Ultrasonic image reconstruction using the Wiener Filter method

by

Sleiman Riad Ghorayeb

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

Major: Biomedical Engineering

Signatures have been redacted for privacy

Signatures have been redacted for privacy

Iowa State University
Ames, Iowa
1988

TABLE OF CONTENTS

I. INTRODUCTION

Many developments in the fields of biomedical research and modern quantitative medicine involving the use of ultrasonic devices have taken place during the last decades. New techniques have been introduced based not only on new discoveries in the physical sciences, but on previous quantities thought to be inaccessible. Modern ultrasound equipment has played a vital role in the detection of echo reflections from targets in human and animal subjects; however, these signals are corrupted by the effects of the 'noise' produced primarily by the body under test and secondarily by other physical factors.

The purpose of any filter is to separate one thing from another. In the electric filter case (i.e., low-pass, high-pass, etc.), this usually refers to passing signals in a specified frequency range and rejecting those outside that range. Here, the filter is simply one of circuit design involving the appropriate choice of resistors, capacitors, and so on.

The problem we are dealing with is of a more fundamental nature. The purpose of this thesis is to suggest a method that would determine the optimal filter's response, in function of its length, that best separates the signal from the noise. The theory involved for that matter is referred to as the Wiener filter theory.

A. Brief Literature Review

In the early years of 1700s, Adrien Marie Legendre, a French mathematician, was the first to develop the method of least squares estimation. After Legendre died in 1783, Karl Friedrich Gauss, a German mathematician, carried that method on, and explored its usage in the course of calculating planetary and comet orbits from telescopic measurement data (Hostetter, 1987). Through the years, various methods using the idea of least squares have become increasingly important in many applications. Communication systems, control systems, navigation, signal and image processing; all were the center of interest in the development of the fundamental ideas of least squares estimation.

The goal of these techniques revolved around providing a solution involving a linear transformation of the measurements to obtain the optimal estimate. In addition, a recursive formulation was derived in which the measurements are processed sequentially (Pratt, 1978). In digital signal processing terms, the method of least squares estimation is a filtering process acting on incoming discrete-time measurement signals to produce discrete-time outputs that represent a close estimate of the measured system parameters.

Norbert Wiener's monumental early work on the extrapolation, interpolation and smoothing of stationary time series (Wiener, 1949) has led to the fundamental minimum mean square estimation theory; that

is, estimating signals from noisy observations.

In 1960, building on the work of Wiener, Rudolph E. Kalman pioneered his first work on linear minimum mean square estimation (Brown, 1983). His result, presently known as the Kalman filter, is a fundamental departure from that of Gauss (Hostetter, 1987) in that it introduces a generalization of recursive least squares. It is especially convenient for digital computer implementation. Nevertheless Kalman's work went hand in hand with the Wiener filter method mentioned earlier.

Both Wiener and Kalman filtering techniques continue to play a prominent role in modern time series analysis. The applications involve the creation of an optimal function that best yields the extraction and the separation of signals buried in noise and sidelobes that confuse the maps and images making them hard to interpret. An example problem which has received much attention in the literature is that of image restoration. Here a blurred image is recorded, and it is desired to remove the blur effect, which ultimately 'restores' the integrity and fidelity of the object being imaged. Image restoration problems of this type occur in a wide variety of fields, including radio astronomy and astrophysics and, recently, in biomedical imaging.

Image restoration by the method of least squares as applied to optical images (Helstrom, 1967) was also an area of research. The solution to such a problem, where the data was corrupted by noise or experimental error, was treated by finding an optimal impulse

response function that minimized the mean squared error between the true image and the estimated one. Here the estimated image depends on assumptions about the spectral densities of the images and the noise.

In 1982, Michael P. Ekstrom published his work on Wiener's mean-square estimation as applied to a two-dimensional imaging system. The extension of Wiener's filtering theory was then expanded to cover the multidimensional case. The problem was addressed by discussing the physical realizability and causality of the Wiener filter as they arise in 2-D. The optimal filter was then derived by solving a 2-D discrete Wiener-Hopf equation, using a 2-D spectral factorization procedure.

In the area of diagnostic ultrasound, the Kalman filter has been employed to accurately determine the locations of tissue structure from observed reflected signals (Kuc, 1979). In their publication, Kuc and his colleagues described the application of Kalman filtering to improve the range resolution of ultrasound signals. The results of their study demonstrated that improving the resolution capability, by applying the Kalman filter, depends upon the quality of the observed signal in terms of the signal-to-noise ratio, and on the accuracy with which the observed waveforms were modelled.

The Wiener filter was then applied to that same area but as a different tool. In 1982, Neal (Neal and Thompson, 1982) explored the use of the Wiener filter as an ultrasonic scattering amplitude es-

timation technique. Once again, the proposed method was based on an estimate of the signal-to-noise ratio, but this time, as a function of frequency for the backscattered signal from the target under test.

In the field of communications, the minimum mean-square error theory was applied (Lu and Wise, 1984) in the context of a symmetric uniform quantization. In their paper, Lu and Wise believed that although digital signals occupy a dominant role in modern communication systems, physical signals are generally analog in nature. Therefore, it is of essential importance to perform uniform quantization when analog-to-digital conversion is taking place. They also observed that, for several different distributions of the input signals, log-log plots of mean-square error versus number of output levels exhibit nearly linear behavior.

In the same field of communications, but in a different approach to test for the minimum mean-square error, graphical communication proved to be a growing area for the application of Kalman filtering to handwriting signal encoding over the telephone channel (Yasuhara and Yasumoto, 1984). The method was used to improve a handwriting signal transmitted in the presence of quantization noise. The Kalman filter improved not only the estimate of the signal, but the signal-to-noise ratio of the reconstructed signal as well.

The Wiener and the Kalman filters also had their share in the field of radio astronomy. In 1986, an adaptive regional Kalman filtering technique (Zheng and Basart, 1986) was used to further improve

noisy radio astronomy maps. Many simulation tests were done using this technique. One test consisted of a noisy radio astronomy map having a 'ring' structure with a dynamic range (maximum intensity of source divided by RMS noise level in background) of 9.6. The details of the ring were blurred. By applying the Kalman filter to the noisy map, the dynamic range was increased to 82.0. The reconstructed map not only showed an increase in the dynamic range but also preserved the edges of the ring.

In the same year, another study was under way to determine the optimal convolving function for creating the least corrupted uniformly spaced data from noisy nonuniformly spaced data using the Wiener filter theory (Ghorayeb, 1986). It was observed that radio astronomy data are collected on a nonuniform basis; therefore requiring the data to be smoothed and then resampled on a regular rectangular grid. The smoothing-resampling process degrades the data even further. In this study, it was shown how the Wiener filter theory can be used to determine the optimal selection of a smoothing function that yields the best estimate of the source's true signal. Two different types of sources were simulated in computer experiments: a point source, and a Gaussian source. The results from this study were significantly improved for a high dynamic range situation; however, for the low dynamic range case, the Wiener filter became unstable and thus non-optimal with respect to the standard smoothing function used for the same purpose at that time.

B. The Goal of the Thesis

To date, the Wiener filter has not been applied explicitly as a mean of 'reconstructing' an ultrasonic noise corrupted image. It is the intent of this thesis to explore how well the Wiener filter works when used as an optimal filtering operation to reduce the amount of noise in the images produced by ultrasonic waves reflected from targets in the human or animal subjects.

This introductory chapter is followed by a review chapter showing the general concept behind the principles and instrumentation of ultrasound. Included in Chapter II also, is a description of the tissue characterization when an ultrasonic wave is applied.

Chapter III provides a brief introduction of the noise characteristics in images produced by ultrasound. It brings to the reader's attention the various forms of noise that a medical image contains. It then progresses to demonstrate the signal processing side of the noise and its effect on the signal and the signal-to-noise ratio.

The body of the thesis revolves around Chapter IV. This chapter contains a description of the smoothing function (i.e., the transfer function) that will best separate the true signal from the corrupting noise. This is done by presenting the Wiener filtering method and by employing the theoretical as well as the practical approaches that lead to the 'optimal' choice of the convolving function.

Chapter V converges into showing the experimental set-up used in this research in addition to various tests and results deduced from these experiments as well as conclusions drawn from their respective analysis.

II. AN OVERVIEW OF ULTRASOUND PRINCIPLES AND INSTRUMENTATION

The theory behind sound waves has been extensively investigated and as a result ultrasound equipment is widely used for clinical imaging. The images produced by those ultrasonic waves are unique since they represent the internal interaction with the mechanical properties of organs and other tissues in the human body, and hence, serve as a powerful diagnostic tool of modern medicine. This chapter is intended to cover the basic principles of ultrasound waves and a brief overview of some of the devices involved in the application of tissue characterization.

A. Physical Principles of Ultrasound

Ultrasound waves used in medical diagnostic equipment propagate longitudinally into the body; that is, the motion of the particle is in the same direction as of the path of transmission. Such a transmission is initiated by a piezoelectric (pressure/electric) transducer, which also plays the role of detecting reflections of the transmitted ultrasound pulses back from the target under test. A basic reflection imaging configuration is shown in Figure 2.1.

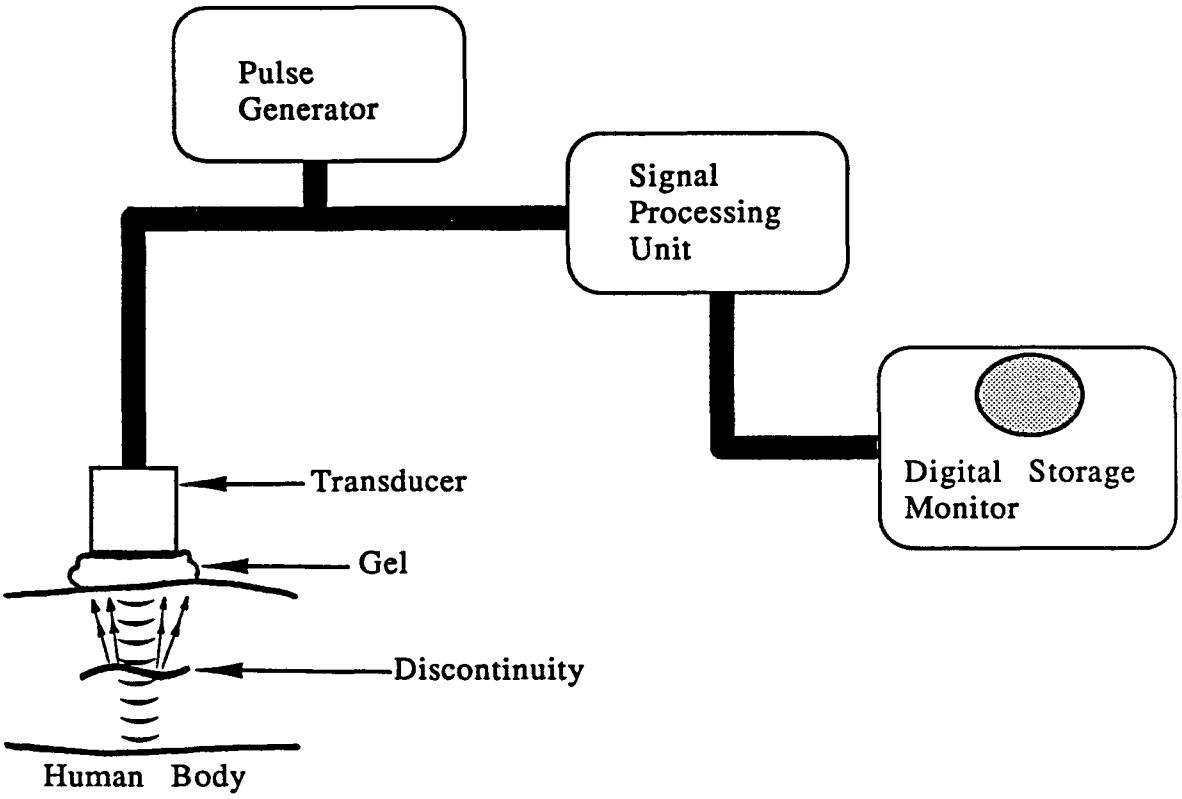


Figure 2.1. Basic reflection imaging system

The pulse generator excites the transducer, which transmits a signal shown in the solid curved lines. As soon as the propagated wavefront hits a discontinuity, a reflected wave is produced, as suggested by the arrowed lines in the diagram. The same transducer receives this reflected wave and the signal is processed by a Signal Processing Unit (SPU) and then displayed on a Digital Storage Monitor (DSM). The SPU usually consists of an amplifier, an A/D

converter, a bandpass filter, and an envelope detector (Macovski, 1983).

An important aspect of sound waves is the speed of propagation, v. It is assumed that the wave propagates at a constant velocity throughout the body. Another important physical concept is the attenuation coefficient, α, which is also uniform through the body. If the body under test is modeled as an array of isotropic scatterers (Hill, Nicholas, and Bamber, 1976), with reflectivity R(x,y,z), the resultant processed signal s(t) is given by

$$s(t) = \sum_x \sum_y \sum_z \frac{e^{-2\alpha z}}{z} R(x,y,z)\ T(x,y)\ p'\left(t-\frac{2z}{v}\right) \qquad (2.1)$$

where

z = distance from the target to the transducer's face

$e^{-2\alpha z}$ = attenuation in the tissue through the round-trip distance of 2z

T = lateral distribution, on the transducer's face, of the propagating wave

$p'\left(t - \frac{2z}{v}\right)$ = received pulse delayed by the round-trip time 2z/v

The received pulse, p'(t), is then the result of convolving the transmitted pulse from the pulse generator, p(t), with the impulse responses of the transducer and the corresponding linear filters in the SPU. The loss in amplitude of the reflected wave due to diffraction spreading from each scatterer, as shown in Figure 2.1, is represented by the 1/z factor in equation (2.1) above.

B. Characterization of Tissue Using Ultrasound

As sound propagates, its intensity, I, generally diminishes with the distance of propagation z (Havlice and Taenzer, 1979) according to

$$I = I_0 e^{-2\alpha z} \tag{2.2}$$

where I_0 is the intensity at z=0 (surface of transducer). The attenuation coefficient, α, depends highly on frequency, unlike the velocity of sound, and is directly and linearly proportional to frequency. This is true for most biological materials in the frequency range 1 to 10 MHz. 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. Table 2.1 shows few typical values for attenuation at 1 MHz. We see, for example, that a 3 MHz sound wave which has traveled a 10-cm distance through fat is 17.40 dB below its initial intensity level, while a 10-MHz sound wave traveling the same distance is 58 dB below its initial intensity level. This explains why low frequency ultrasonic waves are used for imaging structure deep in the body of an obese patient. However, if the organ to be scanned lies just below the skin, such as in a very thin person or in an infant, higher frequencies are used. This limitation on frequency has dif-

Table 2.1. Attenuation coefficient for some materials at 1 MHz

Material	Attenuation Coefficient (dB/cm)
Water	0.0023
Air	11.000
Fat	0.58
Soft Tissue	0.81
Liver	0.95
Kidney	1.1
Muscle	1.70
Bone	12.0

ferent impacts on equipment performance, since the frequency, f, and the velocity of sound, v, in a specific medium determine the wavelength, λ, of the ultrasonic pulse, which is kept as short as possible to improve axial resolution (Haumschild, 1981). The equation is given by

$$\lambda = \frac{v}{f} \tag{2.3}$$

It is seen then, 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 body as the lower frequency one.

As mentioned earlier, it is assumed that the propagation velocity of sound throughout the body is constant. In order to determine the depth of a reflected echo, the round-trip time of the latter is used. This time can be converted to distance, z, from the transducer's surface to the target by knowing the speed of ultrasound in the tissue. Some representative typical propagation mean velocities in various materials are given in Table 2.2.

The soft tissues of the body do not exhibit major changes in their acoustic velocities, but rather, they are limited to a very narrow range. This is fortunate, since fluctuations in velocity can cause little or large geometric distortions in the produced images and, thus, create uncertainties in the final diagnostic. Additional

Table 2.2. Ultrasound propagation velocities in some materials

Material	Mean Velocity (m/sec)
Water	1480
Air	330
Fat	1450
Soft Tissue	1540
Liver	1550
Kidney	1560
Muscle	1590
Bone	4080

geometric errors are caused by deflections of the propagating beam as a result of velocity variations. However, this could be looked upon as a useful technique for detecting malignant tumors which are sites of an increased propagation velocity with respect to their surrounding normal tissues. One last, but important, aspect of sound wave propagation is reflectivity, represented by the term R(x,y,z) when the body tissues are modeled as given by equation (2.1). Reflectivity plays a very unique role as a contrast agent when body images are being produced. It is used, in a narrow sense, as the

simplest behavior occurring at the interface of two adjoining layers. Changes in the characteristic impedance of the materials constituting these layers determine the reflectivity of the surface.

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

$$Z = v\rho \tag{2.4}$$

Listed in Table 2.3 are some values of characteristic impedance for a variety of media.

Table 2.3. Characteristic Impedance for various media

Medium	Characteristic Impedance (10^6 kg .m^{-2} .s^{-1})
Water	1.48
Air	0.0004
Fat	1.37
Soft Tissue	1.62
Liver	1.66
Kidney	1.63
Muscle	1.71
Bone	7.8

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

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1} \tag{2.5}$$

Table 2.4 gives the reflectivity at normal incidence for a variety of tissue interfaces.

It is seen, from Tables 2.3 and 2.4, that the greater the difference of the impedances of the adjoining media, the greater the amount of reflection coefficient at their respective junction.

The amount of sound reflected from an object depends, not only on the difference between the acoustic impedances (Havlice and Taenzer, 1979) of that object and its immediate vicinity, but also on the size, shape and orientation of the object.

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

Media Interface	Reflectivity
Soft tissue - Water	0.05
Soft tissue - Air	0.9995
Muscle - Liver	0.01
Muscle - Kidney	0.03
Fat - Liver	0.09
Fat - Muscle	0.10
Fat - Kidney	0.08
Fat - Bone	0.69
Brain - Skull bone	0.66

III. CHARACTERISTICS OF NOISE IN ULTRASONIC IMAGES

Unlike various types of imaging techniques, where noise is signal-dependent (Macovski, 1983), the noise in ultrasonic systems is an additive random Gaussian process resulting primarily from the transducer set-up, and secondarily from the location of the target under test, especially in the human body. This chapter tackles the concepts necessary to understand the physical characteristics which are of extreme importance in the study of noisy stochastic processes. The discussion will then proceed to introduce the various kinds of noise that can be present during a medical diagnostic situation. The problem of introducing these parameters is presented and developed for the two-dimensional process.

A. A Review

Ultrasonic signals can give numerical information about the structure and function of biological systems. The time taken by the transmitted ultrasonic pulse to travel a round trip path can be used to estimate the distance between the interface of the transducer to the target being scanned; and thus, provides an accurate diagnostic method which has a wide application in the field of clinical medicine. From the earliest use of medical ultrasonics, however, some diagnostic procedures have involved a tremendous amount of

numerical analyses, of varying degrees of complexity, depending on how noisy the produced images were.

In typical medical ultrasonic imaging systems, the observable signal, $S_i(t)$, may, for instance, be the intensity of the ultrasonic pulse. This signal, at any one point in time, consists generally of two components: the information-carrying signal, $S(t)$, and an unwanted noise component, $N(t)$, so that

$$S_i(t) = S(t) + N(t) \tag{3.1}$$

The nature of noise present in an image depends on the way in which the image is generated. Typical forms of noise in medical ultrasonic images are:

1) Fluctuation noise, which occurs when an image is formed by counting the number of reflections arriving from the scatterers originating at organ cells.
2) Fat noise, which is created by adipose tissues which constitute the fat layers located either directly on top of the scanned organ or in the subepidermis region under the skin surface.
3) Computation error noise, due to numerical computations of the image as a result of a series of observations.
4) Systematic error noise, due to instrumentation malfunction (i.e., defect in transducer).

5) Aliasing noise, from other targets which may or may not lie in the field of view and, therefore, are the site of actual emissive reflections that can cause prominent defects to the image.
6) Speckle noise, also known as the earlier stated 'Fluctuation noise', which is seen when the reflectivity function, R(x,y,z), in equation (2.1) is modeled as an array of scatterers. These scattered signals add coherently; that is, they add constructively and destructively depending on the relative phases of each scattered waveform.

B. Characteristics of Image Noise

We just saw, in the previous section, that unwanted signals (i.e., noise) come from a variety of sources, generally classified as man-made interference or naturally occurring noise. By careful engineering, the effects of many undesirable signals can be reduced or even eliminated completely. But there always remain certain inescapable random signals, which present a fundamental limit to systems performance.

Generally speaking, noise can be characterized by two forms: deterministic and stochastic. Deterministic noise is a process such that the noise signal, N(t), at a particular time, is the same at each replication of the observation. The noise is then completely self-determined and self-generated by its own process. On the other hand,

stochastic noise is that process which, at a particular time, is a random variable determined by various locations in the image and is different at each replication of the observation. Only the probability characteristics, and not the actual values of the noise at some specific time, are determined by the generating process.

One type of deterministic noise is designated as 'Gaussian white noise'. Such noise is characterized by its power spectrum which is constant over a wide frequency range, and contains frequency components in equal proportion throughout the spectrum. Now, the reason why this type of noise is classified as 'Gaussian' is because it is known to have a 'Gaussian' probability distribution and possesses the familiar bell-shaped curve, as given by

$$p(x) = \frac{1}{\sqrt{2\pi}\ \sigma} e^{-(x-m)^2/2\sigma^2} \qquad (3.1)$$

where m = mean value

σ = standard deviation

Figure 3.1 describes the continuous random variable p(x) which may take any values in the $[-\infty, +\infty]$ range but is mostly significant near the mean value m.

Because white noise contains all frequencies in equal proportion, it is a convenient process for filter measurements and experimental design work. Consequently, white noise sources with calibrated

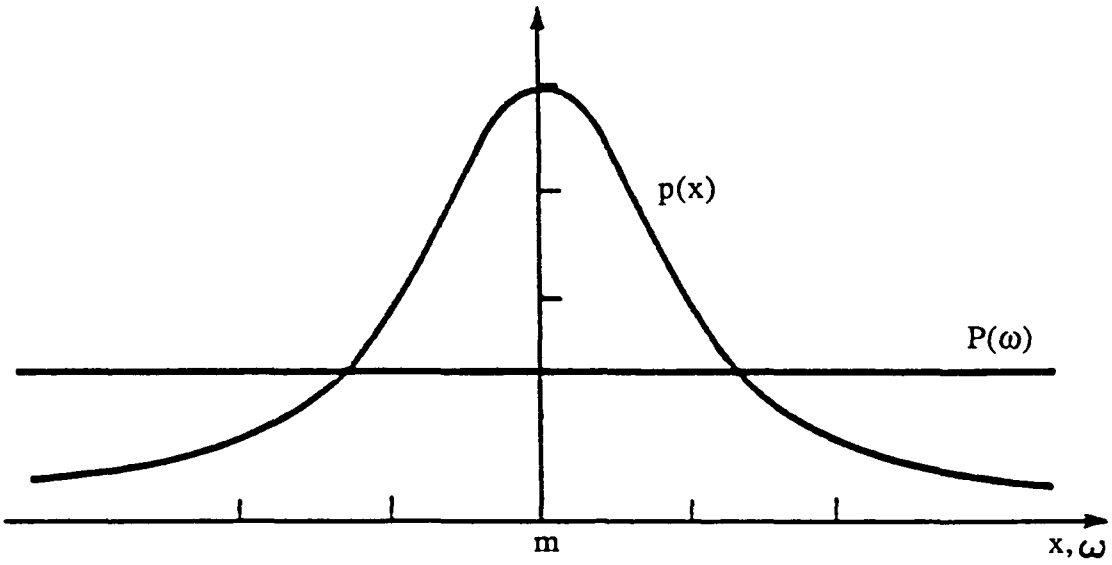


Figure 3.1. Statistical characteristic of Gaussian white noise
p(x): probability distribution
P(ω): power spectrum

power density have become standard laboratory instruments and as will be seen later in this thesis, the noise generator is a computer program (see Appendix) written in FORTRAN 77.

C. Signal-to-Noise Ratio

The signal-to-noise ratio, S/N, resulting from ultrasonic systems governed by additive Gaussian noise, is the ratio of the received signal power at the transducer terminals to the average noise power $\overline{p_n^2}$. In estimating the signal, it must be emphasized that the attenuation compensation takes place beyond the transducer and has generally no effect on the signal-to-noise ratio. Therefore, reflections emanating from greater depths, which experience increased attenuation, result in a reduced S/N ratio.

At a particular depth, z_o, the S/N ratio is defined as the peak signal power received at that depth, divided by the noise power, as given by

$$S/N = \frac{P_o^2}{\overline{p_o^2}} \tag{3.2}$$

where P_o is the peak value of $s_o(t)$, the signal envelope of s(t), derived from depth plane $z = z_o$. This signal $s_o(t)$, in a single transducer system, using steady-state diffraction theory, is given by

$$s_o(t) = \left| \frac{e^{-2\alpha z_o}}{z_o^2} \; p'\left(t-\frac{2z_o}{v}\right) \sum_x \sum_y R(x,y,z_o) \; [T(x,y) \; ** \; e^{j(kr^2/2z_o)}]^2 \right| \tag{3.3}$$

The peak value P_o is given by

$$P_o = \frac{e^{-2\alpha z_o}}{z_o^2} \; P'' \left| \sum_x \sum_y R(x,y,z_o) \; [T(x,y) \; ** \; e^{j(kr^2/2z_o)}]^2 \right| \tag{3.4}$$

where P" is the peak value of p'(t).

The summation expression, in equation (3.4) represents the product of the diffraction patterns of the source and the reflectivity at plane z_o. If the reflectivity function R, representing the object being studied at plane z_o, is small compared to the beam size, the summation is essentially over R itself. Conversely, if the reflectivity function, such as in a tumor, is large compared to the beam pattern, the summation is effectively over the beam pattern and is independent of the size of the object.

IV. OPTIMAL FILTERING

The Wiener Filter is an image restoration technique that uses a statistical procedure in order to correlate the true image to the noisy image and ultimately extract the signal from the noise. Knowing the statistical characteristics of the noisy picture and of the added noise, one can design an 'optimal' filter which can be used along with a digital computer to reconstruct the original source. In practice, we never have the true signal. One possible approach is to calculate the optimum filter for a variety of patients and then use these results inversely as a look-up table. That is, given a 'dirty' image, one could find a similar image in the look-up table and then reconstruct the source using the 'optimal' filter found in the table.

In this chapter, the mathematics behind the Wiener Filter method are presented in preparation of setting up the steps that constitute the procedure to be followed in order to create the 'optimal' function.

A. The Wiener Filter Method

It is the objective of this chapter to explore how well the Wiener Filter technique would lead to the optimal selection of the convolution function that best yields the actual signal distribution from reflected ultrasonic observations made in the presence of random noise. A simulation program is put together to achieve this task.

The problem is dealt with in the one-dimensional space, rather than two dimensions, because the mathematics are simpler. The programming, though, was implemented to meet the task of a multidimensional situation. If the results from the one-dimensional case are promising, the multidimensional case will be tested in future developments.

Consider the linear system shown in Figure 4.1. Let the input to this system be the observed true source noisy signal S_i presented in equation (3.1) and given by

$$S_i(t) = S(t) + N(t) \tag{4.1}$$

where S(t) is the true signal and N(t) is the added noise. Let the output of the system, $S_o(t)$, be the actual measured signal. G(t) is the transfer function of the system or, in more familiar terms, the unknown 'optimal' function that is to be chosen in such a way to make $S_o(t) \sim S(t)$.

The $S_o(t)$ can be given by

$$S_o(t) = \int_o^\infty G(\tau)\ S_i(t-\tau)\ d\tau \tag{4.2}$$

The above equation follows from the standard convolution theorem.

Our goal, then, is to have $S_o(t)$ approximate as closely as possible the time signal S(t). That is, we want to minimize $[S_o(t) - S(t)]$. As a criterion for measuring the difference between $S_o(t)$ and

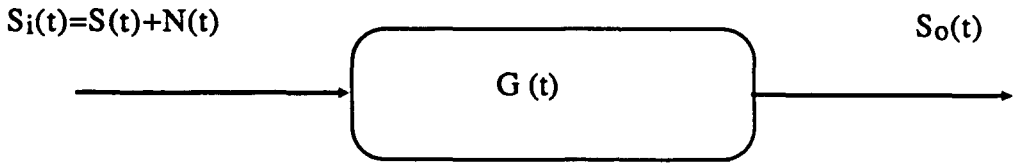


Figure 4.1. 'Optimal' Wiener Filter linear system

S(t), in some period 2T, we shall take the limit as given by

$$\lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} [S_o(t) - S(t)]^2 \, dt \tag{4.3}$$

The above limit gives the square of the rms value of $[S_o(t) - S(t)]$. We now choose a time interval, h, which is small enough to create a $S_i(t)$ that is well characterized by its values at the points $t = kh$ where k refers to the integral values (i.e., $k = -T,\ldots, T$). Letting $S_i(t) = b_k$, then, the observed true source noisy signal can be regarded as a discrete sequence $b_{-T},\ldots, b_o,\ldots, b_T$. Also letting the true signal S(t) be the sequence (a_k), $a_{-T},\ldots, a_o,\ldots, a_T$, the added noise N(t) will have the form (b_k-a_k), that is $(b_{-T}-a_{-T}),\ldots, (b_o-a_o),\ldots, (b_T - a_T)$. Therefore, the linear system shown in Figure 4.1 will be regarded as having an input b_n and an output to approximate the true source sequence, a_n. Equation (4.2) can be approximated by the following summation:

$$S_o(t) = \sum_{n=1}^{\infty} G(nh)\, S_i\,(t-nh) \tag{4.4}$$

Letting $G_n = G(nh)$ for $n \geq 0$ and suitably choosing some value M, (4.4) could be written as

$$S_o(t) = \sum_{n=0}^{M} G_n \, S_i(t-nh) \tag{4.5}$$

Equation (4.5) says that $S_o(t)$ is approximately given by a weighted sum of a number of past values of the input $S_i(t)$. In case $S_o(t)$ and $S_i(t)$ are determined by their values at $t = kh$, we find

$$S_o(kh) = \sum_{n=0}^{M} G_n \, S_i[(k-n)h] \tag{4.6}$$

If we let $S_o(kh) = S_{ok}$ and $S_i[(k-n)h] = S_{i(k-n)}$, equation (4.6) can be rewritten as

$$S_{ok} = \sum_{n=0}^{M} G_n \, S_{i(k-n)} \tag{4.7}$$

Equation (4.7) is now used to determine G_n so that the errors

$$\varepsilon_k = a_k - \sum_{n=0}^{M} G_n \, b_{n-k} \tag{4.8}$$

are as small as possible. For this to happen, G_n should be chosen so that the average of the sum of the squares

$$\sum_{k=-T}^{T} \varepsilon_k^2$$

is a minimum. Stated in formula, we choose G_n so that

$$I = \lim_{T \to \infty} \frac{1}{2T+1} \sum_{k=-T}^{T} \left(a_k - \sum_{n=0}^{M} G_n \, b_{k-n}\right)^2 \tag{4.9}$$

is a minimum.

Equation (4.9) could be presented in a much easier form if we introduce the auto-correlation functions representation of each of the sequences a_k and b_k; that is,

$$R_a(k) = \lim_{T \to \infty} \frac{1}{2T+1} \sum_{l=-T}^{T} a_l \, a_{l-k}$$

$$R_b(k) = \lim_{T \to \infty} \frac{1}{2T+1} \sum_{l=-T}^{T} b_l \, b_{l-k}$$

and the cross-correlation function

$$R_{ba}(k) = \lim_{T \to \infty} \frac{1}{2T+1} \sum_{l=-T}^{T} a_l \, b_{l-k}$$

Note that if the true signal and the added noise are completely uncorrelated, then,

$$R_{ba}(k) = R_a(k)$$

The worst situation that can arise is when $R_{ba}(k) = 0$. This tells us that b_k and a_k have no correlation, which means that the added noise cancels the true signal completely and only random

residue, making it impossible to separate any part of the true signal from the true source noisy signal by a linear system.

Expanding equation (4.9) we get

$$I = \lim_{T\to\infty} \frac{1}{2T+1} \sum_{k=-T}^{T} a_k^2$$

$$- 2 \sum_{n=0}^{M} G_n \lim_{T\to\infty} \frac{1}{2T+1} \sum_{k=-T}^{T} a_k\, b_{k-n}$$

$$+ \sum_{n=0}^{M} \sum_{m=0}^{M} G_n\, G_m \lim_{T\to\infty} \frac{1}{2T+1} \sum_{k=-T}^{T} b_{k-n}\, b_{k-m}$$

Using the above auto- and cross-correlation functions representation, we have

$$I = R_a(0) - 2 \sum_{n=0}^{M} G_n\, R_{ba}(n) + \sum_{m,n=0}^{M} G_n\, G_m\, R_b(m-n) \qquad (4.10)$$

If G_n are chosen to make I a minimum, we must have

$$\frac{\delta I}{\delta G_k} = 0 \qquad k = 0, 1, \ldots, M$$

thus,

$$\frac{\delta I}{\delta G_k} = -2\, R_{ba}(k) + 2 \sum_{n=0}^{M} G_n\, R_b(k-n) = 0$$

So a necessary condition that the G_n make I a minimum is

$$\sum_{n=0}^{M} G_n\, R_b(k-n) = R_{ba}(k) \qquad (4.11)$$

for $k = 0, 1, \ldots, M$.

Equations (4.11) are a linear system of (M+1) equations with (M+1) unknowns. We see that determining Gn depends on the autocorrelation function of b_k and the cross-correlation function of b_k and a_k. It is of absolute necessity for the sequence a_k and b_k to be elements of stationary random processes that are invariant under a translation of time.

In summary, the Wiener Filter method can be characterized by the following (Brown, 1983):

1. Both the true source noisy signal and the added noise should be random processes with known auto- and cross-correlation functions.
2. The goal is to achieve minimum mean-square error for best performance.
3. A solution for the 'optimal' filter weighting function should be based on scalar methods.

B. The Discrete Case

The standard formulation of the discrete, single-channel Wiener Filter problem leads to a system of 'normal linear equations'. In the preceding section, the mathematics that led to these equations (4.11) were developed. The solution of these equations, that gives the 'optimum' function G(n), will be discussed in this section.

Since the system of equations is linear and holds for every n, we take Z-transforms of both sides of equation (4.11).

The definition of Z-transform is

$$\gamma_{xx}(z) = \sum_{n=0}^{\infty} R_{xx}(n)\, z^{-n}$$

Applying this to both sides of equation (4.11) gives

$$\gamma_{mt}(z) = \gamma_{mm}(z)G(z) \qquad (4.12)$$

This is shown in the following manner. Since,

$$R_{mm}(n) \longleftrightarrow \gamma_{mm}(z)$$

and $$G(n) \longleftrightarrow G(z)$$

then, by the convolution property of the Z-transform

$$\sum_{n=0}^{M} G(n)\ R_{mm}(k-n) \longleftrightarrow G(z)\ \gamma_{mm}(z)$$

Hence, the system function of the 'optimum' filter is given by

$$G(z) = \frac{\gamma_{mt}(z)}{\gamma_{mm}(z)} \tag{4.13}$$

where m and t stand for measured and true, respectively; and where $\gamma_{mt}(z)$ is the spectral cross-correlation function and $\gamma_{mm}(z)$ is the spectral autocorrelation function.

This way of solving for the 'optimum' filter could, in fact, have the same problem as that of deconvolution. This could be a problem that negatively affects the output of an 'optimum' filter after noise, with high rms levels, is added to the true signal. On the other hand, this process might be a less favorable path to take for solving the system of normal equations (4.11) for very noisy processes, since the spectral auto- and cross-correlation of these processes will also be very noisy, which will, as a result, introduce noisy 'optimum' filters.

V. THE PROCEDURE, THE TEST AND THE ANALYSIS OF RESULTS

In order to achieve our goal as far as the 'optimal' function is concerned, as presented in Chapter IV, a simulation program was set to constitute the procedure that leads to creating that function and ultimately restore the true signal. The steps of the procedure are:

1. Read in or create the true signal s(t) as it is supposed to be before noise is added.
2. Add some noise with a specific variance and seed, to the true signal in order to simulate the measured signal $S_i(t)$.
3. Take the autocorrelation of $S_i(t)$ with itself, and the cross-correlation of $S_i(t)$ with S(t).
4. Using the results from step 3, find the 'optimal' filter as per equation (4.13).
5. Convolve the latter with the measured signal $S_i(t)$ found in step 2.
6. Lastly, compute the normalized mean-square-error for the computed filter of the true and the restored signal.

As mentioned earlier, in practice, the true signal, due to ultrasonic reflections that characterize a target organ or tumor in the human body, is never known. As a consequence, a direct method to determine the 'optimal' filter cannot be obtained from the only known measured (true signal + noise) signal since one of the condi-

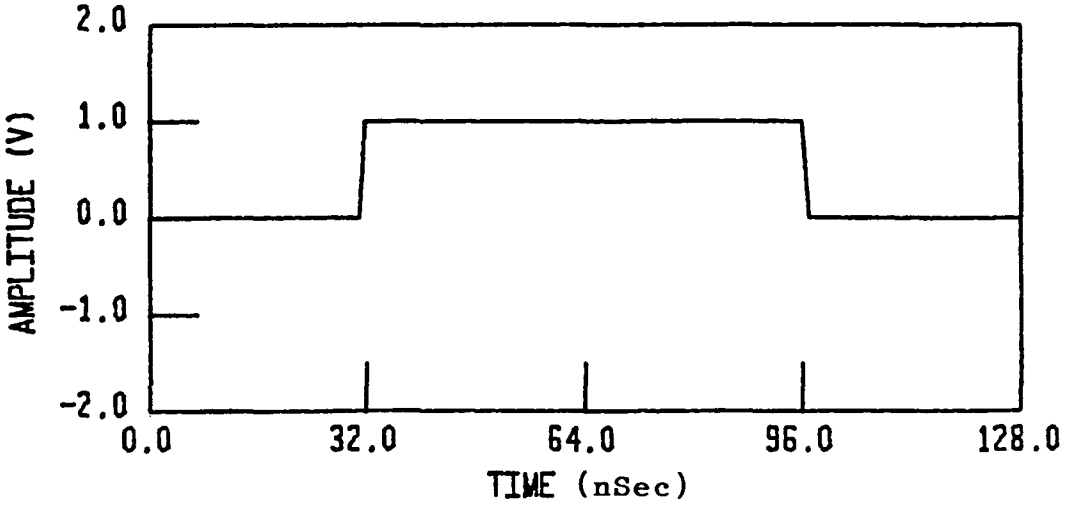
tions of the Wiener filter, as stated earlier, is to assume that both the true signal and the added noise are well-identified processes and with known auto- and cross-correlations functions. So then, one possible approach to acquire is to compute the 'optimal' filter for a variety of patients and then use these results inversely as a look-up table. That is, given a noisy image or signal, one would be able to find a similar image in the table, and then, re-construct the signal by using the corresponding 'optimal' function of the Wiener filter as determined by that look-up table.

A. Experimental Set-up

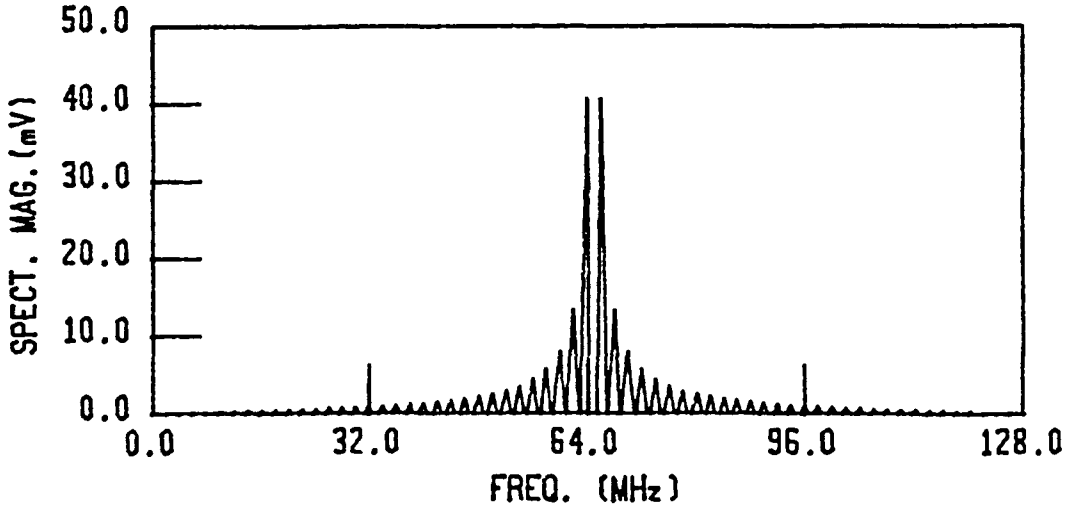
Various types of ultrasonic signatures were simulated using a program (see APPENDIX) that either creates the reflection via a mathematical model or reads in simulated results of pulse-echo waves produced by various transducer prototypes (Brown, 1988).

The mathematical model representing the first simulative pulse-echo wave was simply a 'pill-box' or in a more familiar term, a rectangular function whose amplitude and width are two varying parameters. Figure 5.1 shows the 'pill-box' in the time domain and its corresponding frequency domain spectral magnitude.

Four simulated pulse-echo performances corresponding to four various piezo film transducer prototypes #P10, #P13, #P17 and #P20 (Brown, 1988) were also used. It is not the intent of this work



(A)



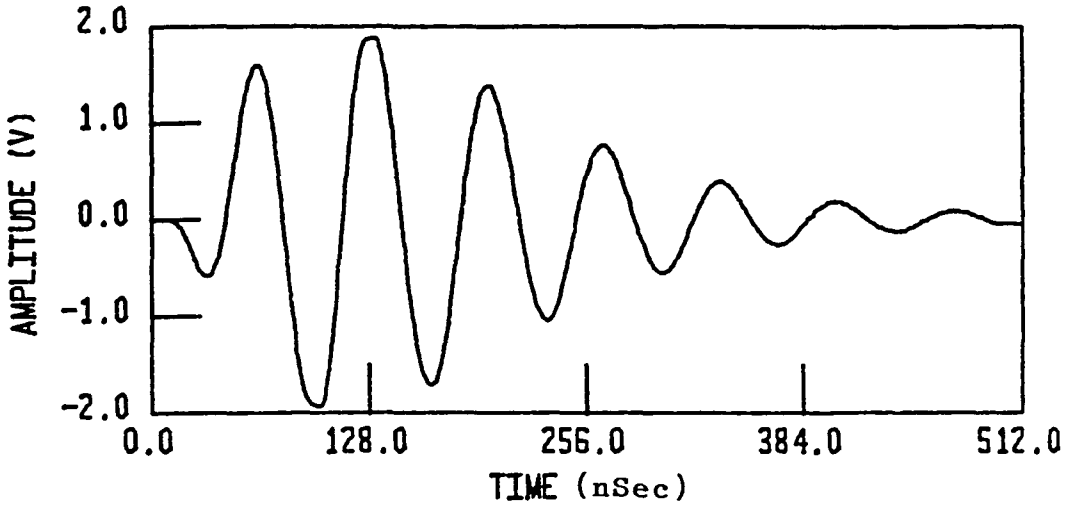
(B)

Figure 5.1. Simulated pulse-echo response given by a 'pill-box'
(A) time-domain
(B) spectral magnitude

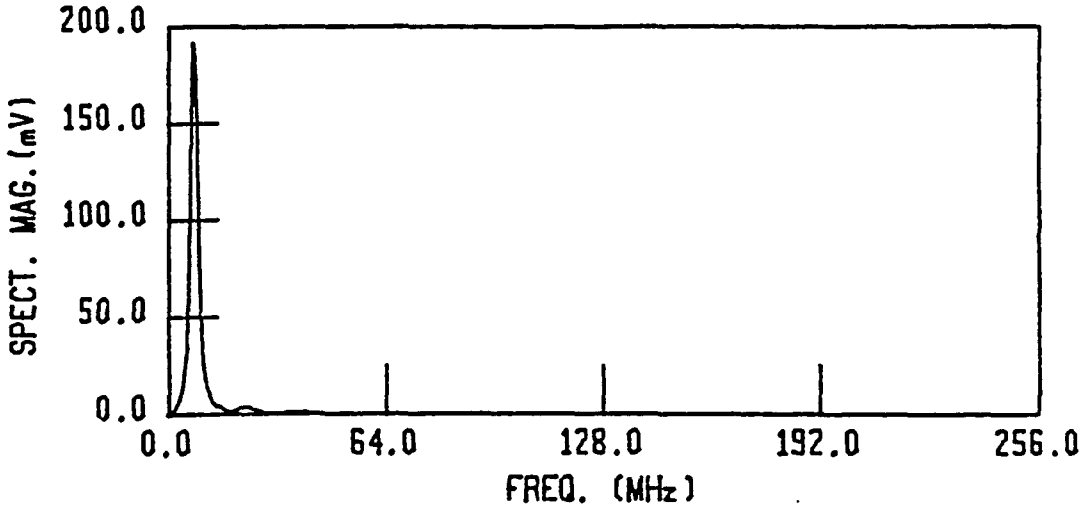
to go into a detailed explanation of the physical/chemical design of these transducers. The only thing I would like to point out, though, is that the transducers were constructed with copolymer probes which used gold metallization. The testing and simulation work on #P10 and #P13 showed very high sensitivity and a narrow band response. On the other hand, #P17 and #P20 were designed to show a more broad band pulse-echo response. Figures 5.2-5.5 show the time domain and the spectral magnitude simulation results of pulse-echo performances for these transducers.

B. Tests and Results

As a first attempt to test for the effectiveness of the Wiener filter from an optimal point of view, using the software (see Appendix) developed for that purpose, a simulated waveform representing an 'ideal' pulse-echo reflection was used. This waveform was created by adding a signal of Gaussian white noise, with zero mean and unity variance, to the rectangular pulse shown in Figure 5.1. The program used to generate the Gaussian noise (see Appendix C) was implemented to create a particular noise signal which depends on an input variable called the 'seed'. The seed determines how flat the spectral density of the noise would be over a given range of frequencies. It has been found, by trial and error, that a seed value of

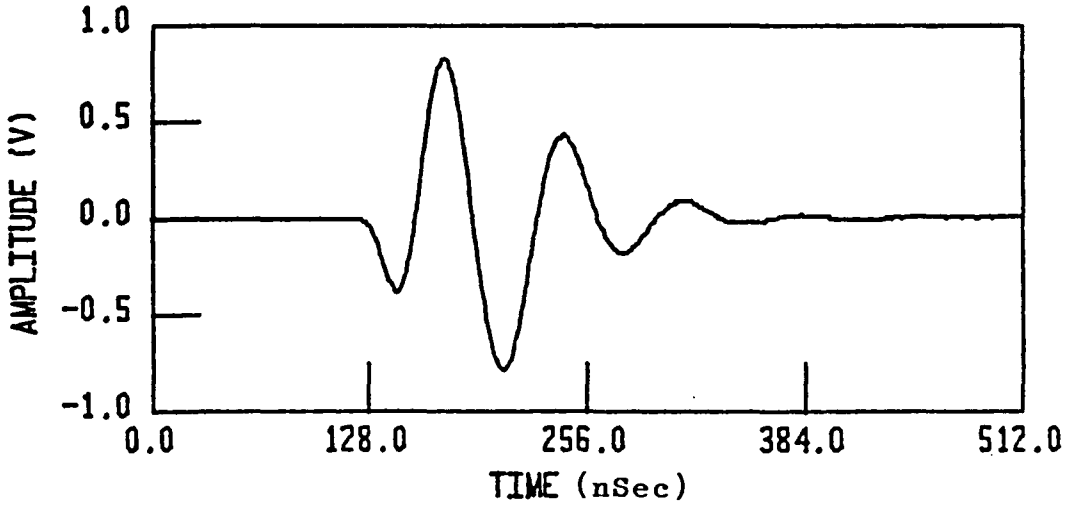


(A)

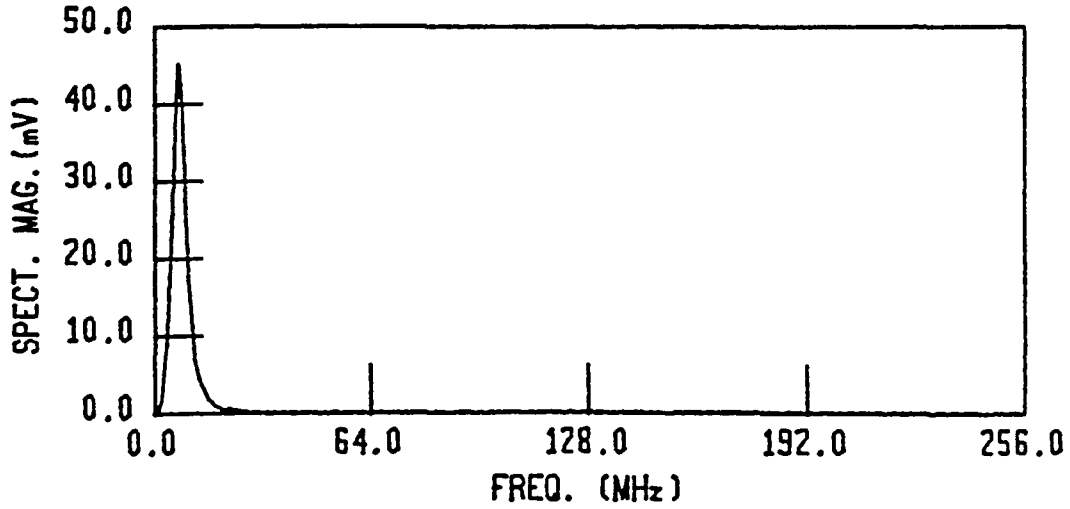


(B)

Figure 5.2. Actual pulse-echo return waveform for #P10 transducer (narrow-band)
(A) time-domain waveform
(B) spectral magnitude

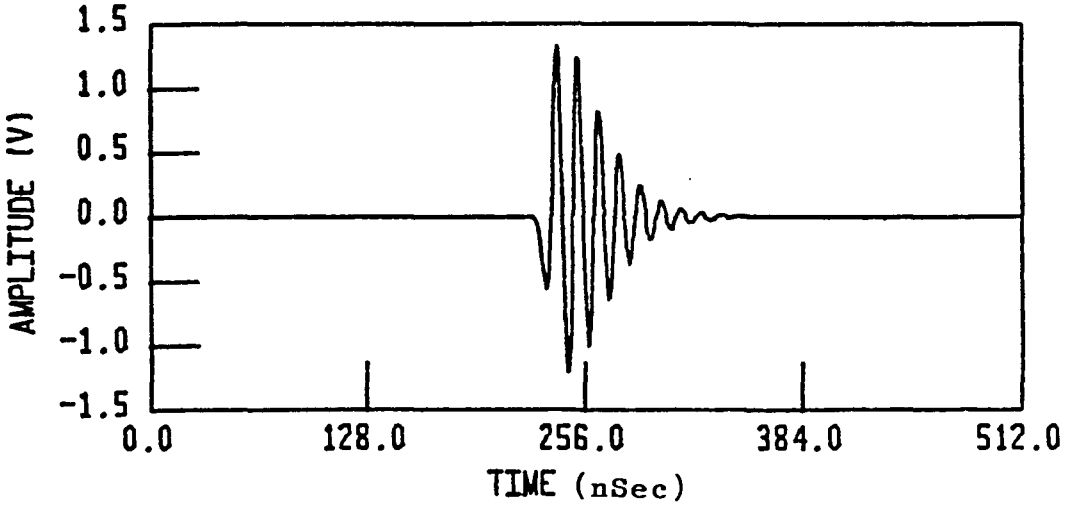


(A)

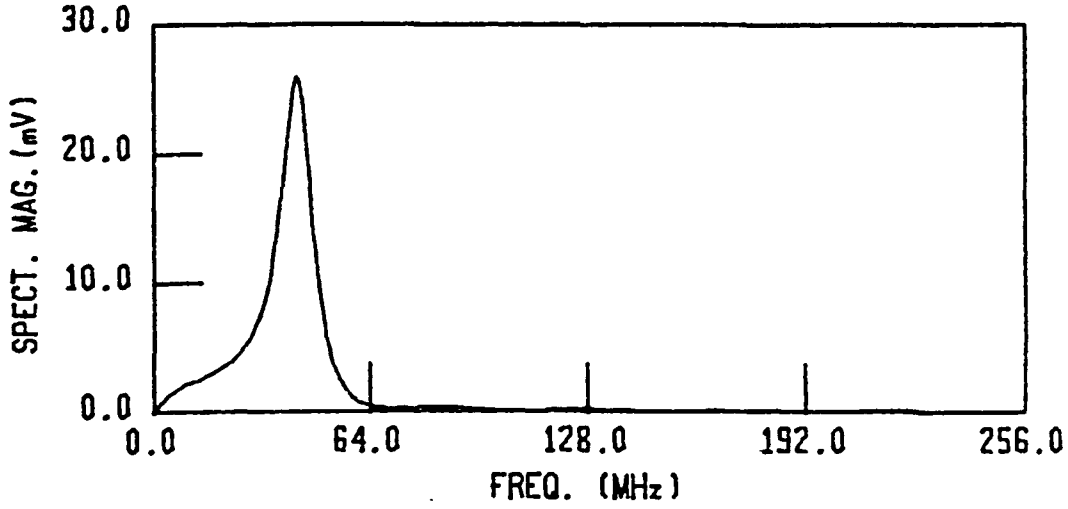


(B)

Figure 5.3. Actual pulse-echo return waveform for #P13 transducer (narrow-band)
(A) time-domain waveform
(B) spectral magnitude

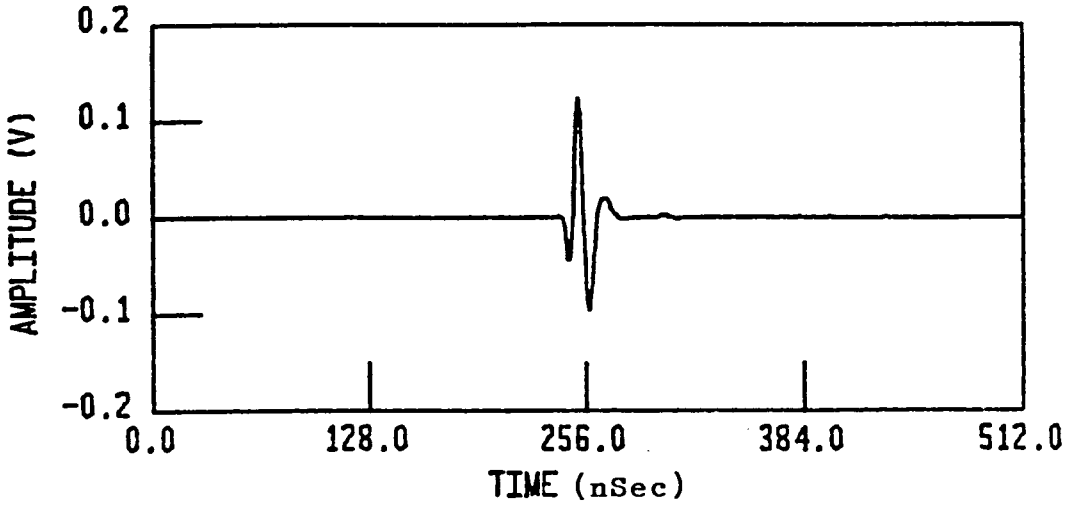


(A)

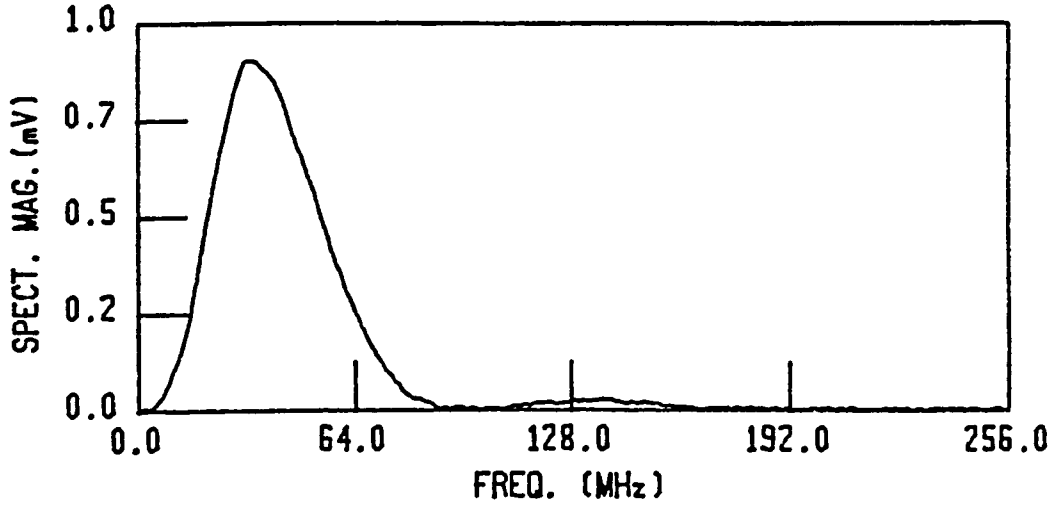


(B)

Figure 5.4. Actual pulse-echo return waveform for #P17 transducer (broad-band)
(A) time-domain waveform
(B) spectral magnitude



(A)



(B)

Figure 5.5. Actual pulse-echo return waveform for #P20 transducer (broad-band)
(A) time-domain waveform
(B) spectral magnitude

52.0 would provide a noise signal with a very wide-band spectrum, which is exactly what is needed in order for the noise to be characterized as 'white'. This value of the seed was used at all times during all tests.

This preliminary test was used to search for the occurrence of an optimal finite length (i.e., finite number of coefficients) Wiener filter. Two sets of computations were performed on the rectangular pulse. In the first one, the rms level of the noise was arbitrarily chosen to be 0.5 and in the second one, the rms was reduced to 0.4. A filter length of 6, in the first test, proved to be optimal. At that length, a minimum mean-square error (M.S.E.) of 0.108 was achieved as opposed to higher M.S.E. values attained after convolving the noisy data with the same filter but having a shorter or a longer length. In the second test, a minimum M.S.E. of 0.038 was reached using a filter length of 3.

Note that step 6 in the procedure, shown earlier in the chapter, that leads to creating the optimal Wiener filter and thus restoring the true signal, could be named the 'DECISION MAKER'. This 'DECISION MAKER' provides the information about whether or not an optimal Wiener filter, with a specific length, was accomplished. This step requires the computation of the M.S.E. of the true and the restored signals by the following equation

$$\text{M.S.E.} = \frac{1}{N} \sum_{i=0}^{N} (S_{it} - S_{ir})^2$$

where S_{it} and S_{ir} are the true and the restored pulse-echoes; respectively. As one of the characteristics of the Wiener filter, the M.S.E. should be as minimal as possible for optimal performance.

The assumption made, then, with respect to the presence of an optimal Wiener filter length at which a minimum M.S.E. is reached, proved to be realizable.

The testing went on next to cover more realistic situations. The four simulated pulse-echo reflections shown in Figures 5.2-5.5 were used. As mentioned earlier, the first two pulse-echoes, corresponding to transducers #P10 and #P13, showed to have a narrow-band response; while the last two reflections, corresponding to transducers #P17 and #P20, were designed to provide a more wide-band pulse-echo response (Brown, 1988).

Various tests were made on these reflections. All computations were performed in the time domain. Once again, the measured signal (i.e., noisy signal) was simulated by adding Gaussian white noise, with zero mean and unity variance, to the desired true pulse-echo. Seven Wiener filter lengths (2 to 8) were used, during each of the four tests, to check for the optimal situation when the M.S.E. reached a relatively minimum value. The rms level value for the added noise was again picked randomly for each case, just enough

for the true signal to appear pretty corrupted. Noise, with rms levels of 0.50, 0.14, 0.34 and 0.025, was added to the pulse-echo produced by each of #P10, #P13, #P17 and #P20 transducers; respectively.

A test that one can perform to determine how large the signal is, with respect to background noise, in the image is via what is called the 'DYNAMIC RANGE' (D.R.). The D.R. is given by the following formula

$$\text{D.R.} = \frac{\text{Maximum intensity of the signal}}{\text{RMS noise level (in background)}}$$

Table 5.1 shows the D.R. for each of the four pulse-echo signals after the noise signal was added with its respective rms level.

Table 5.1. Dynamic Range of the noisy signals shown in Figures 5.6(A) - 5.13(A)

Transducer	D.R.
#P10	3.841388
#P13	5.711462
#P17	3.952838
#P20	4.900750

Following the pattern given by the procedure at the beginning of the chapter, the Wiener filter was created using equation (4.13) and then convolved with the noisy pulse-echoes in order to produce an estimate of the true ones. Table 5.2 shows the results of each test performed on the noisy pulse-echo waveforms in addition to the minimum M.S.E.s achieved in each situation as a function of filter length (LFIL). Note that all M.S.E.s were computed after the restored signals were amplified and normalized to their corresponding true pulse-echoes.

Figures 5.6-5.13 show the noisy pulse-echo waveforms used in each test with their respective restored waveforms. Figures 5.6, 5.8, 5.10, and 5.12 show the filtered signals resulting form the convolution process of the noisy signal with the optimal length Wiener filter as indicated by Table 5.2. Figures 5.7, 5.9, 5.11 and 5.13 show the results of a non-optimal situation for the Wiener filter (LFIL = 7) which led to a non-minimal M.S.E.

The following section will elaborate on these results by presenting a thorough explanation on what the Wiener filter method has achieved vis-a-vis improving the distorted pulse-echo reflections.

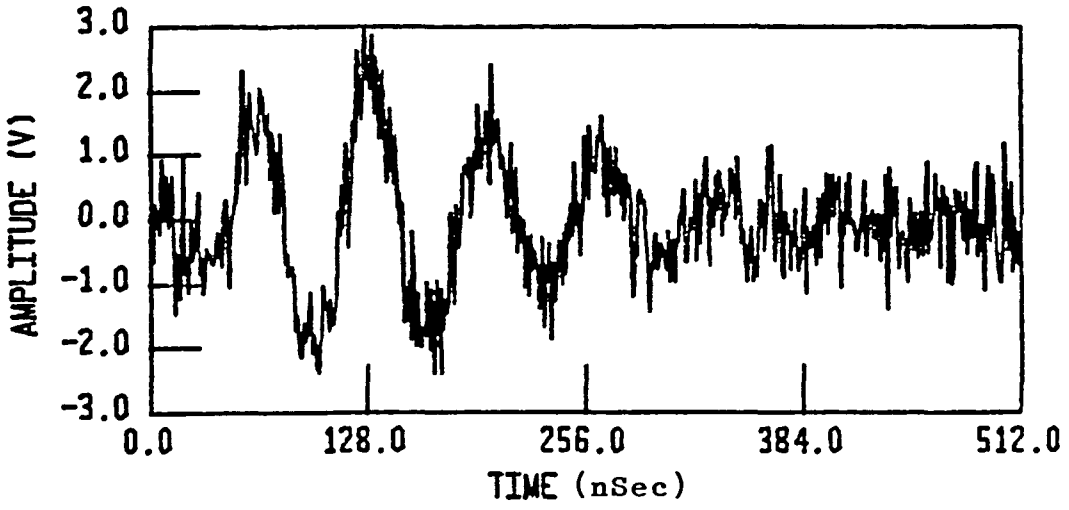
C. Analysis and Conclusions

A few things have to be discussed in order to evaluate the performance of the Wiener Filter just applied to ultrasonic pulse-echo

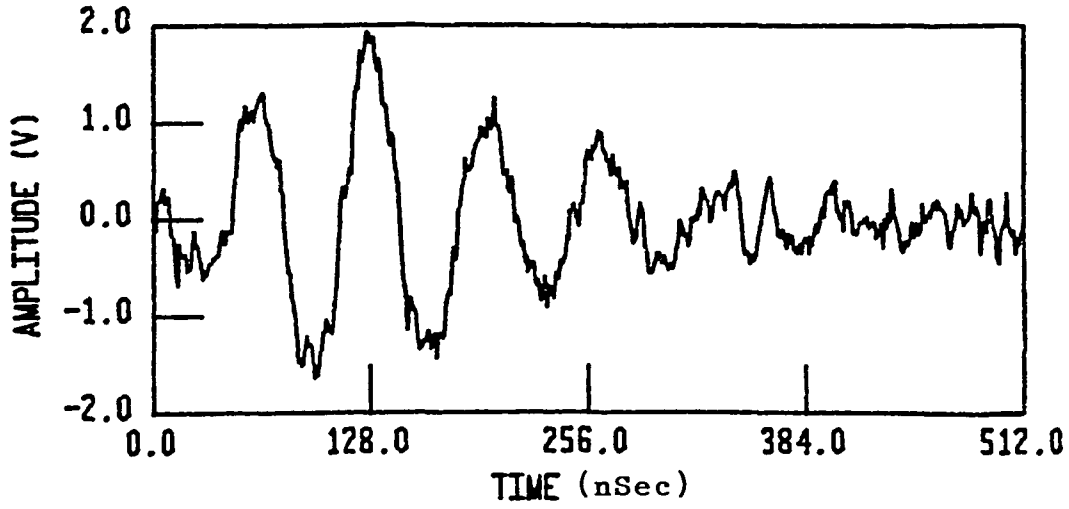
Table 5.2. Mean-square error vs. Wiener filter length after a normalized amplification of the restored signal for #P10, #P13, #P17 and #P20 transducers

Transducer	Filter Length	M.S.E.
#P10	2	11.28404 E-02
	3	8.69661 E-02
	4	7.96607 E-02
	5*	7.89403 E-02
	6	8.80743 E-02
	7	22.56284 E-02
	8	22.68874 E-02
#P13	2	9.03072 E-03
	3	6.67758 E-03
	4	5.78372 E-03
	5*	5.46910 E-03
	6	5.83497 E-03
	7	33.54119 E-03
	8	26.61132 E-03
#P17	2	5.07252 E-02
	3*	4.87135 E-02
	4	6.35663 E-02
	5	8.08125 E-02
	6	10.26430 E-02
	7	13.34327 E-02
	8	15.75842 E-02
#P20	2	5.04773 E-04
	3*	4.37144 E-04
	4	4.68149 E-04
	5	4.90305 E-04
	6	5.47612 E-04
	7	8.39851 E-04
	8	10.00594 E-04

* Optimal filter length

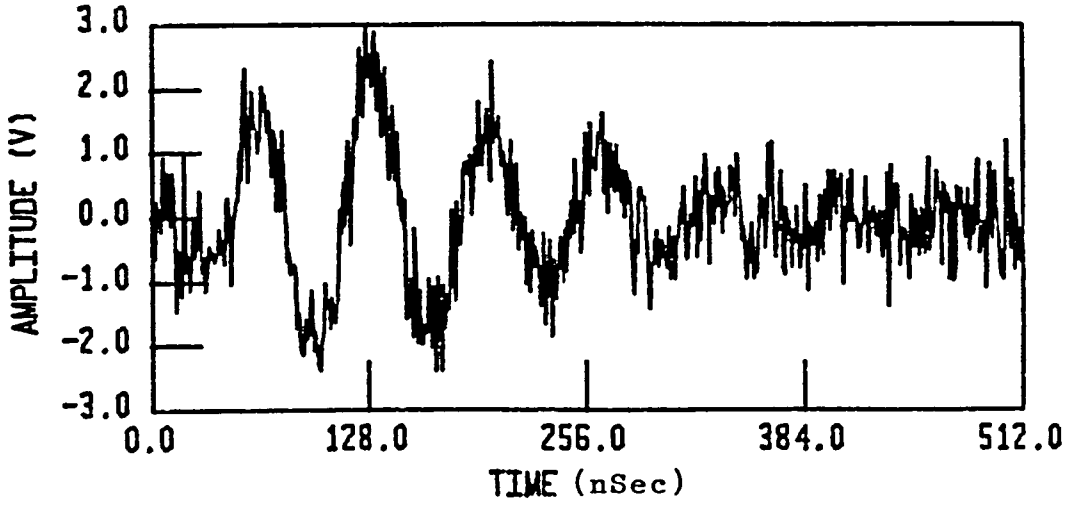


(A)

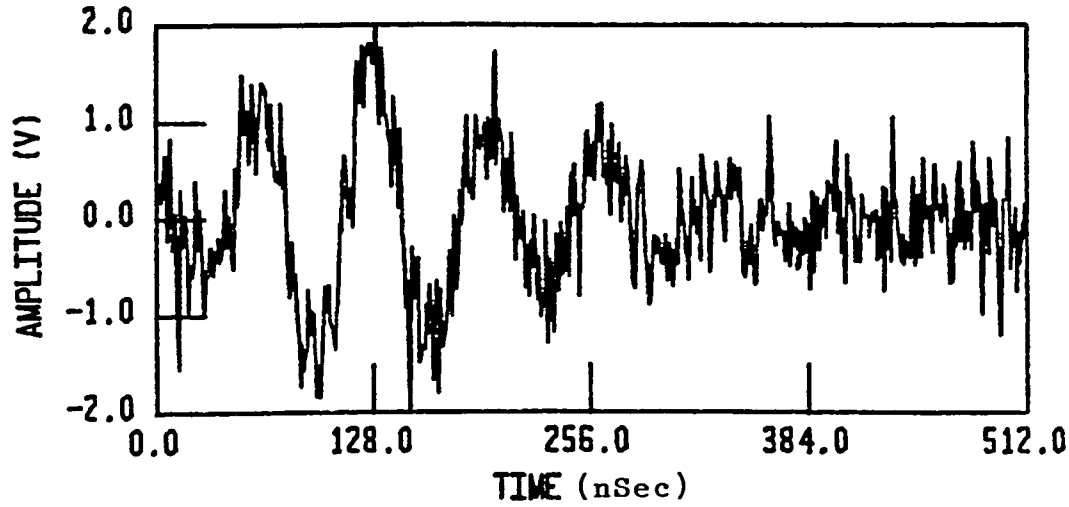


(B)

Figure 5.6. Wiener filtering (LFIL=5) of noisy pulse-echo for #P10 transducer
(A) noisy pulse-echo waveform
(B) filtered waveform

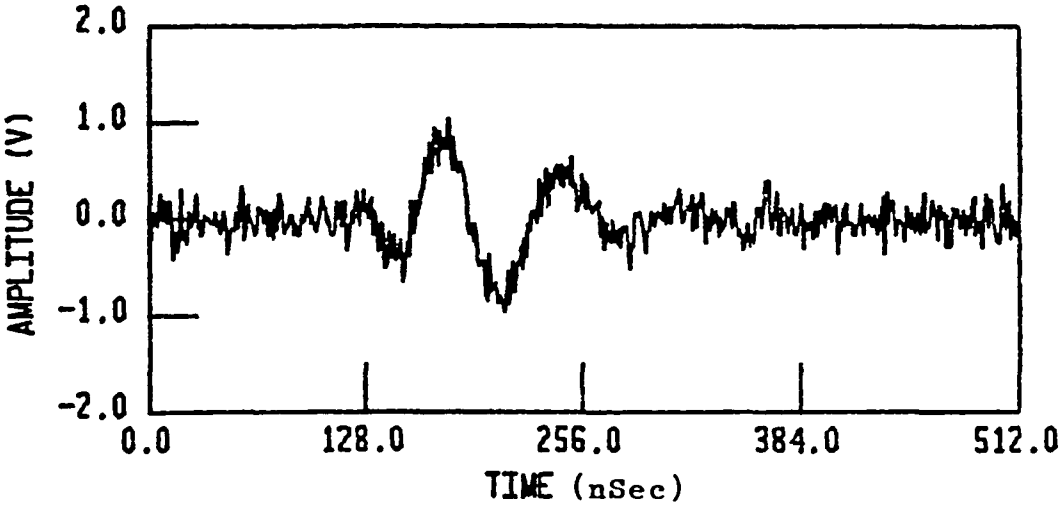


(A)

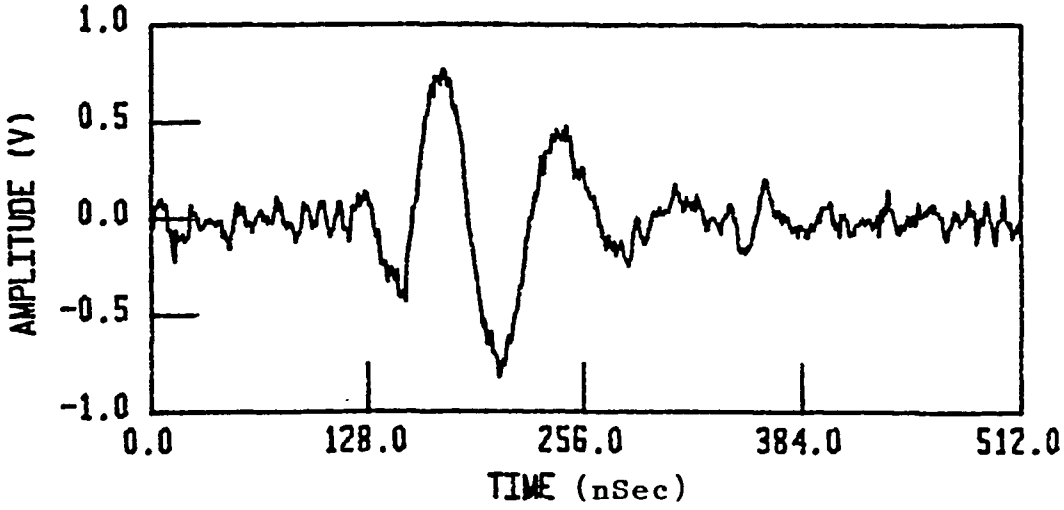


(B)

Figure 5.7. Wiener filtering (LFIL=7) of noisy pulse-echo for #P10 transducer
(A) noisy pulse-echo waveform
(B) filtered waveform

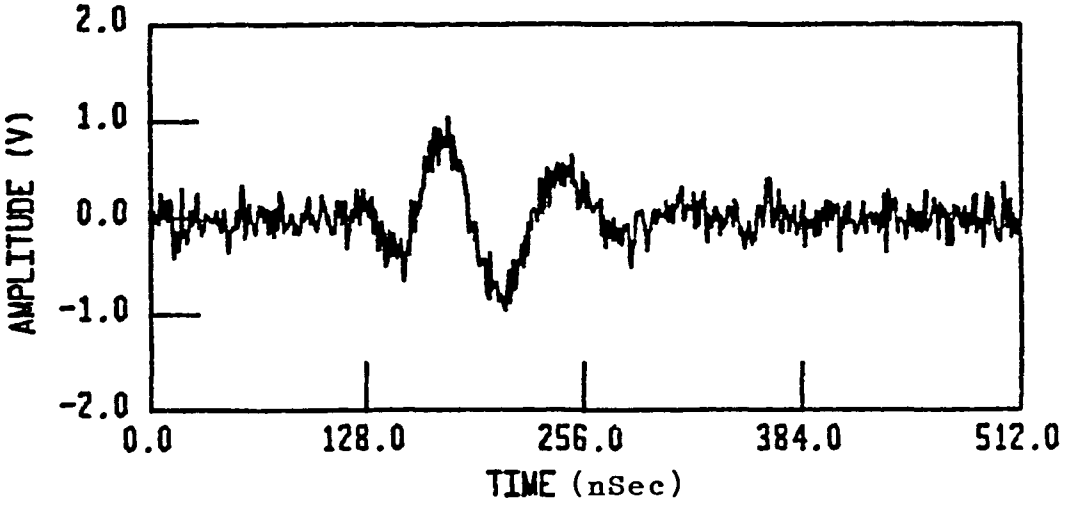


(A)

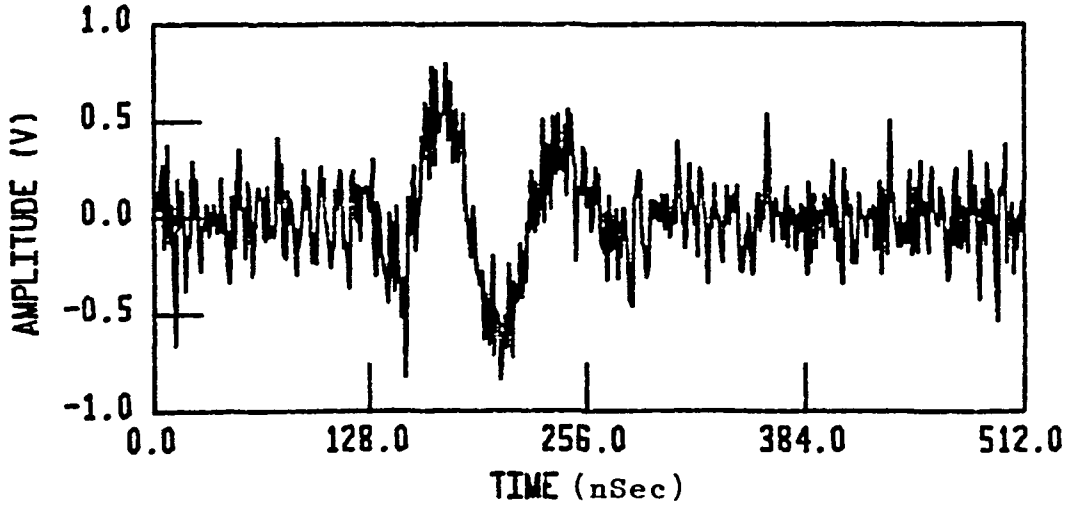


(B)

Figure 5.8. Wiener filtering (LFIL=5) of noisy pulse-echo for #P13 transducer
(A) noisy pulse-echo waveform
(B) filtered waveform

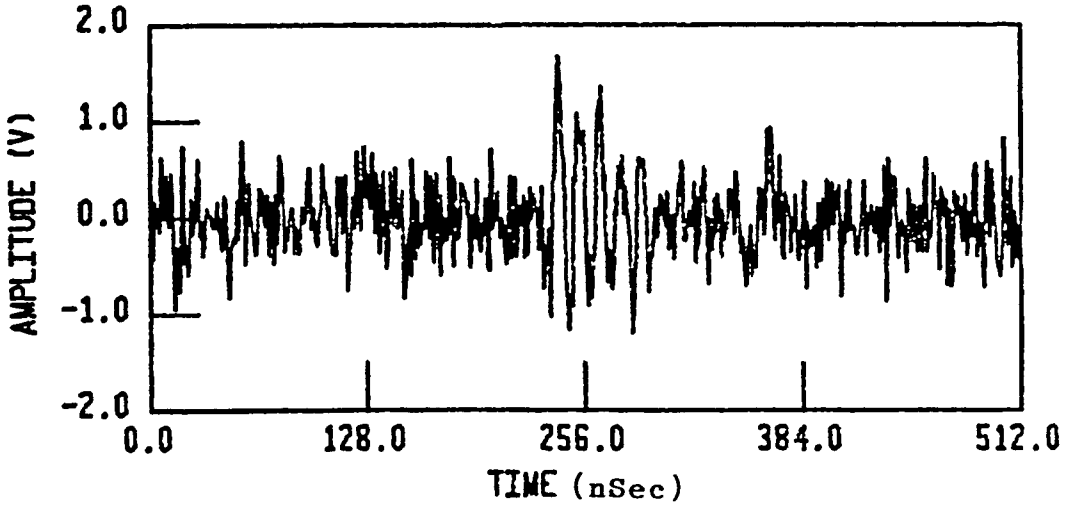


(A)

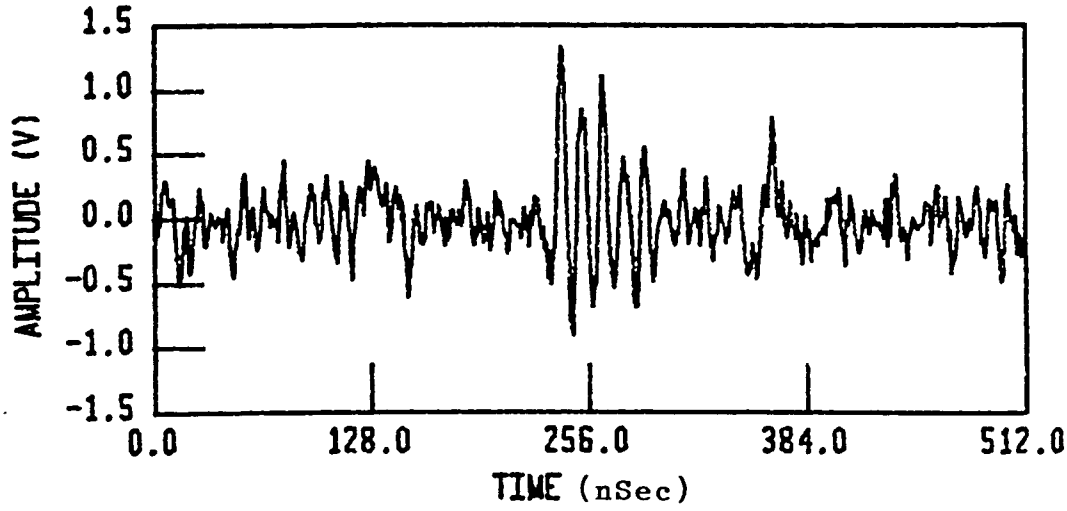


(B)

Figure 5.9. Wiener filtering (LFIL=7) of noisy pulse-echo for #P13 transducer
(A) noisy pulse-echo waveform
(B) filtered waveform

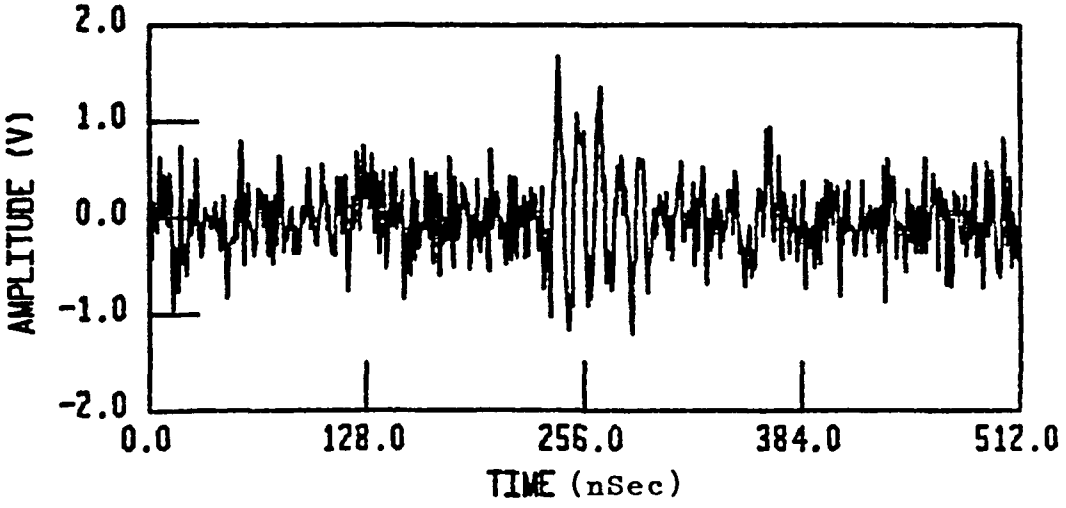


(A)

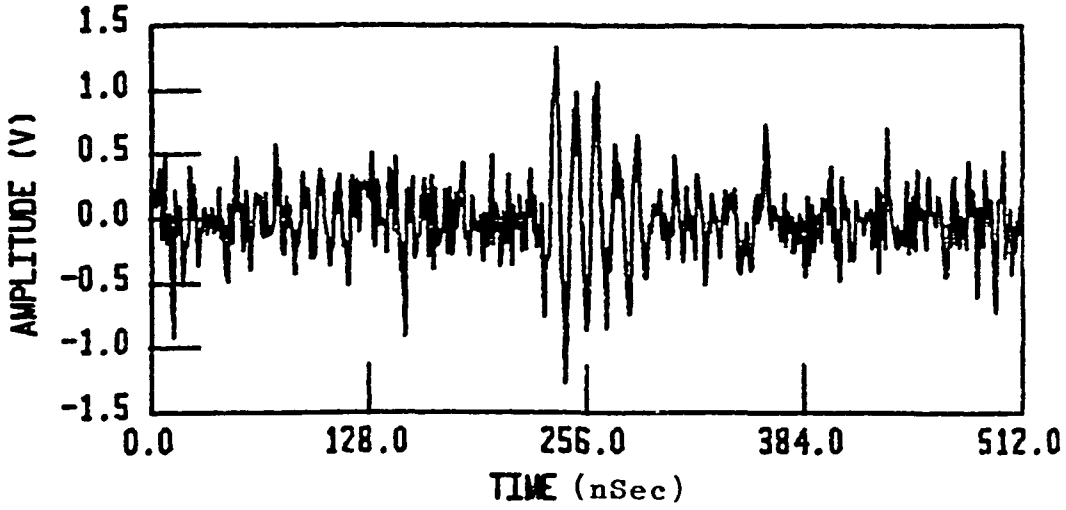


(B)

Figure 5.10. Wiener filtering (LFIL=3) of noisy pulse-echo for #P17 transducer
(A) noisy pulse-echo waveform
(B) filtered waveform

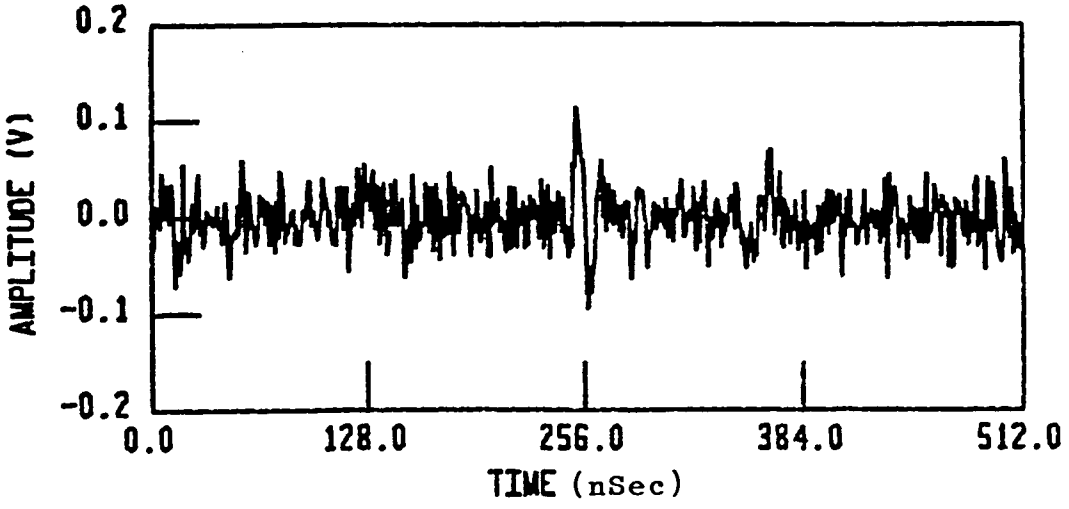


(A)

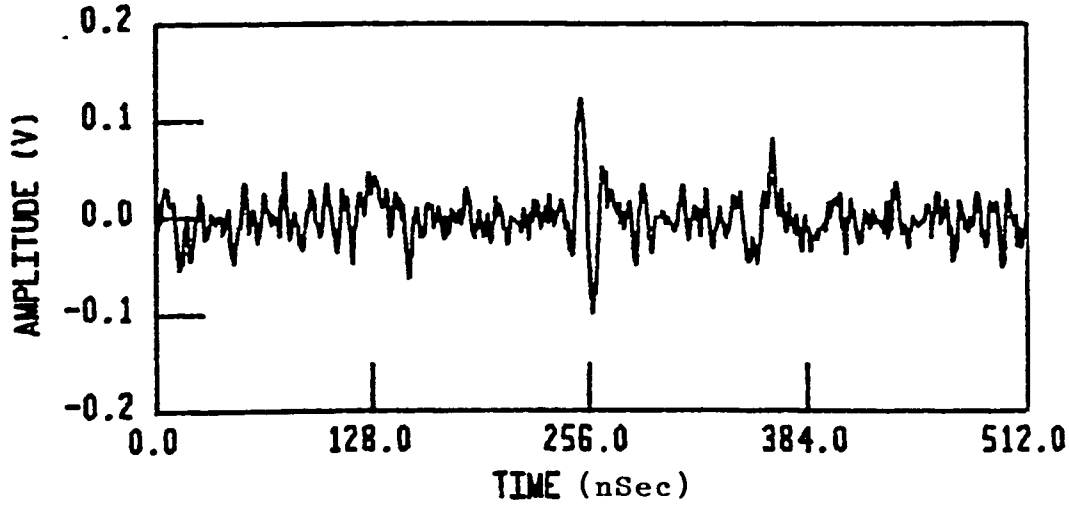


(B)

Figure 5.11. Wiener filtering (LFIL=7) of noisy pulse-echo for #P17 transducer
(A) noisy pulse-echo waveform
(B) filtered waveform

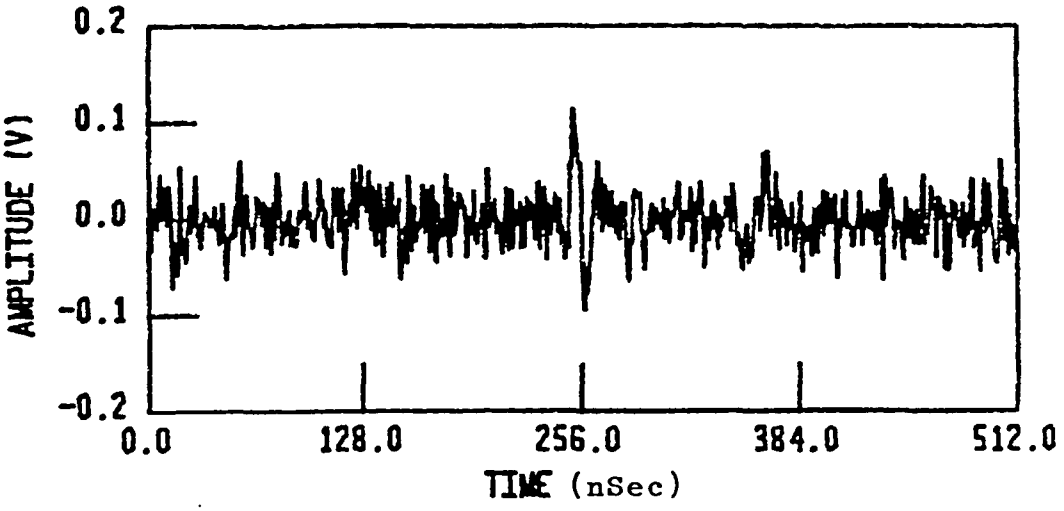


(A)

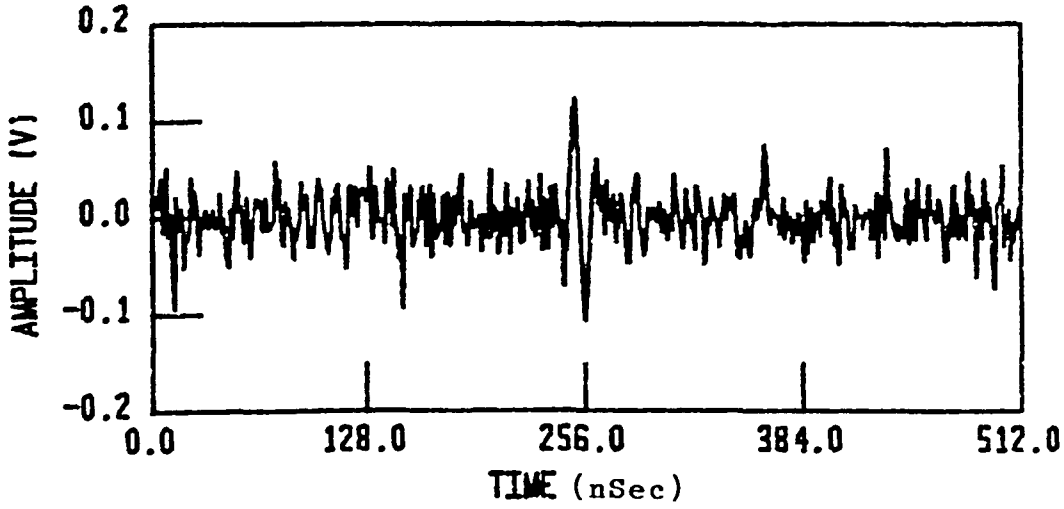


(B)

Figure 5.12. Wiener filtering (LFIL=3) of noisy pulse-echo for #P20 transducer
(A) noisy pulse-echo waveform
(B) filtered waveform



(A)



(B)

Figure 5.13. Wiener filtering (LFIL=7) of noisy pulse-echo for #P20 transducer
(A) noisy pulse-echo waveform
(B) filtered waveform

reflections as shown in the previous section.

By examining Table 5.2, it is well noticed that the M.S.E. in each of the four tests has reached a minimum (as expected) but then started to increase as a function of filter length. It has been determined (Koopmans, 1974) that this is a normal behavior of the Wiener filter. The reason for that behavior is not very clear; however, one interpretation can be proposed; rounding-off errors when more filter coefficients are computed by the long division process discussed earlier, is probably the cause of this discrepancy. At any rate, one and only one optimal filter length exists that leads to a minimum M.S.E. When larger filter lengths (>8) were tested, it was noticed that fluctuations in the M.S.E. occurred all along, but never reached again the minimum value provided by the optimal length.

One other observation worth mentioning is the rms noise level and its effect on the M.S.E. As an increasing amount of noise (higher rms) is added to the true pulse-echo, the M.S.E. appears to increase; and as one might expect, the overall restored signal gets very noisy. In fact, as the dynamic range of the noisy pulse-echo decreases, the 'reconstructed' signal becomes essentially useless since the noise components, after amplification, are also amplified by this operation causing a monotonic deterioration of the filter's output.

For the given situation, though, it is seen form Figures 5.6-5.13 that the Wiener filter has definitely improved the signals of the noisy pulse-echo reflections. Table 5.3 summarizes the content of

those figures by showing the dynamic range of the restored signals for the optimal and the non-optimal filter lengths. These values can be compared to their corresponding counterpart D.R. values given by Table 5.1 before the filtering took place.

One other note about the filter's length and its relation to the M.S.E. By looking at the overall results in Table 5.2, one can notice that, in the narrow-band response (i.e., #P10 and #P13) situations, a longer filter length was required to achieve the minimum M.S.E. as opposed to a shorter one in the broad-band response case. More work and consequently more computation time were necessary to smooth the first two noisy pulse-echoes than to perform the same operation on the ones belonging to #P17 and #P20 transducers. On the other hand, it has been observed that as the rms noise level changed (increasingly or decreasingly), the length of the Wiener filter required for best performance (i.e., minimum M.S.E.) also changed accordingly. For example, in testing the pulse-echo for #P20 transducer, the noise rms level was slightly decreased to 0.01. A filter length of only 2 was required to reach an even lower M.S.E. of 8.089 E-05 as compared to 4.371 E-04 for the larger rms level of 0.025. This criterion leads us to believe that low M.S.E. is indeed a function of low rms noise level as well, or in a more familiar term, high signal-to-noise ratio.

As a conclusive remark about the application of the Wiener filter to ultrasonic pulse-echo reflections, it is a well established fact that,

Table 5.3. Dynamic range of the restored signals for the optimal and the non-optimal filter lengths shown in Figures 5.6-5.13

Transducer	Filter Length	D.R.
#P10	5	11.31512
	7	5.65473
#P13	5	15.45008
	7	4.95239
#P17	3	7.95490
	7	5.79924
#P20	3	7.04860
	7	5.23619

with the presence of noise, this method could still perform properly as far as true signal feature enhancement is concerned, as long as the overall shape of the true image is not completely unrecognized. If the signal-to-noise ratio is low, the 'reconstruction' capability will be degraded because the noise components can be amplified during that operation.

In this thesis, the Wiener filter technique was considered for improvement of noisy ultrasonic pulse-echo reflections produced by narrow-band and broad-band transducers. Although more testing of the technique is required to determine the full extent of the technique's ability to improve ultrasonic images under a variety of conditions, the results obtained in this research work indicate that Wiener filtering is a useful technique for increasing the recognition of wanted details contained in ultrasonic images and, thus, may facilitate any kind of ultrasonic image understanding in mostly all fields utilizing this tool.

It is worthwhile to continue developing this approach, perhaps in the future on real-life experiments using real-time digital image processing. One application that might be a good laboratory experiment would involve a subjective evaluation of beef grading, since developments in that field were accomplished with the use of ultrasound to determine texture differentiation of marbling in beef.

VI. REFERENCES

Ahmed, N., and T. Natarajan. 1983. Discrete-time signals and systems. Reston Publishing Company, Inc., Reston, VA.

Box, G. E. P., and G. M. Jenkins. 1976. Time Series Analysis: Forecasting and Control. Holden-Day, Oakland, CA.

Bracewell, R. N. 1978. The Fourier transform and its application. McGraw-Hill, New York, NY.

Brown, L. F. 1986. Design of wideband ultrasound instrumentation for tissue characterization. M.S. thesis. Iowa State University, Ames, IA.

Brown, L. F. 1988. Electrochemical modeling, performance testing, and design of piezoelectric polymer film ultrasound transducers. Ph.D. dissertation, Iowa State University, Ames, Iowa.

Brown, R. G. 1983. Introduction to random signal analysis and Kalman Filtering. John Wiley & Sons, New York, NY.

Cobbold, R. S. C. 1974. Transducers for Biomedical Measurements: Principles and Applications. John Wiley & Sons, New York, NY.

Davenport, W. B., and W. C. Root. 1958. An introduction to the theory of random signals and noise. McGraw-Hill, New York, NY.

Dudgeon, D. E., and R. M. Mersereau. 1984. Multidimensional digital signal processing. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Ekstrom, M. P. 1982. Realizable Wiener Filtering in two dimensions. IEEE Trans. Acoust., Speech, Signal Processing, ASSP-30: 31-40.

Fink, M. A., and J. F. Cardoso. 1984. Diffraction effects in pulse-echo measurement. IEEE Trans. Sonics Ultrasonics, SU-31:313-329.

Fry, F. J. 1978. Ultrasound: Its application in medicine and Biology. Elsevier Scientific Publishing Company, New York, NY.

Gabor, D. 1965. The smoothing and filtering in two-dimensional images. Progress in Biocyb., 2:2-9.

Gammell, P. M. 1980. Beef Grading by Ultrasound. NASA Tech Brief, Vol. 5, No. 4.

Ghorayeb, S. R. 1986. Applying Wiener Filter theory to nonuniformly sampled noisy data. M.S. thesis, Iowa State University, Ames, IA.

Gull, S. F., and E. Daniell. 1978. Image reconstruction from noisy and incomplete data. Nature 272:686-690.

Haumschild, D. J. 1981. An Ultrasonic Bragg Scattering Technique for the Quantitative Characterization of Marbling in Beef. Ph.D. dissertation, Iowa State University, Ames, IA.

Haumschild, D. J., and D. L. Carlson. 1983. An Ultrasonic Bragg Scattering Technique for the Quantitative Characterization of Marbling in Beef. Ultrasonics, 21, No. 5:226-233.

Havlice, J. F., and J. C. Taenzer. 1979. Medical Ultrasonic Imaging: An Overview of Principles and Instrumentation. Proc. IEEE 67:620-641.

Hay, G. A. 1976. Medical Images: Formation, Perception and Measurement. The Institute of Physics and John Wiley & Sons, Great Britain.

Helstrom, C. W. 1967. Image restoration by the method of least squares. J. Opt. Soc. Am., 57:297-304.

Hill, C. R., D. Nicholas, and J. C. Bamber. 1976. Backscattering Analysis and Ultrasonic Imaging. John Wiley & Sons, Great Britain.

Hostetter, G. H. 1987. Recursive estimation. Handbook of digital signal processing. Academic Press, New York, NY.

Hunt, J. W., M. Arditi, and F. S. Foster. 1983. Ultrasound Transducers for Pulse-Echo Medical Imaging. IEEE Trans. on Biomedical Engineering, BME-30, No. 8:453-481.

Koch, I. 1982. Ultrasonic Backscattering from Non-Uniform Scatterers. Ultrasonic Imaging, 4, No. 2:140-162.

Koopmans, L. H. 1974. The spectral analysis of time series. Academic Press, New York, NY.

Kuc, R. B. 1979. Application of Kalman filtering techniques to diagnostic ultrasound. Ultrasonic Imaging, 1, No. 2:105-120.

Lu, F., and G. L. Wise. 1984. A simple approximation for minimum mean-square error symmetric uniform quantization. IEEE Trans. on Communications, COM-32, No. 4:470-472.

Macovski, A. 1983. Medical Imaging Systems. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Neal, S., and D. O. Thompson. 1982. An examination of the application of Wiener filtering to ultrasonic scattering amplitude estimation. Unpublished paper. Ames Laboratory, Iowa State University.

Orfanidis, S. J. 1985. Optimum Signal Processing. Macmillan Publishing Company, New York, NY.

Papoulis, A. 1977. Signal Analysis. McGraw-Hill Book Company, New York, NY.

Pratt, W. K. 1978. Digital image processing. Wiley, New York, NY.

Robinson, E. A. 1967. Multichannel Time Series Analysis. 2nd Edition. Goose Pond Press, Houston, TX.

Robinson, E. A. 1981. Time Series Analysis and Applications. Goose Pond Press, Houston, TX.

Srinatu, M. D., and P. K. Rajasekaran. 1979. An introduction to statistical signal processing with applications. John Wiley & Sons, New York, NY.

Wainstein, L. A., and V. D. Zubakov. 1962. Extraction of signals from noise. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Wells, P. N. T. 1969. Physical Principles of Ultrasonic Diagnosis. Academic Press, New York, NY.

Whalen, A. D. 1971. Detection of signals in noise. Academic Press, New York, NY.

Wiener, N. 1949. Extrapolation, interpolation, and smoothing of stationary time series. Wiley, New York, NY.

Yasuhara, M., and Y. Yasumoto. 1984. An improved adaptive predictor in DPCM based on the Kalman filter and its application to handwriting signal encoding. IEEE Trans. on Communications, COM-32, No. 4:484-488.

Zheng, Y., and J. P. Basart. 1986. Enhancing synthetic radio images by a bank of adaptive regional Kalman filters. Abstracted in Bul. of Am. Astr. Soc. 168th meeting, 18, No. 2:701.

VII. ACKNOWLEDGEMENTS

I would like to express a special thank you to my major professor Dr. David Carlson whose expertise in ultrasonic systems and continuous advise in that field have led to the selection of this research topic.

I would like to thank Dr. William Brockman and once again Dr. Robert Lambert for their enthusiasm and their interest to serve on my graduate committee.

I am deeply grateful for the Electrical Engineering Department for its generous support in providing me with the Teaching Assistantship throughout the continuance of this second master's degree.

I would like to express my sincere gratitude to Mark Mehl who spent hours helping me allocate enough computer memory (on the PDP-11) for my programming package to run successfully.

And last, but not least, a very special word to the four people who once again stood patiently behind me, supported me and provided me with continuous encouragement throughout my educational career. My father, Riad, my mother, Sonia, my sister, Ghada, and my brother, Samer; I thank you for being so close to me even though you were thousands of miles away.

VIII. APPENDIX A: THE MAIN PROGRAM

```
      PROGRAM PULSE
C*************************************************************************
C*                                                                       *
C*      This is the main program that creates or reads in a simu-        *
C*      lated pulse-echo waveform. The main feature of this program      *
C*      is to apply the Wiener filter technique to the noisy pulse-      *
C*      echo; that is, after the actual pulse-echo has been corrupted    *
C*      by an additive Gaussian white noise. The procedure followed      *
C*      in order to create the optimal Wiener filter and to restore      *
C*      the true pulse-echo signal can be summarized as follows:         *
C*                                                                       *
C*      (1) Take the autocorrelation of the noisy pulse-echo             *
C*          with itself.                                                 *
C*                                                                       *
C*      (2) Take the cross-correlation of the noisy pulse-echo           *
C*          with the actual one.                                         *
C*                                                                       *
C*      (3) Use the results from steps (1) & (2) to determine            *
C*          the optimal Wiener filter via a long division mani-          *
C*          pulation as given by equation (4.13) in chapter IV.          *
C*                                                                       *
C*      (4) Convolve the computed optimal filter with the noisy          *
C*          pulse-echo to get the restored signal.                       *
C*                                                                       *
C*      (5) Finally, compute the normalized Mean-Square-Error            *
C*          for the computed filter of the true and the restored         *
C*          pulse-echos.                                                 *
C*                                                                       *
C*************************************************************************
      VIRTUAL XMEAS(1024),TRUE(1024),AUTO(1024),CROIX(1024)
      VIRTUAL XNOISE(1024),TIMAG(1024)
      VIRTUAL ERR(1024),FLTR(1024),ESTIM(2*1024-1),T(1024)

999   TYPE *,' '
      TYPE *,' THE MENU OF THIS PROGRAM IS:'
      TYPE *,' '
      TYPE *,' 1. CREATE A RECTANGULAR PULSE, ADD NOISE TO IT,'
      TYPE *,'    THEN WIENER FILTER IT.'
      TYPE *,' '
      TYPE *,' 2. READ A PULSE-ECHO RESPONSE PRODUCED BY ONE OF FOUR'
      TYPE *,'    TRANSDUCERS, ADD NOISE TO THE RESPONSE, THEN WIENER'
      TYPE *,'    IT.'

      TYPE *,' '
      TYPE *,' WHICH MENU WOULD YOU LIKE? (1 OR 2)'
      READ(5,*) MENU

      IF (MENU.EQ.1) GOTO 887
      IF (MENU.EQ.2) GOTO 888

887   TYPE *,' '
      TYPE *,' THIS THE RECTANGULAR PULSE GENERATING MENU...'
      TYPE *,' '
      TYPE *,' WHAT IS THE MAGNITUDE OF THE PULSE? (REAL #)'
      READ(5,*) XMAG
      TYPE *,' HOW MANY SAMPLE POINTS ARE THERE? (POWER OF 2)'
      READ(5,*) NPTS
      TYPE *,' WHAT IS THE DUTY CYCLE? (0.0,...,1.0)'
      READ(5,*) XDUTY
```

```
      GOTO 889
888   TYPE *,' '
      TYPE *,' THIS IS THE PULSE-ECHO SIMULATION MENU...'
      TYPE *,' '
      TYPE *,' THE PULSE-ECHO THAT WILL BE READ WAS GENERATED BY ONE'
      TYPE *,' OF 4 TRANSDUCERS (NARROW-BAND OR BROAD-BAND)'
      TYPE *,' '
      TYPE *,' PLEASE SELECT ONE OF THE FOLLOWING XDCRS: (10,13,17 OR 20)'
      TYPE *,' '
      TYPE *,' #P10 (narrow-band)'
      TYPE *,' #P13 (narrow-band)'
      TYPE *,' #P17 (broad-band)'
      TYPE *,' #P20 (broad-band)'
      READ(5,*) IXDCR
      TYPE *,' '
      TYPE *,' PLEASE ENTER THE TOTAL NUMBER OF POINTS: (POWER OF 2)'
      READ(5,*) NPTS
889   TYPE *,' '
      TYPE *,'*****************************************************************'
      TYPE *,' '
      TYPE *,'        THIS IS THE GAUSSIAN NOISE GENERATING PROGRAM'
      TYPE *,' '
      TYPE *,'*****************************************************************'
      TYPE *,' '
      TYPE *,' PLEASE ENTER A REAL NUMBER FOR THE SEED:'
      READ(5,*) SEED
      TYPE *,' PLEASE ENTER THE DESIRED STANDARD DEVIATION:'
      READ(5,*) STDEV
      TYPE *,' PLEASE ENTER THE FACTOR YOU WOULD LIKE TO DIVIDE'
      TYPE *,' THE NOISE MAGNITUDE BY:'
      READ(5,*) FACTOR
      TYPE *,' '
      TYPE *,'*****************************************************************'
      TYPE *,' '
      TYPE *,'    THE OUTPUT WILL CONTAIN ZERO-MEAN GAUSSIAN NOISE'
      TYPE *,' '
      TYPE *,'*****************************************************************'
C.....................Generate the noise and read it...................

      TYPE *,' '
      TYPE *,' I AM IN GNOISE'
      CALL GNOISE(SEED,NPTS,STDEV)
      TYPE *,' I AM OUT OF GNOISE'
      OPEN(UNIT=1,TYPE='OLD',NAME='VM:GNOISE.TMP')
      DO 4 I = 1 , NPTS
         READ(1,*) XNOISE(I)
         XNOISE(I) = XNOISE(I) / FACTOR
4     CONTINUE
```

```
        CLOSE(UNIT=1)
C.....................end of noise generation.....................
        LNX = INT ((ALOG (FLOAT(NPTS))) / (ALOG (2.0)))
        IF (MENU.EQ.1) GOTO 110
        IF (MENU.EQ.2) GOTO 111
C.....................generate the rectangular pulse.................
110     TYPE *,' '
        TYPE *,' I AM CREATING THE RECTANGULAR PULSE AND ADDING NOISE TO IT'
        TYPE *,' '
        TYPE *,' .....................PATIENCE !!!.....................'
        TYPE *,' '
        XRISE = (NPTS - NPTS * XDUTY) / 2
        XFALL = XRISE + NPTS * XDUTY
        DO 6 I = 1 , NPTS
           T(I) = 1. * I
           IF ((I.GE.XRISE).AND.(I.LE.XFALL)) GOTO 100
           GOTO 200
100        TRUE(I)  = XMAG
           XMEAS(I) = TRUE(I) + XNOISE(I)
           GOTO 300
200        TRUE(I)  = 0.0
           XMEAS(I) = XNOISE(I)
300        TIMAG(I) = 0.0
6       CONTINUE
        GOTO 117
C.....................end of rectangular pulse generation.................
C.....................begin reading pulse-echo data.....................
111     TYPE *,' I AM READING THE PULSE-ECHO AND ADDING NOISE TO IT...'
        TYPE *,' '
        TYPE *,' ....................PATIENCE !!!.....................'
        TYPE *,' '
        IF (IXDCR.EQ.10) GOTO 112
        IF (IXDCR.EQ.13) GOTO 114
        IF (IXDCR.EQ.17) GOTO 113
        IF (IXDCR.EQ.20) GOTO 115
112     OPEN(UNIT=1,TYPE='OLD',NAME='P10.TIM')
        GOTO 116
113     OPEN(UNIT=1,TYPE='OLD',NAME='P13.TIM')
        GOTO 116
114     OPEN(UNIT=1,TYPE='OLD',NAME='P15.TIM')
```

```
        GOTO 116
115     OPEN(UNIT=1,TYPE='OLD',NAME='BP20.TIM')
116     DO 7 I = 1 , NPTS
           T(I) = 1. * I
           READ(1,*) TRUE(I)
           XMEAS(I) = TRUE(I) + XNOISE(I)
           TIMAG(I) = 0.0
7       CONTINUE
        CLOSE(UNIT=1)
C.....................end of data reading.....................
117     TYPE *,' '
        TYPE *,'************************************************************'
        TYPE *,' '
        TYPE *,'              ONE LAST SET OF QUESTIONS...'
        TYPE *,' '
        TYPE *,'************************************************************'
        TYPE *,' '
        TYPE *,' PLEASE ENTER THE LENGTH OF THE AUTOCORRELATION:'
        READ(5,*) LAUTO
        TYPE *,' '
        TYPE *,' NOW ENTER THE LENGTH OF THE CROSS-CORRELATION:'
        READ(5,*) LCROSS
        TYPE *,' '
        TYPE *,' AND FINALLY... PLEASE ENTER THE LENGTH OF THE WIENER FILTER:'
        TYPE *,' NOTE THAT LENGTH OF FILTER <= LENGTH OF CORRELATIONS'
        READ(5,*) LFIL
        TYPE *,' '
        TYPE *,'************************************************************'
        TYPE *,' '
        TYPE *,'          THANK YOU FOR YOUR COOPERATION !!!!'
        TYPE *,' '
        TYPE *,'************************************************************'
        TYPE *,' '
C.................find auto- and cross-correlations...................

        TYPE *,' I AM COMPUTING THE AUTOCORRELATION'
        CALL CROSS(NPTS,XMEAS,NPTS,XMEAS,LAUTO,AUTO)
        TYPE *,' I AM COMPUTING THE CROSS-CORRELATION'
        CALL CROSS(NPTS,XMEAS,NPTS,TRUE,LCROSS,CROIX)

C.....................find the WIENER filter.....................
        TYPE *,' I AM COMPUTING THE WIENER FILTER'
        CALL WIENER(LFIL,AUTO,CROIX,FLTR,ERR)
```

```
C.........convolution of WIENER filter with measured signal.............
        TYPE *,' I AM CONVOLVING THE FILTER WITH THE NOISY PULSE-ECHO'
        TYPE *,' '
        TYPE *,' ....................PATIENCE !!!....................'
        CALL CONV(NPTS,XMEAS,LFIL,FLTR,ESTIM)
C       DO 898 I = 1 , NPTS+LFIL-1
C898    ESTIM(I) = AMP * ESTIM(I)
C.........find the mean-square-error between estimated signal............
C                       and the true signal

        TYPE *,' I AM COMPUTING THE MEAN-SQUARE-ERROR'
        CALL MSE(NPTS,TRUE,ESTIM,XMSE)
        TYPE *,' '
        TYPE *,' .....................SUCCESSFUL TEST !!!.................'
        TYPE *,' '
        TYPE *,' MEAN-SQUARE-ERROR = ',XMSE
C....compute the dynamic range of the noisy and the restored signals....
        TYPE *,' '
        TYPE *,' I AM COMPUTING THE DYNAMIC RANGE OF THE NOISY SIGNAL'
        TYPE *,' '
        CALL PEAK(NPTS,TRUE,XPEAK)
        CALL RMS(NPTS,XNOISE,XRMS)
        TYPE *,' THE LARGEST VALUE IN THE NOISY SIGNAL IS =',XPEAK
        TYPE *,' THE RMS OF THE BACKGROUND NOISE IS =',XRMS
        DR = XPEAK / XRMS
        TYPE *,' '
        TYPE *,' DYNAMIC RANGE OF THE NOISY SIGNAL = ',DR
        TYPE *,' '
        TYPE *,' I AM COMPUTING THE DYNAMIC RANGE OF THE RESTORED SIGNAL'
        TYPE *,' '
        CALL PEAK(NPTS,ESTIM,XXPEAK)
        CALL RMS(NPTS,ESTIM,XXRMS)
        TYPE *,' THE LARGEST VALUE IN THE RESTORED SIGNAL IS =',XXPEAK
        TYPE *,' THE RMS OF THE BACKGROUND NOISE IS =',XXRMS
```

```
      DR = XXPEAK / XXRMS
      TYPE *,' '
      TYPE *,' DYNAMIC RANGE OF THE RESTORED SIGNAL = ',DR
C.....check if restored signal needs any amplification.....
      AMP = XPEAK / XXPEAK
      IF (AMP.GT.1.0) GOTO 210
      TYPE *,' '
      TYPE *,' THE RESTORED SIGNAL DOES NOT HAVE TO BE AMPLIFIED !!!'
      GOTO 340
210   TYPE *,' '
      TYPE *,' THE RESTORED SIGNAL MUST BE AMPLIFIED BY ',AMP
      TYPE *,' '
      TYPE *,' WOULD YOU LIKE TO DO SO ? (1 -- YES)'
      READ(5,*) IAMP
      IF (IAMP.EQ.1) GOTO 118
      GOTO 340
118   DO 898 I = 1 , NPTS+LFIL-1
898   ESTIM(I) = AMP * ESTIM(I)
C.....find the new mean-square-error.....
      TYPE *,' I AM COMPUTING THE NEW MEAN-SUARE-ERROR'
      CALL MSE(NPTS,TRUE,ESTIM,XXMSE)
      TYPE *,' '
      TYPE *,' NEW M.S.E. AFTER AMPLIFICATION = ',XXMSE
C.....compute the new dynamic range of the restored signal.....
      CALL PEAK(NPTS,ESTIM,XXXPEAK)
      CALL RMS(NPTS,ESTIM,XXXRMS)
      TYPE *,' '
      TYPE *,' THE PEAK VALUE OF THE AMPLIFIED SIGNAL IS =',XXXPEAK
      TYPE *,' THE RMS OF THE BACKGROUND NOISE IS =',XXXRMS
      DR = XXXPEAK / XXXRMS
      TYPE *,' '
```

```
      TYPE *,' DYNAMIC RANGE OF THE AMPLIFIED SIGNAL =',DR
C.....Plot the NOISY and the RESTORED signal.....
340   TYPE *,' '
      TYPE *,' WOULD YOU LIKE TO PLOT THE NOISY AND RESTORED SIGNALS ?'
      TYPE *,' (1 -- YES)'
      READ(5,*) IPLOT
      IF (IPLOT.EQ.1) GOTO 350
      GOTO 600
350   IF (MENU.EQ.1) GOTO 400
      IF (MENU.EQ.2) GOTO 500
400   CALL PLOT2(NPTS,T,XMEAS,ESTIM)
      GOTO 600
500   CALL PLOT1(IXDCR,T,XMEAS,ESTIM)
600   TYPE *,' '
      TYPE *,' WOULD YOU LIKE TO PERFORM ANOTHER TEST? (1--Y)'
      READ(5,*) IANS
      IF (IANS.EQ.1) GOTO 999
      STOP
      END
```

IX. APPENDIX B: FFT SUBROUTINE

```
      SUBROUTINE FFT1(X,Y,TABLE,M,LL,ISN)
C
C
C     FFT is IN-PLACE DFT computation using SANDE ALGORITHM
C     and MARKEL PRUNING modification.
C
C     X is an array of length 2**M used to hold REAL part of
C          COMPLEX input.
C     Y is an array of length 2**M used to hold IMAGINARY part of
C          COMPLEX input.
C     TABLE is an array of length (N/4)+1, where N=2**M. TABLE
C          contains QUARTER-LENGTH cosine table.
C     M = integer. Size of FFT to be performed is given by
C          N=2**M.
C
C     (Note that the bit reverse table is set for a maximum of
C     N=2**12=4096)
C
C     LL = integer. There are 2**LL actual data points.
C     ISN is either -1 or 1. Set ISN to -1 for FORWARD DFT and
C          set ISN to 1 for INVERSE DFT.
C
C
      DIMENSION X(1096),Y(1096),TABLE(1025),L(12)
      EQUIVALENCE (L12,L(1)),(L11,L(2)),(L10,L(3)),
     $            (L9,L(4)),(L8,L(5)),(L7,L(6)),
     $            (L6,L(7)),(L5,L(8)),(L4,L(9)),
     $            (L3,L(10)),(L2,L(11)),(L1,L(12))

      N=2**M
      ND4=N/4
      ND4P1=N/4 + 1
      ND4P2=ND4P1 + 1
      ND2P2=ND4 + ND4P2
      LLL=2**LL

      DO 8 LO=1,M
         LMX=2**(M-LO)
         LMM=LMX
         LIX=2*LMX
         ISCL=N/LIX

C     Test for PRUNING

         IF(LO-M+LL) 1,2,2
1        LMM=LLL
2     DO 8 LM=1,LMM
         IARG=(LM-1)*ISCL+1
         IF(IARG.LE.ND4P1) GOTO 4
         K1=ND2P2-IARG
         C=-TABLE(K1)
         K3=IARG-ND4
         S=ISN*TABLE(K3)
         GOTO 6
4        C=TABLE(IARG)
         K2=ND4P2-IARG
         S=ISN*TABLE(K2)
6        CONTINUE
      DO 8 LI=LIX,N,LIX
         J1=LI-LIX+LM
```

```
            J2=J1+LMX
            T1=X(J1)-X(J2)
            T2=Y(J1)-Y(J2)
            X(J1)=X(J1)+X(J2)
            Y(J1)=Y(J1)+Y(J2)
            X(J2)=C*T1-S*T2
            Y(J2)=C*T2+S*T1
8         CONTINUE
C         Perform BIT REVERSAL
          DO 40 J=1,12
             L(J)=1
             IF(J-M) 31,31,40
31           L(J)=2**(M+1-J)
40        CONTINUE
          JI=1
          DO 60 J1=1,L1
          DO 60 J2=J1,L2,L1
          DO 60 J3=J2,L3,L2
          DO 60 J4=J3,L4,L3
          DO 60 J5=J4,L5,L4
          DO 60 J6=J5,L6,L5
          DO 60 J7=J6,L7,L6
          DO 60 J8=J7,L8,L7
          DO 60 J9=J8,L9,L8
          DO 60 J10=J9,L10,L9
          DO 60 J11=J10,L11,L10
          DO 60 JR=J11,L12,L11
             IF(JI-JR) 51,51,54
51           R=X(JI)
             X(JI)=X(JR)
             X(JR)=R
             FI=Y(JI)
             Y(JI)=Y(JR)
             Y(JR)=FI
54           IF(ISN) 53,53,52
52           X(JR)=X(JR)/FLOAT(N)
             Y(JR)=Y(JR)/FLOAT(N)
53           JI=JI+1
60        CONTINUE
          RETURN
          END
C*******************************************************************************
C*******************************************************************************
          SUBROUTINE COSQT(M,TABLE)
C         This subroutine generates QUARTER-LENGTH cosine table.
          DIMENSION TABLE(2)
          N=2**M
          ND4P1=N/4 + 1
          SCL=6.283185307/FLOAT(N)
          DO 10 I=1,ND4P1
             ARG=FLOAT(I-1)*SCL
10        TABLE(I)=COS(ARG)
          RETURN
          END
```

X. APPENDIX C: WIENER FILTER SUBROUTINES

```
      SUBROUTINE GNOISE(SEEDI,IK,STDEV)
C
C     GAUSSIAN WHITE NOISE PROGRAM
C
      VIRTUAL G1(1024),G2(1024)
      REAL SEEDI,STDEV
      INTEGER IK

      N    = IK
      PI   = 3.141592654
      BIG  = 1.E8
      M    = N/2
      VAR  = STDEV*STDEV
      SEED = SEEDI

      OPEN(UNIT=1,TYPE='NEW',NAME='VM:GNOISE.TMP')

      DO 10 I=1,M
        IF(I.EQ.1)GO TO 99
        IF(SEED.NE.SEEDI) GOTO 99
        SEED = SEED + I
99    DO 20 K=1,10
        S    = SEED
        SEED = AMOD(61.*S+7.,BIG)
        SCL1 = SEED / BIG
        S    = SEED
        SEED = AMOD(61.*S+7.,BIG)
        SCL2 = SEED / BIG
20    CONTINUE
        SCL1  = ABS(SCL1)
        RN    = SQRT(2.*VAR*ALOG(1./SCL1))
        G1(I) = RN * COS(2.*PI*SCL2)
        G2(I) = RN * SIN(2.*PI*SCL2)
        WRITE(1,*) G1(I)
        WRITE(1,*) G2(I)
10    CONTINUE
      CLOSE(UNIT=1)

      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE CROSS(LX,X,LY,Y,LG,G)

      VIRTUAL X(LX),Y(LY),G(LG)

      DO 1 J = 1 , LG

      LOW = MINO(LY,LX-J+1)

1     G(J) = DOT(LOW,J,X,Y)

      RETURN
      END
C*********************************************************************
C*********************************************************************
```

```
      FUNCTION DOT(L,J,X,Y)
C
CC    THE PURPOSE OF THIS SUBROUTINE IS TO COMPUTE THE DOT PRODUCT
CC    OF TWO VECTORS.
CC    THE SUBROUTINE INPUTS ARE:
CC        L = LENGTH OF X = LENGTH OF Y = N
CC        X = VECTOR OF LENGTH N
CC        Y = VECTOR OF LENGTH N
CC
CC    THE SUBROUTINE OUTPUT IS:
CC        P = THE DOT PRODUCT
C
      VIRTUAL X(1024),Y(1024)
      DOT = 0.0
      IF(L.LE.0) RETURN
      DO 1 IJ = J , L+J-1
1     DOT = DOT + X(IJ) * Y(IJ)
      RETURN
      END
C*************************************************************************
C*************************************************************************
      SUBROUTINE WIENER(LR,R,G,F,A)
C*************************************************************************
C*                                                                       *
C*    This program creates the Wiener filter via a long division         *
C*    procedure using the coefficients of the Z-transform of both        *
C*    the cross-correlation (numerator) and the autocorrelation          *
C*    (denominator) of the noisy pulse-echo with the actual one,         *
C*    and the noisy pulse-echo with itself) respectively.                *
C*                                                                       *
C*************************************************************************
      VIRTUAL R(LR),G(LR),F(LR),A(LR)
      V    = R(1)
      D    = R(2)
      A(1) = 1.
      F(1) = G(1) / V
      Q    = F(1) * R(2)
      IF (LR.EQ.1) RETURN
C.....................................................................
      DO 4 L = 2 , LR
      A(L) = -D / V
      IF (L.EQ.2) GOTO 2
      L1 = (L - 2) / 2
      L2 = L1 + 1
```

```
        IF (L2.LT.2) GOTO 5
C.......................................................................
        DO 1 J = 2 , L2
        HOLD = A(J)
        K    = L - J + 1
        A(J) = A(J) + A(L) * A(K)
1       A(K) = A(K) + A(L) * HOLD
5       IF (2*L1.EQ.L-2) GOTO 2
C.......................................................................
        A(L2+1) = A(L2+1) + A(L) * A(L2+1)
2       V    = V + A(L) * D
        F(L) = (G(L) - Q) / V
        L3   = L - 1
C.......................................................................
        DO 3 J = 1 , L3
        K = L - J + 1
3       F(J) = F(J) + F(L) * A(K)
C.......................................................................
        IF (L.EQ.LR) RETURN
        D = 0.0
        Q = 0.0
C.......................................................................
        DO 4 I = 1 , L
        K = L - I + 2
        D = D + A(I) * R(K)
4       Q = Q + F(I) * R(K)
C.......................................................................
        RETURN
        END
C***********************************************************************
C***********************************************************************
        SUBROUTINE CONV(LA,A,LB,B,C)
        VIRTUAL A(1024),B(1024),C(2*1024-1)
        LC = LA + LB - 1
        CALL ZERO(LC,C)
```

```
      DO 1 I = 1 , LA
      DO 1 J = 1 , LB
      K = I + J - 1
1     C(K) = C(K) + A(I) * B(J)
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE ZERO(LX,X)
C
C     THIS SUBROUTINE STORES THE FLOATING-POINT NUMBER ZERO, 0.0,
C     IN EACH STORAGE LOCATION OF AN ARRAY.
C
      VIRTUAL X(LX)
      IF(LX.LE.0) RETURN
      DO 1 I = 1 , LX
1     X(I) = 0.0
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE MSE(N,A,B,XMS)
C
C     THIS SUBROUTINE COMPUTES THE MEAN SQUARE ERROR BETWEEN
C     THE TRUE (W/O NOISE) SIGNAL AND THE RESTORED SIGNAL CREATED
C     BY THE 'OPTIMAL' WIENER FILTER.
C
C     THE SUBROUTINE INPUTS ARE:
C
C             A = TRUE SIGNAL WITHOUT NOISE = (A1,...,An)
C             B = RESTORED SIGNAL = (B1,...,Bn)
C             N = TOTAL NUMBER OF POINTS
C
C     THE SUBROUTINE OUTPUT IS:
C
C            MS = MEAN SQUARE ERROR
C
      VIRTUAL A(1024),B(2*1024-1)
      SUM = 0.0
      DO 1 I = 1 , N
      SUM = SUM + (A(I) - B(I)) * (A(I) - B(I))
1     CONTINUE
      XMS = SUM / N
      RETURN
```

```
        END
C*************************************************************************
C*************************************************************************
        SUBROUTINE PLOT1(IPLOT,T,XMEAS,ESTIM)
C*************************************************************************
C*                                                                       *
C*      This routine is used to plot the pulse-echo noisy data           *
C*      and the restored actual pulse-echo reflection using the          *
C*      HGRAPH software on the PDP-11.                                   *
C*                                                                       *
C*************************************************************************
        VIRTUAL XMEAS(1024),ESTIM(2*1024-1),T(1024)
        INTEGER IPLOT
        CALL INIPLT(99,6.5,9.)
        CALL FRAME(.5,6.5,4.5,9.)
        CALL WINDOW(1.,5.5,2.,4.)
C.....................Plot the noisy signal...................
        IF (IPLOT.EQ.10) GOTO 10
        IF (IPLOT.EQ.13) GOTO 13
        IF (IPLOT.EQ.17) GOTO 17
        IF (IPLOT.EQ.20) GOTO 20
10      CALL SCALE(0.,512.,-3.,3.)
        CALL AXIS(128.,1.,'TIME (uSec)',11,1,1,'AMPLITUDE (V)',13,1,1)
        GOTO 100
13      CALL SCALE(0.,512.,-2.,2.)
        CALL AXIS(128.,1.,'TIME (uSec)',11,1,1,'AMPLITUDE (V)',13,1,1)
        GOTO 100
17      CALL SCALE(0.,512.,-2.,2.)
        CALL AXIS(128.,1.,'TIME (uSec)',11,1,1,'AMPLITUDE (V)',13,1,1)
        GOTO 100
20      CALL SCALE(0.,512.,-.2,.2)
        CALL AXIS(128.,.1,'TIME (uSec)',11,1,1,'AMPLITUDE (V)',13,1,1)
100     CALL VDASHLN(T,XMEAS,512,0,0,1,0,0)
C.....................Plot the restored signal.................
        CALL FRAME(.5,6.5,.9,5.)
        IF (IPLOT.EQ.10) GOTO 1
        IF (IPLOT.EQ.13) GOTO 2
        IF (IPLOT.EQ.17) GOTO 3
        IF (IPLOT.EQ.20) GOTO 4
1       CALL SCALE(0.,512.,-2.,2.)
        CALL AXIS(128.,1.,'TIME (uSec)',11,1,1,'AMPLITUDE (V)',13,1,1)
        GOTO 200
2       CALL SCALE(0.,512.,-1.,1.)
        CALL AXIS(128.,.5,'TIME (uSec)',11,1,1,'AMPLITUDE (V)',13,1,1)
```

```
        GOTO 200
3       CALL SCALE(0.,512.,-1.5,1.5)
        CALL AXIS(128.,.5,'TIME (uSec)',11,1,1,'AMPLITUDE (V)',13,1,1)
        GOTO 200
4       CALL SCALE(0.,512.,-.2,.2)
        CALL AXIS(128.,.1,'TIME (uSec)',11,1,1,'AMPLITUDE (V)',13,1,1)
200     CALL VDASHLN(T,ESTIM,512,0,0,1,0,0)

        CALL ENDPLT

        RETURN
        END
C*************************************************************************
C*************************************************************************

        SUBROUTINE PLOT2(NPTS,T,XMEAS,ESTIM)

C*************************************************************************
C*                                                                       *
C*      This routine is used to plot the data of the measured            *
C*      noisy rectangular pulse and the restored true pulse              *
C*      using the HGRAPH software on the PDP-11.                         *
C*                                                                       *
C*************************************************************************
        VIRTUAL XMEAS(1024),ESTIM(2*1024-1),T(1024)
        INTEGER NPTS

        XN   = 1. * NPTS
        XTIC = XN / 4.

        CALL INIPLT(99,6.5,9.)
        CALL FRAME(.5,6.5,4.5,9.)
        CALL WINDOW(1.,5.5,2.,4.)

C.....................Plot the noisy signal...................

        CALL SCALE(0.,XN,-3.,3.)
        CALL AXIS(XTIC,1.,'TIME (uSec)',11,1,1,'AMPLITUDE (V)',13,1,1)

        CALL VDASHLN(T,XMEAS,NPTS,0,0,1,0,0)

C.....................Plot the restored signal.................

        CALL FRAME(.5,6.5,.9,5.)

        CALL SCALE(0.,XN,-2.,2.)
        CALL AXIS(XTIC,1.,'TIME (uSec)',11,1,1,'AMPLITUDE (V)',13,1,1)

        CALL VDASHLN(T,ESTIM,NPTS,0,0,1,0,0)

        CALL ENDPLT

        RETURN
        END
```

```
C*************************************************************************
      SUBROUTINE PEAK(NPTS,XFUNC,PK)
C*************************************************************************
C*                                                                       *
C*      This subroutine finds the absolute magnitude of the largest      *
C*      peak stored in the array XFUNC. The result will then be sto-     *
C*      red in the variable PK (abbreviation for PEAK).                  *
C*                                                                       *
C*************************************************************************
      VIRTUAL XFUNC(1024)
      REAL PK
      PK = ABS(XFUNC(1))
      DO 1 I = 2 , NPTS
         IF (ABS(XFUNC(I)) .GT. ABS(PK)) GOTO 2
         GOTO 1
2        PK = ABS(XFUNC(I))
1     CONTINUE
      RETURN
      END
C*************************************************************************
C*************************************************************************
      SUBROUTINE RMS(NPTS,XFUNC,ROOT)
C*************************************************************************
C*                                                                       *
C*      This subroutine finds the RMS noise level in the array XFUNC.    *
C*                                                                       *
C*************************************************************************
      VIRTUAL XFUNC(1024)
      REAL ROOT
      TOT = 0.0
      XMS = 0.0
      DO 1 I = 384 , NPTS
         TOT = TOT + XFUNC(I)
1     CONTINUE
      XMEAN = TOT / FLOAT(NPTS/4)
      DO 2 I = 384 , NPTS
         XMS = XMS + (XFUNC(I) - XMEAN) * (XFUNC(I) - XMEAN)
2     CONTINUE
      ROOT = SQRT(XMS / FLOAT(NPTS/4))
      RETURN
      END
```