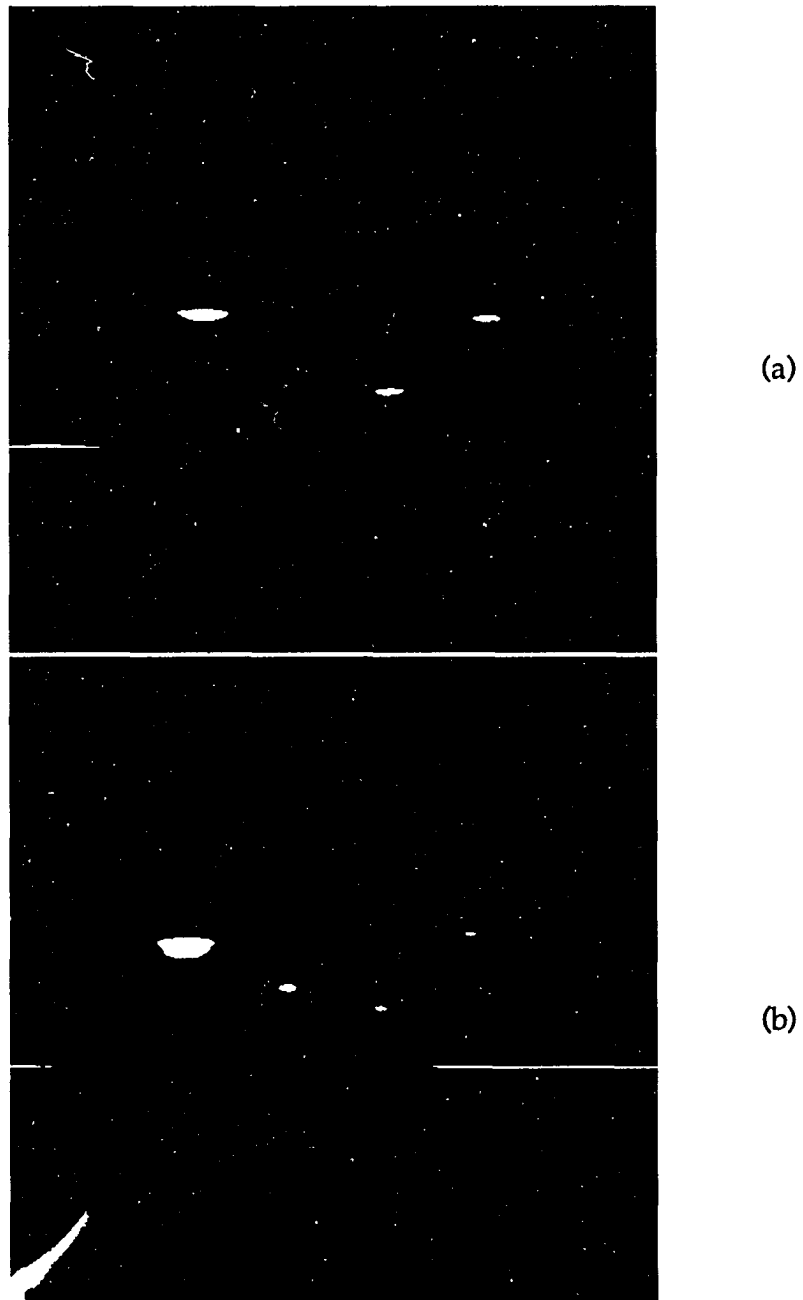


Figure 4.5. Rectified (-6 dB threshold) SAFT B-scan images for
(a) Plexiglass
(b) Graphite/Epoxy

but rather was positioned in such a way that the beam of insonification did not completely cover that flaw, giving rise to a weak reflection. Two, the location of that target, due to its depth, was shadowed by the adjacent slot, and therefore was not completely illuminated by the beam from the various transducer positions along the line scan, resulting in a minimal amount of information about that hole.

One note worth mentioning about the SAFT algorithm, deals with the type of computational approximations involved during the SAFT image production. As seen from the block diagram in Figure 4.4, two conditions, depicted by the questions inside the diamond shaped blocks, were being checked for throughout the shifting/summing process. When the time-of-flight is computed for every transducer position to each pixel in the "focus grid", as described earlier, the first condition requires that every pixel that does not lie within the cone of insonification (COI) of the propagating beam, be excluded from the computation process. Knowing the wave velocity, v_0 , in the coupling medium, the radius, r , and the focal distance, f , of the transducer, a formula can be derived [47] to calculate the width of the COI inside the material, as given by the following expression

$$\text{cone width} = 2D \cdot \tan \left\{ \sin^{-1} \left[K \cdot \sin \left(\tan^{-1} \left(\frac{r}{f} \right) \right) \right] \right\} \quad (4.4)$$

where D = depth in the "focus grid"
and $K = v_c / v_0$

Once the above condition is satisfied, the second condition further checks to see if the computed pathlength, corresponding to that particular pixel lying within the COI, is less than the number of points (2048 in this case) collected on a per A-scan basis.

Tables 4.1 (a) and (b) show a comparison between the actual measured targets (depth location and width size) and measurements computed from the

Table 4.1. Comparison between actual and experimental depth and width dimensions. (a) Plexiglass™ (b) Graphite/Epoxy

(a)	Depth Dimension		Width Dimension	
	Actual	Post SAFT	Actual	Post SAFT
Flaw Type				
Elongated F.B.	0.75"	0.76"	0.5"	0.48"
Slot	0.5"	0.52"	0.125"	0.125"
Side-Drilled	0.6"	0.61"	0.125"	0.11"
F. B. #1	0.67"	0.66"	0.125"	0.125"
F. B. #2	0.5"	0.52"	0.125"	0.125"

(b)	Depth Dimension		Width Dimension	
	Actual	Post SAFT	Actual	Post SAFT
Flaw Type				
Elongated F.B.	0.75"	0.767"	0.5"	0.56"
Slot	0.5"	0.52"	0.125"	0.14"
Side-Drilled	0.6"	0.603"	0.125"	0.09"
F. B. #1	0.67"	0.644"	0.125"	0.14"
F. B. #2	0.5"	0.493"	0.125"	0.14"

SAFT images. Table 4.1 (a) gives the results for the Lucite piece and Table 4.1 (b) gives data for the composite sample. These data were obtained using a ruler and estimating the size of the flaws relative to the size of the image. This method of sizing the defects is not very accurate but was the only available means at the time of the experiment. Both tables indicate that SAFT did prove to be an adequate tool at providing enough information about the targets. However, due to its constraint, as far as the nature of the sample under test is concerned; that is, the material must be homogeneous and isotropic, SAFT was not quite successful in reconstructing defects in the composite block. As is discussed in the following section, one way to improve the quality of the conventional SAFT when applied to composites, is to consider the different velocities of the ultrasonic wave in the various directions of propagation, instead of assuming the same velocity in all directions.

Anisotropy Effects

The performance of conventional reconstruction methods (i.e., SAFT), used in ultrasonic NDT, has been quite impressive as far as the detection and sizing of flaws and defects in various materials is concerned. However, as previously stated, assumptions were made about the type of material used in the restoration process. The two major assumptions dealt with representing the medium as isotropic and homogeneous. These suppositions are less likely to hold true when composite materials are tested. This section describes a scheme that can be used in this situation, as a modified version of the formally presented SAFT algorithm.

As already discussed, the manner in which SAFT can be applied to composites is not as trivial as the isotropic/homogeneous case of reconstruction. Although a preliminary application of the conventional SAFT algorithm [47] appeared to have given acceptable results, more work had to be devoted in order to improve the quality of the SAFT-processed images, for a complete defect diagnosis and a thorough understanding of

materials with a higher degree of complex anisotropy.

Consider the situation where an infinitely long fibrous composite specimen (Figure 4.6) is immersed in water and subjected to a longitudinal plane wave, originating from a focused transducer, and propagating at normal angle of incidence. The wave interaction with the material results in displacement fields which can be described by stress-strain relations [49]. The non-constant nature of the phase velocities, v_p , in the direction of propagation of the corresponding plane-wave vectors for anisotropic media, results in three non-spherical slowness curves (Figure 4.7). The outer and the middle curves represent the two shear waves and the innermost one corresponds to the longitudinal wave. If we now consider looking at the latter slowness surface (Figure 4.8), one can determine the various group velocities, v_g , which correspond to the various phase velocities in the different directions of propagation (θ_i) as shown in Figure 4.6. These represent the family of stress-wave velocities in those directions, needed when SAFT is applied. Group velocities can be determined by projecting their vectors, on the surface of the slowness curve, normal to the corresponding phase velocities vectors. Using some simple trigonometry, the group velocities can then be computed by the following

$$v_p = v_g \cdot \cos \varphi_i \quad (4.5)$$

where φ_i is the angle formed between the group velocity and the phase velocity vectors. From this equation, a look-up table can be constructed for all possible angles in the field of view of the transducer. The calculated group velocities could then be used along with Equations (4.1 - 4.3) to reconstruct the plane in which the focusing takes place.

Note that this method is only a proposal and has not been applied yet to either a simulation nor a laboratory experiment, but will be further studied in future work.

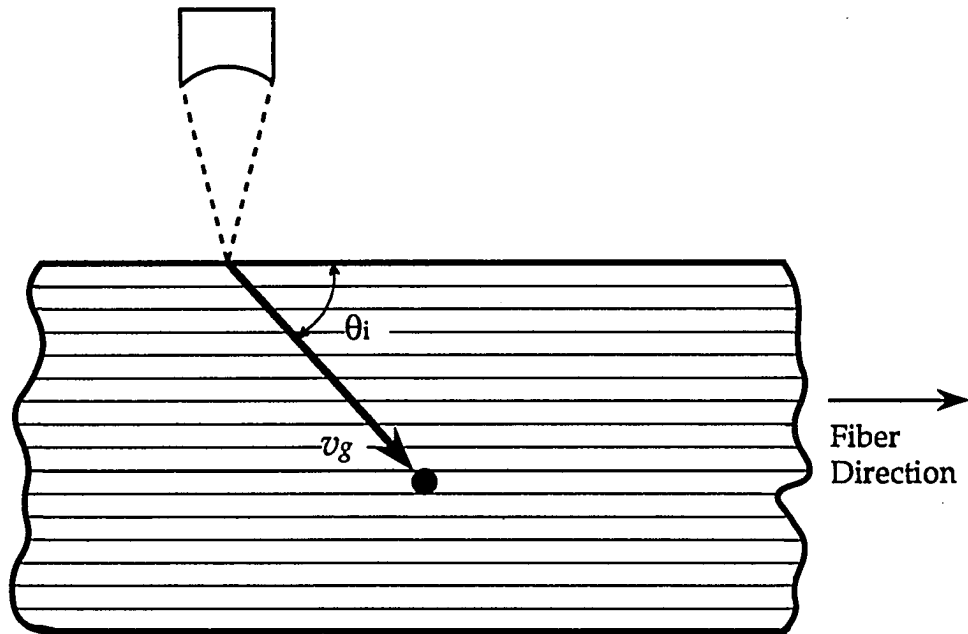


Figure 4.6. Example of a fibrous composite material under test

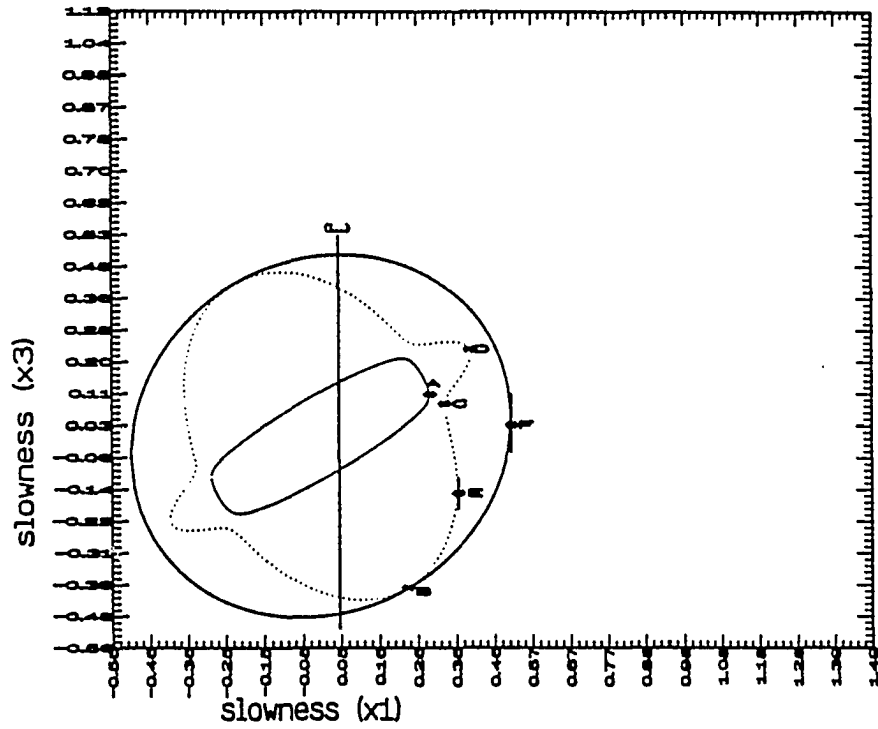


Figure 4.7. Example of slowness curves when the fiber direction is at 60 degrees from the x_1 axis (Chedid-Helou, 1991)

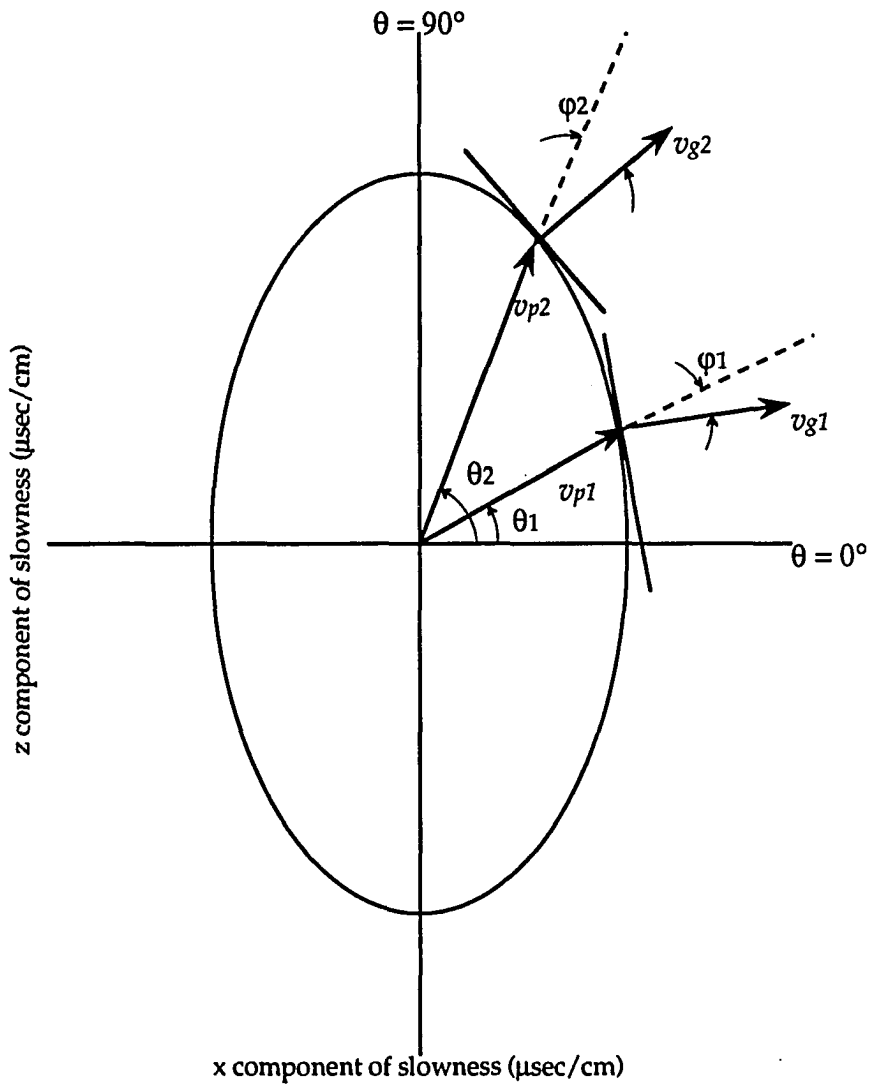


Figure 4.8. Slowness surface for a longitudinal wave

Finite Element Simulation

A particularly important aspect in ultrasonic NDT research is the detailed understanding of the ultrasonic energy interaction with the material under test. Since the finite element method (FEM) can be used, as seen in the previous chapter, to solve the hyperbolic partial differential equations [38, 39, 50, 51], which govern ultrasonic wave propagation, FEM solutions can be used to mimic a SAFT measurement.

The goal of this section is to use FEM in order to simulate the action of a transducer array, and then use the resulting signals, along with SAFT, to reconstruct the reflectors under study.

To repeat, in brief, what was said earlier, the finite element method is based on two fundamental principles: (1) the weak formulation of a boundary value problem; and (2) the domain decomposition into subdomains or elements [39]. Equations (3.1 - 3.8) constitute the scheme followed for the direct computer code implementation for the computation of the displacement vectors, and hence, the simulated A-scans at each transducer time step.

For the purpose of describing the overall picture, the sample to be modeled is presumed to be of an isotropic nature (i.e., Plexiglass™) and of infinite extent. Practical considerations dictated a block geometry of 1x1x2 inches. A set of two slots (0.5 inch apart) are located just beneath the line scan. The width of the slots is the same (0.125 inch); however, their depth from the surface of the sample varied (0.5 inch and 0.75 inch). Figure 4.9 shows the geometry of the sample. The sample is considered to be placed in a water tank. The transducer is simulated as being 1 inch in diameter, having a center frequency of 2 MHz, and producing a very broad spectrum. A transducer separation of one half a wavelength ($\lambda/2$) is used to meet the criterion required by SAFT. A line scan of 1 inch would then require 38 time domain A-scan traces to be produced. In order to achieve this requirement, the finite

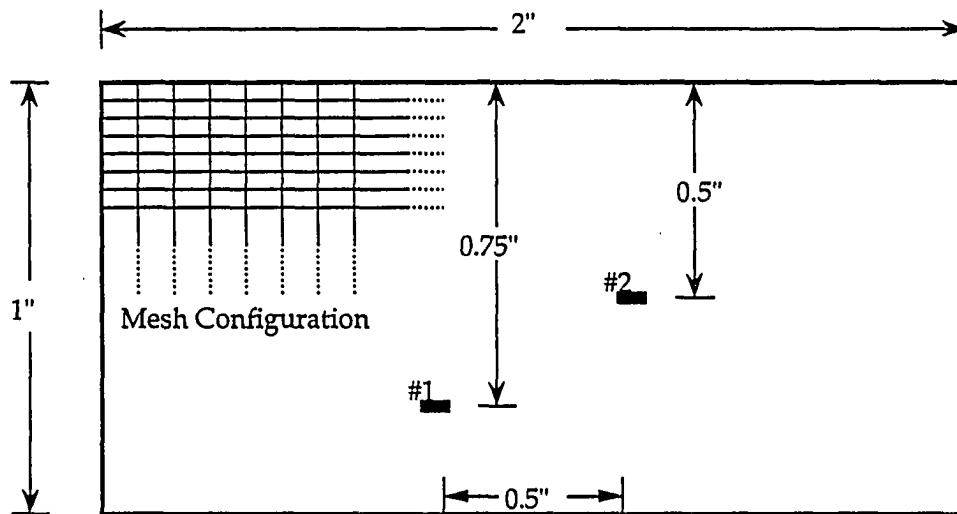


Figure 4.9. Sample geometry used in the finite element simulation

element program is set with the following parameters:

Longitudinal wave velocity	= 2740 m/s
Shear wave velocity	= 1360 m/s
Density (Plexiglass™)	= 1188 Kg/m ³
Number of elements	= 178802
Number of nodes	= 179700
Number of unknowns (Displ.)	= 359400
Number of points/A-scan	= 700

As expected, the program provided a set of data representing the displacements in the longitudinal and lateral directions at each of the 179,700 nodes. Figure 4.10 depicts some of the resulting longitudinal A-scans, after the front and back walls have been removed. Theoretically, only these longitudinal displacements are needed for the SAFT algorithm, since the lateral displacements do not carry much information to enhance the resolution of the medium under study. Nevertheless, once the data was acquired, it was found out that there might be enough information in the raw lateral displacements, for the SAFT to be not only capable of reconstructing the image from the longitudinal displacements, but also from the lateral displacements too.

Figures 4.11 (a) and 4.12 (a) show the resulting raw images produced by the FE code. Figure 4.11 (a) is the result of the longitudinal displacements, while Figure 4.12 (a) is obtained from the lateral displacements.

The final step of this study is to run the FE data through the SAFT algorithm in order to further focus on the flaws and to be able to classify them in terms of shape, size, orientation and location.

Since the scanning process of the sample took place in one direction and not in a 2-D fashion (i.e., raster scan), then 2-D reconstruction (B-scan) instead of 3-D (C-scan) is expected to result from the SAFT processing. This fact, then, provides us with a flaw characterization in terms of size, orienta-

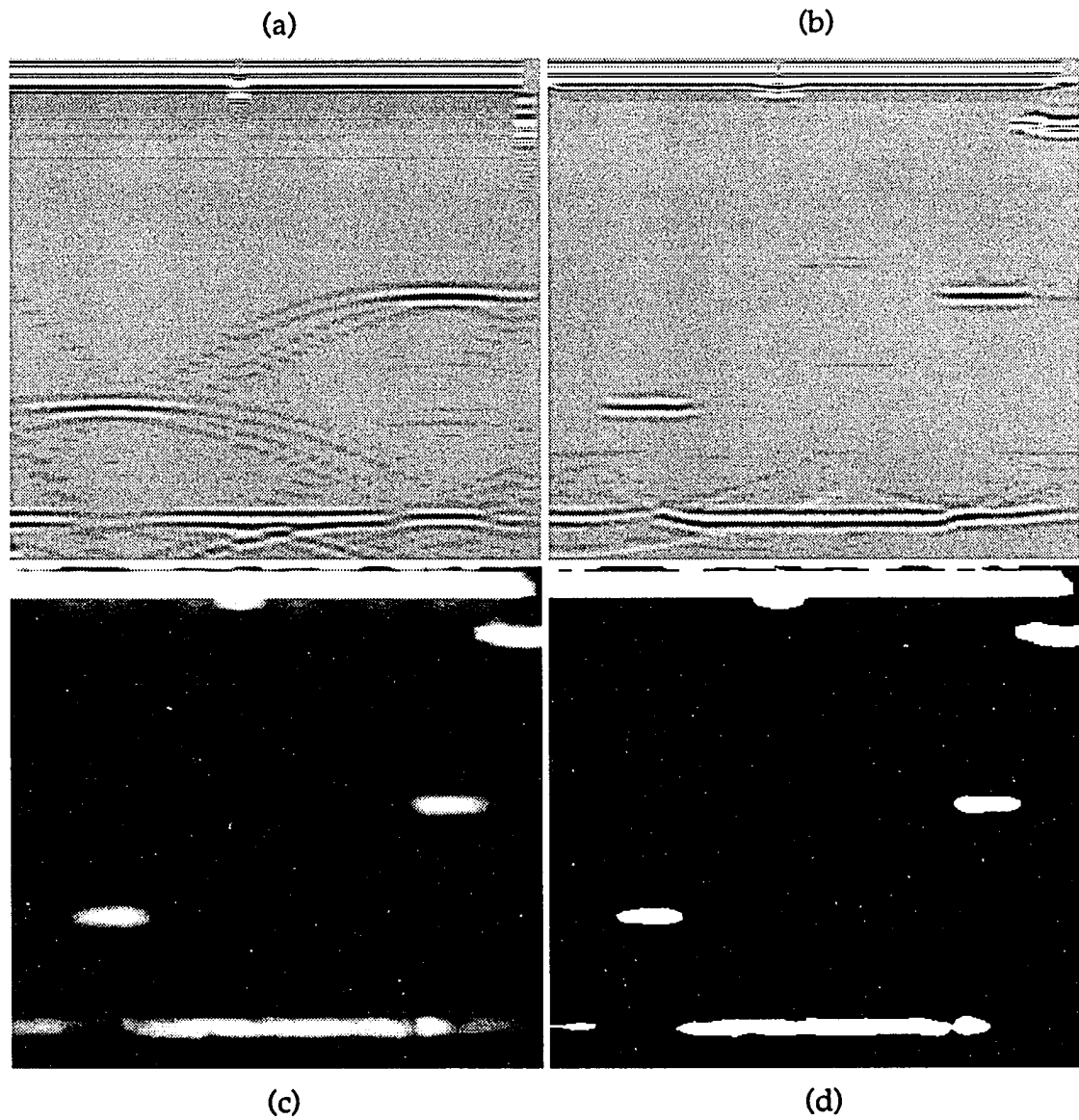


Figure 4.11. Finite element longitudinal displacements
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

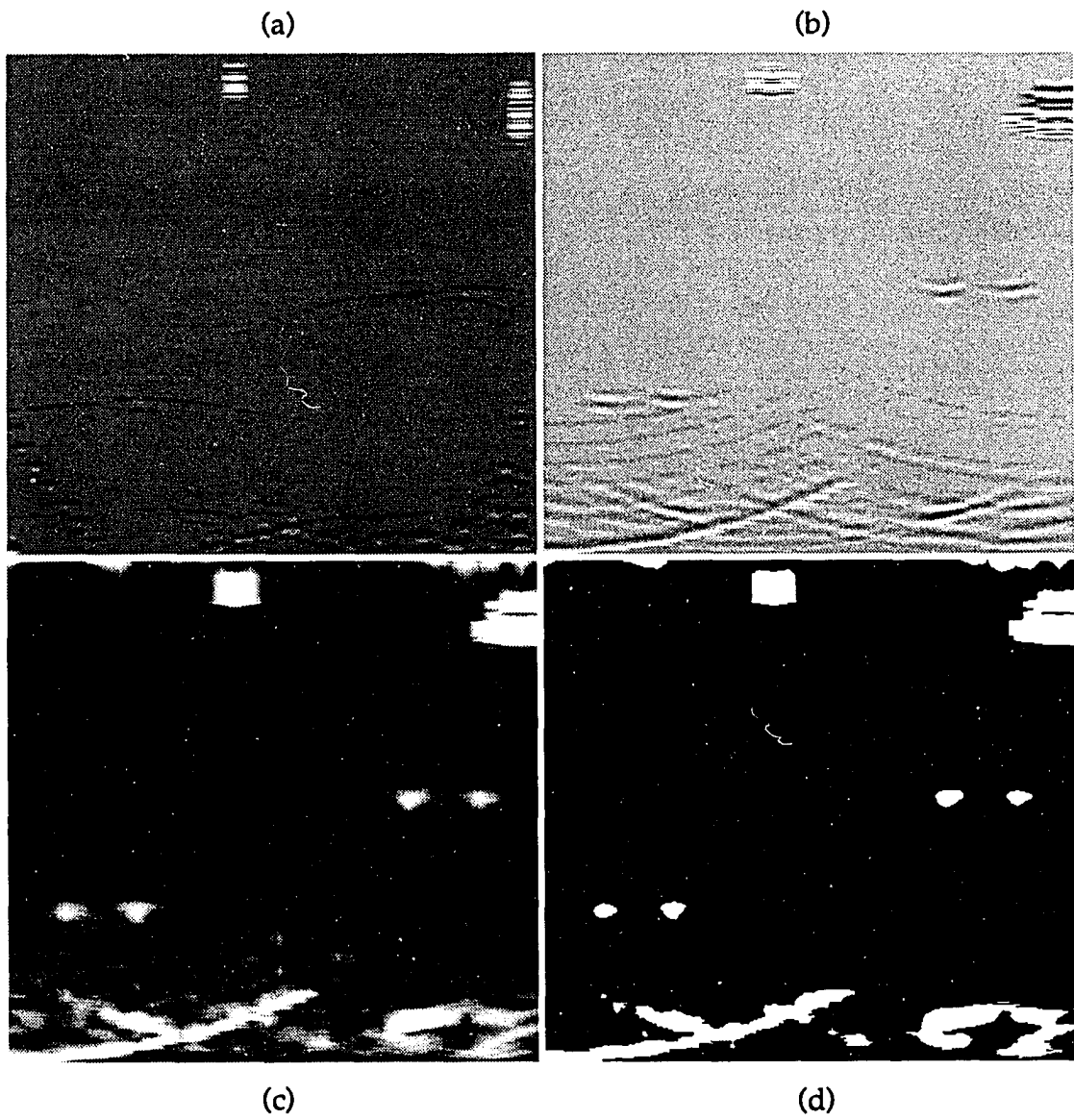


Figure 4.12. Finite element lateral displacements
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

tion and location only.

Once the data set was collected, different SAFT tests were performed on the A-scans in order to, once again, portray both qualitative as well as quantitative assessments of the actual and of the restored samples. The resulting B-scan images are shown in Figures 4.11 (b) and 4.12 (b), for the longitudinal and the lateral cases; respectively. Again the darkest grey-level represents the zero amplitude. The horizontal direction (left to right) corresponds to the transducer movement over the surface of the test specimen, while the vertical downward direction corresponds to the depth in within the sample.

The results shown, clearly indicate the existence of both slots and that the SAFT process has increased the resolution of these slots. Figures 4.11 (c) and 4.12 (c) show the same images as in Figures 4.11 (b) and 4.12 (b), after the magnitude of the images has been calculated to produce rectified versions without loss of resolution.

In order for the performance of SAFT to be evaluated, as far as sizing of the flaws is concerned, a 6 dB threshold has been applied to Figures 4.11 (c), and 4.12 (c). The resulting images are shown in Figures 4.11 (d) and 4.12 (d). Again, the superiority of the SAFT technique is indicated. Table 4.2 shows a comparison of the actual measured targets (depth and width) and those estimated from the FE SAFT images. Note that this particular sizing is based on the rectified longitudinal images and not on the rectified lateral images, although the latter would give a similar measurement determination.

The application of SAFT, using the finite element modeling technique, has been described. Results indicate the validity of the FEM model as a *Test Bed* for SAFT sensitivity studies and point out the possible use of lateral displacement data for SAFT reconstruction.

Table 4.2. Comparison between actual and FEM SAFT dimensions in both the longitudinal and the lateral directions

	Depth Dimension		Width Dimension	
	Actual	FEM SAFT	Actual	FEM SAFT (-6 dB)
SLOT #1	0.75"	0.76"	0.125"	0.125"
SLOT #2	0.50"	0.51"	0.125"	0.125"

CHAPTER 5. BEAMFORMING: A DECONVOLUTION ALGORITHM

The general definition of the term *Beamforming* refers to the reception of energy propagating in a particular direction while rejecting energy propagating in other directions [16, 17]. Signals within the beam are transmitted with minimum distortion, and those lying outside the beam are isolated and attenuated. This form of signal processing is *adaptive* in the sense that it employs a certain weight adjustment algorithm to minimize the output variance in the desired look direction. Some of its practical applications include [83] radar, sonar, geophysical exploration, astronomy and astrophysics, and biomedical image and signal processing. From an ultrasonic NDT viewpoint, beamforming can be interpreted as spatial filtering which operates on the outputs of an array of transducers in order to enhance the amplitude of a coherent wavefront, emanating from a certain direction, and propagating in a medium such as Plexiglass™ or Graphite/Epoxy. In fact, these transducers provide a means of *sampling* the received echoes in space [83], and the collected outputs, at any specific instant of time, can be interpreted as a *snapshot* of the medium under test. Therefore, in the case of a line-scan, this data snapshot is similar to a set of tap inputs present in the design of a finite-impulse-response (FIR) filter [16, 83] at a particular instant of time.

The following section describes the concept of beamforming and introduces some of the terminology that is pertinent to this research work.

Description of General Beamforming

Conventional time-domain beamforming, in its simplest form, such as SAFT, is accomplished by appropriately delaying and adding the outputs of an array of transducers, as indicated in Figure 5.1. In general, the delays are correlated with the anticipated propagation delays of every individual pressure field incident from a specific direction. The discrete implementation of this beamformer requires the sampling of the time-domain signals at a rate

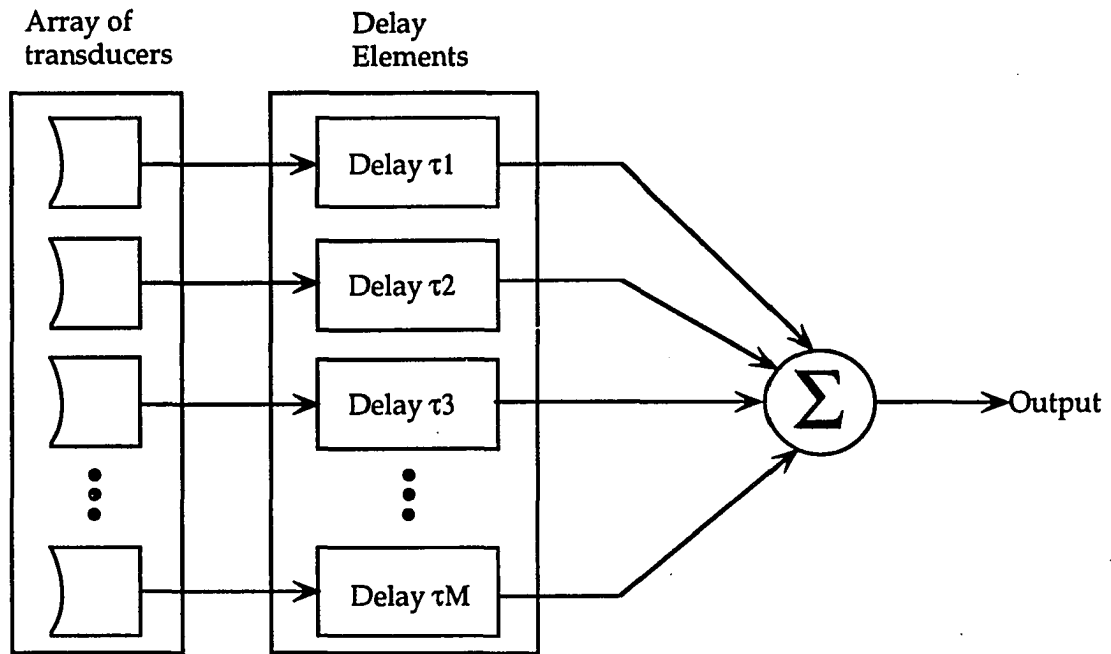


Figure 5.1. Delay-and-sum beamformer structure (SAFT)

consistent with the Nyquist criterion in order for the reconstruction process to be successful. This is discussed shortly in this section. The family of signals are then time delayed and summed in the beamforming operation. Note that prior to this operation, the outputs of the transducer are weighted so as to improve the beam's spatial response. These weights are sometimes assumed to be unity for simplicity, as is the case in Figure 5.1. This type of beamforming systems is called the *delay-and-sum* beamformer [17, 83]. As an example, in NDT, if a sample containing a defect is scanned, the average power at the output of this beamformer (SAFT) is maximized when the center of the aperture lies directly above the target (i.e., the defect is completely covered by the steering beam). As seen in Chapter 4, one of the major drawbacks of the SAFT beamformer, however, is that its performance weakens in the case when interfering signals are present, such as those signals resulting from the effect of anisotropy.

Due to the fact that the delay-and-sum beamformer response is highly dependent not only on the power of the incoming wavefront, but also on the interference contributions from undesirable sources, a different technique used for solving beamforming problems, adaptively, was proposed in 1969 by Capon [84]. The design of this newly developed beamformer is based on two major requirements that the beamformer has to satisfy. These two requirements are (1) the target being scanned is always covered by the steering beam, and (2) the effects of noise and interfering signals, from other sources in the field of view, are to be minimized.

In order for these requirements to be met, a $(M \times 1)$ *weight vector* W is chosen to minimize the variance (i.e., average power) of the beamformer output. In its turn, the chosen weight vector has to satisfy the condition

$$W^H S(\phi) = 1 \quad (5.1)$$

where $S(\phi)$ is a $(M \times 1)$ *steering vector*. The superscript H represents Hermitian

(i.e., complex conjugate transpose). The above condition (5.1) ensures that, for any *look* direction, the target signal is always covered by the beam, by maintaining a constant array response, regardless of the assigned values for the weight vector coefficients. The steering vector $S(\phi)$ can be expressed in terms of its elements as

$$S(\phi) = \begin{bmatrix} 1 & e^{-j\phi} & \dots & e^{-j(M-1)\phi} \end{bmatrix}^H \quad (5.2)$$

The angle ϕ can be determined by the direction of the target via the following relationship

$$\phi = 2\pi \frac{d}{\lambda} \sin \theta \quad (5.3)$$

where d is the spacing between adjacent transducer positions, λ is the wavelength, and θ is the incidence angle of a plane wave, measured relative to the normal of the linear array. In other words, as an ultrasonic plane wave reflects from a discontinuity or a defect within a material and propagates toward a linear array of transducers, a spatial delay of $(d \sin \theta)$ exists between the received echo signals at any pair of neighboring transducers [83]. This spatial delay can be converted to an electrical angular shift as described by the above expression (5.3). Figure 5.2 describes the above plane-wave-array geometry. The incidence angle θ can only be contained within the $[-\pi/2, \pi/2]$ range. Therefore, according to Equation (5.3), ϕ can be permitted to vary only within the $[-\pi, \pi]$ range. This fact, then, restricts the spacing, d , between the transducer positions to be less than half of the wavelength value (i.e., $\lambda/2$), in order for a proportional correspondence between ϕ and θ , to exist [83]. This condition constitutes the basis of the sampling theorem, namely, the Nyquist criterion. The constrained variance minimization just described can be implemented using an algorithm referred to as the *minimum-variance distortionless response* (MVDR) [84, 85]. A basic MVDR adaptive beamformer

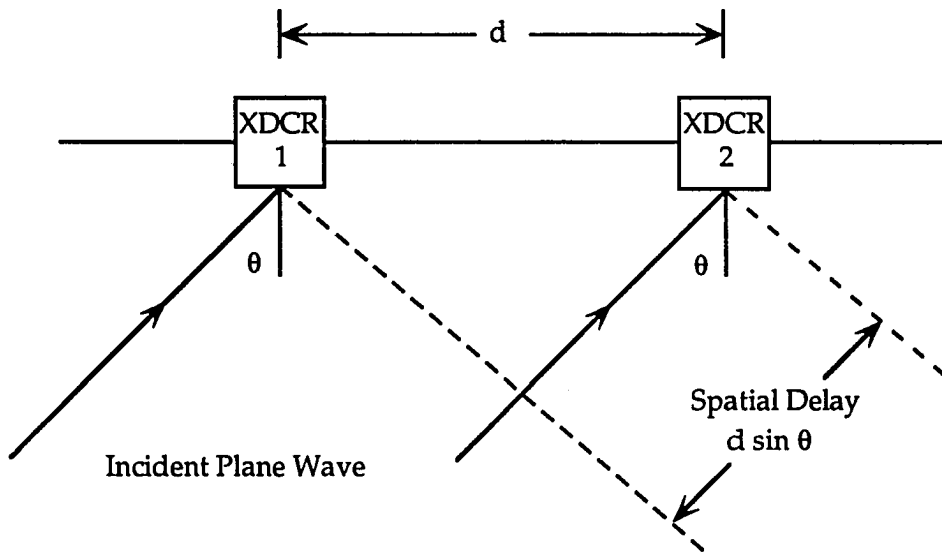


Figure 5.2. The geometry of a plane wave incident on a linear array and its effect on the spatial delay incurrence

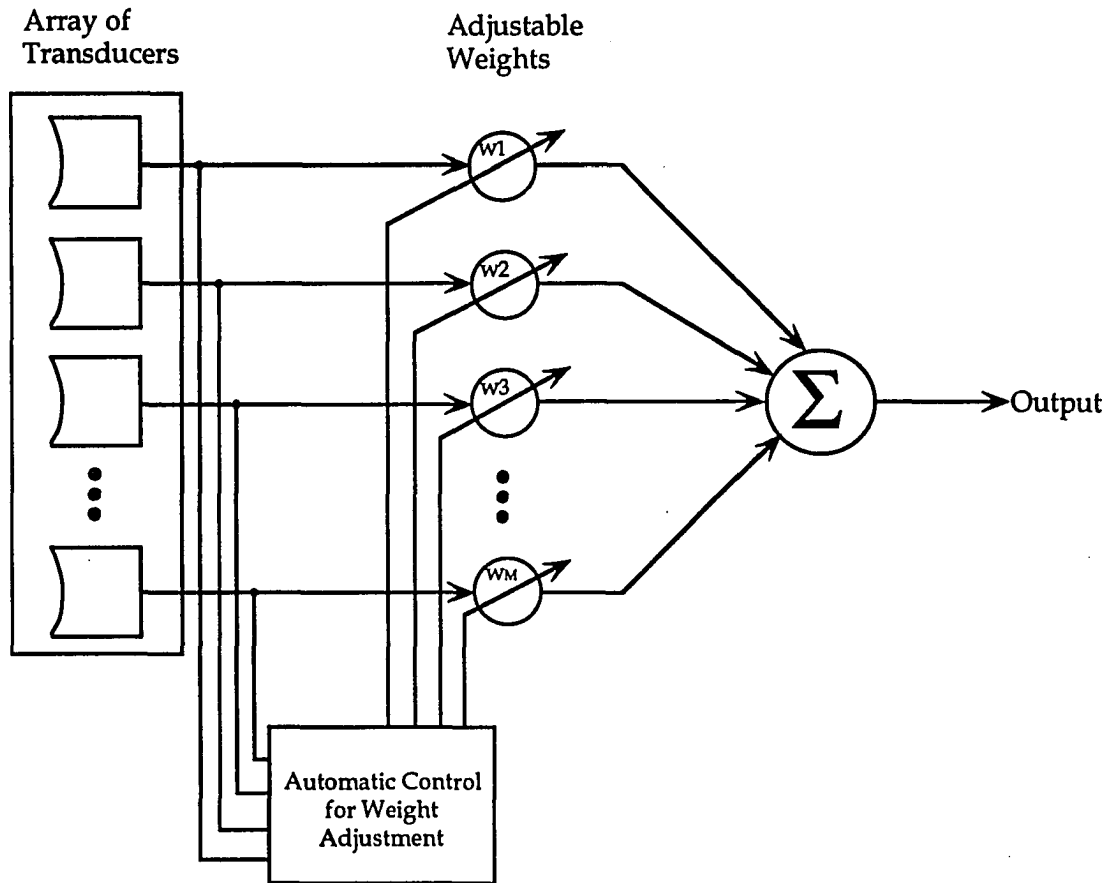


Figure 5.3. Basic MVDR adaptive beamformer (Haykin, 1991)

is shown in Figure 5.3 [27, 83].

Note that MVDR beamforming is considered to be a special case of what is referred to as the *linearly constrained minimum variance* (LCMV) beamforming system [16, 83]. The basic idea behind LCMV is to minimize the beamformer output variance or power subject to the response constraint

$$\mathbf{W}^H \mathbf{S}(\phi) = g \quad (5.4)$$

where g is a complex constant. By linearly constraining the weights to satisfy (5.4), the LCMV beamformer insures that any incoming signal from an angle θ (or ϕ) is passed to the output with response g . If $g=1$, then (5.4) is termed the response constraint of the MVDR beamformer, as given by Equation (5.1).

Data Independent Beamformer

One major contribution to adaptive arrays was made in 1967 [27]. Here, the variable weights were automatically calibrated (adjusted) by a simple adaptive technique based on the *least-mean squares* (LMS) algorithm. From a *minimum-mean-square error* (MMSE) sense, the resulting array adapts itself to form a main beam in one direction while rejecting noise signals occurring outside that main beam.

Another algorithm that deals with weight adjustments in adaptive antenna systems, was put forth in 1976 [28]. In a classic report, the governing control system that covers the operation of adaptive antenna arrays, was derived. The proposed algorithm was based on the maximization of the signal-to-noise ratio (SNR) at the output of the antenna array, in the presence of noise.

Both, the LMS (MMSE) algorithm [27] and the maximum SNR algorithm [28], are rather similar. They adaptively adjust the weights in the

antenna array by correlating antenna signals. The examples shown using both algorithms, illustrate the convergence of the LMS and the SNR adaptation procedures toward the optimum Wiener solution.

In a *data independent* (DI) beamformer [16], the weights are designed in a similar, but not exactly the same, fashion to the SNR and the LMS algorithms, so that the beamformer response approximates a desired response, from an arbitrary direction, independent of the array data statistics. This method constitutes the main theme of the present research topic.

The choice of the weights is such that the actual response

$$\mathbf{R}(\phi) = \mathbf{W}^H \mathbf{S}(\phi) \quad (5.5)$$

approximates a desired response $\mathbf{R}_d(\phi)$. The selection of these weights can be based on a number of techniques similar to those used in FIR filter design. The only difference in this analogy, is that the weights here are chosen to minimize the L_p norm [16] of the difference between the desired and the actual response. One of the most commonly used norms is the L_2 (least-squares).

The method of least-squares (LS) is used to solve a linear filtering problem, without invoking any assumptions on the statistics of the inputs applied to the beamformer (i.e., filter). LS may be viewed as an alternative to the Wiener filter theory. However, Wiener filtering is sometimes referred to as least-squares filtering [86]. This is somewhat of an oversimplification since the Wiener filter is based on ensemble averages where the resulting optimum filter, in a probabilistic sense, is obtained for all realizations of the operational medium, hence minimizing the average squared-error and not just the squared-error, as in least-squares filtering. There exists, however, some sort of a coincidental connection between Wiener filtering and deterministic least-squares. The following section demonstrates the link betw-

een the two methods through the description of the LS method in its basic form.

Least-Squares Estimation

In order to describe the *liaison* between the Wiener filter and the LS filter, assume that an $(N \times N)$ matrix quantity $[F]$ is to be determined from an $(M \times N)$ matrix, $[Y]$, of linearly related known measurements

$$[X] \cdot [F] = [Y] \quad (5.6)$$

where $[X]$ and $[Y]$ are complex and assumed to be given. $[X]$ is a matrix with dimensions $(M \times N)$. Both $[X]$ and $[Y]$ are analogous to $S(\phi)$ and $R(\phi)$, respectively. It is also assumed that $M > N$, and that $[F]$ is *overdetermined* by the system of equations represented by (5.6). The estimate of $[F]$ is defined as the "best" or the "optimum", $[F_{\text{opt}}]$, solution that minimizes the sum of the squared residuals. That is, if $[Y]$ is moved to the left of (5.6) and $[F]$ is substituted by $[F_{\text{opt}}]$, the residual error matrix, $[\epsilon]$, can be obtained by

$$[X] \cdot [F_{\text{opt}}] - [Y] = [\epsilon] \quad (5.7)$$

$[F_{\text{opt}}]$ can then be estimated by minimizing the performance measure, which is the sum of squares of errors

$$[J] = [\epsilon]^H [\epsilon] \quad (5.8)$$

The solution is found by expressing $[\epsilon]$ in terms of $[F_{\text{opt}}]$ then setting the derivatives of $[J]$, with respect to each component of $[F]$, to zero, as follows

$$[J] = [\epsilon]^H [\epsilon] = ([X] \cdot [F_{\text{opt}}] - [Y])^H ([X] \cdot [F_{\text{opt}}] - [Y]) \quad (5.9)$$

This expression may be expanded and differentiated term-by-term and then set equal to zero. If the brackets [] are dropped, for simplicity, this will lead to the following

$$\begin{aligned} \frac{\partial}{\partial F_{\text{opt}}} &= \left(F_{\text{opt}} (X^H X) F_{\text{opt}} - Y^H X F_{\text{opt}} - F_{\text{opt}}^H X^H Y + Y^H Y \right) \\ &= 2 (X^H X) F_{\text{opt}} - (Y^H X)^H - X^H Y = 0 \end{aligned} \quad (5.10)$$

which can be rewritten as

$$(X^H X) F_{\text{opt}} = X^H Y \quad (5.11)$$

Equation (5.11) may now be solved for F_{opt} . The result is

$$F_{\text{opt}} = (X^H X)^{-1} X^H Y \quad (5.12)$$

This is the solution of the deterministic least-squares problem and is often called the *minimum error-norm solution*. Equation (5.12) shows that the least-squares estimate F_{opt} is linearly related to the measurements Y . This is not surprising since derivatives of quadratic functions are linear functions.

A more generalized form [86] of the least-squares (Wiener) problem can be stated as follows. It is required to minimize

$$[J] = [\epsilon]^H [W] [\epsilon] \quad (5.13)$$

where $[W]$ is a symmetric and positive definite weighting matrix. If it is desired to have equal weightings of the residuals, $[W]$ will simply be set to the identity matrix $[I]$. However, in this case, a weighting matrix is used because, in some situations, more confidence is acquired in the accuracy of

some measurements than of others; hence, the elements of $[W]$ are chosen accordingly to weigh these measurements more heavily than others. As a result, the solution to the least-squares estimate (Wiener filter) is modified to (again after dropping the brackets [])

$$F_{\text{opt}} = (X^H W X)^{-1} X^H W Y \quad (5.14)$$

which is the solution of the general problem.

In a slightly different approach [86] for determining the solution for the optimal filter, let $[R_{xx}]$ represent the time-averaged autocorrelation matrix of the tap inputs, and let $[R_{xy}]$ represent the time-averaged cross-correlation matrix between the tap inputs and the desired response. Using the above auto- and cross- correlation matrices representation, and assuming that $[R_{xx}]$ is nonsingular (i.e., the inverse matrix $[R_{xx}]^{-1}$ exists), then the solution for the optimal filter can be written as

$$[F_{\text{opt}}] = [R_{xx}]^{-1} [R_{xy}] \quad (5.15)$$

Singular-Value Decomposition

In the previous section, the system of *normal equations* for computing the linear least-squares solution, was developed. The formulations presented for this solution were given by Equations (5.12) and (5.15) directly in terms of data matrices. To recapitulate, $[X]$ is the data matrix representing the signals as the input vectors and $[Y]$ is the data matrix representing the signals of the desired response. The two forms (5.12) and (5.15) are mathematically equivalent. However, their computational procedures for evaluating $[F_{\text{opt}}]$, are different. Equation (5.15) involves the computation of the product $X^H X$ to determine the autocorrelation matrix $[R_{xx}]$, while Equation (5.12) requires that the entire term $(X^H X)^{-1} X^H$ be evaluated by means of the singular-value

decomposition (SVD) to be applied directly to the input data matrix $[X]$. If a new matrix is defined

$$X^+ = (X^H X)^{-1} X^H \quad (5.16)$$

then (5.12) may be rewritten as

$$F_{\text{opt}} = X^+ \cdot Y \quad (5.17)$$

The main interest in the SVD method, here, is to then formulate a general definition for the pseudoinverse X^+ . Given the $(M \times N)$ data matrix X , there exist two unitary matrices V and U , such that

$$U^H X V = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \quad (5.18)$$

where Σ is a diagonal matrix

$$\Sigma = \text{diag} (\sigma_1, \sigma_2, \dots, \sigma_r) \quad (5.19)$$

and where the subscript r denotes the rank of the matrix X . The rank is defined as the number of linearly independent columns in the matrix X . the σ 's represent the singular values of the matrix X^+ , and are ordered as

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$$

Note that V is an $(N \times N)$ unitary matrix whose columns are made up of a set of orthonormal eigenvectors of $X^H X$ that are associated with the eigenvalues σ_j^2 's. Equation (5.18) establishes the theorem of the singular value decomposition for overdetermined systems.

In practice, it is often seen that the data matrix X contains linearly dependent

columns [83]. Consequently, there exists an infinite number of solutions to the least-squares problem. Therefore, deciding which of these solutions is to be adopted as the ultimate choice, is cumbersome. However, if the SVD technique is used, this issue can indeed be resolved, even when the $\text{null}(X) \neq 0$. The pseudoinverse X^+ is defined as [83, 87, 88]

$$X^+ = V \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^H \quad (5.20)$$

where

$$\Sigma^{-1} = \text{diag} \left(\sigma_1^{-1}, \sigma_2^{-1}, \dots, \sigma_r^{-1} \right)$$

If the matrix V is partitioned as

$$V = (V_1, V_2)$$

where V_1 is an $(N \times K)$ matrix, and V_2 is an $(N \times (N-K))$ matrix, where $K=r$ (the rank of X), with

$$V_1^H V_2 = 0$$

it can be shown, without proof, that

$$\begin{aligned} V_1^H X^H X V_1 &= \Sigma^2 \\ \Rightarrow \Sigma^{-1} V_1^H X^H X V_1 \Sigma^{-1} &= I \end{aligned} \quad (5.21)$$

Now if a new $(M \times K)$ matrix

$$U_1 = X V_1 \Sigma^{-1} \quad (5.22)$$

is defined, then from (5.21) and (5.22) it is noted that

$$(X^H X)^{-1} = V_1 \Sigma^{-2} V_1^H$$

and

$$X^H = V_1 \Sigma U_1^H$$

where

$$\Sigma^{-2} = \text{diag} \left(\sigma_1^{-2}, \sigma_2^{-2}, \dots, \sigma_r^{-2} \right)$$

In the next chapter, all of the above theory is put to the test. The implementation of the data independent beamformer is described by a block diagram. In addition, the defect prediction and the reconstruction procedures using finite element simulated data are also presented. Furthermore, the effects of transducer baseline nonuniformity as well as the material geometry, on the performance of the beamformer, are discussed via a few examples.

CHAPTER 6. RESULTS OF DEFECT PREDICTION

The concepts of two adaptive systems with the task to minimize the output variance or power under the linear constrained response to specified directions were introduced in Chapter 5. In addition, a brief theoretical analysis on the data independent beamformer based on the method of least-squares (LS) was also derived.

In this chapter, the algorithm for the data independent beamformer system is presented. It is seen that this technique is capable of adjusting the weights of the interpolating filter (beamformer), in function of the array of transducers, to predict an incoming signal from a desired direction while discriminating against other signals from different directions. The transducer array input data are simulated using the finite element method, as shown in Chapters 4 and 5. Computer implementation confirms that the filter is indeed able to not only predict defect locations from a set of data produced by a nonuniformly spaced transducer baseline, based on one material geometry, but also predict defect locations, to a lesser extent, from a data set produced by a nonuniformly spaced transducer baseline based on a different material geometry. This geometry is taken to be the *reverse* of the training *actual* geometry. On the other hand, further to the tests performed to determine the *optimal* filters with the actual geometry, playing the role of the *training* data, similar filter studies were conducted assuming the reversed geometry as the basis for the training data, in order to ascertain the robustness of the proposed method. Results due to the effects of transducer spacing nonuniformity and of material geometry are shown. Furthermore, in order to assess the performance of this beamforming method, plots of the relative mean-squared error (RMSE) between the resulting B-scan images and the desired response are also presented.

This problem is classified as a minimum least-squares (MLS) problem where no *a priori* knowledge of the statistics of the input signals is assumed.

Filter Implementation and Procedure

As seen in Chapter 5, the standard formulation for the discrete, multi-channel data independent beamformer leads to a system of *normal linear equations*. The solution to these equations gives the optimal filter that minimizes the mean-squared error (MSE) between the reconstructed signal and the true desired signal. The intent of this section is to outline the basic steps which are utilized as a means to determine the optimal filter and to confirm the numerical predictions produced by the filter. These numerical predictions are directly compared to the training finite element model for both the actual and the reversed geometries.

The simulation test system consists of the following items:

1. Start out with the set of finite element simulated time-domain ultrasonic signals, as shown in Chapter 4, serving as the training data. Here, 38 signals are obtained with the finite element test bed.
2. Take the Fourier transform of these time-domain signals.
3. Read in the same ultrasonic signals but in reversed order with respect to the signals read in step 1; and again take the Fourier transform of these signals.
4. Initialize two temporary arrays X and X_1 to zero. These arrays represent the actual (or reversed) and the reversed (or actual) inputs to the beamformer, respectively.
5. Determine the percentage number of records to be kept as the input to the beamformer, with the rest of the records assumed to be missing and need to be reconstructed. Five choices are available:

(a) 50% (keep the odd records)

14. Finally, compare the resulting output time-domain images with their corresponding desired images in steps 1 and 3, by computing the relative mean-squared error (RMSE) according to

$$\text{RMSE (\%)} = 100 \sum_{i=1}^{nr} \sum_{j=1}^{np} \frac{(y_r(i,j) - y_t(i,j))^2}{y_t^2(i,j)}$$

where y_r is the reconstructed output and y_t is the true desired output; and where nr and np are the number of records and number of points per record, respectively. The computer code for the above procedure is given in Appendix B.

In practice, we never have the desired signal distribution. As a consequence, a direct method to determine the optimal filter cannot be obtained from only the nonuniformly spaced signals. So then, the goal of this approach is to calculate a number of filters, F_{opt} , for a variety of nonuniform transducer baselines and use these results inversely as a *look-up table*. That is, given a set of nonuniformly spaced ultrasonic signatures from an experimental inspection of a material with known geometry and known type (i.e., Plexiglass™, Steel, Aluminum etc.), one could find a similar material in the look-up table, and then reconstruct the irregular test image using the corresponding optimal filter found in the table.

Prediction and Reconstruction: Tests and Results

Two sets of tests were performed using the finite element code to provide the training data for both the actual and the reversed geometries. The simulated material is considered to be isotropic and having the same characteristics as Plexiglass™. The material geometry and the location of the flaws are shown in Figure 4.9. The resulting B-scan records are shown in Figure 4.11, and depicted again in Figure 6.1, for convenience. Let this image be of the actual geometry and let Figure 6.2 be of the reversed geometry.

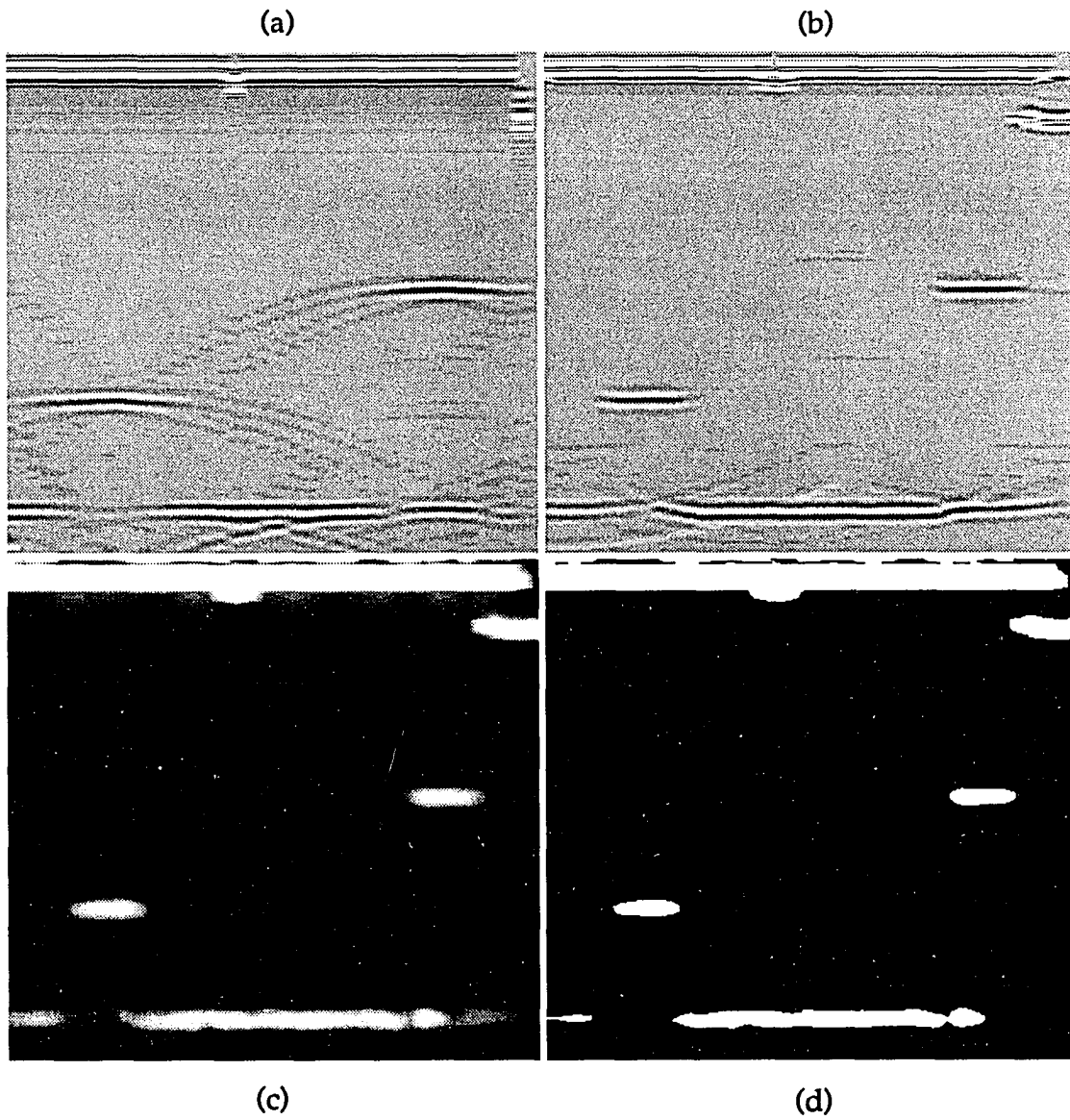


Figure 6.1. Finite element actual geometry
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

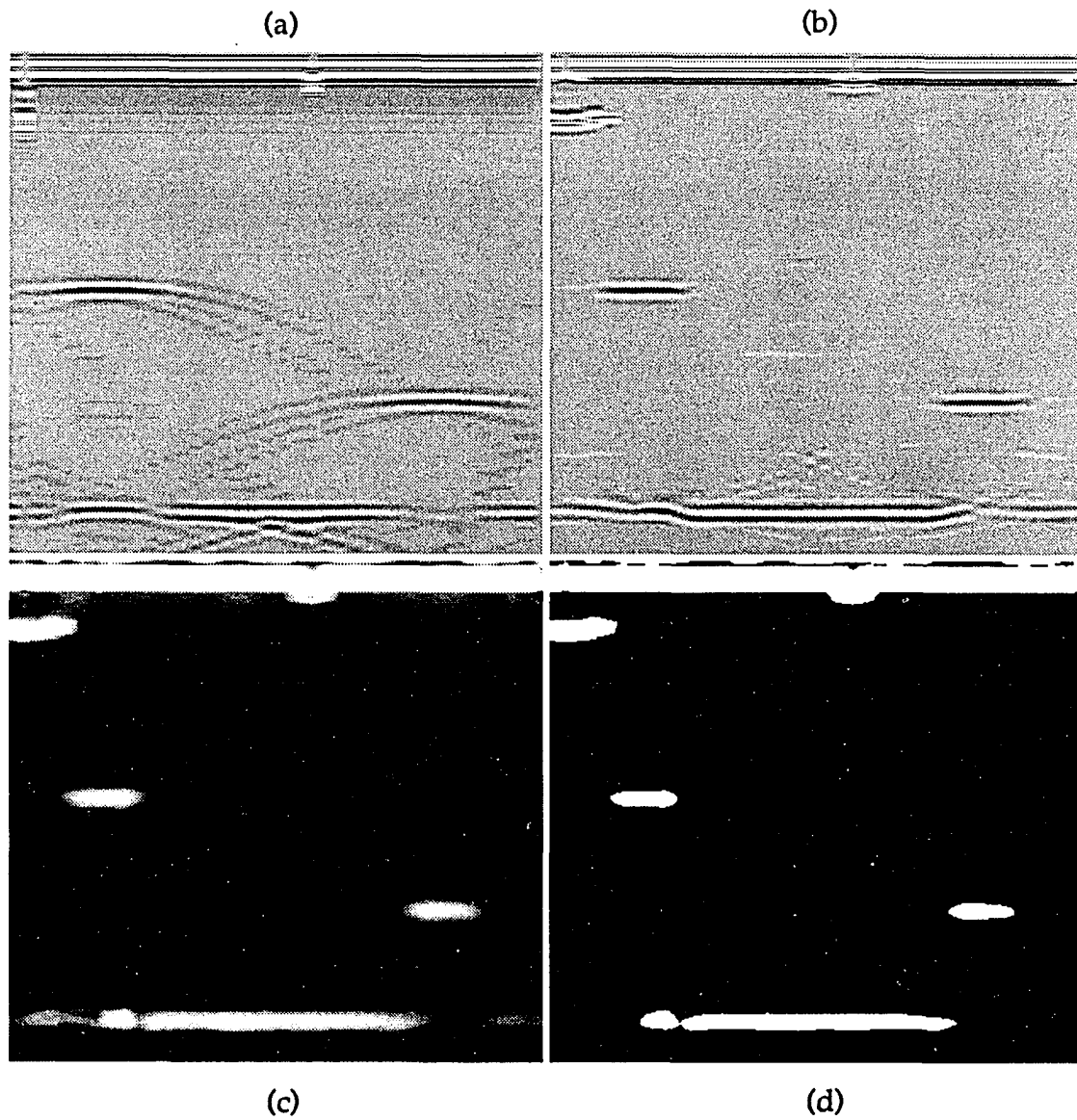


Figure 6.2. Finite element reversed geometry
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

The first set of tests are conducted on the actual geometry with Figure 6.1 being our desired image. That is, the filter is computed based on nonuniform signals taken from the actual geometry. Once the filter weights are determined, they are then convolved with both the irregular actual image and with the irregular reversed image (having same number of missing records). Again, the total number of records in both the actual and the reversed geometries is 38; from which 50% (odd or even), 26%, 13%, or 3% are kept so to make the transducer baseline equally spaced (in the case of the 50% arrangements) or unequally spaced (in the case of the 26%, 13%, or 3%).

The tests in the second set are basically the same as the tests achieved in the first set, except that now the filters are the result of using the reversed geometry as the training data, and applying these filters on the nonuniform signals of the actual geometry.

Following each prediction process in either set of tests, SAFT is used to determine the size and location of the two flaws in the resulting B-scan images, similar to the way it was applied in Chapter 4.

The total CPU time required for each of the test runs (Beamforming + SAFT) is approximately 4 minutes and 24 seconds on the Stellar™ computer system.

As a first attempt to check for the effectiveness of the DI beamformer from an optimal point of view, using the software (Appendices A - C) developed for that purpose, the first case of keeping the odd 50% of the ultrasonic signals in the actual geometry, is tested for. Figure 6.3 shows the magnitude of the frequency response filter produced as a result of this preliminary trial. The darkest grey level in the image corresponds to the zero amplitude. All (38 x 38) filter images were padded with zeros (on the right and on the bottom), throughout this work, to produce (64 x 64) portrayals of these images so that their inverse FFT can be determined. One particularly intriguing pattern detected in the data comprised within these filter images, is

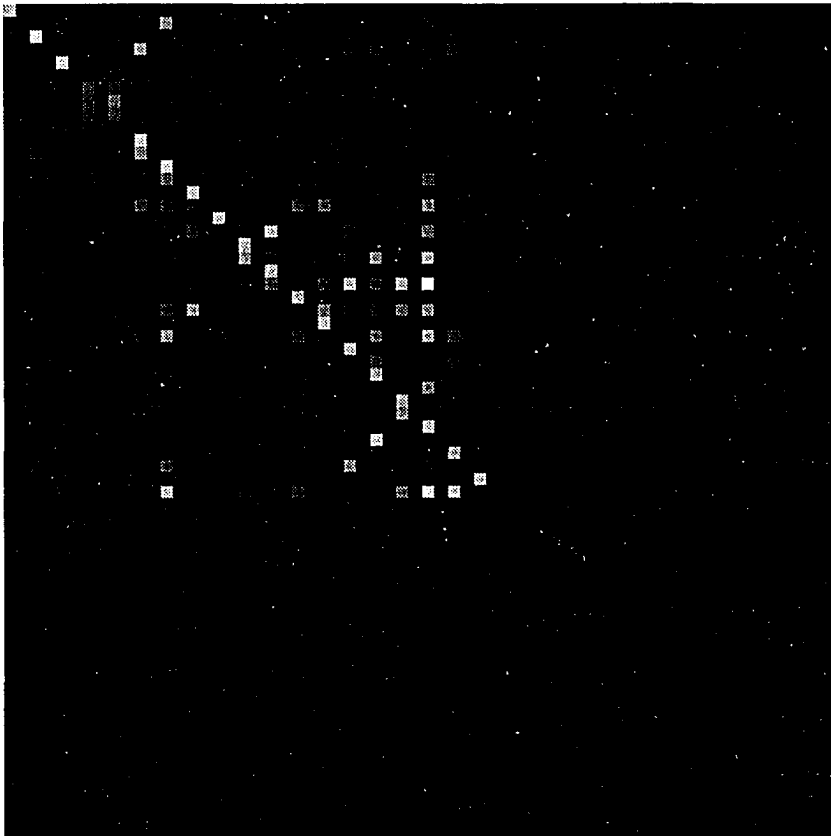


Figure 6.3. Magnitude of frequency domain filter based on 50% (Odd) of the FEM actual geometry

that the only columns containing any significant values, are those which correspond to the signals that were kept in the input data array X , with the largest value in each column lying on, or close to, the diagonal of the filter matrix. For example, in Figure 6.3, the only columns that contain any meaningful data are the odd columns (1, 3, 5, ... , 37), which constitute 50% of the record. Similar patterns are also noticeable in the subsequent filter images, for the rest of the cases to be studied. The other interesting feature in these filter images, is that they reflect the order of the filter. As described in Chapter 5, the way the optimal filters are solved for, is by differentiating the second-order error expression given by Equation (5.9). This operation leads to the first-order relationship (Equation (5.12)) for the filters. Therefore, the optimal filters obtained are linear filters. This can actually be confirmed by taking a vertical slice along any of the significant columns of any of the filter images, and noting that the profile is analogous to that of a single-pole finite-impulse response (FIR) filter. The next step is to convolve the complex filter, whose magnitude is shown in Figure 6.3, with the nonuniform arrays X and X_1 , for both the actual and the reversed geometries. Figures 6.4 and 6.5 illustrate the outcome of the convolution. Figures 6.4 (a) and 6.5 (a) show the reconstructed raw actual and reversed images, while Figures 6.4 (b) - (d) and 6.5 (b) - (d) exhibit the SAFT B-scan images and their corresponding rectified version after a -6 dB threshold has been applied in order to determine the lateral size of the flaws as well as their respective location. Quantitative evaluation of these and following images will be analyzed in the next section. In the very same manner, the other filters are computed based on keeping the 50% (even), 26%, 13% and 3% of the ultrasonic record of the actual geometry, and once again convolved with the individual arrays X and X_1 . Figures 6.6, 6.9, 6.12 and 6.15 display the magnitude of the filter response for each of the cases; and Figures 6.7, 6.8, 6.10, 6.11, 6.13, 6.14, 6.16 and 6.17 all demonstrate the individual reconstructed actual and reversed records and their respective SAFT B-scan versions, in the same orderly fashion presented for the first case.

Next, the filter is trained on the reversed geometry data and then applied to the nonuniform actual data. This is done to verify the performance

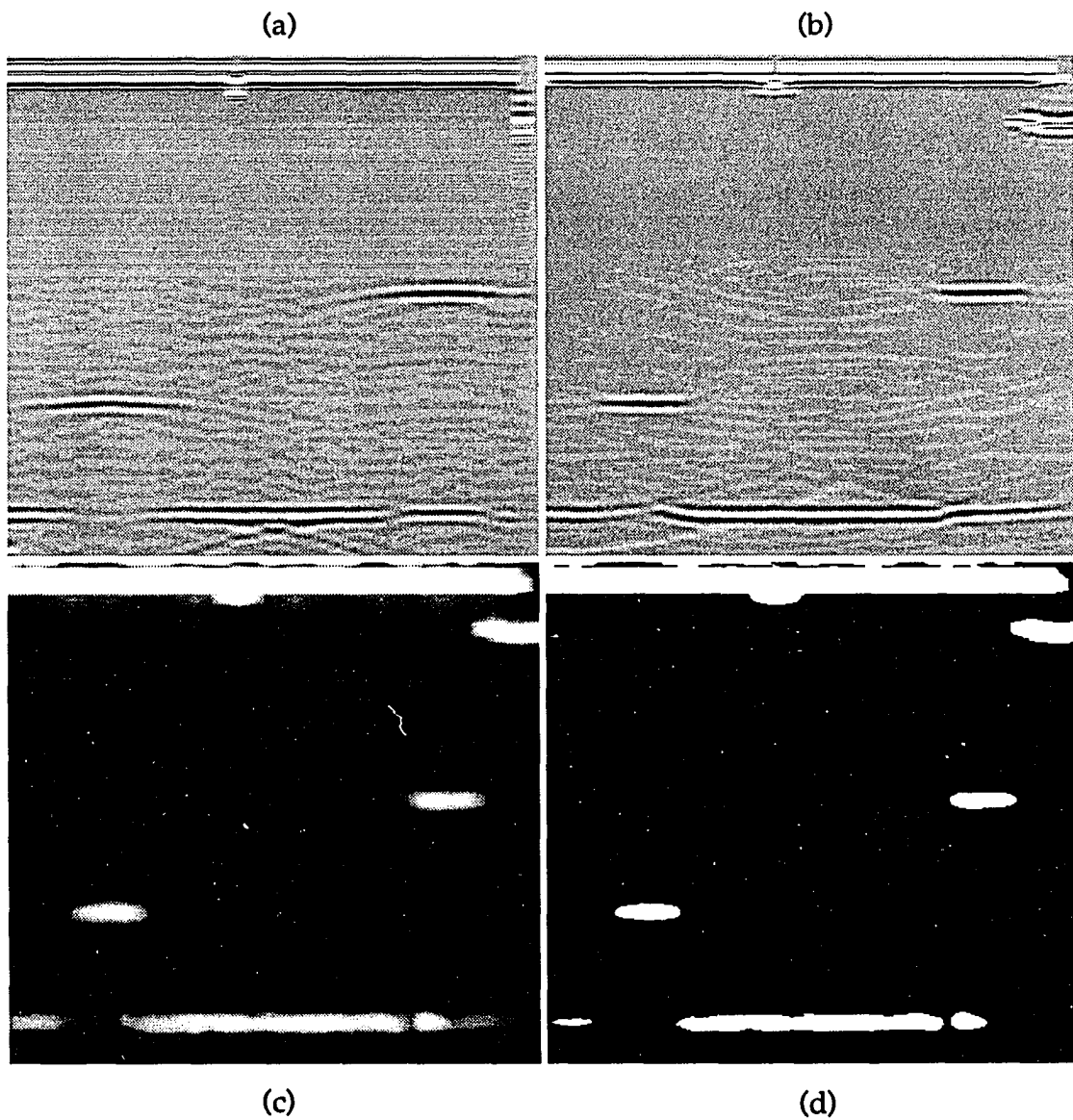


Figure 6.4. Reconstructed FEM actual geometry using filter in Fig. 6.3
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

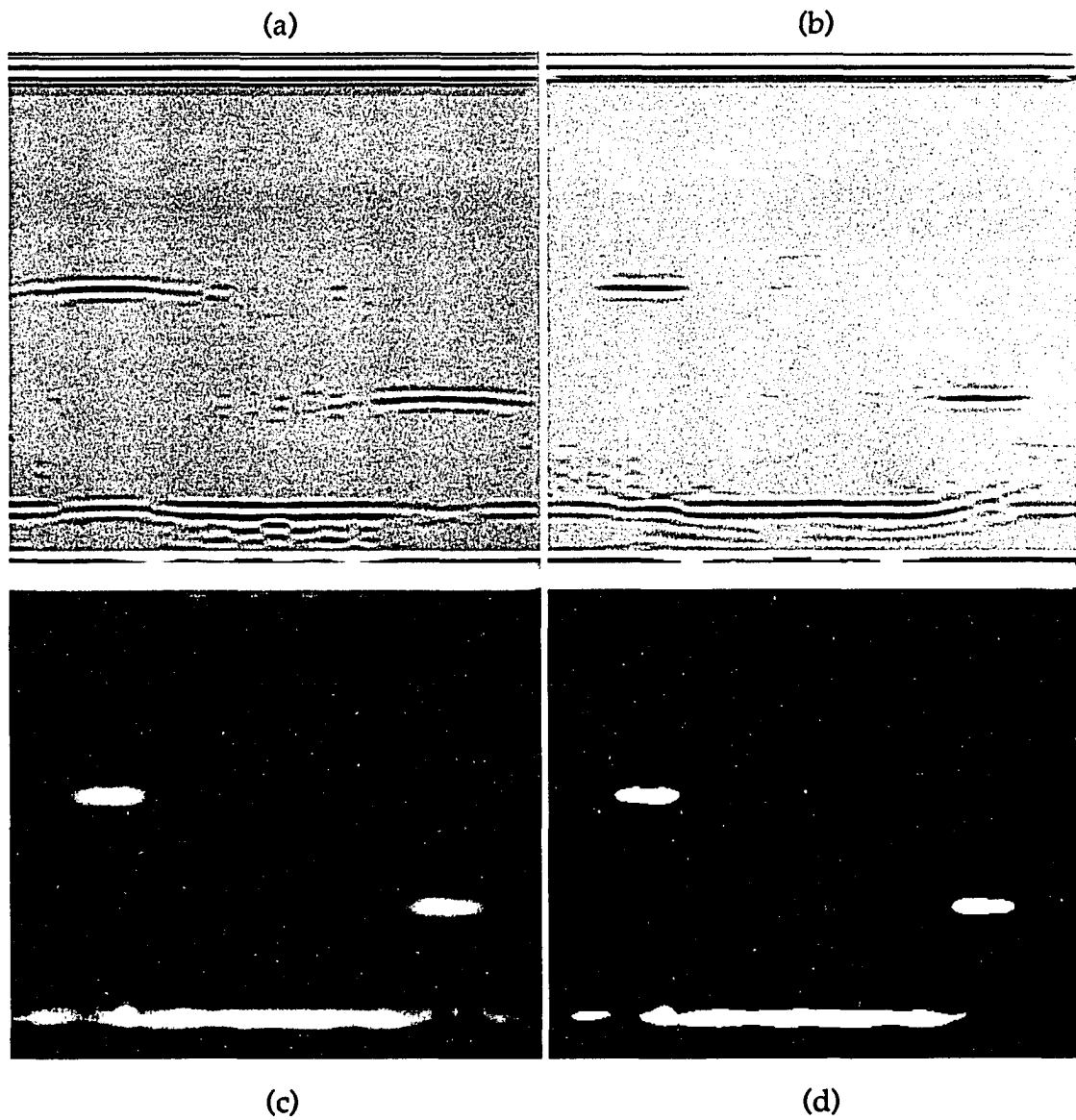


Figure 6.5. Reconstructed FEM reversed geometry using filter in Fig. 6.3

- (a) Raw B-scan image
- (b) SAFT B-scan image
- (c) Magnitude of SAFT
- (d) Rectified (-6 dB threshold) SAFT image

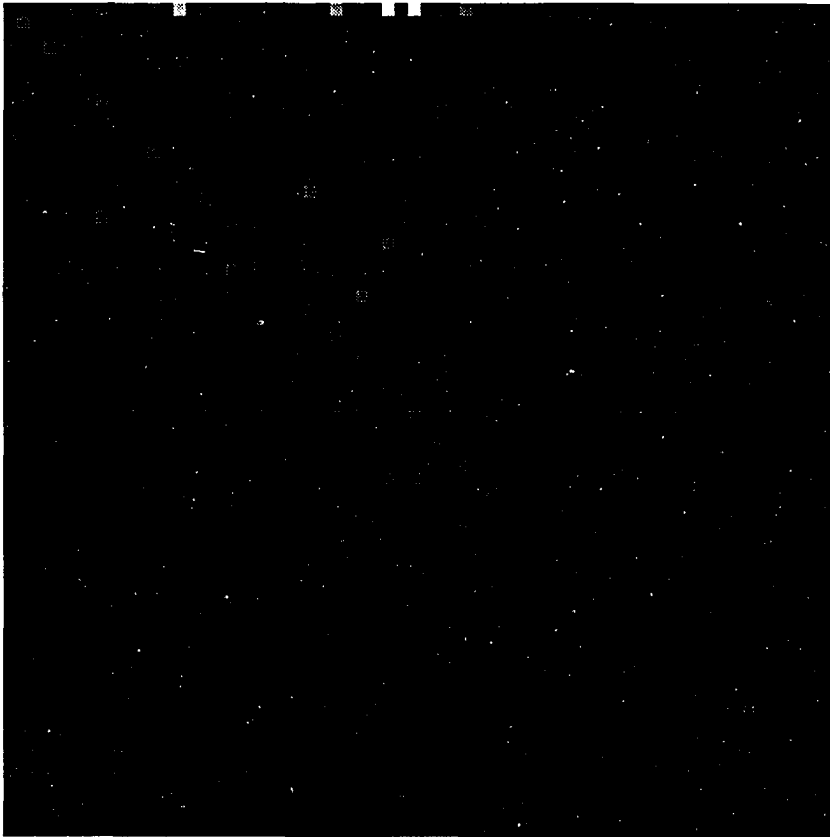


Figure 6.6. Magnitude of frequency domain filter based on 50% (Even) of the FEM actual geometry

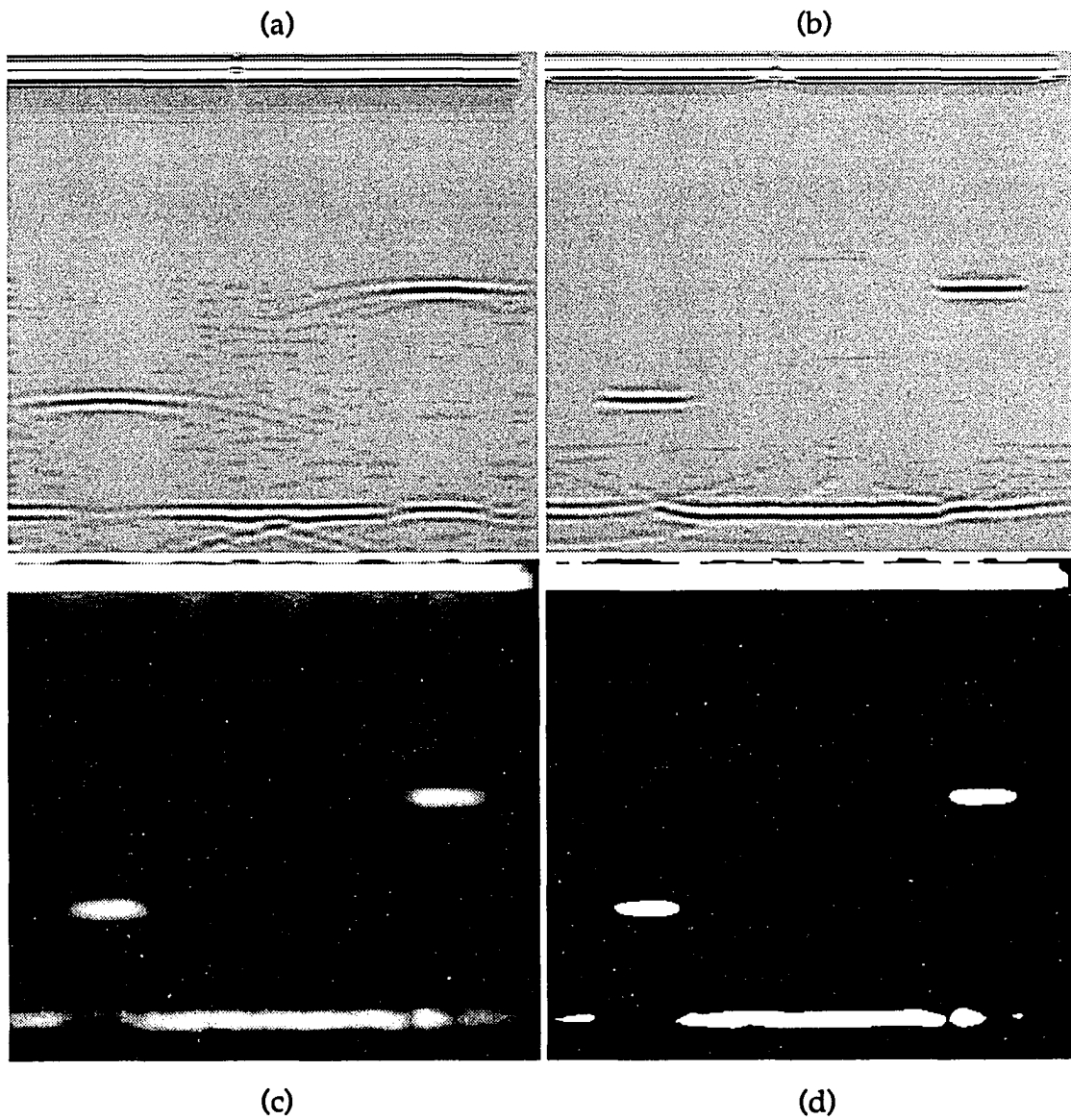


Figure 6.7. Reconstructed FEM actual geometry using filter in Fig. 6.6
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

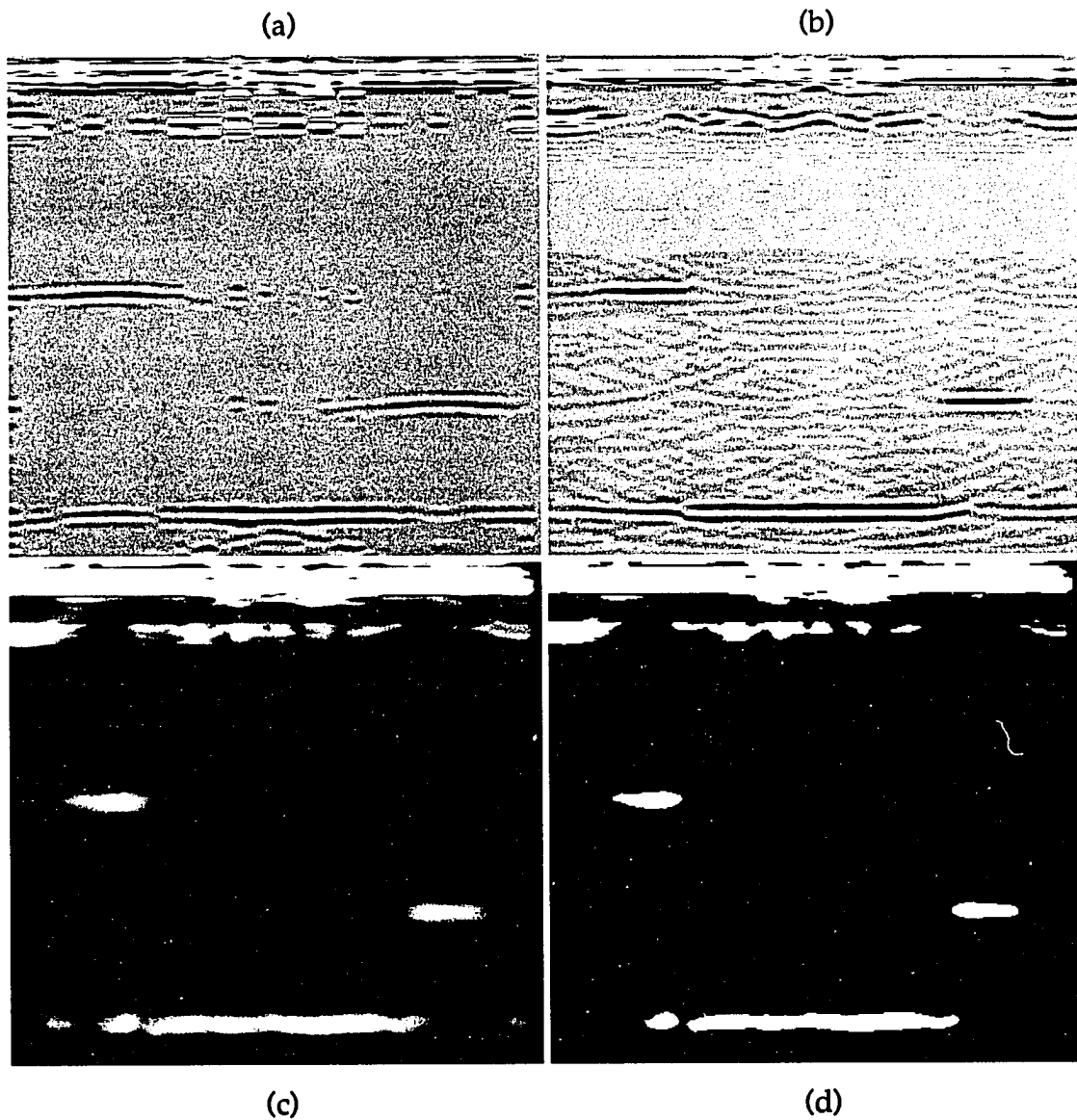


Figure 6.8. Reconstructed FEM reversed geometry using filter in Fig. 6.6
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

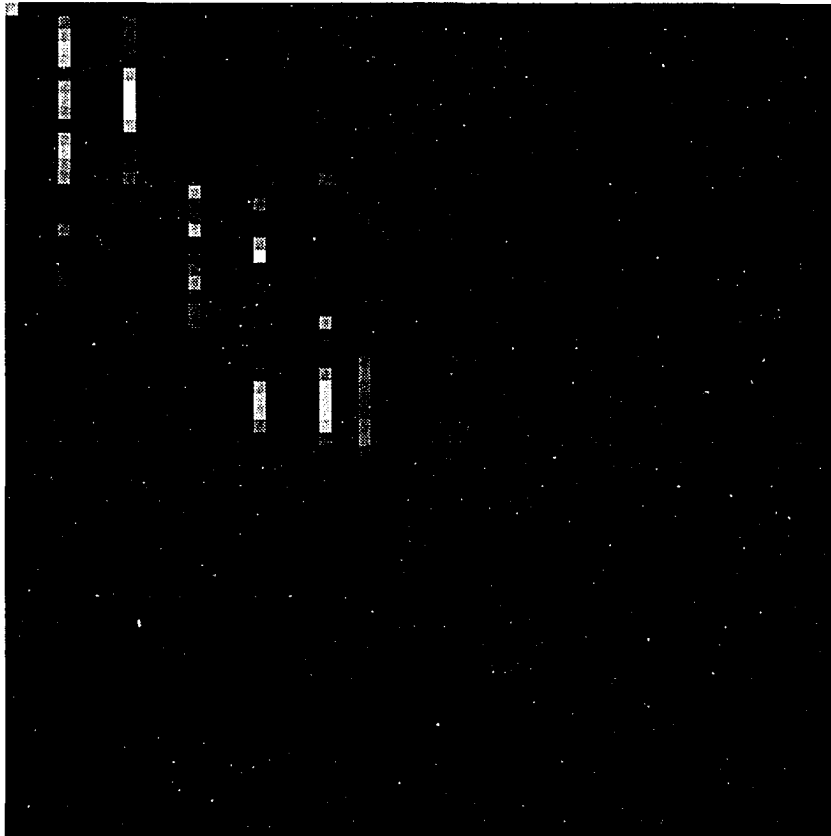


Figure 6.9. Magnitude of frequency domain filter based on 26% (1, 5, 10, 15, 20, 25, 28, 30, 35, 38) of the FEM actual geometry

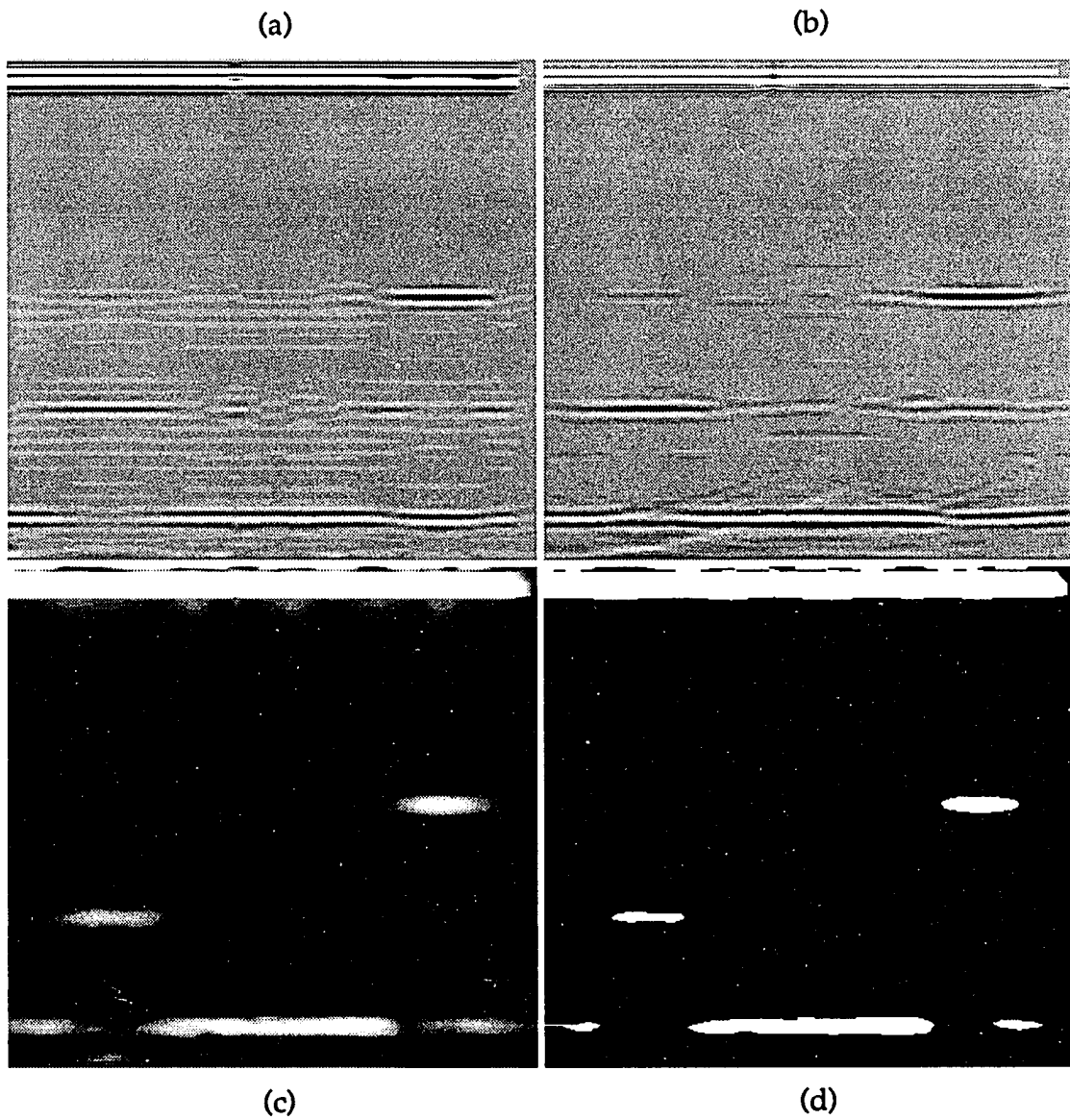


Figure 6.10. Reconstructed FEM actual geometry using filter in Fig. 6.9
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

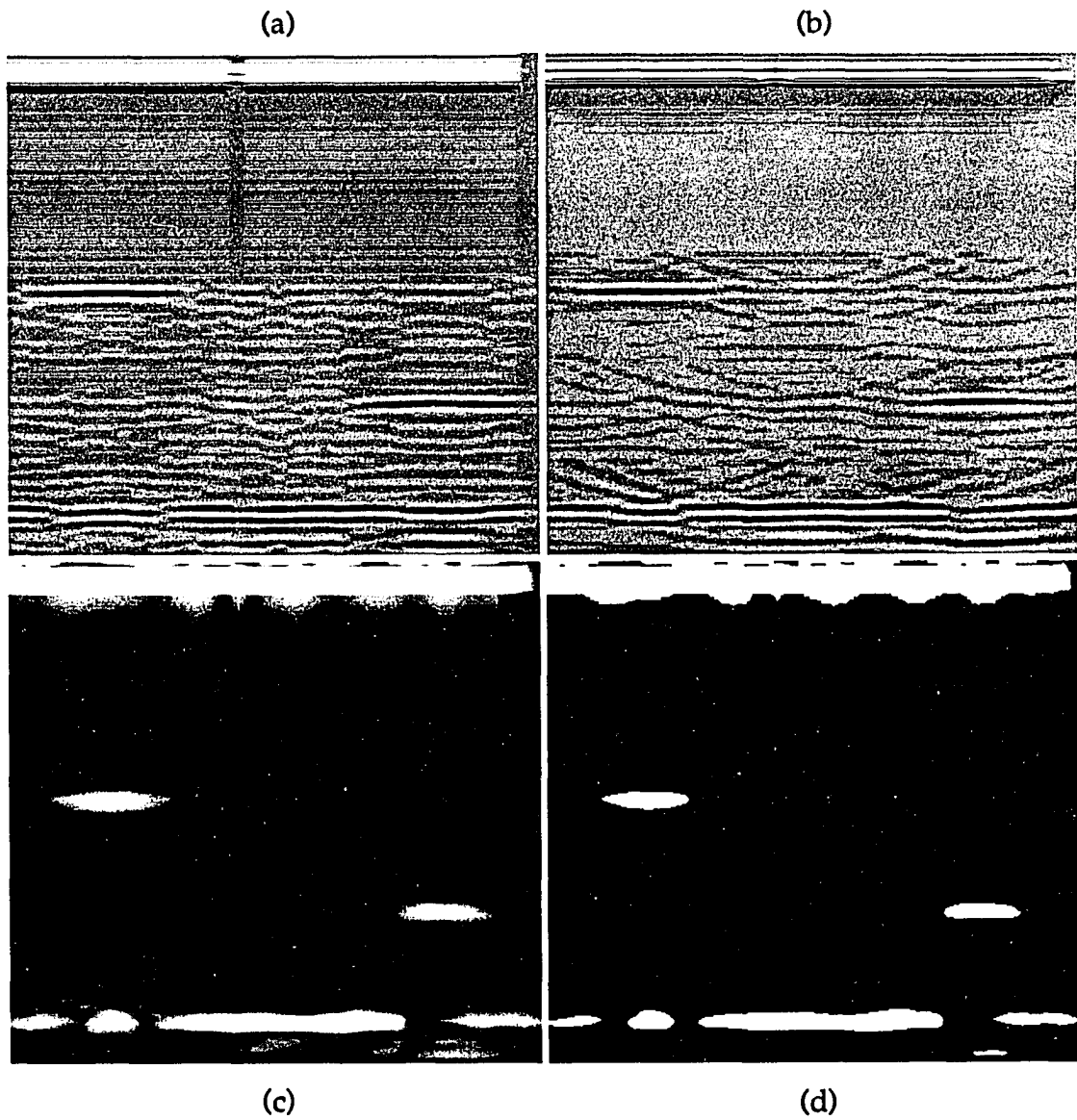


Figure 6.11. Reconstructed FEM reversed geometry using filter in Fig. 6.9

(a) Raw B-scan image

(b) SAFT B-scan image

(c) Magnitude of SAFT

(d) Rectified (-6 dB threshold) SAFT image

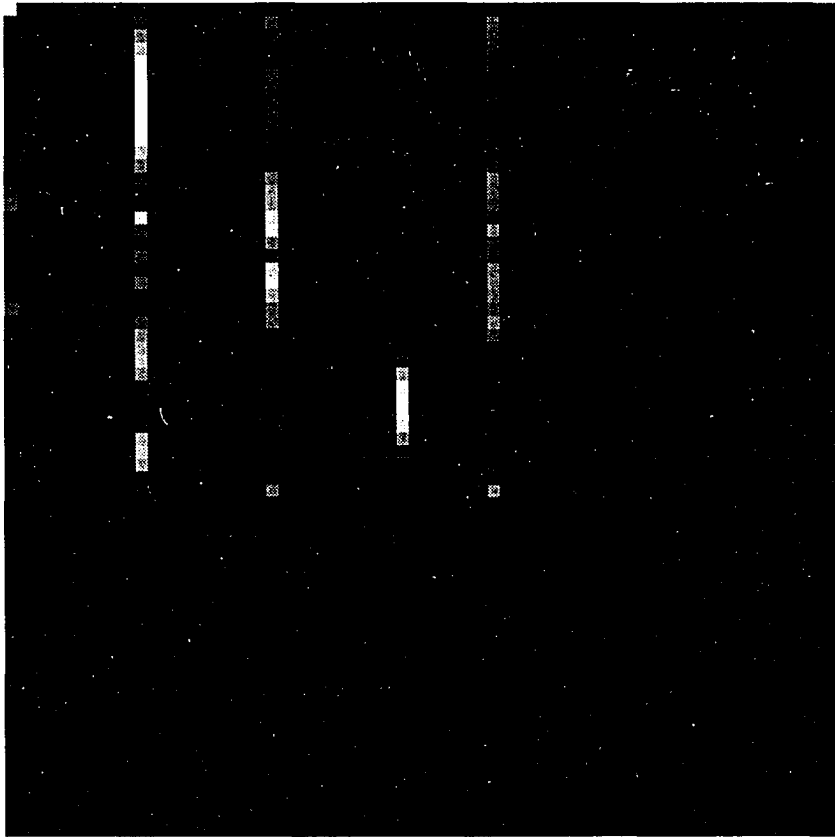


Figure 6.12. Magnitude of frequency domain filter based on 13% (1, 11, 21, 31, 38) of the FEM actual geometry

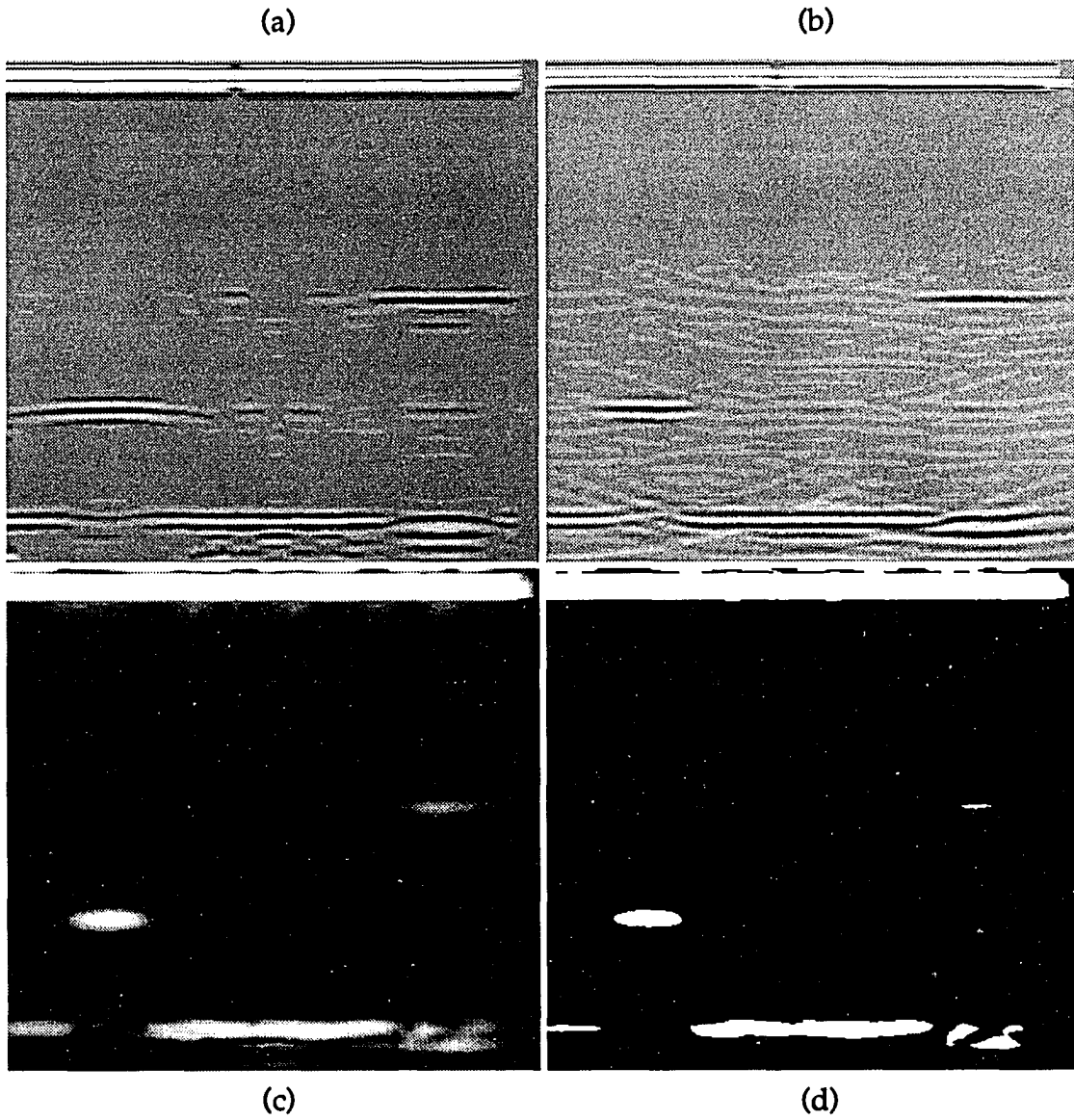


Figure 6.13. Reconstructed FEM actual geometry using filter in Fig. 6.12
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

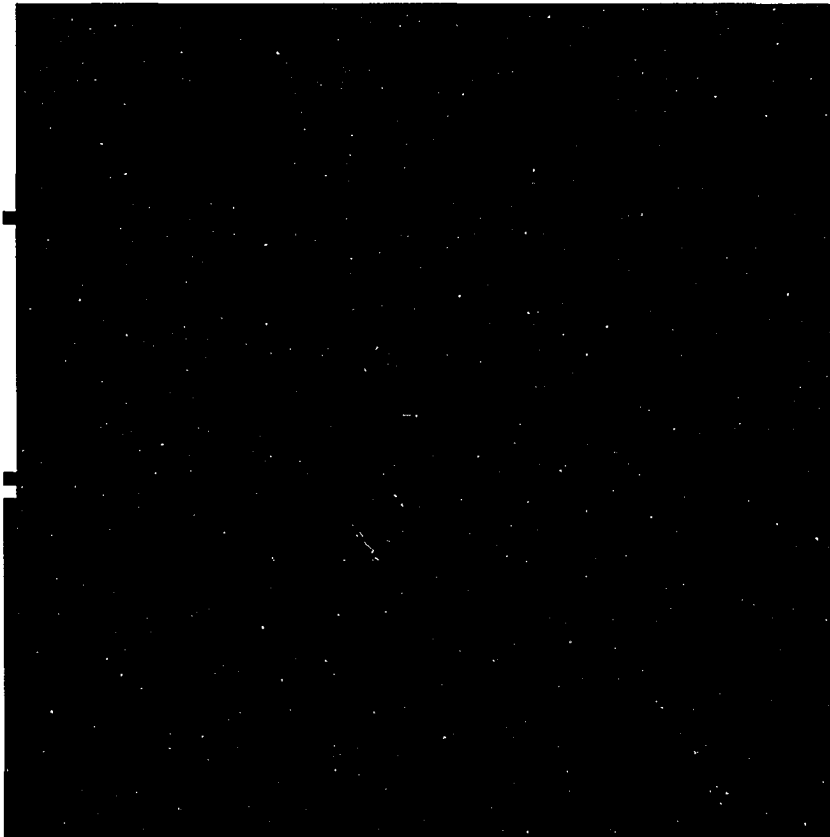


Figure 6.15. Magnitude of frequency domain filter based on 3% (1) of the FEM actual geometry

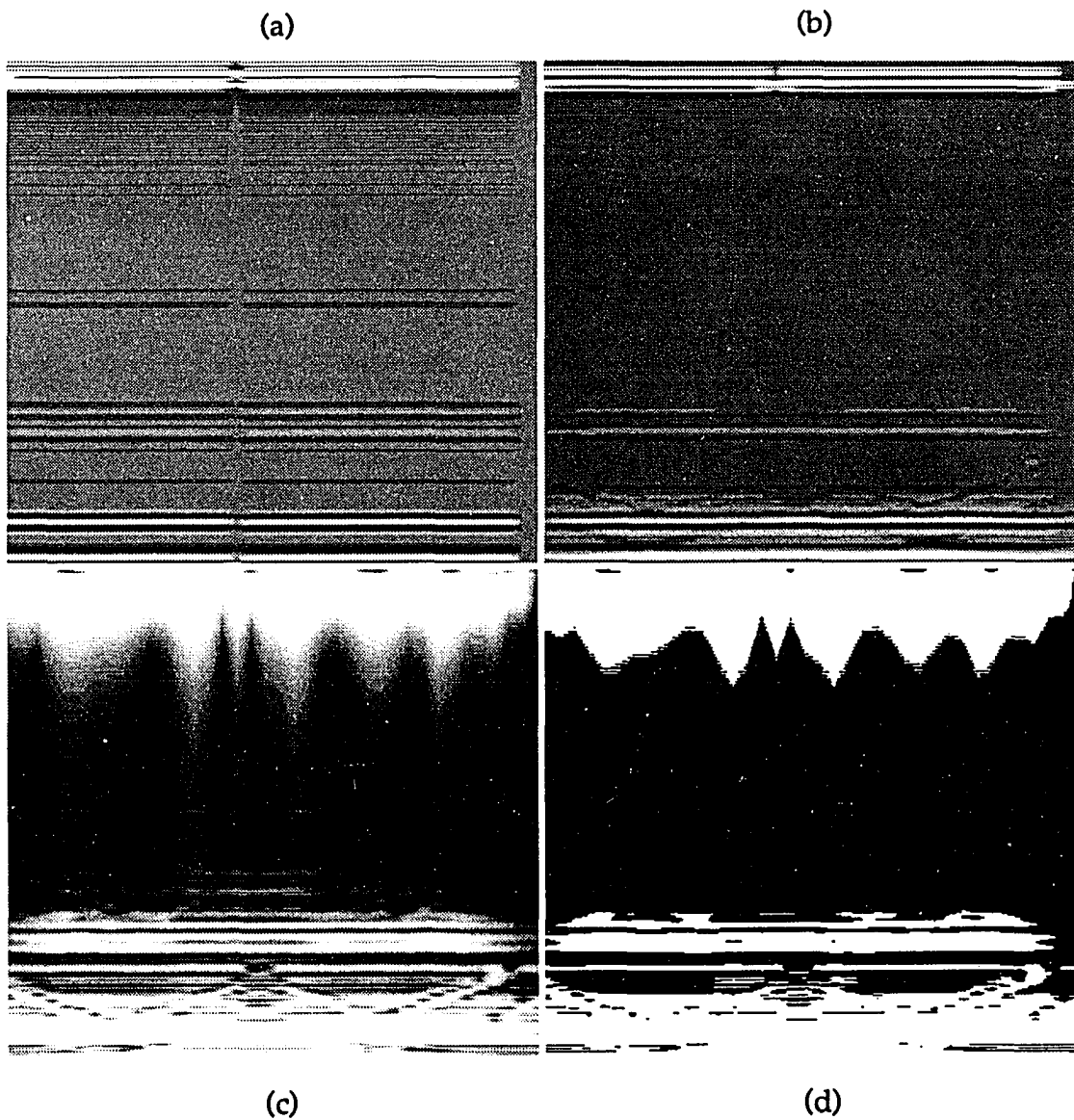


Figure 6.16. Reconstructed FEM actual geometry using filter in Fig. 6.15
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

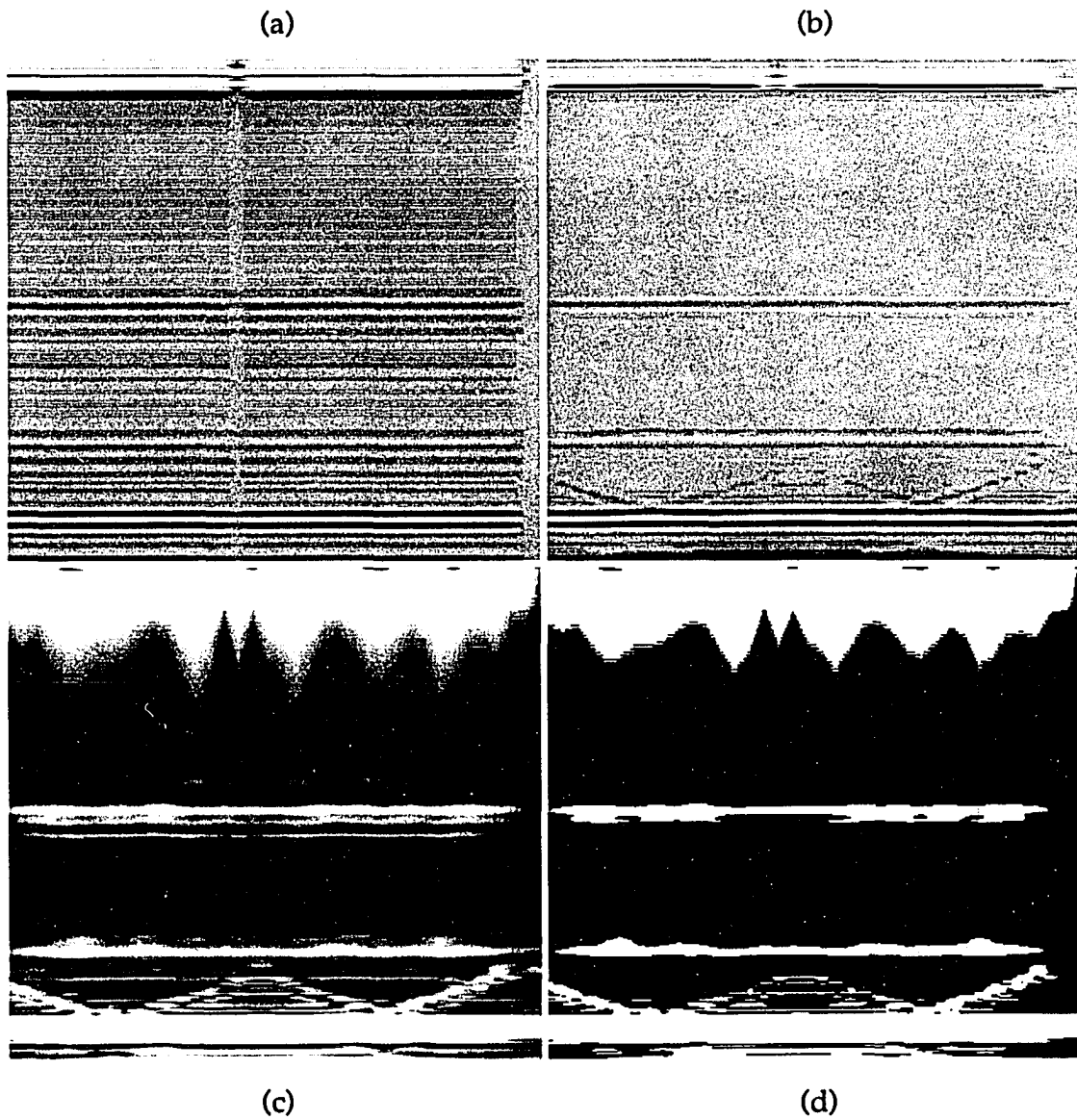


Figure 6.17. Reconstructed FEM reversed geometry using filter in Fig. 6.15
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

of the beamformer when operating on a modified version of the ultrasonic data. Again, the same type of tests were directed as far as the percentage of missing signals is concerned, and once more the filters proved indeed to be robust enough at providing satisfactory reconstruction of the missing signals in the various look directions. Figures 6.18 - 6.29 illustrate the different filter images with their corresponding medium reconstruction in both the reversed and the actual geometries. The last case of keeping 3% (1 record) of the record is not reported since the results resemble the ones shown in Figures 6.15 - 6.17.

Discussion of Results

A number of points have to be discussed in order to evaluate the performance of the DI beamformer just applied as a restoration technique in the presence of missing ultrasonic pulse-echo signals from a typical nondestructive test, as shown in the previous section.

First, two comments need to be made in relation to the images in both sets of tests where 26% and 13% keep situations are considered. These comments can be stated in form of questions. How are the 10 signals (26%) or the 5 signals (13%) chosen? And, would the choice of these signals affect the reconstruction scheme? During the testing procedure, it has been noticed that care must be taken when choosing the 26% or the 13% signals. If these signals are selected in a way that they are adjacent to each other (i.e., #17, #18, #19, ... etc.), that is, in a clustered lay-out, then the reconstruction of the overall images becomes very poor and the resulting raw and reversed geometries are similar to the situation where only 1 signal (3%) is kept, in which case only the front and the back walls are recovered, and no distinctive flaw signals are present. A few runs of the program were executed with bulks of signals (10 or 5) chosen from the beginning, the middle, and the end of the ultrasonic record. All six runs (three on the actual training data and three on the reversed training data) resulted in similar deteriorated outputs. No specific interpretation can be thought of at present for this discrepancy, except that the

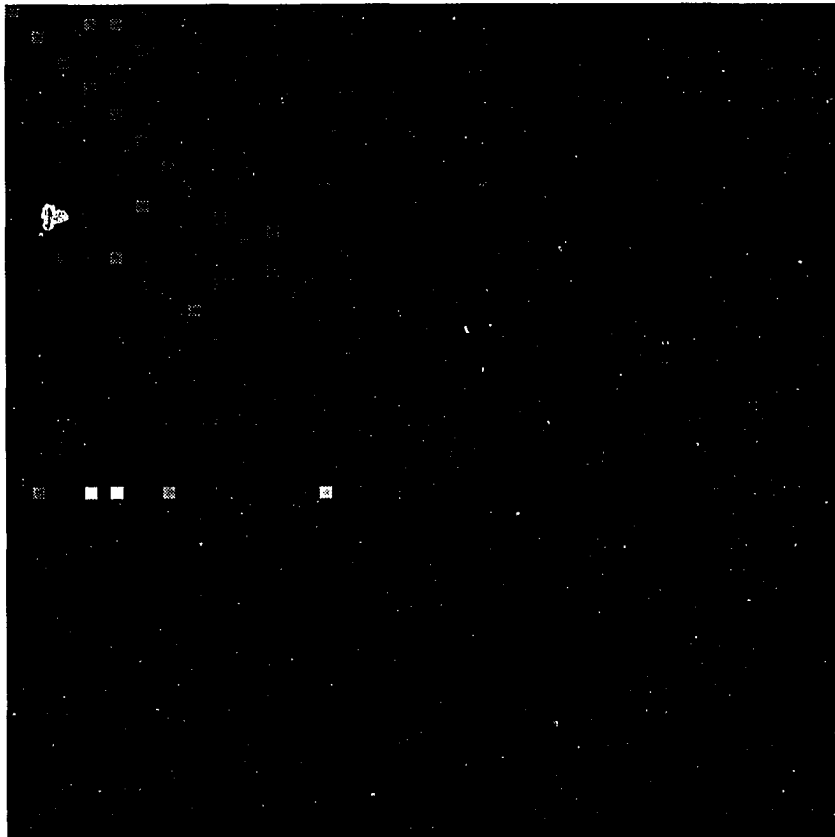


Figure 6.18. Magnitude of frequency domain filter based on 50% (Odd) of the FEM reversed geometry

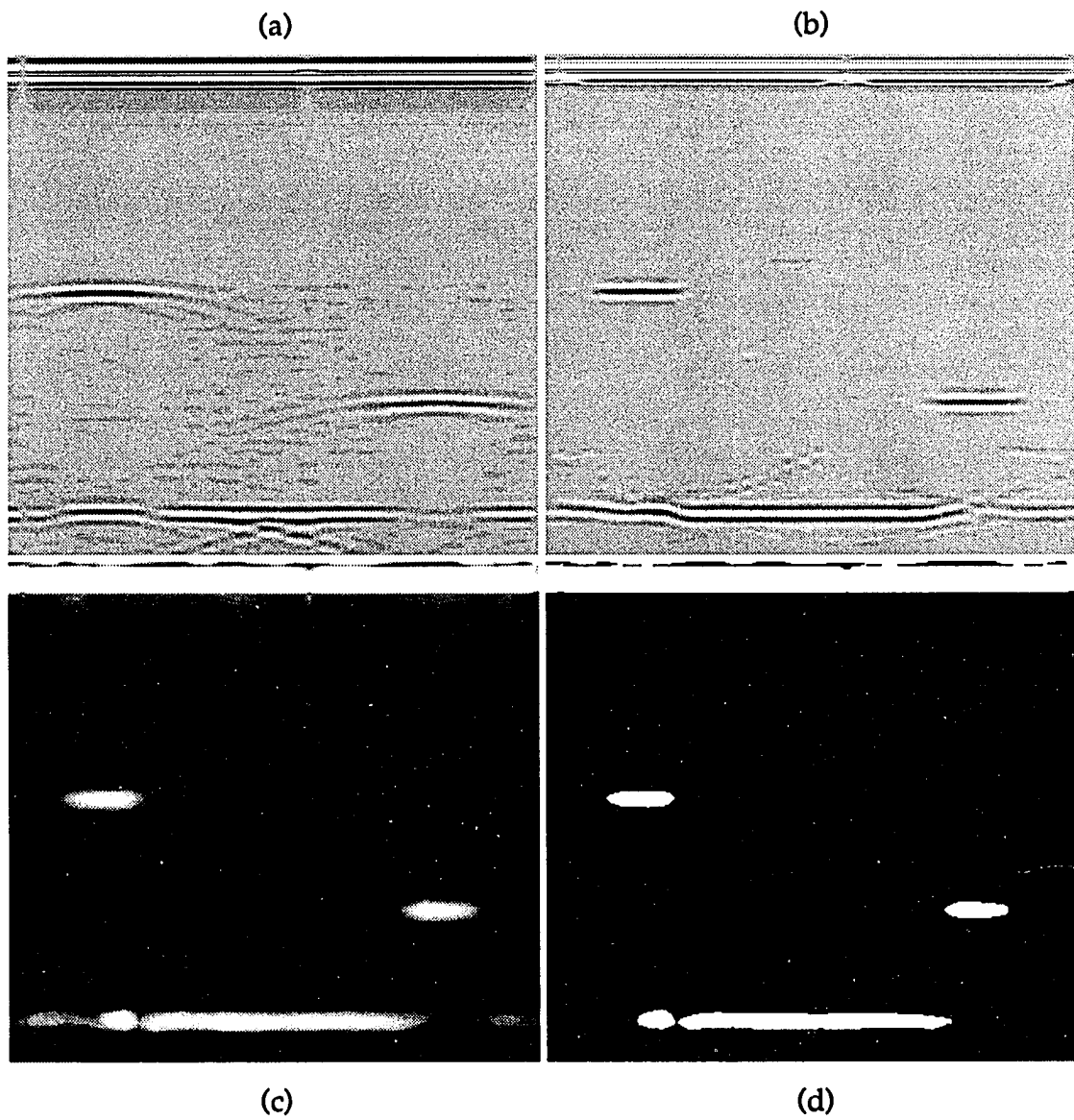


Figure 6.19. Reconstructed FEM reversed geometry using filter in Fig. 6.18
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

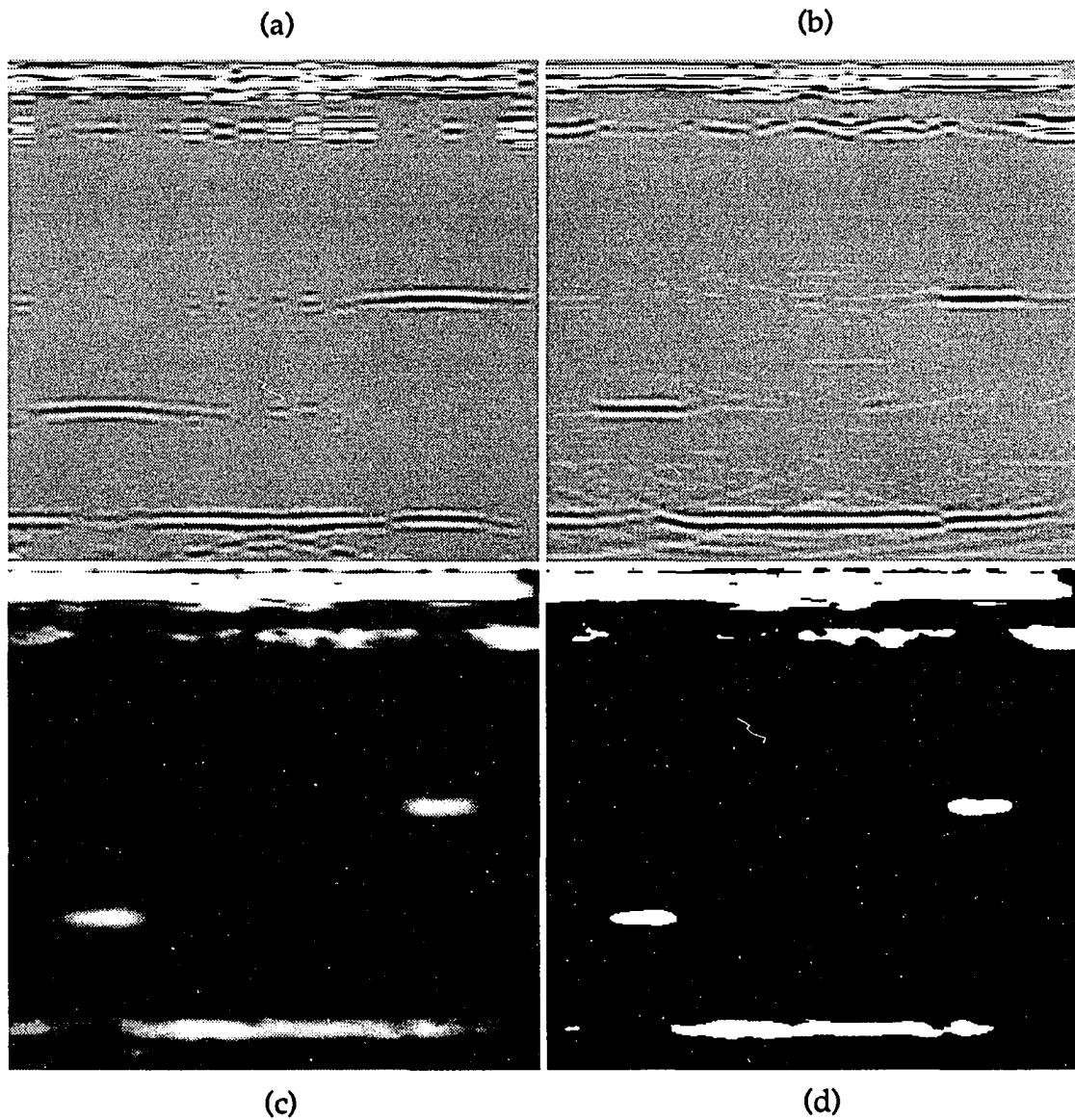


Figure 6.20. Reconstructed FEM actual geometry using filter in Fig. 6.18
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

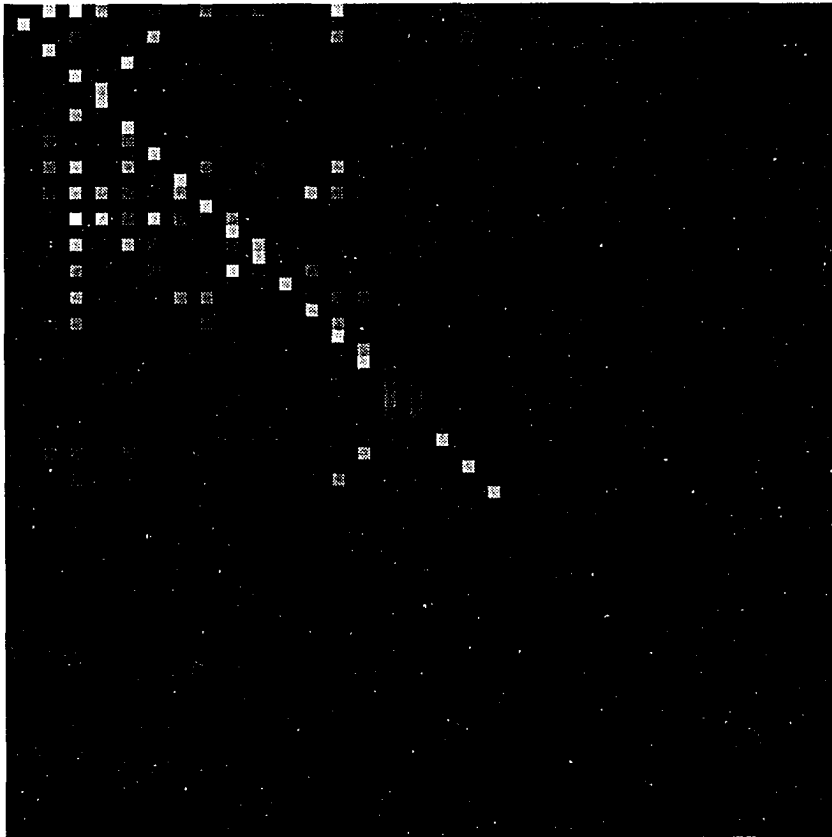


Figure 6.21. Magnitude of frequency domain filter based on 50% (Even) of the FEM reversed geometry

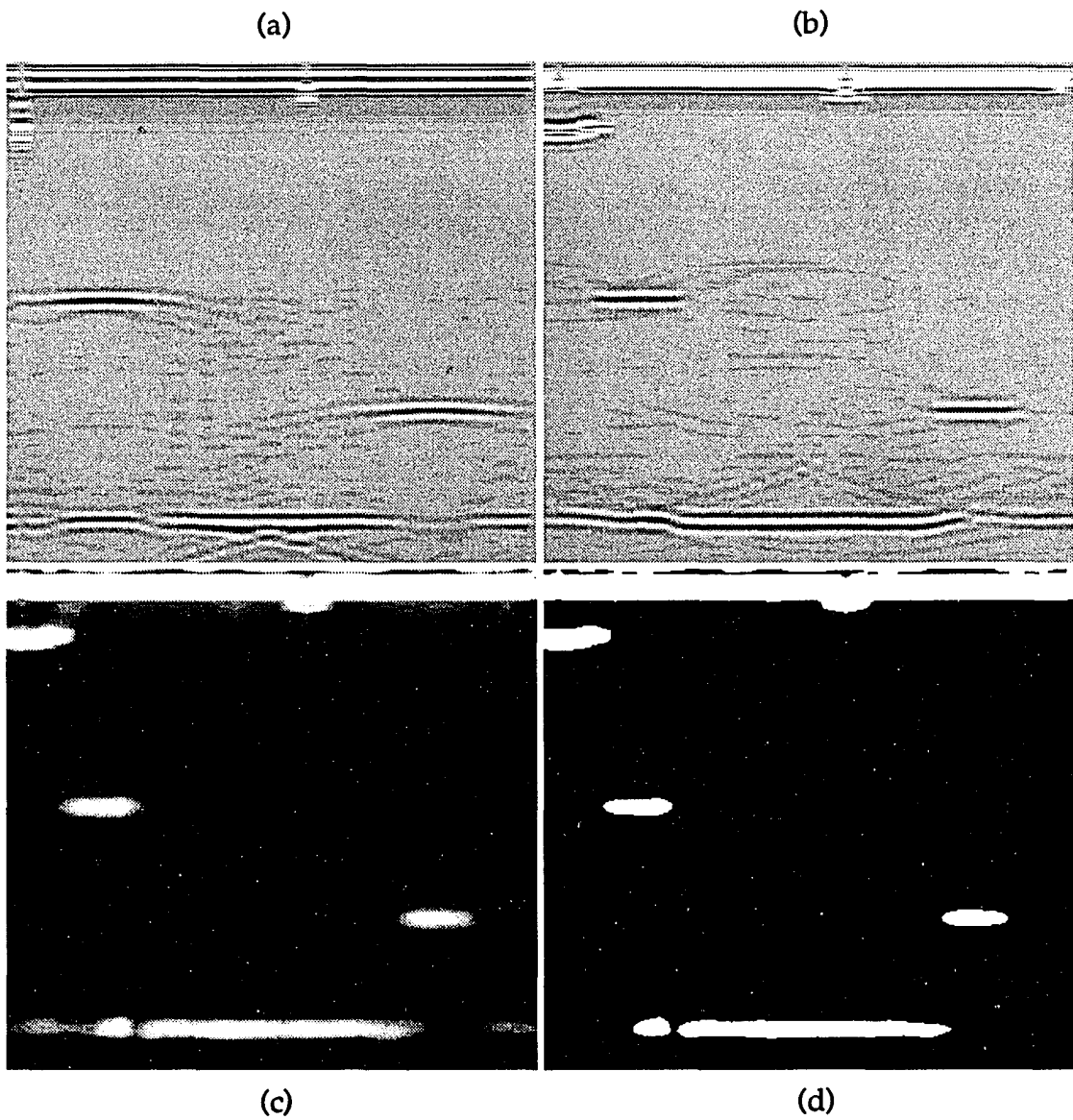


Figure 6.22. Reconstructed FEM reversed geometry using filter in Fig. 6.21
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

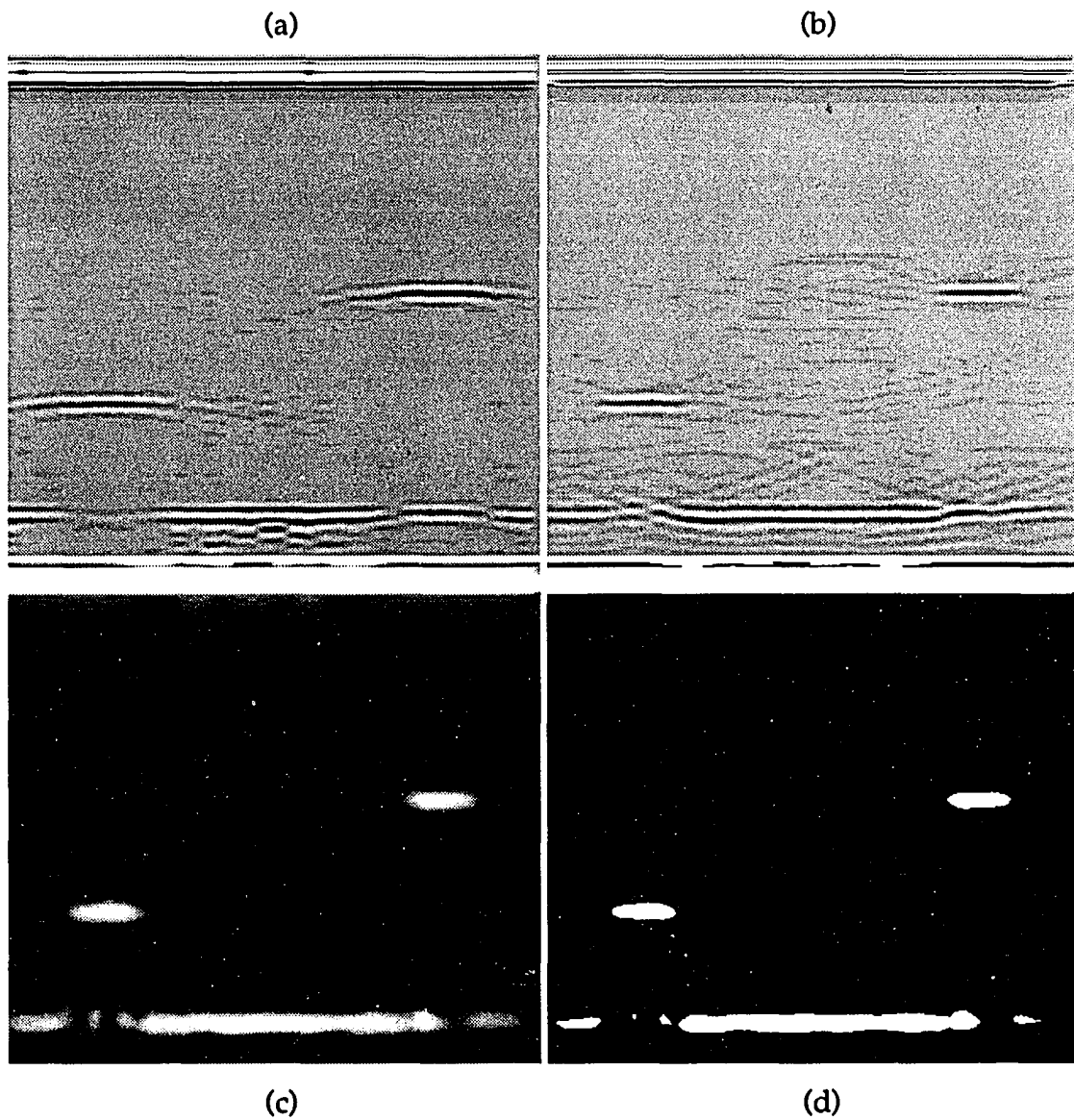


Figure 6.23. Reconstructed FEM actual geometry using filter in Fig. 6.21
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

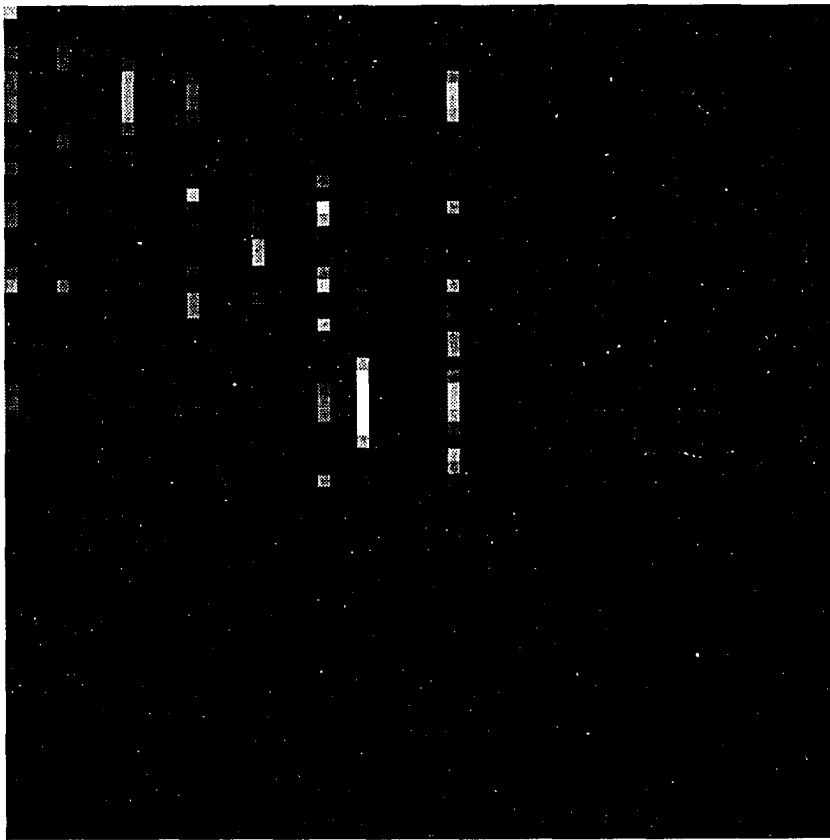


Figure 6.24. Magnitude of frequency domain filter based on 26% (1, 5, 10, 15, 20, 25, 28, 30, 35, 38) of the FEM reversed geometry

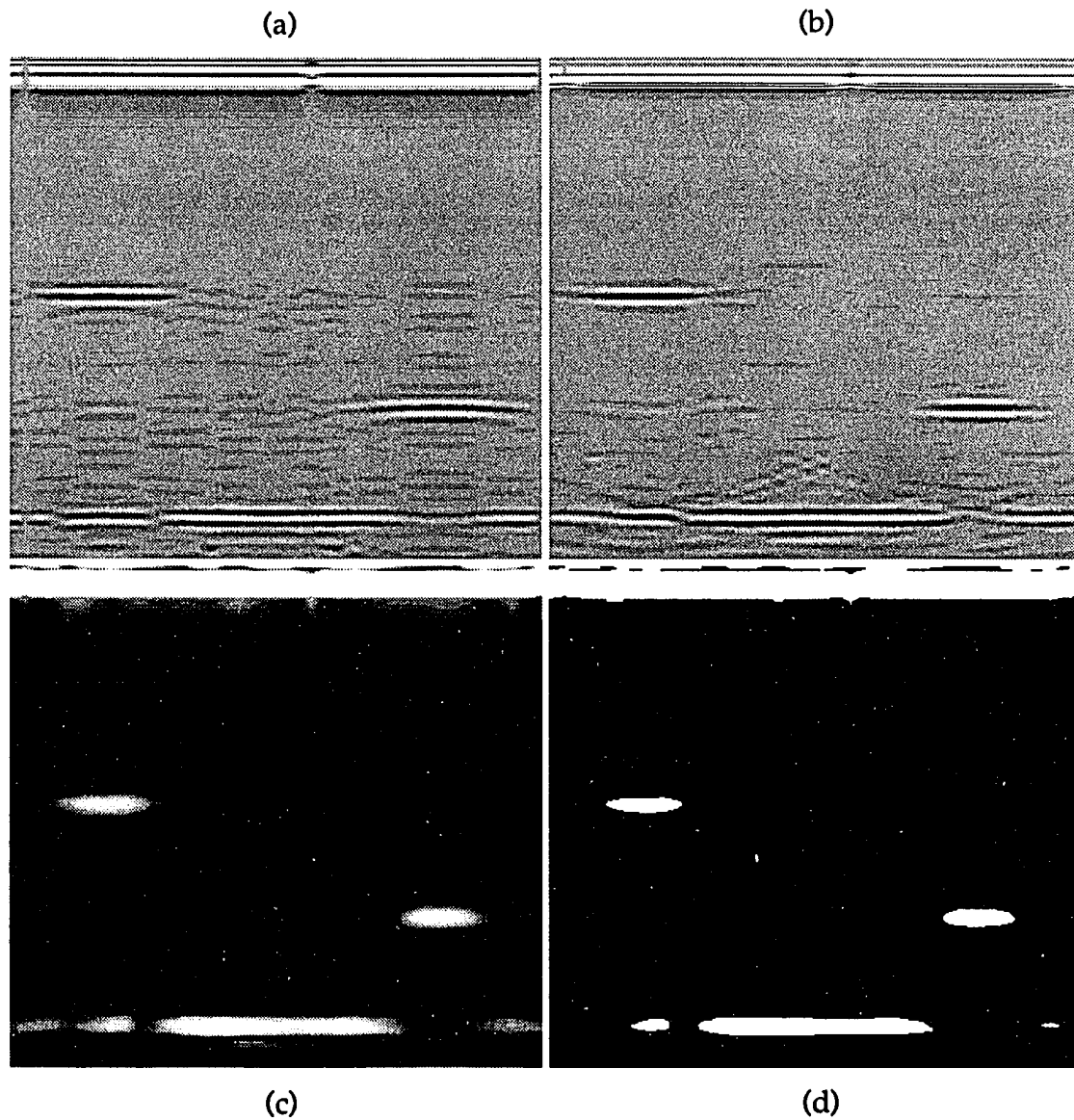


Figure 6.25. Reconstructed FEM reversed geometry using filter in Fig. 6.24
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

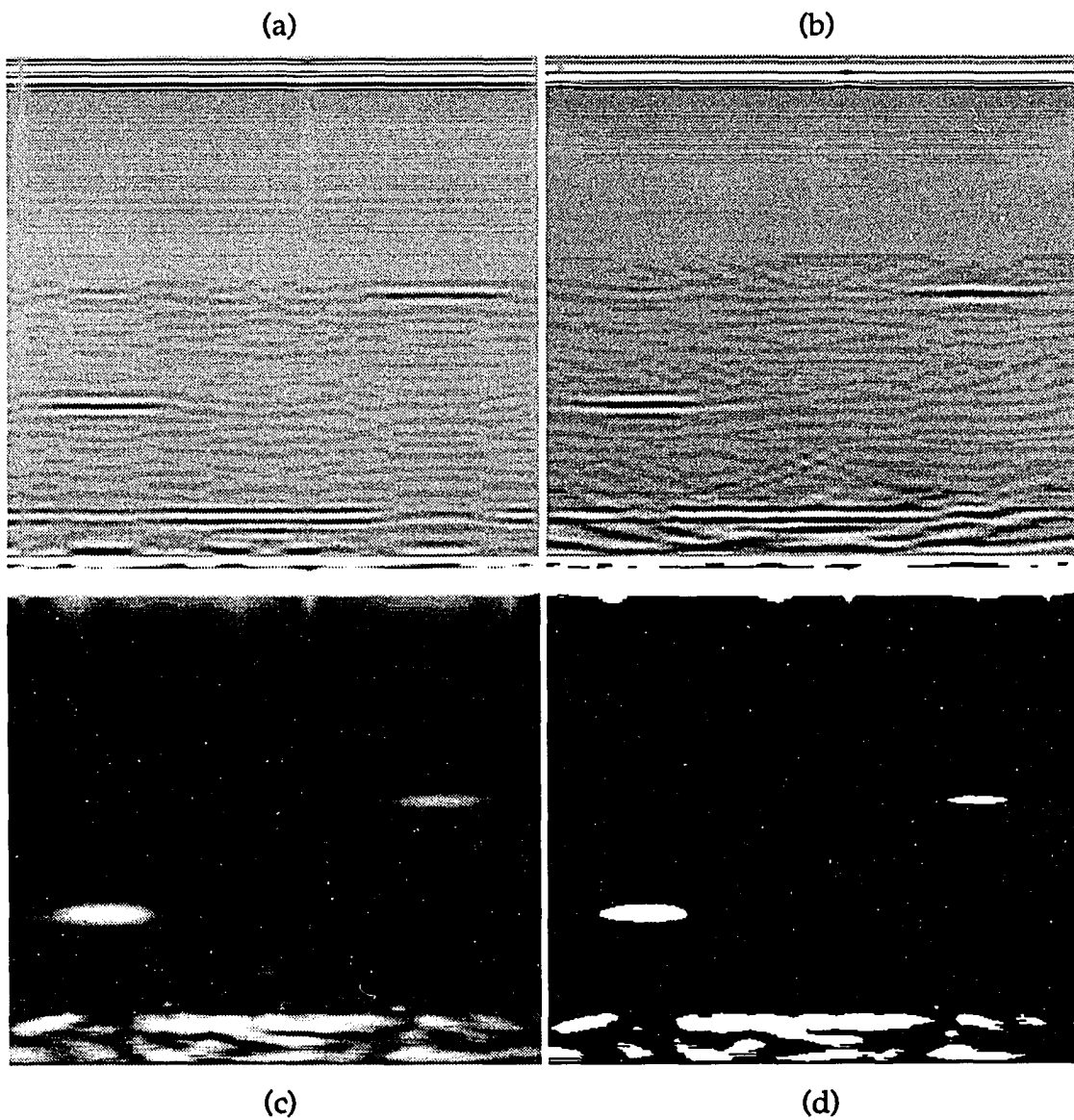


Figure 6.26. Reconstructed FEM actual geometry using filter in Fig. 6.24
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

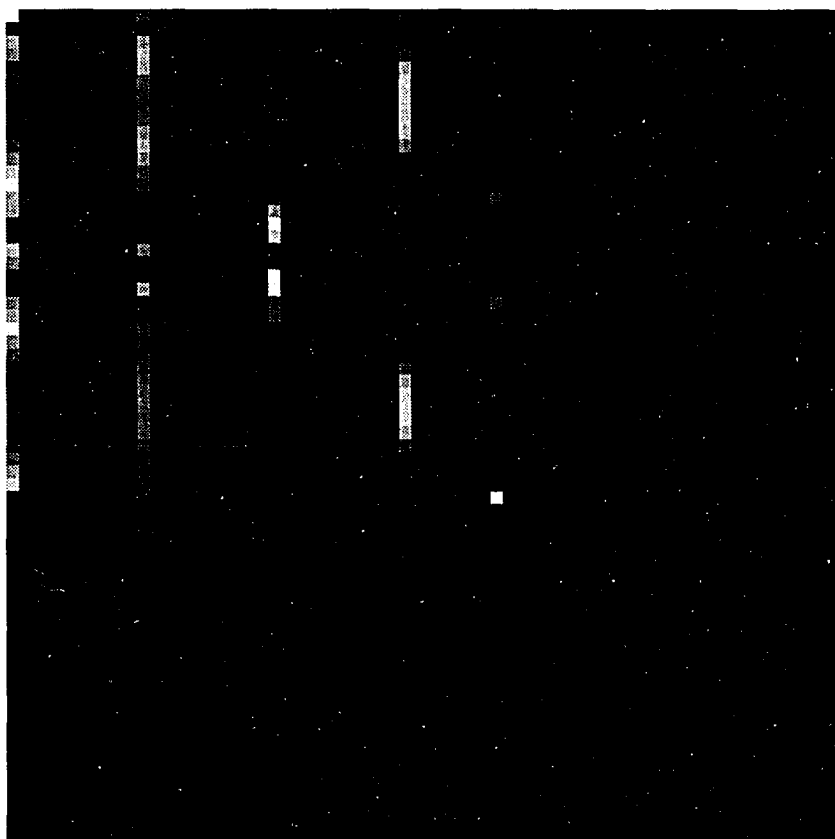


Figure 6.27. Magnitude of frequency domain filter based on 13% (1, 11, 21, 31, 38) of the FEM reversed geometry

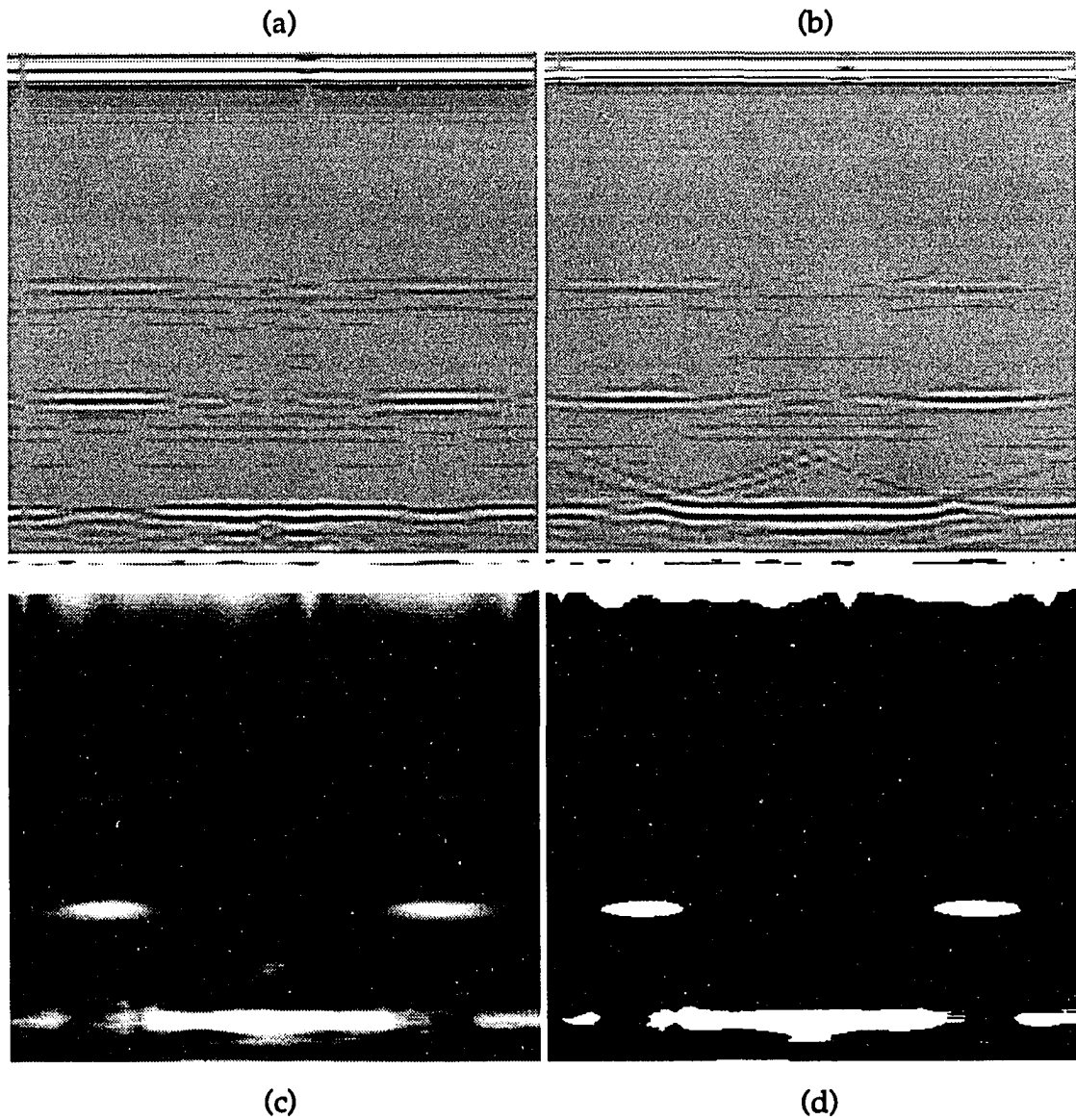


Figure 6.28. Reconstructed FEM reversed geometry using filter in Fig. 6.27
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

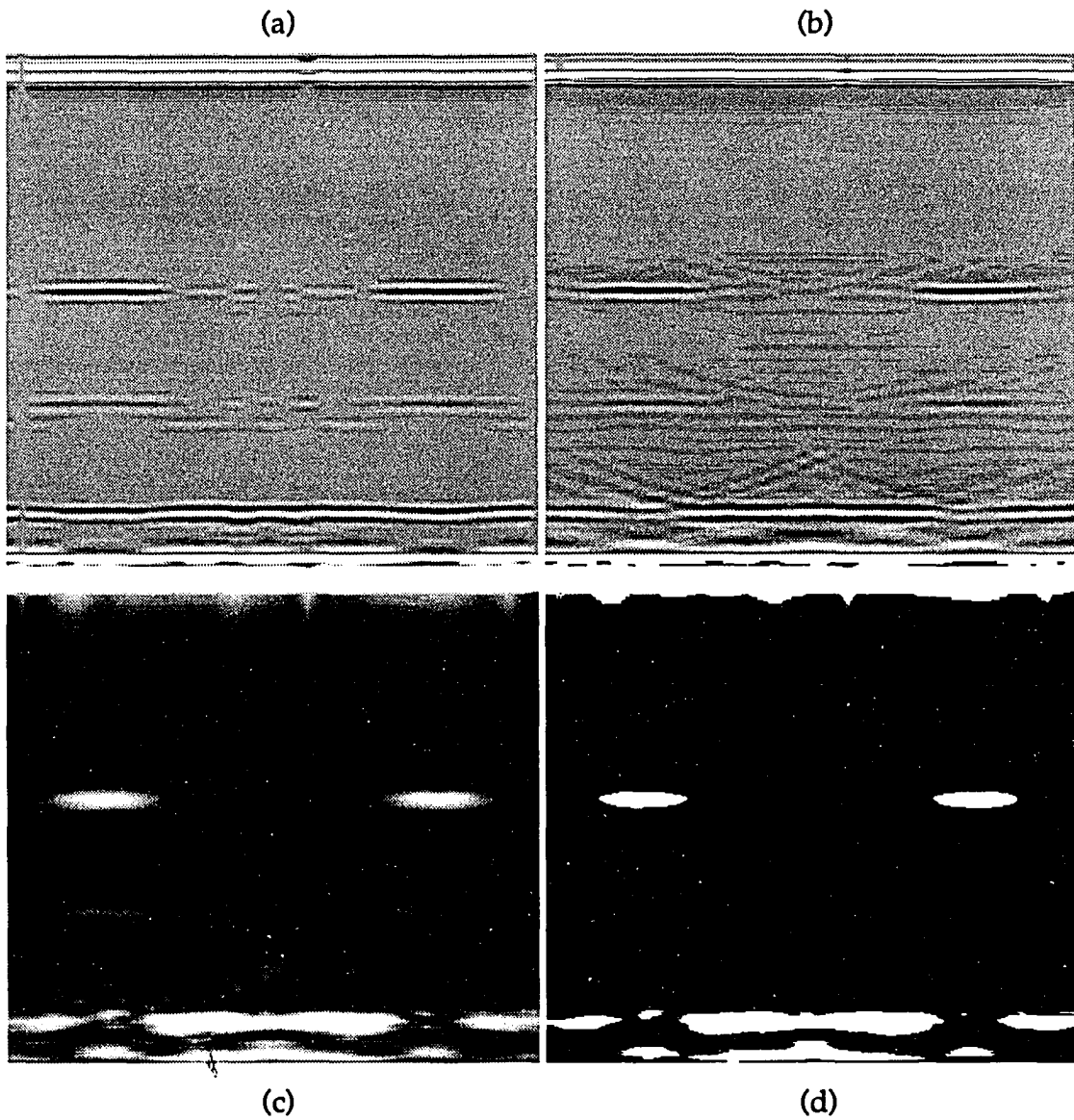


Figure 6.29. Reconstructed FEM actual geometry using filter in Fig. 6.27
(a) Raw B-scan image
(b) SAFT B-scan image
(c) Magnitude of SAFT
(d) Rectified (-6 dB threshold) SAFT image

signals have to be chosen in a scattered arrangement so that the whole medium is covered by the individual insonifying beams of the transducers.

Second, the assessment of the DI beamformer may be looked upon in terms of computing the relative mean-squared error (RMSE) between the reconstructed and the desired images. Tables 6.1 and 6.2 show the RMSE results of each of the tests performed on the nonuniform actual and reversed records, respectively. Since the numerical values of the front and the back walls in the images greatly exceed those of the flaw signals, the RMSE's are calculated using only that portion of the signals containing the flaw signatures. Figures 6.30 and 6.31 show the plots of these RMSE values. Figure 6.30 corresponds to when the optimal filter is based on the actual geometry (dotted line A) and then convolved with the reversed geometry (solid line B); and Figure 6.31 corresponds to when the filter is trained using the reversed apparatus (dotted line A) and then convolved with the actual geometry (solid line B). All four plots in the figures seem to make sense as far as the increasing trend the curves are acquiring as more signals are missing from the records. However, when lines B in both figures are compared, a few remarks are revealed about each of the filters. One, the filters in Figure 6.30 appear to operate better when 50% (odd), 26% or 13% of the reversed geometry are kept than the filters in Figure 6.31 as they operate on the actual geometry when the same percentages of the signals are kept. On the other hand, both sets of filters seem to have similar behaviors when they are convolved with their respective training records. The only mismatch appears when 13% of the data are kept. Here, the filter in Figure 6.30 provides a better smoothing effect than its counterpart in Figure 6.31.

Third, the rectified images displayed in the previous section provide another means of concretely measuring the level of achievement the DI beamformer is capable of attaining as far as flaw sizing is concerned. The widths of the flaws as well as their depths from the surface of the specimen are measured using a straight edge and compared relative to the width (1" line scan) and the height (1" thick) of the images. These dimensions are

Table 6.1. Relative Mean-Squared Error when the filters are based on the FEM actual geometry

Records Kept	Relative Mean-Squared Error (%)	
	Actual	Reversed
50% (odd)	3.29	12.94
50% (even)	4.36	32.36
26%	38.39	42.77
13%	44.00	56.54
3%	129.09	137.78

Table 6.2. Relative Mean-Squared Error when the filters are based on the FEM reversed geometry

Records Kept	Relative Mean-Squared Error (%)	
	Actual	Reversed
50% (odd)	32.54	4.40
50% (even)	13.29	3.38
26%	49.14	29.67
13%	82.37	81.04
3%	129.39	137.69

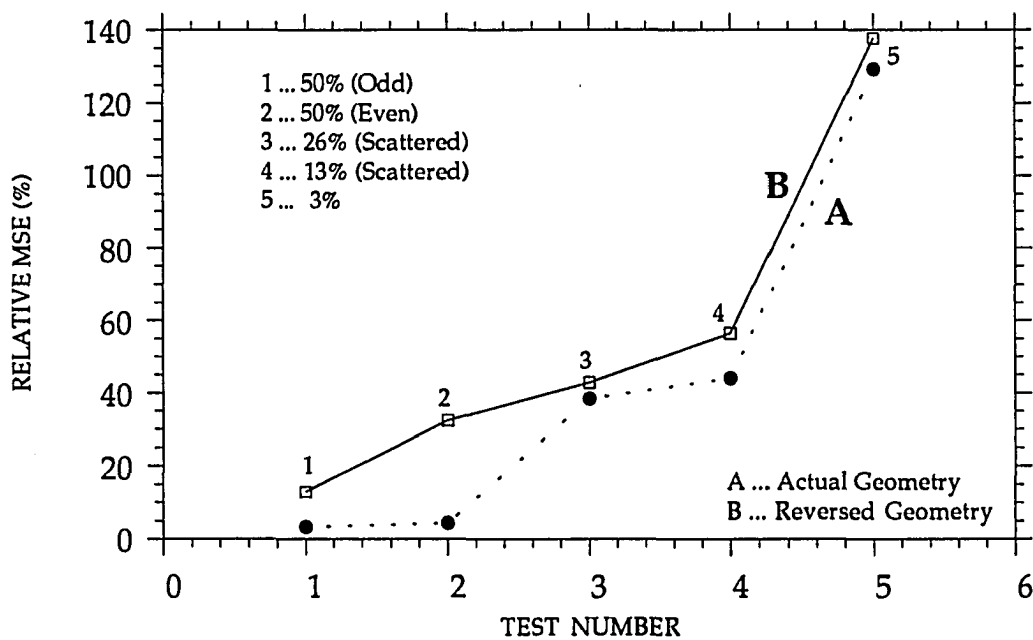


Figure 6.30 Plot comparing RMSE values when filters are trained using FEM actual geometry and applied to FEM actual geometry and to FEM reversed geometry.

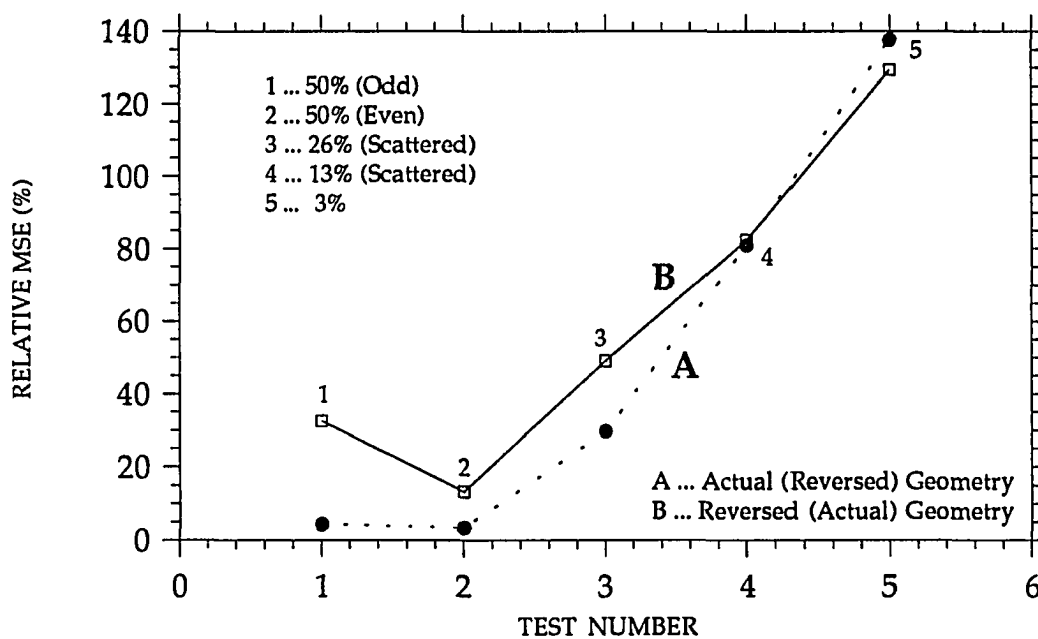


Figure 6.31 Plot comparing RMSE values when filters are trained using FEM reversed geometry and applied to FEM reversed geometry and to FEM actual geometry.

Table 6.3. Flaw sizing in both the longitudinal and the lateral directions when filters are designed using the actual geometry and applied to the reversed geometry

Kept Records	Slot #	Depth Dimension		Width Dimension	
		Actual Geometry	Reversed Geometry	Actual Geometry	Reversed Geometry
50% (Odd)	1	0.74"	0.73"	0.125"	0.125"
	2	0.50"	0.50"	0.125"	0.125"
50% (Even)	1	0.74"	0.73"	0.125"	0.129"
	2	0.50"	0.50"	0.125"	0.129"
26%	1	0.72"	0.71"	0.135"	0.146"
	2	0.49"	0.50"	0.146"	0.167"
13%	1	0.72"	0.71"	0.129"	0.167"
	2	0.47"	0.48"	0.0625"	0.104"
3%	1	---	---	---	---
	2	---	---	---	---

actual and the reversed geometries, in either test. This is only coincidental since the cross-over is otherwise independent of the amount of missing signals for the cases studied.

The following chapter gives a brief summary of the achievements presented in this research topic. Also in that chapter, special emphasis is put forth on remarks pertaining to the beamformer technique just described, followed by a short discussion of experimental suggestions and further work developments.

CHAPTER 7. CONCLUSION

The major objective of this research work has been to investigate a number of methods ranging from the modeling of ultrasonic NDT wave phenomena and using the resulting model to test the SAFT algorithm by mimicking the transducer measurements required by SAFT, to the investigation and application of a newly developed beamforming technique, namely, the data independent beamformer, used to predict ultrasonic signatures from nonuniformly spaced transducer baselines. This chapter summarizes the main contributions and suggests further areas for future study.

Milestones and Major Work Accomplishments

This dissertation has illustrated that the finite element method can indeed be adopted as a test bed for studying the efficaciousness of the SAFT algorithm. One of the advantages of this modeling technique lies in its flexibility as far as the material properties are concerned. Although FEM is used in this work to mimic the SAFT measurements in a Plexiglass™ sample, other materials, such as composites, can also be modeled. Due to time constraints, only Lucite was used as the test piece. With this in mind, the various factors affecting the reconstruction scheme, such as material type, flaw size, flaw depth within the material, and material geometry, are incorporated as variables in the apparatus, and enable one to evaluate the final reconstructed image. Tests show that the flaws were not only sized in the longitudinal direction, but also in the lateral direction, using the shear components produced by the FEM model. This task may become difficult if measurements are collected during a laboratory experiment. However, using the finite element model, a careful study can be made to determine the sensitivity of the parameters involved in the SAFT reconstruction algorithm to changes in material variables.

Using the same finite element model as a test bed, another study was made to examine how well the data independent beamformer works in the presence of nonuniformly spaced transducer locations scanning across the surface of a test specimen. By assuming various baseline configurations to the collected ultrasonic A-scan signals, the weights of the beamformer are designed, and test comparisons confirm that the resulting optimal filters are capable of not only predicting those missing signals in the training record geometry, but also in a different geometry, considered here to having the reversed shape of the training record, with the same number of missing A-scans.

In summary, the following few points can be identified to describe the overall assessment of the beamforming method:

1. The data independent beamformer is a very useful technique, especially when prompt and accurate testing are major requirements in a manufacturing environment of structural components.
2. The beamformer is very simple to understand and implement, and requires no *a priori* knowledge of the statistics of the input signals. However, care must be taken when the filter is presented with a data record. The record must contain the same number of missing signals similar to the record used to train the filter. Otherwise, erroneous results will occur.
3. Although the performance of the beamformer is excellent even when 87% of the signals are missing, except for a few discrepancies, it is wise to note that this high level of missing signals constitutes an upper threshold of the maximum allowable degree of freedom in which the beamformer can operate with a relatively minimum level of error. Therefore, if time is not a major issue, testing an ultrasonic data set with 26% available A-scan signals can be as efficient and more reliable than testing a record with 13% or less available A-scans.

4. The one advantage of this beamforming technique lies within its practicality as far as the areas in which it can be used. Beside nondestructive testing of materials, the fields of radar (i.e., synthetic aperture radar, SAR), and medical ultrasound would make adequate grounds for the application of this beamformer.
5. Finally, and as a general remark, when using this deconvolution scheme, the *Rule of Thumb*, which states that the transducer separation between adjacent positions over a test sample must be at least one-half-wavelength (i.e., $\lambda/2$) in order to achieve good longitudinal and lateral resolutions, is no longer true. Note that by keeping 10 A-scans (26%), the transducer separation is increased to approximately 2λ ; and when 5 A-scans (13%) are kept, this separation is further increased to about 5λ , reducing the scanning time by 87% in the latter case, and by 74% in the previous case.

Future Work

The work reported in this dissertation about the finite element modeling technique, SAFT, and the data independent beamformer, clearly shows that these methodologies are quite powerful at achieving their respective goals. However, further development and refinement of these processes is desirable, in order to consider all practical factors. Proposals for additional studies may include the following avenues:

1. Use finite element modeling to simulate more complicated materials, such as composites; as well as more complex material geometries.
2. Again use the resulting model from 1. as a test bed to check for the sensitivity of the parameters of the SAFT algorithm.
3. In a similar fashion, acquire the same model, once more as a test bed, to

BIBLIOGRAPHY

- [1] J. J. Flaherty, K. R. Erickson and V. M. Lund. "Synthetic Aperture Ultrasonic Imaging Systems." US Patent 3, 548-642. 1967.
- [2] C. B. Burckhardt, P. A. Grandchamp and H. Hoffman. "An Experimental 2 MHz Synthetic Aperture Sonar System Intended For Medical Use." IEEE Trans. on Sonics and Ultrasonics SU-21, No. 1 (1974): 1-6.
- [3] J. R. Frederick, J. A. Seydel and R. C. Fairchild. USNRC Report NUREG-0007-1. 1976.
- [4] J. R. Frederick, R. C. Fairchild and B. H. Anderson. USNRC Report NUREG-0007-2. 1977.
- [5] J. R. Frederick, R. C. Fairchild, C. J. H. Vanden Broek and M. B. Elzinga. USNRC Report NUREG/CR-0135. 1978.
- [6] J. R. Frederick, C. Vanden Broek, M. Elzinga, M. Dixon, D. Papworth, N. Hamano and K. Ganapathy. USNRC Report NUREG/CR-0581. 1979.
- [7] J. Seydel. "Ultrasonic Synthetic-Aperture Focusing Techniques in NDT." In Research Techniques in Nondestructive Testing Vol. VI, ed. R. S. Sharpe, 1-47. New York: Academic Press, 1982.
- [8] J. Johnson. "Parameter Study of Synthetic-Aperture Focusing in Ultrasonics." In Review of Progress in Quantitative NDE Vol. 1, eds. D. O. Thompson and D. E. Chimenti, 735-752. New York: Plenum Press, 1982.
- [9] H. G. Kraus. "Generalized Synthetic Aperture, Focused Transducer, Pulse-Echo, Ultrasonic Scan Data Processing for Non-Destructive Inspection." Ultrasonics 21, No. 1 (1983): 9-15.
- [10] R. N. Thomson. "Transverse and Longitudinal Resolution of the Synthetic Aperture Focusing Technique." Ultrasonics 22, No. 1 (1984): 9-15.

- [11] S. F. Burch and J. T. Burton. "Ultrasonic Synthetic Aperture Focusing Using Planar Pulse-echo Transducers." Ultrasonics 22, No. 6 (1984): 275-281.
- [12] S. R. Doctor, T. E. Hall and L. D. Reid. "SAFT - The Evolution of a Signal Processing Technology for Ultrasonic Testing." NDT International 19, No. 3 (1986): 163-168.
- [13] K. J. Langenberg, M. Berger, Th. Kreutter, K. Mayer and V. Schmitz. "Synthetic Aperture Focusing Technique Signal Processing." NDT International 19, No. 3 (1986): 177-189.
- [14] K. Mayer, R. Marklein, K. J. Langenberg and T. Kreutter. "Three-Dimensional Imaging System Based on Fourier Transform Synthetic Aperture Focusing Technique." Dept. Electrical Engineering, FB 16, University of Kassel, D-3500, Kassel, FRG, April 1989.
- [15] J. A. Högbom. "Aperture Synthesis with Non-Regular Distribution of Interferometer Baselines." Astronomy and Astrophysics Supplement 15 (1974): 417-426.
- [16] B. D. Van Veen and K. M. Buckley. "Beamforming: A Versatile Approach to Spatial Filtering." IEEE ASSP MAGAZINE (April 1988): 4-24.
- [17] D. E. Dudgeon and R. M. Mersereau. *Multidimensional Digital Signal Processing*. 293-315. New Jersey: Prentice-Hall, 1984.
- [18] T. W. Parks and C. S. Burrus. *Digital Filter Design*. New York: Wiley-Interscience, 1987.
- [19] L. O. Frost. "An Algorithm for Linearly Constrained Adaptive Array Processing." Proc. IEEE 60 (1972): 926-935.
- [20] K. M. Buckley. "Spatial/Spectral Filtering with Linearly-Constrained Minimum Variance Beamformer." IEEE Trans. on Acoustic, Speech and Signal Processing ASSP-35 (1987): 249-266.
- [21] B. D. Van Veen and R. A. Roberts. "Partially Adaptive Beamformer Design Via Output Power Minimization." IEEE Trans. on Acoustic, Speech and Signal Processing. ASSP-35 (1987): 1534-1542.

- [22] B. D. Van Veen. "Improved Power Minimization Based Partially Adaptive Beamforming." IEEE Trans. on Acoustic, Speech and Signal Processing ASSP-88 (1988): 2793-2796.
- [23] K. Takao, M. Fujita and T. Nishi. "An Adaptive Antenna Array Under Directional Constraints." IEEE Trans. on Antennas and Propagation AP-24 (1976): 662-669.
- [24] M. Buckley and L. J. Griffiths. "An Adaptive Generalized Sidelobe Canceller With Derivative Constraints." IEEE Trans. on Antennas and Propagation AP-34, No. 3 (1986): 311-319.
- [25] M. H. Er and A. Cantoni. "Derivative Constraints for Broad-Band Element Space Antenna Array Processors." IEEE Trans. on Acoustic, Speech and Signal Processing ASSP-31 (1983): 1378-1393.
- [26] L. J. Griffiths and C. W. Jim. "An Alternative Approach to Linearly Constrained Adaptive Beamforming." IEEE Trans. on Antennas and Propagation AP-30 (1982): 27-34.
- [27] B. Widrow, P. E. Mantey, L. J. Griffiths and B. B. Good. "Adaptive Antenna Systems." Proc. IEEE 55 (1967): 2143-2159.
- [28] S. P. Applebaum and D. J. Chapman. "Adaptive Arrays with Main Beam Constraints." IEEE Trans. on Antennas and Propagation AP-24 (1976): 650-662.
- [29] T. Shan and T. Kailath. "Adaptive Beamforming for Coherent Signals and Interference." IEEE Trans. on Acoustic, Speech and Signal Processing ASSP-33 (1985): 527-536.
- [30] B. Widrow, K. M. Duvall, R. P. Gooch and W. C. Newman. "Signal Cancellation Phenomena in Adaptive Arrays: Causes and Cures." IEEE Trans. on Antennas and Propagation AP-30 (May 1982): 469-478.
- [31] H. Cox, R. M. Zeskind and M. M. Owen. "Robust Adaptive Beamforming." IEEE Trans. on Acoustic, Speech and Signal Processing ASSP-35 (1987): 1365-1375.
- [32] B. G. Wahlberg, I. M. Y. Mareels and I. Webster. "Experimental and Theoretical Comparison of Some Algorithms for Beamforming in Single Receiver Adaptive Arrays." IEEE Trans. on Antennas and Propagation AP-39, No. 1 (1991): 21-28.

- [33] B. Hull and V. John. *Non-destructive Testing*. London: Macmillan Education, 1988.
- [34] L. W. Kessler. "Acoustic Microscopy". Metals Handbook 17. Materials Park: ASM International, 1989.
- [35] L. Brown. *Design of Wideband Ultrasound Instrumentation for Tissue Characterization*. M.S. Thesis, Iowa State University, Ames, Iowa, 1986.
- [36] J. F. Havlice and J. C. Taenzer. "Medical Ultrasonic Imaging: An Overview of Principles and Instrumentation." Proc. IEEE 67 (1979): 620-641.
- [37] D. J. Haumschild and D. L. Carlson. "An Ultrasonic Bragg Scattering Technique for the Quantitative Characterization of Marbling in Beef." Ultrasonics 21, No. 5 (1983): 226-233.
- [38] R. Ludwig. *The Finite Element Modeling of Ultrasonic NDT Phenomena*. Ph.D. Dissertation, Colorado State University, Fort Collins, Colorado, 1986.
- [39] Z. You. *Finite Element Study of Ultrasonic imaging*. Ph. D. Dissertation Iowa State University, Ames, Iowa, 1991.
- [40] A. Macovski. *Medical Imaging Systems*. New Jersey: Prentice-Hall, 1983.
- [41] P. N. T. Wells. *Physical Principles of Ultrasonic Diagnosis*. London: Academic Press, 1969.
- [42] J. P. Woodcock. *Ultrasonics: Medical Physics Handbook 1*. Bristol: Adam Hilger, 1979.
- [43] S. F. Burch. "Comparison of SAFT and Two-dimensional Deconvolution Methods for the Improvement of Resolution in Ultrasonic B-scan Images." Ultrasonics 25 (1987): 259-266.
- [44] T. E. Hall, S. R. Doctor and L. D. Reid. "A Real-time SAFT System Applied to the Ultrasonic Inspection of Nuclear Components." In Review of Progress in Quantitative NDE Vol. 6A, eds. D. O. Thompson and D. E. Chimenti, 509-517. New York: Plenum Press, 1986.

- [45] Y. Ozaki, H. Sumitani, T. Tomoda and M. Tanaka. "A New System for Real-Time Synthetic Aperture Ultrasonic Imaging." IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control UFFC-35, No. 6 (1988): 828-838.
- [46] S. Kramer. "Ultrasonic Weld Defect Sizing Using the Synthetic Aperture Focusing Technique." In Review of Progress in Quantitative NDE Vol. 8B, eds. D. O. Thompson and D. E. Chimenti, 1995-2002. New York: Plenum Press, 1982.
- [47] S. R. Ghorayeb, M. S. Hughes, D. K. Holger and L. W. Zachary. "A Preliminary Application of SAFT on Composites." In Review of Progress in Quantitative NDE Vol. 10A, eds. D. O. Thompson and D. E. Chimenti, 1027-1034. New York: Plenum Press, 1991.
- [48] S. Ghorayeb, M. Yamano, Z. You and W. Lord. "Ultrasonic SAFT Reconstruction using the Finite Element Modeling." In Review of Progress in Quantitative NDE Vol. 11, eds. D. O. Thompson and D. E. Chimenti. New York: Plenum Press, 1992.
- [49] F. A. Chedid-Helou and J. H. Hemann. "Mathematical Modeling of Wave Propagation in Anisotropic Media." Materials Evaluation (June 1991): 708-715.
- [50] R. Ludwig and W. Lord. "Development in the Finite Element Modeling of Ultrasonic NDT Phenomena." In Review of Progress in Quantitative NDE Vol. 5A, eds. D. O. Thompson and D. E. Chimenti, 73-81. New York: Plenum Press, 1986.
- [51] R. Ludwig and W. Lord. "A Finite Element Study of Ultrasonic Wave Propagation and Scattering in an Aluminum Block." Materials Evaluation 46, No. 1 (1988): 108-113.
- [52] Z. J. Cendes. "Unlocking the Magic of Maxwell's Equations." IEEE Spectrum (April 1989): 29-33.
- [53] J. Filipczynski. "Scattering of a Plane Longitudinal Wave on a Free Surface of a Disc in a Solid Medium." In Proc. of Vibration Problems Vol. 2, No. 1 (6), 41-54. Warsaw, Poland, 1961.
- [54] L. W. Schmerr, A. Sedov and C-P Chiou. "A Unified Constrained Inversion Model for Ultrasonic Flaw Sizing." Research in Nondestructive Evaluation 1 (1989): 77-97.

- [55] C. F. Ying and R. Truell. "Scattering of a Plane Longitudinal Wave by a Spherical Obstacle in an Isotropically Elastic Solid." Journal of Applied Physics 27 (1956): 1086.
- [56] Y-H Pao and C. C. Mow. "Theory of Normal Modes and Ultrasonic Spectral Analysis of the Scattering of Waves in a Solid." Journal of Acoustical Society of America 59 (1976): 1046-1056.
- [57] R. M. White. "Elastic Wave Scattering at a Cylindrical Discontinuity in a Solid." Journal of Acoustical Society of America 30 (1958): 771-785.
- [58] T. S. Lewis, D. W. Kraft and N. Hom. "Scattering of Elastic Waves by a Cylindrical Cavity in a Solid." Journal of Applied Physics 47 (1976): 1795-1798.
- [59] U. Aulenbacher and K. J. Langenberg. "Transient Radiation Pattern of Ultrasonic Transducer on Elastic Half-Space." In New Procedures in Nondestructive Testing, ed. P. Holler, 81-92. Berlin: Springer-Verlag, 1983.
- [60] R. Ludwig, D. Moore and W. Lord. "An Analytical and Numerical Study of Transient Force Excitation on an Elastic Half-Space." IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control UFFC-36, No. 3 (1989): 342-350.
- [61] Lecture notes from EM 574: Ultrasonic Nondestructive Measurement Principles. Fall 1990. Iowa State University, Ames, Iowa.
- [62] A. Sedov and L. W. Schmerr. "Pulse Distortion and the Elastodynamic Kirchhoff Approximation for Cracks. The Direct and Inverse Problems." Society for Industrial and Applied Mathematics 47, No. 6 (1987): 1201-1215.
- [63] J. E. Gurbenatis, E. Domany, J. A. Krumhansl and M. Huberman. "The Born Approximation in the Theory of the Scattering of Elastic Waves by Flaws." Journal of Applied Physics 48 (1977): 2812.
- [64] C. A. Brebbia. *New Developments in Boundary Element Methods*. Southampton, England: CML Publications, 1980.
- [65] C. A. Brebbia, J. C. F. Telles and L. C. Wrobel. *Boundary Element Techniques - Theory and Applications in Engineering*. Berlin: Springer-Verlag, 1984.

- [66] F. J. Rizzo, D. J. Shippy and M. Rezayat. "A Boundary Integral Equation Method for Radiation and Scattering of Elastic Waves in Three Dimensions." International Journal of Numerical Methods in Engineering 21 (1985): 115-129.
- [67] D. E. Budreck and J. D. Achenbach. "3-D Ultrasonic Scattering from Planar Cracks by the Boundary Element Method." In Review of Progress in Quantitative NDE, Vol. 7A, eds. D. O. Thompson and D. E. Chimenti, 103-111. New York: Plenum Press, 1988.
- [68] E. Isaacson and H. B. Keller. *Analysis of Numerical Methods*. New York: John Wiley, 1966.
- [69] K-J Bathe. *Finite Element Procedures in Engineering Analysis*. New Jersey: Prentice-Hall, 1982.
- [70] L. J. Bond. "Methods for the Computer Modeling of Ultrasonic Waves in Solids." In Research Techniques in NDT Vol. VI, ed. R. S. Sharpe, 107-150. New York: Academic Press, 1982.
- [71] L. J. Bond. "Computer Modeling in Ultrasonics." Ultrasonics 18 (1980): 43.
- [72] L. J. Bond, M. Punjani and N. Saffari. "Ultrasonic Wave Propagation and Scattering Using Explicit Finite Difference Methods." In Mathematical Modeling in Nondestructive Testing, eds. M. Blakemore and G. A. Georgious, 81-124. Oxford: Clarendon Press, 1988.
- [73] A. H. Harker. *Elastic Waves in Solids - With Applications to Nondestructive Testing of Pipelines*. Bristol, UK: Adam Hilger, 1988.
- [74] K. Harumi. "Computer Simulation of Ultrasonics in a Solid." Materials Evaluation 44 (1986): 1086.
- [75] K. Harumi, T. Saito and T. Fujimori. "Motion Picture of the Computer Simulation of Elastic Waves from Transducers." In Proceedings of the 10th World Conference on Nondestructive Testing. Moscow, 1982.
- [76] W. B. Bickford. *A First Course in the Finite Element Method*. Boston: Richard D. Irwin, 1990.
- [77] N. Kituchi. *Finite Element Methods in Mechanics*. Cambridge: Cambridge University Press, 1986.

- [78] T. J. R. Hughes. *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Englewood Cliffs: Prentice-Hall, 1987.
- [79] J. Lysmer and L. A. Drake. "A Finite Element for Seismology." Methods in Computational Physics 11 (1972): 181-216.
- [80] Z. You, M. Lusk, R. Ludwig and W. Lord. "Numerical Simulation of Ultrasonic Wave Propagation in Anisotropic and Attenuative Solid Materials." IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control UFFC-38, No. 5 (1991): 436-445.
- [81] J. Krautkrämer and H. Krautkrämer. *Ultrasonic Testing of Materials*. New York: Springer-Verlag, 1977.
- [82] R. G. Pridham and R. A. Mucci. "A Novel Approach to Digital Beamforming." Journal of Acoustical Society of America 63, No. 2 (1978): 425-434.
- [83] S. Haykin. *Adaptive Filter Theory*. Englewood Cliffs: Prentice-Hall, 1991.
- [84] J. Capon. "High-Resolution Frequency-Wavenumber Spectrum Analysis." Proc. IEEE 57 (1969): 1408-1418.
- [85] N. L. Owsley. "Sonar Array Processing." In Array Signal Processing, ed. S. Haykin, 115-193. Englewood Cliffs: Prentice-Hall, 1985.
- [86] R. G. Brown. *Introduction to Random Analysis and Kalman Filtering*. New York: John Wiley, 1983.
- [87] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Baltimore: The Johns Hopkins University Press, 1989.
- [88] G. W. Stewart. *Introduction to Matrix Computations*. New York: Academic Press, 1973.
- [89] R. F. Harrington. "Sidelobe Reduction by Nonuniform Element Spacing." IRE Transactions on AP AP-9 (1961): 187-192.
- [90] A. Ishimaru. "Theory of Unequally-Spaced Arrays." IRE Transactions on AP AP-10 (1962): 691-702.
- [91] D. J. Chapman. "Partial Adaptivity for the Large Array." IEEE

Transactions on AP AP-24 (Sept. 1976): 685-696.

- [92] R. G. Pridham and R. A. Mucci. "Digital Interpolation Beamforming for Low-Pass and Bandpass Signals." Proc. IEEE 67, No. 6 (1979): 904-919.
- [93] T. J. Cornwell and K. F. Evans. "A Simple Maximum Entropy Deconvolution Algorithm." Astronomy and Astrophysics 143 (1985): 77-83.
- [94] N. Ahmed and T. Natarajan. *Discrete-Time Signals and Systems*. Reston: Reston, 1983.
- [95] A. A. Giordano and F. M. Hsu. *Least Squares Estimation with Applications to Digital Signal Processing*. New York: John Wiley, 1985.
- [96] S. J. Orfanidis. *Optimum Signal Processing: An Introduction*. New York: Mc Millan, 1985.
- [97] L. L. Scharf. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Reading: Addison-Wesley, 1991.
- [98] B. D. Steinberg and H. M. Subbaram. *Microwave Imaging Techniques*. New York: John Wiley, 1991.
- [99] V. W. Hansen. *Numerical Solution of Antennas in Layered Media*. Somerset: Research Studies Press, 1989.
- [100] R. C. Gonzalez and P. Wintz. *Digital Image Processing*. Reading: Addison-Wesley, 1987.

APPENDIX A. SAFT PROGRAMS

```

/* ##### */
/* ###          Synthetic Aperture Focusing Technique      ### */
/* ###          in Ultrasonic NDE                          ### */
/* ###          For                                        ### */
/* ###          EXPERIMENTAL DATA                        ### */
/* ###          Synthetic Aperture Focusing Technique has been adopted ### */
/* ###          to attain high resolution in both direction, and chara- ### */
/* ###          cterization defect, such as size, shape and orientation. ### */
/* ###          This routine produces corelated A-scans by summing up ### */
/* ###          received ultrasonic signals which are delayed to com- ### */
/* ###          pensate for ultrasonic path delay.         ### */
/* ###          processing flow                            ### */
/* ###          1) initial parameter set up                ### */
/* ###          2) inputting raw A-scan data                ### */
/* ###          3) directivity correction                   ### */
/* ###          4) delay and sum                            ### */
/* ###          5) output correlated A-scans                ### */
/* ###          Note: This program is designed to work ONLY on experi- ### */
/* ###          mental data (Plexiglass or Graphite/Epoxy) ### */
/* ###          ##### */

```

```

#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <stdio.h>
#include <malloc.h>
#include <math.h>
#define SQR(a) ((a)*(a))
#define PI 3.14159265358979323846

```

```

Display *d;
Window win,w,w1,w2;
GC gc;
XPoint p[700][2048];

```

```

main()
{
    int ndat,nscan,nbyte,fp1,nr1,i,h,j,k,gdati,c_flag,**ipdat;
    int sa_size,q_flag,ndir,t_delay,a_flag,dir_angle,add_no,f_flag;
    int **imatix(),sp;
    char *file_name,*file_name1,*file_name2;
    char *file_name3,*file_name4;
    float **a_dat,cond[30],imax,imin,scale,dumy[2048],dumy1,dumy2;
    float dx,dt,u_vel,*dir_dat,ls_angle;
    float **c_dat,limit,fs;

```

```

float *vector(),**matrix();
double pass;
void nrerror(),free_vector(),free_matrix(),free_imatrix();
FILE *fp, *fopen();

unsigned int w_width,w_height,w_width1,w_height1;
int x_st,y_st,gdat;
int x_st0,y_st0,x_st1,y_st1;
float w_max;

/* ### parameters setting for X-Window ### */
/*
d = XOpenDisplay (NULL);

x_st = 40 ,y_st = 20;
x_st0 = 40 ,y_st0 = 500;
x_st1 = 600,y_st1 = 20;
w_width = 512,w_height = 400;
w_width1= 400,w_height1= 400;

w = XCreateSimpleWindow
(d,RootWindow(d,0),x_st,y_st,w_width,w_height,2,0,1);
w1= XCreateSimpleWindow
(d,RootWindow(d,0),x_st0,y_st0,w_width,w_height,2,0,1);
w2= XCreateSimpleWindow
(d,RootWindow(d,0),x_st1,y_st1,w_width1,w_height1,2,0,1);
*/
/* ### 1) initial parameter set ### */

printf("\n Choose raw data filename: \n  1) Plexiglass data
\n  2) Graphite/Epoxy Data\n");
scanf("%d",&f_flag);

if (f_flag == 1) {
    file_name = "plsechplex.dat";
    file_name1 = "plexraw.dat";
    file_name2 = "plexsaf.dat";
    file_name3 = "plexenv.dat";
}

if (f_flag == 2) {
    file_name = "plsechcomp.dat";
    file_name1 = "compraw.dat";
    file_name2 = "compsaf.dat";
    file_name3 = "compenv.dat";
}

printf("\n Enter the transducer separation (mm)\n");

```

```

scanf("%f",&dx);
printf("\n Enter the sampling frequency (MHz)\n");
scanf("%f",&fs);
dt = 1/fs;

printf("\n Enter the ultrasonic sound velocity in specimen (mm/usec)\n");
scanf("%f",&u_vel);

printf("\n Enter the number of data points per A-scan\n");
scanf("%d",&ndat);

printf("\n Enter the total number of A-scans\n");
scanf("%d",&nscan);

printf("\n Choose the method to compute synthetic aperture: \n  1) user chosen
aperture or \n  2) aperture computed by the method of fixed angle\n");
scanf("%d",&a_flag);

if (a_flag == 1) {
    printf("\n Enter synthetic aperture size (number of A-scans) \n");
    scanf("%d",&sa_size);
}
else {
    printf("\n Enter the critical angle of the beam (degrees)\n");
    scanf("%f",&ls_angle);
}

/* ### 2) Open file to input raw A-scans data ### */

if (a_flag == 1) { /* 0 padding for extension of edge data */
    add_no = (int) (sa_size-1)/2;
}
else {
    add_no = (int) (u_vel*dt*ndat*tan(ls_angle/180*PI)/2/dx);
}

a_dat = matrix(1,nscan+2*add_no,1,ndat);

for(h=1;h<=nscan+2*add_no;h++) { /* 0 padding for edge data */
    for(i=1;i<=ndat;i++) {
        a_dat[h][i] = 0;
    }
}
printf("\n Reading the raw data \n");
fp = fopen(file_name,"r"); /* inputting raw data */

for(i=0;i<ndat;i++) {

```

```

        fscanf(fp,"%f",&dumy[i]);
    }
    fscanf(fp,"%f%f",&dumy1,&dumy2);
    for(i=0;i<ndat;i++) {
        fscanf(fp,"%f",&dumy[i]);
    }
    fscanf(fp,"%f%f",&dumy1,&dumy2);

    for(h=1;h<=nscan-1;h++) { /* h:xducer posi. i:data points per A-scan */
        if(h!=1) {fscanf(fp,"%f%f",&dumy1,&dumy2);}
        for(i=1;i<=ndat;i++) {
            fscanf(fp,"%f",&a_dat[h+add_no][i]);
        }
    }
    fclose(fp);
    printf("\n Finished reading the raw data \n");

    /* ### Convert Raw Data to Image Processor Format ## */

    ipdat = imatrix(1,nscan,1,ndat);

    imax = a_dat[add_no+1][1] , imin = a_dat[add_no+1][1];

    for(h=add_no+1;h<=add_no+nscan;h++) {
        for(i=1;i<=ndat;i++) {
            if(imax<a_dat[h][i]) imax = a_dat[h][i];
            if(imin>a_dat[h][i]) imin = a_dat[h][i];
        }
    }

    scale = 255./(imax-imin);

    printf("\n max = %f  min = %f  scale = %f\n",imax,imin,scale);

    for(h=1;h<=nscan;h++) {
        for(i=1;i<=ndat;i++) {
            ipdat[h][i] = (int)(scale*(a_dat[h+add_no][i] - imin));
        }
    }

    fp = fopen(file_name1,"w"); /* output raw data of happi format */
    fprintf(fp,"%d\n%d\n",512,512);

    for(i=1;i<=ndat/4;i++) {
        for(h=40;h<40+512;h++) {
            fprintf(fp,"%d\n",ipdat[h][i]);
        }
    }

```

```

fclose(fp);

printf("\n Would you like to plot the raw data using X-Window?
(yes = 1)\n");
scanf("%d",&q_flag);

if(q_flag == 1) {

mapping(&a_dat[add_no+1],nscan,ndat,&gdat,w_width,w_height,&w_max,0,&sp);
    graph(d,w,gdat,nscan,&sp);
}

printf("\n Would you like to continue this process? (yes = 1)\n");
scanf("%d",&q_flag);
if (q_flag != 1) exit(0);

/*
for(h=add_no;h<=nscan+add_no-1;h++) {
    printf("scanning no = %d\n",h);
for(i=1;i<=ndat;i++) {
    printf("%d %f\n",i,a_dat[h][i]);
}
}
*/

/* ### 3) directivity correction ### */

printf(" Would you like to account for the directivity of the
transducer (yes=1 no=0)\n");
scanf("%d",&q_flag);

ndir = 90 ;      /* dir_angle from 0 to 90 deg a deg pitch */
dir_dat = vector(0,ndir);

dir_cal(dir_dat,q_flag,0,ndir);

/* ### 4) delay and sum ### */

printf("\n SAFT in progress \n");

c_dat = matrix(1,nscan,1,ndat);

if (a_flag == 1) { /* fixed number aperture */
for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        c_dat[h][i] = a_dat[h+add_no][i];
        for(j=1;j<=(sa_size-1)/2;j++) {

```



```

        pass = SQR(i)+SQR(2*dx*j/u_vel/dt);
        t_delay = (int) sqrt(pass);
        dir_angle = (int)fabs((acos((double)(i)/t_delay)*180/PI));
        c_dat[h][i] = c_dat[h][i]+( a_dat[h-j+add_no][t_delay]
        + a_dat[h+j+add_no][t_delay] )/dir_dat[dir_angle];
    }
}
}
else { /* fixed angle aperture */
for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        c_dat[h][i] = a_dat[h+add_no][i];
        limit = u_vel*i*dt*tan(ls_angle/180.0*PI)/2/dx;
        for(j=1;j<=limit;j++) {
            pass = SQR(i)+SQR(2*dx*j/u_vel/dt);
            t_delay = (int) sqrt(pass);
            dir_angle = (int)fabs((acos((double)(i)/t_delay)*180/PI));

            c_dat[h][i] = c_dat[h][i]+( a_dat[h-j+add_no][t_delay]
            + a_dat[h+j+add_no][t_delay] )/dir_dat[dir_angle];
        }
    }
}
}

printf("\n SAFT has successfully ended \n");

/* ### Convert data to Image Processor Format ### */

imax = c_dat[1][1] , imin = c_dat[1][1];

for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        if(imax<c_dat[h][i]) imax = c_dat[h][i];
        if(imin>c_dat[h][i]) imin = c_dat[h][i];
    }
}

scale = 255./(imax-imin);

for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        ipdat[h][i] = (int)(scale*(c_dat[h][i] - imin));
    }
}

```

```

/* ### 5) output correlated A-scans ### */

printf("\n Writing data out to files \n");

fp = fopen(file_name2,"w");
fprintf(fp,"%d\n%d\n",512,512);

for(i=1;i<=ndat/4;i++) {
    for(h=40;h<40+512;h++) {
        fprintf(fp,"%d\n",ipdat[h][i]);
    }
}
fclose(fp);

/* ### graphic routine using X-Window ### */

printf("\n Would you like to plot the SAFT data using X-Window?
(yes = 1)\n");
scanf("%d",&q_flag);
if(q_flag == 1) {
    mapping(c_dat,nscan,ndat,&gdatt,w_width,w_height,&w_max,0,&sp);
    graph(d,w1,gdatt,nscan,&sp);
}

printf("\n Computing the envelope of the SAFT image \n");

envlp(c_dat,nscan,ndat);

/* ### output envelop detected data ### */

imax = c_dat[1][1] , imin = c_dat[1][1];

for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        if(imax<c_dat[h][i]) imax = c_dat[h][i];
        if(imin>c_dat[h][i]) imin = c_dat[h][i];
    }
}

scale = 255./(imax-imin);

for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        ipdat[h][i] = (int)(scale*(c_dat[h][i] - imin));
    }
}

```

```

fp = fopen(file_name3,"w");
fprintf(fp,"%d\n%d\n",512,512);
for(i=1;i<=ndat/4;i++) {
    for(h=40;h<40+512;h++) {
        fprintf(fp,"%d\n",ipdat[h][i]);
    }
}
fclose(fp);

printf("\n Would you like to plot the envelope of SAFT image using
X-Window? (yes = 1)\n");
scanf("%d",&q_flag);
if(q_flag ==1) {
    mapping(c_dat,nscan,ndat,&gdat,w_width1,w_height1,&w_max,0,&sp);
    graph(d,w2,gdat,nscan,&sp);
}
scanf("%d",&h);
XCLOSEDisplay (d);

/* ### memory free ### */

free_vector(dir_dat,0,ndir);
free_matrix(a_dat,1,nscan+2*add_no,1,ndat);
free_matrix(c_dat,1,nscan,1,ndat);
free_imatrix(ipdat,1,nscan,1,ndat);
}
graph(d,win,gdat,ntime,sp)
int gdat,ntime,*sp;

{
    int i;
    XSetWindowAttributes a;
    unsigned int l_width;

    a.override_redirect = 1;
    XChangeWindowAttributes (d,win,CWOverrideRedirect,&a);

    XMapWindow (d,win);

    gc = XCreateGC (d,win,0,0);

    l_width = 1;

    XSetLineAttributes(d,gc,l_width,LineSolid,CapRound,JoinRound);
    printf("step = %d\n",*sp);
}

```

```

for(i=0;i<ntime/(*sp);i++) {
    XDrawLines (d,win,gc,p[i],gdat,CoordModeOrigin);
}

XFlush(d);

}
mapping(dat,nscan,ndat,gdat,w_width,w_height,w_max,flag,sp)
unsigned int w_height,w_width;
int nscan,ndat,*gdat,flag,sp[1];
float **dat,*w_max;

{
    float ndd,step,nc;
    int nd,i,h,i_sift,ns;

    i_sift = 10;

    printf(" Enter the stepsize of the plot (once per ? A-scans)\n");
    scanf("%d",sp);

    ns = (int)(((float) w_width)/nscan*sp[0]);
    ndd = ((float) (ndat))/(w_height-i_sift);

    if(ndd<=1.0) {
        nd = 1;
        step = ndd;
    }
    else {
        nd = (int)ceil(ndd);
        step = 1;
    }
    if(flag == 0) {
        *w_max = 0;

        for(h=1;h<=nscan;h++) {
            for(i=1;i<=ndat;i++) {
                if(*w_max<fabs(dat[h][i])) *w_max = fabs(dat[h][i]);
            }
        }
    }

    nc = *w_max/ns*2;

    *gdat = (int)(ndat/nd) + ndat%nd;

```

```

    for(h=0;h<nscan/sp[0];h++) {
        for(i=0;i<*gdat;i++) {
            p[h][i].x = (short)(dat[h*sp[0]+1][i*nd+1]/nc*3+ns*(h+0.5));
            p[h][i].y = (short)(i/step+i_sift);
        }
    }
}
envlp(dat,nscan,ndat)
float **dat;
int ndat,nscan;
{
    int fft_flag,i,h;
    float *fft_dat;
    float *vector();
    void four1(),free_vector();

    fft_dat = vector(1,2*ndat);

    for(h=1;h<=nscan;h++) {
        for(i=1;i<=ndat;i++) {
            fft_dat[2*i-1] = dat[h][i];
            fft_dat[2*i] = 0;
        }

        fft_flag = 1;

        four1(fft_dat,ndat,fft_flag);

        for(i=ndat+1;i<=2*ndat;i++) {
            fft_dat[i] = 0;
        }

        fft_dat[1] = 0 , fft_dat[2] = 0;

        fft_flag = -1;

        four1(fft_dat,ndat,fft_flag);

        for(i=1;i<=ndat;i++) {
            dat[h][i] = sqrt(fft_dat[2*i-1]*fft_dat[2*i-1] +
                fft_dat[2*i]*fft_dat[2*i]);
        }
    }

    free_vector(fft_dat,1,2*ndat);
}

```

```

/* ##### */
/* ####                                     #### */
/* ####      Synthetic Aperture Focusing Technique      #### */
/* ####              in Ultrasonic NDE                  #### */
/* ####              For                                #### */
/* ####      FINITE ELEMENT SIMULATED DATA            #### */
/* ####                                               #### */
/* ####                                               #### */
/* #### Synthetic Aperture Focusing Technique has been adopted #### */
/* #### to attain high resolution in both direction, and charac- #### */
/* #### terization defect, such as size, shape and orientation. #### */
/* #### This routine produces correlated A-scans by summing up #### */
/* #### received ultrasonic signals which are delayed to com- #### */
/* #### pensate for ultrasonic path delay.              #### */
/* ####                                               #### */
/* #### processing flow                                #### */
/* #### 1) initial parameter set up                    #### */
/* #### 2) Reading raw A-scans data                    #### */
/* #### 3) directivity correction                      #### */
/* #### 4) delay and sum                               #### */
/* #### 5) output correlated A-scans                   #### */
/* ####                                               #### */
/* #### Note: This program is designed to work ONLY on Finite #### */
/* ####      Element simulated data (regular or reversed geo- #### */
/* ####      metries)                                   #### */
/* ####                                               #### */
/* ####                                               #### */
/* ##### */

```

```

#include <stdio.h>
#include <malloc.h>
#include <math.h>
#define SQR(a) ((a)*(a))
#define PI 3.14159265358979323846

main()
(
int ndat,nscan,nbyte,fp1,nr1,i,h,j,k,gdati,c_flag,**ipdat;
int sa_size,q_flag,ndir,t_delay,a_flag,dir_angle,add_no,f_flag;
int **imatrix(),sp,ndat1,xs,zs;
char file_name[50],*file_name1,*file_name2;
char *file_name3,*file_name4;
float **a_dat,cond[30],imax,imin,scale,**dummy,dummy1,dummy2;
float dx,dt,u_vel,*dir_dat,ls_angle;
float **c_dat,limit,fs;
float *vector(),**matrix();
double pass;
void nerror(),free_vector(),free_matrix(),free_imatrix();

```

```

FILE *fp, *fopen();

float w_max;

/* ### 1) initial parameter setup ### */

/* file_name = "finite.dat, finite1.dat, ascan1, or ascan2"; */
printf("\n Enter the name of the FEM data file: \n");
scanf("%s",file_name);

file_name1 = "finraw2.dat";
file_name2 = "finsaf2.dat";
file_name3 = "finenv2.dat";

printf("\n Enter the transducer separation (mm)\n");
scanf("%f",&dx);

printf("\n Enter the sampling frequency (MHz)\n");
scanf("%f",&fs);

dt = 1/fs;

printf("\n Enter the ultrasonic sound velocity in specimen (mm/usec)\n");
scanf("%f",&u_vel);

printf("\n Enter the number of data points in each A-scan & cal points
(cal must be power of 2)\n");
scanf("%d%d",&ndat1,&ndat);

printf("\n Enter the total number of A-scans\n");
scanf("%d",&nscan);

printf("\n Choose the method to compute synthetic aperture: \n  1) user chosen
aperture or \n  2) aperture computed by the method of fixed angle\n");
scanf("%d",&a_flag);

if (a_flag == 1) {
    printf("\n Enter the synthetic aperture size (number of A-scans) \n");
    scanf("%d",&sa_size);
}
else {
    printf("\n Enter the critical angle of the beam (degrees)\n");
    scanf("%f",&ls_angle);
}

printf("\n Enter the skip separation of the grid pixels (in the x & z directions) \n");
scanf("%d%d",&xs,&zs);

```

```

/* ### 2) Open file to input raw A-scans data ### */

if (a_flag == 1) { /* 0 padding for extension of edge data */
    add_no = (int) (sa_size-1)/2;
}
else {
    add_no = (int) (u_vel*dt*ndat*tan(ls_angle/180*PI)/2/dx);
}

dummy = matrix(1,nscan,1,ndat);

printf("\n Reading the raw data \n");
fp = fopen(file_name,"r"); /* inputting raw data */

for(h=1;h<=nscan-1;h++) { /* h: xducer posi. i: data points per A-scan */
    for(i=1;i<=ndat;i++) {

/* Execute the following statement if using the longitudinal displ. */

        fscanf(fp,"%f%f",&dummy[h][i],&dummy1);

/* Execute the following statement if using the lateral displacements */

        /* fscanf(fp,"%f%f",&dummy1,&dummy[h][i]); */
    }
}
fclose(fp);
printf("\n Finished reading the raw data \n");

a_dat = matrix(1,nscan+2*add_no,1,ndat);

for(h=1;h<=nscan+2*add_no;h++) { /* 0 padding edge of data */
    for(i=1;i<=ndat;i++) {
        a_dat[h][i] = 0;
    }
}

for(h=1;h<=nscan/xs;h++) {
    for(i=1;i<=ndat/zs;i++) {
        a_dat[h+add_no][i] = dummy[h*xs][i*xs];
    }
}

/* ### Convert Raw Data to Image Processor Format ### */

ipdat = imatrix(1,nscan,1,ndat);

```



```

imax = a_dat[add_no+1][200] , imin = a_dat[add_no+1][200];

for(h=add_no+1;h<=add_no+nscan;h++) {
    for(i=200;i<=ndat1;i++) {
        if(imax<a_dat[h][i]) imax = a_dat[h][i];
        if(imin>a_dat[h][i]) imin = a_dat[h][i];
    }
}

scale = 255./(imax-imin);

printf("\n max = %e  min = %e  scale = %e\n",imax,imin,scale);

for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        ipdat[h][i] = (int)(scale*(a_dat[h+add_no][i] - imin));
        if(ipdat[h][i]>255) ipdat[h][i] = 255;
    }
}

fp = fopen(file_name1,"w"); /* output raw data of happi format */
fprintf(fp,"%d\n%d\n",ndat1/2,nscan/xs);

for(i=1;i<=ndat1/2;i++) {
    for(h=1;h<=nscan/xs;h++) {
        fprintf(fp,"%d\n",ipdat[h][i*2]);
    }
}
fclose(fp);

printf("\n Would you like to continue this process? (yes = 1)\n");
scanf("%d",&q_flag);
if (q_flag != 1) exit(0);

/*
for(h=add_no;h<=nscan+add_no-1;h++) {
    printf("scanning no = %d\n",h);
for(i=1;i<=ndat;i++) {
    printf("%d  %f\n",i,a_dat[h][i]);
}
}
*/

/* ### 3) directivity correction ### */

printf("\n Would you like to account for the directivity of the
transducer? (yes=1 no=0)\n");
scanf("%d",&q_flag);

```

```

ndir = 90 ;          /* dir_angle from 0 to 90 deg a deg pitch */
dir_dat = vector(0,ndir);

dir_cal(dir_dat,q_flag,0,ndir);

/* ### 4) delay and sum ### */

printf("\n SAFT in progress \n");

c_dat = matrix(1,nscan,1,ndat);
for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        c_dat[h][i] =0;
    }
}

if (a_flag == 1) { /* fixed number aperture */
for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        c_dat[h][i] = a_dat[h+add_no][i];
        for(j=1;j<=(sa_size-1)/2;j++) {
            pass = SQR(i)+SQR(2*dx*j/u_vel/dt);
            t_delay = (int) sqrt(pass);
            dir_angle = (int)fabs((acos((double)(i)/t_delay)*180/PI));
            c_dat[h][i] = c_dat[h][i]+( a_dat[h-j+add_no][t_delay]
                + a_dat[h+j+add_no][t_delay] )/dir_dat[dir_angle];
        }
    }
}
}
else { /* fixed angle aperture */
for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        c_dat[h][i] = a_dat[h+add_no][i];
        limit = u_vel*i*dt*tan(ls_angle/180.0*PI)/2/dx;
        for(j=1;j<=limit;j++) {
            pass = SQR(i)+SQR(2*dx*j/u_vel/dt);
            t_delay = (int) sqrt(pass);
            dir_angle = (int)fabs((acos((double)(i)/t_delay)*180/PI));

            c_dat[h][i] = c_dat[h][i]
                +( a_dat[h-j+add_no][t_delay]
                + a_dat[h+j+add_no][t_delay] )
                +
                /dir_dat[dir_angle];
        }
    }
}
}
}

```

```

}
}

printf("\n SAFT has successfully ended\n");

/* ### Convert data to Image Processor Format ### */

imax = c_dat[1][200] , imin = c_dat[1][200];

for(h=1;h<=nscan;h++) {
    for(i=200;i<=ndat1;i++) {
        if(imax<c_dat[h][i]) imax = c_dat[h][i];
        if(imin>c_dat[h][i]) imin = c_dat[h][i];
    }
}

scale = 255./(imax-imin);

for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        ipdat[h][i] = (int)(scale*(c_dat[h][i] - imin));
        if(ipdat[h][i]>255) ipdat[h][i] = 255;
    }
}

/* ### 5) output correlated A-scans ### */

printf("\n Writing data out to files \n");

fp = fopen(file_name2,"w");
fprintf(fp,"%d\n%d\n",ndat1/2,nscan/xs);
for(i=1;i<=ndat1/2;i++) {
    for(h=1;h<=nscan/xs;h++) {
        fprintf(fp,"%d\n",ipdat[h][i*2]);
    }
}
fclose(fp);

printf("\n Computing the envelope of the SAFT image \n");

envlp(c_dat,nscan,ndat);

/* ### output envelop detected data ### */

imax = c_dat[1][200] , imin = c_dat[1][200];

```

```

for(h=1;h<=nscan;h++) {
    for(i=200;i<=ndat1-200;i++) {
        if(imax<c_dat[h][i]) imax = c_dat[h][i];
        if(imin>c_dat[h][i]) imin = c_dat[h][i];
    }
}

scale = 255./(imax-imin);

for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {
        ipdat[h][i] = (int)(scale*(c_dat[h][i] - imin));
        if(ipdat[h][i]>255) ipdat[h][i] = 255;
    }
}

fp = fopen(file_name3,"w");
fprintf(fp,"%d\n%d\n",ndat1/2,nscan/xs);
for(i=1;i<=ndat1/2;i++) {
    for(h=1;h<=nscan/xs;h++) {
        fprintf(fp,"%d\n",ipdat[h][i*2]);
    }
}
fclose(fp);

/* ### memory free ### */

free_vector(dir_dat,0,ndir);
free_matrix(a_dat,1,nscan+2*add_no,1,ndat);
free_matrix(c_dat,1,nscan,1,ndat);
free_imatrix(ipdat,1,nscan,1,ndat);
}
envlp(dat,nscan,ndat)
float **dat;
int ndat,nscan;
{

int fft_flag,i,h;
float *fft_dat;
float *vector();
void four1(),free_vector();

fft_dat = vector(1,2*ndat);

for(h=1;h<=nscan;h++) {
    for(i=1;i<=ndat;i++) {

```

```
        fft_dat[2*i-1] = dat[h][i];
        fft_dat[2*i] = 0;
    }

    fft_flag = 1;

    four1(fft_dat,ndat,fft_flag);

    for(i=ndat+1;i<=2*ndat;i++) {
        fft_dat[i] = 0;
    }

    fft_dat[1] = 0 , fft_dat[2] = 0;

    fft_flag = -1;

    four1(fft_dat,ndat,fft_flag);

    for(i=1;i<=ndat;i++) {
        dat[h][i] = sqrt(fft_dat[2*i-1]*fft_dat[2*i-1] +
            fft_dat[2*i]*fft_dat[2*i]);
    }
}

free_vector(fft_dat,1,2*ndat);
}
```

```

C*****
C**
C**          BEAMFORMING IN NDE          **
C**
C** The main objective of this program is to use a beamforming **
C** technique to estimate an ultrasonic signal (or signals) **
C** arriving from a desired direction during a nondestructive **
C** evaluation of isotropic and anisotropic materials. **
C** A data independent Beamformer is adopted for this study. **
C**
C** Written on: Tuesday 27 November, 1990 **
C** Updated: Friday 13 December, 1991 **
C**
C*****

```

```

parameter (n1rec=38,n2rec=600,nnpts=700,llpts=1024,kpts=2048)
parameter (nn=38*38,ll=1024*1024,nl=38*1024,kp=2*2048)
character*15 femname,femname1,filename,expdata
integer vc,vs
real di,focl,xpeak,xxpeak,xmse,sum,amp

```

```

real map(n2rec,kpts),map1(n2rec,kpts)
real data(kpts),data1(kpts),rdata(kpts),indx(n1rec),ipvt(n1rec)
real ecco(llpts),ecco2(n1rec,nnpts)
real lon(nnpts),lat(nnpts)
real llon(n1rec,llpts),llat(n1rec,llpts)
real ecco3(n1rec,llpts),necco(n1rec,llpts),necco1(n1rec,nnpts)
real xr(nl),xi(nl),ur(nn),ui(nn),vr(nn),vi(nn)
real lon1(kpts),dat1(kp),dat2(kp)
real ecc2(n1rec,kpts),necc1(n1rec,kpts)

```

```

complex ecco1(llpts,n1rec),x(llpts,n1rec),y(llpts,n1rec)
complex ecc1(llpts,n1rec),x1(llpts,n1rec)
complex xt(n1rec,llpts),xx(llpts,n1rec),iden(n1rec,n1rec)
complex filter(n1rec,n1rec),out(llpts,n1rec),out1(llpts,n1rec)
complex work(n1rec),s(n1rec),e(n1rec),u(n1rec,n1rec),v(n1rec,n1rec)
complex ut(n1rec,n1rec),vt(n1rec,n1rec),ss(n1rec,n1rec)
complex subinv(n1rec,n1rec),xinv(n1rec,n1rec)
complex prod(n1rec,n1rec),prod1(n1rec,n1rec),prod2(n1rec,llpts)
complex image(llpts)

```

```

type*,' '
type*,'
type*,' '
type*,' '
type*,'Enter the velocity of sound (m/sec) in the coupling medium:'
read*,vc
type*,' '

```

WELCOME TO BEAMFORMING '

```

type*, 'Enter the velocity of sound (m/sec) in the specimen:'
read*, vs
type*, ' '
type*, 'Enter the number of points per A-scan record:'
read*, npts
type*, ' '
type*, 'Enter the number of collected A-scan records:'
read*, nrec
type*, ' '

```

C-----

```

if(npts.le.32) lpts = 32
if(npts.gt.32 .and. npts.le.64) lpts = 64
if(npts.gt.64 .and. npts.le.128) lpts = 128
if(npts.gt.128 .and. npts.le.256) lpts = 256
if(npts.gt.256 .and. npts.le.512) lpts = 512
if(npts.gt.512 .and. npts.le.1024) lpts = 1024
if(npts.gt.1024 .and. npts.le.2048) lpts = 2048

```

```

type*, ' '
type*, '          lpts =', lpts
type*, ' '

```

C-----

```

type*, ' '
type*, 'Would you like to work with the:'
type*, ' '
type*, ' 1. U-Displacements (Longitudinal). '
type*, ' 2. V-Displacements (Lateral). '
type*, ' 3. The magnitude of the above two.'
type*, ' '
read*, iwork
type*, ' '
type*, ' '
type*, 'Enter the name of the Finite Element data file'
type*, '(regular or reversed geometry)'
type*, ' '
read '(a)', femname

```

C-----

C-----READ THE PULSE-ECHO RECORDS-----

C-----

```

type*, ' '
type*, '          READING THE PULSE-ECHOS .... PATIENCE !!!'
type*, ' '

```

```

C-----
  open(1,file=femname)
  do 115 i = 1 , nrec
    do 114 j = 1 , npts
      read(1,*) lon(j), lat(j)
      if(iwork.eq.1) ecco2(i,j) = lon(j)
      if(iwork.eq.2) ecco2(i,j) = lat(j)
      if(iwork.eq.3) ecco2(i,j) = sqrt(lon(j)*lon(j) + lat(j)*lat(j))
      if(iwork.eq.1) ecco3(i,j) = lon(j)
      if(iwork.eq.2) ecco3(i,j) = lat(j)
      if(iwork.eq.3) ecco3(i,j) = sqrt(lon(j)*lon(j) + lat(j)*lat(j))
114  continue
      do 213 jj = npts+1 , lpts
        lon(jj) = 0.0
213  continue
        do 212 jjj = 1 , lpts
          data(2*jjj-1) = lon(jjj)
          data(2*jjj) = 0.0
212  continue
          call four1(data,lpts,1)
C-----
      if (i.eq.16) goto 650
      goto 652
650  open(13,file='ascan_four')
      do 651 j = 1 , lpts

```



```

rdata(2*j-1) = data(2*j-1)
rdata(2*j) = data(2*j)

write(13,*) j,sqrt((rdata(2*j-1))**2 + (rdata(2*j))**2)
651 continue

call four1(rdata,lpts,-1)           ! Inverse FFT of record 16.

open(14,file='ascan_inv')
do 751 j = 1 , npts
751 write(14,*) j , rdata(2*j-1)/lpts

open(15,file='ascan_raw')
do 752 j = 1 , npts
752 write(15,*) j,lon(j)

C-----

652 do 214 ji = 1 , lpts

      ecco1(ji,i) = cmplx(data(2*ji-1),data(2*ji))

214 continue

115 continue

      close(1)

C-----

      open(17,file='ecco1.d')

do 177 j = 1 , nrec
177 write(17,*) (real(ecco1(j,i)),i=1,nrec)

      close(17)

C-----
C-----
C-----

type*,' '
type*,' Which geometry would you like to use in order'
type*,' to test the robustness of the interpolating'
type*,' filter created by the beamformer using the '
type*,' finite element data:'
type*,' '

```

```

type*, '          1. Experimental data (Plexi or Gr/Ep)'
type*, '          2. Reversed FE geometry'
type*, ' '
read*, igeom

if(igeom.eq.1) goto 210
goto 211

210  type*, ' '
      type*, 'Enter the name of the sectioned experimental data file'
      type*, ' '
      read '(a)',filename

      open(2,file=filename)

      mpts = 2048

      goto 290

211  type*, ' '
      type*, 'Enter the name of the reversed FEM geometry data file'
      type*, ' '
      read '(a)',femname1

      open(2,file=femname1)      ! This file contains the same
                                ! FEM data but in reversed record
                                ! order (38,37,...,1)

      mpts = lpts

      goto 291

C-----
290  type*, ' '
      type*, '          READING THE EXPERIMENTAL DATA....'
      type*, ' '

      read(2,*) nr1,nr2

      type*, ' '
      write(*,*) 'This set contains records',nr1,' to',nr2
      type*, ' '

      do 990 i = 1 , nrec
      do 991 j = 1 , mpts

          read(2,*) lon1(j)

```

```

        ecc2(i,j) = lon1(j)
991  continue
        do 992 j = 1 , mpts
            dat1(2*j-1) = lon1(j)
            dat1(2*j) = 0.0
992  continue
        call four1(dat1,mpts,1)
        do 993 j = 1 , mpts
            ecc1(j,i) = cmplx(dat1(2*j-1),dat1(2*j))
993  continue
990  continue
        goto 292

```

C-----

```

291  type*, ' '
     type*, ' '
     type*, ' '
                                READING THE REVERSED FEM DATA...!
        do 790 i = 1 , nrec
        do 791 j = 1 , npts
            read(2,*) lon1(j),dummy
            ecc2(i,j) = lon1(j)
791  continue
        do 792 j = npts+1 , lpts
            lon1(j) = 0.0
792  continue
        do 793 j = 1 , lpts
            dat1(2*j-1) = lon1(j)
            dat1(2*j) = 0.0

```

```

793  continue
      call four1(dat1,lpts,1)
      do 794 j = 1 , lpts
          ecc1(j,i) = cmplx(dat1(2*j-1),dat1(2*j))
794  continue
790  continue
C-----

292  close(2)
C-----
C-----

      type*,' '
      type*,' '           DONE READING THE PULSE-ECHOS....'
      type*,' '
C-----

      type*,' '
      type*,'Are you willing to perform BEAMFORMING? (1--yes)'
      type*,' '
      read*,ibeam

      if(ibeam.eq.1) goto 67
      goto 68

C-----
C-----BEAMFORMING PORTION OF PROGRAM-----
C-----

67   type*,' '
      type*,' '
      type*,' '           BEAMFORMING IS IN PROGRESS....PATIENCE!!!'
      type*,' '
C-----

      do 225 i = 1 , nrec           ! Initializing input and output.
      do 224 j = 1 , lpts

          x(j,i) = 0.0
          y(j,i) = 0.0

```

```

224  continue
225  continue

      do 994 i = 1 , nrec
      do 995 j = 1 , mpts

          x1(j,i) = 0.0

995  continue
994  continue

      type*,' '
      type*,' What percentage of the whole record array'
      type*,' would you like to keep:'
      type*,' '
      type*,'          0. 100% (Keep the full record)'
      type*,'          1. 50% (Keep the odd records)'
      type*,'          2. 50% (Keep the even records)'
      type*,'          3. 26% (Approximately)'
      type*,'          4. 13% (Approximately)'
      type*,'          5.  3% (Approximately)'
      type*,' '
      type*,' in the X matrix.'
      type*,' '
      read*,iperc

      if (iperc.eq.0) goto 814
      if (iperc.eq.1) goto 815
      if (iperc.eq.2) goto 816
      if (iperc.eq.3) goto 817
      if (iperc.eq.4) goto 830
      if (iperc.eq.5) goto 818

```

C.....

```

814  do 624 i = 1 , nrec
      do 625 j = 1 , lpts

          x(j,i) = ecco1(j,i)

625  continue
624  continue

      do 996 i = 1 , nrec
      do 997 j = 1 , mpts

          x1(j,i) = eccl(j,i)

```

997 continue
 996 continue

goto 819

C.....

815 do 227 i = 1 , nrec/2
 do 226 j = 1 , lpts

x(j,2*i-1) = ecco1(j,2*i-1) ! Odd Records.

226 continue
 227 continue

do 998 i = 1 , nrec/2
 do 999 j = 1 , mpts

x1(j,2*i-1) = ecc1(j,2*i-1)

999 continue
 998 continue

goto 819

C.....

816 do 909 i = 1 , nrec/2
 do 910 j = 1 , lpts

x(j,2*i) = ecco1(j,2*i) ! Even Records.

910 continue
 909 continue

do 890 i = 1 , nrec/2
 do 891 j = 1 , mpts

x1(j,2*i) = ecc1(j,2*i)

891 continue
 890 continue

goto 819

C.....

817 type*, ' '

```

type*, '26% of the 38 records means you can'
type*, 'only keep 10 records.'
type*, '
type*, ' Please enter the 10 records you would like to keep:'
type*, '(All on one line)'
type*, '
read*, ir1, ir2, ir3, ir4, ir5, ir6, ir7, ir8, ir8, ir10

```

```

do 912 j = 1 , lpts

```

```

  x(j, ir1) = eccol(j, ir1)
  x(j, ir2) = eccol(j, ir2)
  x(j, ir3) = eccol(j, ir3)
  x(j, ir4) = eccol(j, ir4)
  x(j, ir5) = eccol(j, ir5)
  x(j, ir6) = eccol(j, ir6)
  x(j, ir7) = eccol(j, ir7)
  x(j, ir8) = eccol(j, ir8)
  x(j, ir9) = eccol(j, ir9)
  x(j, ir10) = eccol(j, ir10)

```

```

912  continue

```

```

do 892 j = 1 , mpts

```

```

  x1(j, ir1) = ecc1(j, ir1)
  x1(j, ir2) = ecc1(j, ir2)
  x1(j, ir3) = ecc1(j, ir3)
  x1(j, ir4) = ecc1(j, ir4)
  x1(j, ir5) = ecc1(j, ir5)
  x1(j, ir6) = ecc1(j, ir6)
  x1(j, ir7) = ecc1(j, ir7)
  x1(j, ir8) = ecc1(j, ir8)
  x1(j, ir9) = ecc1(j, ir9)
  x1(j, ir10) = ecc1(j, ir10)

```

```

892  continue

```

```

      goto 819

```

```

C.....

```

```

830  type*, '
type*, '13% of the 38 records means you can '
type*, 'only keep 5 records.'
type*, '
type*, ' Please enter the 5 records you would like to keep:'
type*, '(All on one line)'

```

```

type*,' '
read*,ir1,ir2,ir3,ir4,ir5

do 832 j = 1 , lpts

    x(j,ir1) = eccol(j,ir1)
    x(j,ir2) = eccol(j,ir2)
    x(j,ir3) = eccol(j,ir3)
    x(j,ir4) = eccol(j,ir4)
    x(j,ir5) = eccol(j,ir5)

832  continue

do 893 j = 1 , mpts

    x1(j,ir1) = ecc1(j,ir1)
    x1(j,ir2) = ecc1(j,ir2)
    x1(j,ir3) = ecc1(j,ir3)
    x1(j,ir4) = ecc1(j,ir4)
    x1(j,ir5) = ecc1(j,ir5)

893  continue

goto 819

C.....

818  type*,' '
      type*,' 3% of the 38 records means you can '
      type*,' only keep 1 record.'
      type*,' '
      type*,' Please enter the one record you would like to keep:'
      type*,' '
      read*,ir

do 913 j = 1 , lpts

    x(j,ir) = eccol(j,ir)

913  continue

do 894 j = 1 , mpts

    x1(j,ir) = ecc1(j,ir)

894  continue

C-----

```



```
819 do 750 i = 1 , nrec
    do 760 j = 1 , lpts
```

```
    xx(j,i) = x(j,i)
```

```
760 continue
```

```
750 continue
```

```
C-----
C-----
```

```
type*,' '
type*,' Would you like to keep:'
type*,' '
type*,'          0. The full material geometry (i.e., all A-scans)'
type*,'          1. 50% of the A-scans (Keep the odd records)'
type*,'          2. 50% of the A-scans (Keep the even records)'
type*,'          3. 26% of the A-scans (Approximately)'
type*,'          4. 13% of the A-scans (Approximately)'
type*,'          5. 3% of the A-scans (Approximately)'
type*,' '
type*,' for the Y matrix.'
type*,' '
type*,' Suggestion: Keep the full geometry (i.e., choose 0)'
type*,' '
read*,jkeep

if(jkeep.eq.0) goto 820
if(jkeep.eq.1) goto 821
if(jkeep.eq.2) goto 822
if(jkeep.eq.3) goto 823
if(jkeep.eq.4) goto 884
if(jkeep.eq.5) goto 885
```

```
C.....
```

```
820 do 414 i = 1 , nrec
    do 413 j = 1 , lpts
```

```
    y(j,i) = eccol(j,i)
```

```
413 continue
```

```
414 continue
```

```
goto 824
```

```
C.....
```

```

821  do 914 i = 1 , nrec/2
      do 915 j = 1 , lpts
          y(j,2*i-1) = ecco1(j,2*i-1)      ! Odd records.
915  continue
914  continue

      goto 824

C.....

822  do 916 i = 1 , nrec/2
      do 917 j = 1 , lpts
          y(j,2*i) = ecco1(j,2*i)      ! Even Records.
917  continue
916  continue

      goto 824

C.....

823  type*,' '
      type*,' 26% of the 38 records means you can'
      type*,' only keep 10 records.'
      type*,' '
      type*,' Please enter the 10 records you would like to keep:'
      type*,' (All on one line)'
      type*,' '
      read*,ir1,ir2,ir3,ir4,ir5,ir6,ir7,ir8,ir8,ir10

      do 919 j = 1 , lpts
          y(j,ir1) = ecco1(j,ir1)
          y(j,ir2) = ecco1(j,ir2)
          y(j,ir3) = ecco1(j,ir3)
          y(j,ir4) = ecco1(j,ir4)
          y(j,ir5) = ecco1(j,ir5)
          y(j,ir6) = ecco1(j,ir6)
          y(j,ir7) = ecco1(j,ir7)
          y(j,ir8) = ecco1(j,ir8)
          y(j,ir9) = ecco1(j,ir9)
          y(j,ir10) = ecco1(j,ir10)

919  continue

```

goto 824

C.....

```
884  type*,' '
      type*,' 13% of the 38 records means you can '
      type*,' only keep 5 records.'
      type*,' '
      type*,' Please enter the 5 records you would like to keep:'
      type*,' (All on one line)'
      type*,' '
      read*,ir1,ir2,ir3,ir4,ir5
```

do 882 j = 1 , lpts

```
      y(j,ir1) = ecco1(j,ir1)
      y(j,ir2) = ecco1(j,ir2)
      y(j,ir3) = ecco1(j,ir3)
      y(j,ir4) = ecco1(j,ir4)
      y(j,ir5) = ecco1(j,ir5)
```

882 continue

goto 824

C.....

```
885  type*,' '
      type*,' 3% of the 38 records means you can'
      type*,' only keep 1 record.'
      type*,' '
      type*,' Please enter the one record you would like to keep:'
      type*,' '
      read*,ir
```

do 886 j = 1 , lpts

```
      y(j,ir) = ecco1(j,ir)
```

886 continue

C-----

```
824  open(70,file='x.d')
      open(80,file='y.d')
      open(81,file='x1.d')
```

do 777 j = 1 , nrec

```

777 write(70,*) (real(x(j,i)),i=1,nrec)
    close(70)

    do 888 j = 1 , nrec
888 write(80,*) (real(y(j,i)),i=1,nrec)
    close(80)

    do 828 j = 1 , lpts
828 write(81,*) (real(x1(j,i)),i=1,nrec)
    close(81)

```

C-----

```

    k = 0

    do 228 i = 1 , nrec          ! X is Hermitian.
    do 229 j = 1 , lpts

        k = k + 1

        xr(k) = real(x(j,i))
        xi(k) = -aimag(x(j,i))

229 continue
228 continue

```

C-----

```

    type*, ' '
    type*, '.....TRANSPOSING X.....'

    call matrans(lpts,nrec,xr)
    call matrans(lpts,nrec,xi)

    type*, ' '
    type*, '.....FINISHED TRANSPOSING X.....'

```

C-----

```

    k = 0

    do 231 j = 1 , lpts
    do 232 i = 1 , nrec

        k = k + 1

        xt(i,j) = cmplx(xr(k),xi(k))

```

232 continue
231 continue

C-----

```

open(90,file='xt.d')
do 799 i = 1 , nrec
799 write(90,*) (real(xt(i,j)),j=1,nrec)
close(90)

```

C-----

```

type*, ' '
type*, ' Multiplying ..... xt * x'
type*, ' '

call brainy(nrec,lpts,1,xt,lpts,nrec,1,x,prod)

do 567 i = 1 , nrec
do 568 j = 1 , nrec

    prod1(j,i) = prod(j,i)

568 continue
567 continue

```

C-----

```

type*, ' '
type*, '          Determining the Pseudo-Inverse '
type*, ' '
type*, '          x+ = inv(xt * x)'
type*, ' '
type*, '          The Pseudo-inverse can be written as:'
type*, ' '
type*, '          x+ = V * diag(1/s) * Ut'
type*, ' '
type*, '          Calling the Singular Value Decomposition program'
type*, ' '
job = 11          ! Subroutine CSVDC is taken from LINPACK.

call csvdc(prod,nrec,nrec,nrec,s,e,u,nrec,v,nrec,work,job,info)

```

C-----

```

type*, ' '

```

```

type*,'          job = 'job
type*,'          info = 'info
type*,' '
type*,' '

```

C-----

```

open(40,file='svd_u.d')
do 401 j = 1 , nrec
401 write(40,*) (real(u(j,i)),i=1,nrec)
close(40)

```

```

open(21,file='svd_v.d')
do 402 j = 1 , nrec
402 write(21,*) (real(v(j,i)),i=1,nrec)
close(21)

```

```

open(22,file='svd_s.d')
write(22,*) (real(s(i)),i=1,nrec)
close(22)

```

```

open(71,file='svd_e.d')
write(71,*) (real(e(i)),i=1,nrec)
close(71)

```

C-----

```

k = 0

```

```

do 741 i = 1 , nrec          ! U is also Hermitian.
do 742 j = 1 , nrec

```

```

k = k + 1

```

```

ur(k) = real(u(j,i))
ui(k) = -aimag(u(j,i))

```

```

742 continue

```

```

741 continue

```

C-----

```

type*,' '
type*,' Transposing the Right Singular matrix V'
type*,' '

```

```

call matrans(nrec,nrec,ur)
call matrans(nrec,nrec,ui)

```



```

type*, '.....INVERSE FFT.....'
type*, '
do 235 i = 1 , nrec      ! Regular FEM Geometry.
  do 233 j = 1 , lpts
    data1(2*j-1) = real(out(j,i))
    data1(2*j) = aimag(out(j,i))
233  continue
    call four1(data1,lpts,-1)
    do 234 jj = 1 , npts
      necco(i,jj) = data1(2*jj-1) / lpts
234  continue
235  continue
C-----
    if(igeom.eq.1) goto 690
    goto 691
C-----
690  do 895 i = 1 , nrec      ! Experimental Geometry.
    do 896 j = 1 , mpts
      dat2(2*j-1) = real(out1(j,i))
      dat2(2*j) = aimag(out1(j,i))
896  continue
      call four1(dat2,mpts,-1)
      do 897 j = 1 , mpts
        necc1(i,j) = dat2(2*j-1) / mpts
897  continue
895  continue

```

```

        goto 692
C-----
691  do 490 i = 1 , nrec          ! Reversed FEM Geometry.
        do 491 j = 1 , mpts
            dat2(2*j-1) = real(out1(j,i))
            dat2(2*j) = aimag(out1(j,i))
491  continue
        call four1(dat2,mpts,-1)
        do 492 jj = 1 , npts
            necc1(i,jj) = dat2(2*jj-1) / lpts
492  continue
490  continue
C-----
692  type*, ' '
        type*, '.....INVERSE FFT DONE.....'
        type*, ' '
        type*, ' '
        type*, '.....WRITING DATA OUT TO FILES.....'
C-----
        open(11,file='ascan1')          ! Regular FEM Geometry.
        do 447 i = 1 , nrec
        do 446 j = 1 , npts
            write(11,*) ecco2(i,j),necco(i,j)
446  continue
447  continue
        close(11)
        suma = 0.0
        sumb = 0.0
        sum1 = 0.0

```

```

do 601 i = 1 , nrec
do 602 j = 200 , 600          ! MSE is computed for only the points
                              ! between 200 and 600 per record.

    a = ecco2(i,j)
    b = necco(i,j)

    aabs = abs(a)
    aa = aabs * aabs

    babs = abs(b)
    bb = babs * babs

    suma = suma + aa
    sumb = sumb + bb

    sum1 = sum1 + (a - b) * (a - b)

602 continue
601 continue

xmse1 = sum1 / (nrec*npts)

ymse1 = 100. * (sum1 / suma)

type*,' '
type*,' '
type*,'           Actual Geometry'
type*,' '
type*,'           POWER OF RAW IMAGE = ',suma
type*,'           POWER OF NEW IMAGE = ',sumb
type*,' '
type*,'           MEAN-SQUARE ERROR = ',xmse1
type*,'           RELATIVE MSE = ',ymse1,'%'
```

C-----

```

    if(igeom.eq.1) goto 693
    goto 694

C-----
```

```

693 type*,' '
type*,' Enter the name of the file for the reconstructed'
type*,' EXPERIMENTAL data'
type*,' '
read '(a)',expdata
```

```

open(12,file=expdata)           ! Experimental Geometry.

do 695 i = 1 , nrec
do 696 j = 1 , mpts

    write(12,*) ecc2(i,j),necc1(i,j)
696 continue
695 continue

close(12)

sumb = 0.0
sum2 = 0.0

do 603 i = 1 , nrec
do 604 j = 1 , mpts

    c = ecc2(i,j)
    d = necc1(i,j)

    absc = abs(c)
    csq = absc * absc

    sumb = sumb + csq

    sum2 = sum2 + (c - d) * (c - d)

604 continue
603 continue

xmse2 = 100 * (sum2 / sumb)

type*, ' '
type*, ' '
type*, ' RELATIVE MEAN-SQUARE ERROR (experimental) = ',xmse2,'% '

goto 494

```

C-----

```

694 open(12,file='ascan2')       ! Reversed FEM Geometry.

do 697 i = 1 , nrec
do 698 j = 1 , npts

    write(12,*) ecc2(i,j),necc1(i,j)

```

```

698 continue
697 continue

close(12)

sumc = 0.0
sumd = 0.0
sum2 = 0.0

do 605 i = 1 , nrec
do 606 j = 200 , 600

    c = ecc2(i,j)
    d = necc1(i,j)

    xcabs = abs(c)
    cc = xcabs * xcabs

    dabs = abs(d)
    dd = dabs * dabs

    sumc = sumc + cc
    sumd = sumd + dd

    sum2 = sum2 + (c - d) * (c - d)

606 continue
605 continue

xmse2 = sum2 / (nrec * npts)
ymse2 = 100. * (sum2 / sumc)

type*,' '
type*,' '
type*,' '           Reversed Geometry'
type*,' '
type*,' '           POWER OR RAW IMAGE =' ,sumc
type*,' '           POWER OF NEW IAMGE =' ,sumd
type*,' '
type*,' '           MEAN-SQUARE ERROR =' ,xmse2
type*,' '           RELATIVE MSE =' ,ymse2,'%'
type*,' '

C-----

494 type*,' '
type*,' .....FINISHED WRITING DATA.....'

```


APPENDIX C. OTHER PROGRAMS AND SUBROUTINES


```
subroutine four1(data,nn,isign)
```

```

c*****
c
c This subroutine replaces 'data' by its Discrete Fourier
c Transform, if 'isign' is 1; or replaces 'data' by 'nn' times
c its inverse DFT, if 'isign' is -1. 'data' is a complex array
c of length 'nn' or, equivalently, a real array of length 2*nn.
c 'nn' MUST be an integer power of two.
c
c From: Numerical recipes in FORTRAN.
c*****

real*8 wr,wi,wpr,wpi,wtemp,theta ! double precision for the trigo-
                                ! nometric recurrences.
dimension data(2*nn)

n = 2 * nn
j = 1

do 11 i = 1 , n , 2           ! Perform bit-reversal.

    if(j.gt.1) then

        tempr = data(j)           ! Exchange the two complex numbers.
        tempi = data(j+1)
        data(j) = data(i)
        data(j+1) = data(i+1)
        data(i) = tempr
        data(i+1) = tempi

    end if
    m = n / 2

1   if((m.ge.2) .and. (j.gt.m)) then

        j = j - m
        m = m / 2

        goto 1

    end if

    j = j + m

11  continue

mmax = 2                       ! Here begins the Danielson-

```

```

2      if(n.gt.mmax) then
                                ! Lanczos section of the routine.
                                ! Outer loop executed log(base2)nn times.

        istep = 2 * mmax
        theta = 6.28318530717959D0 / (isign*mmax)      ! initializing for the
        wpr = -2.D0 * dsin(0.5D0 * theta) ** 2        ! trig recurrence.
        wpi = dsin(theta)
        wr = 1.D0
        wi = 0.D0

        do 13 m = 1 , mmax , 2
            do 12 i = m , n , istep

                j = i + mmax

                ! This is the Danielson-Lanczos formula:

                tempr = snl(wr) * data(j) - snl(wi) * data(j+1)
                tempi = snl(wr) * data(j+1) + snl(wi) * data(j)
                data(j) = data(i) - tempr
                data(j+1) = data(i+1) - tempi
                data(i) = data(i) + tempr
                data(i+1) = data(i+1) + tempi

12      continue

        wtemp = wr
        wr = wr * wpr - wi * wpi + wr
        wi = wi * wpr + wtemp * wpi + wi
                                ! trigonometric recurrence.

13     continue

        mmax = istep

        goto 2

    end if

    return
end

```

```

C*****
C*****

```

```

function dot(l,j,x,y)
real x(2048),y(2048)

```

```

dot = 0.0
if(1.le.0) return
do 1 ij = j , l+j-1
1 dot = dot + x(ij) * y(ij)
return
end

```

```

C*****
C*****

```

```

subroutine zero(lx,x)
complex x(lx)
if(lx.le.0) return
do 1 i = 1 , lx
1 x(i) = 0.0
return
end

```

```

C*****
C*****

```

```

subroutine matrans(m,n,matrix)

```

```

C-----
c-- This subroutine transposes a rectangular matrix. --
c--                                     --
c-- m = number of rows in matrix.      --
c-- n = number of columns in matrix.   --
C-----

```

```

dimension matrix(1)
k = m * n - 1
do 1 i = 2 , k
1 matrix(i) = (matrix(i) / 2 ) * 2
do 3 l = 2 , k

```

```

      if(matrix(l) .ne. (matrix(l)/2)*2) goto 3
      keep = matrix(l)
      ij = 1
2     jless1 = (ij - 1) / m
      i = ij - jless1 * m
      j = jless1 + 1
      ji = j + (i - 1) * n
      katch = matrix(ji)
      matrix(ji) = keep + 1
      keep = katch
      ij = ji
      if(ij.ne.1) goto 2
3     continue
      return
      end

```

```

C*****
C*****

```

subroutine matrinv(n,a,ainv,det,adjug,p)

```

C-----
c-- This subroutine inverts a (not necessarily symmetric) --
c-- n x n matrix, a, by a method given by Faddeev and --
c-- Sominskii. The method is described in Gantmacher. In --
c-- addition to the inverting the matrix, the method also --
c-- yields the determinant, the adjugate, and the coeffi- --
c-- cients of the characteristic polynomial, det(LI-a). --
c-- --
c-- n = order of matrix. --
c-- a = n x n matrix. --
c-- ainv = inverse of a. --
c-- det = determinant of a. --
c-- adjug = adjugate of a. --
c-- p = coefficients of characteristic polynomial. --
C-----

```

```

complex a(n,n),ainv(n,n),det,adjug(n,n),p(n)
call move(n,a,ainv)
do 4 k = 1 , n
  p(k) = 0.0
  do 2 i = 1 , n
2    p(k) = p(k) + ainv(i,i)
    p(k) = p(k) / float(k)
    if(k.eq.n) goto 5
  call move(n,ainv,adjug)
  do 3 i = 1 , n
3    adjug(i,i) = ainv(i,i) - p(k)
4  call brainy(n,n,1,a,n,n,1,adjug,ainv)
5  call move(n,adjug,ainv)
  if(cabs(p(n)) .lt. 1.0e-30) goto 7
  do 6 i = 1 , n
  do 6 j = 1 , n
6  ainv(i,j) = ainv(i,j) / p(n)
7  det = p(n)
  if(mod(n,2) .eq. 1) return
  det = -det
  do 8 i = 1 , n
  do 8 j = 1 , n
8  adjug(i,j) = -adjug(i,j)
  return
end

```

C*****

C*****

```

subroutine inverse(n,a,b)

complex ek
complex a(n,n),b(n,n)

do 5 i = 1 , n
do 5 j = 1 , n
5   b(i,j) = 0.0

b(1,1) = 1. / a(1,1)

if(n.eq.1) return

do 40 m = 2 , n

k = m - 1

ek = a(m,m)

do 10 i = 1 , k
do 10 j = 1 , k
10   ek = ek - a(m,i) * b(i,j) * a(j,m)

b(m,m) = 1. / ek

do 30 i = 1 , k
do 20 j = 1 , k
20   b(i,m) = b(i,m) - b(i,j) * a(j,m) / ek
30   b(m,i) = b(i,m)

do 40 i = 1 , k
do 40 j = 1 , k
40   b(i,j) = b(i,j) + b(i,m) * b(m,j) * ek

return
end

```

C*****
C*****

```

subroutine move(lx,x,y)

```

```

C-----
C-- The purpose of this subroutine is to move an array --
C-- from one storage location to another. --
C-----

```

```

      complex x(lx,lx),y(lx,lx)

```

```

      do 1 i = 1 , lx
      do 1 j = 1 , lx

```

```

1      y(i,j) = x(i,j)

```

```

      return
      end

```

```

C*****
C*****

```

```

      subroutine brainy(nra,nca,la,a,nrb,ncb,lb,b,c)

```

```

C-----
C-- This subroutine performs matrix polynomial multipli- --
C-- cation. Because polynomial multiplication corresponds --
C-- to complete transient convolution, then this subroutine --
C-- performs the complete transient convolution of two --
C-- multichannel signals. --
C-- --
C-- This subroutine allows the use of rectangular matrices. --
C-- --
C-- nra = number of rows of matrix a = p. --
C-- nca = number of columns of matrix a = q. --
C-- la = length of multichannel signal a = m+1. --
C-- a = (a0,a1,...,am), where each coefficient ai, --
C-- is a p x q constant matrix. --
C-- --
C-- nrb = number of rows of matrix b = q. --
C-- ncb = number of columns of matrix b = r. --
C-- lb = length of multichannel signal b = n+1. --
C-- b = (b0,b1,...,bn), where each coefficient bi, --
C-- is a q x r constant matrix. --
C-- --
C-- c = (c0,c1,...,c(m+n)), is the output. --
C-----

```

```

      complex a(nra,nca,la),b(nca,ncb,lb),c(nra,ncb,1)

```

```

      lc = la + lb - 1

```

```
call zero(nra*ncb*lc,c)

do 1 i = 1 , la
do 1 j = 1 , lb

k = i + j - 1

do 1 m = 1 , nra
do 1 n = 1 , ncb
do 1 l = 1 , nca

1  c(m,n,k) = c(m,n,k) + a(m,l,i) * b(l,n,j)

return
end
```