NUSC Technical Report 7203
19 June 1984

Analysis of Time Delay Estimator Performance

Kent N. Scarbrough
Surface Ship Sonar Department

Naval Underwater Systems Center
Newport, Rhode Island / New London, Connecticut

Preface

This study was prepared as a dissertation in partial fulfillment of the requirements for the degree Doctor of Philosophy in Electrical Engineering at Kansas State University. Portions of this work were performed under NUSC Project No. 733PBO, "Mathematical Model of Target Localization Error Due to Uncertainties in Sensor Position," Principal Investigator, Kent Scarbrough (funded by the Technical Director's New Professional Bid & Proposal Program), and NUSC Project No. B12611, "Inter-Array Processing," Principal Investigator, Dr. J. P. Beam, NUSC Program Manager, Dr. G. C. Carter, Program Manager, R. T. Cockerill, NAVSEA-63D.

The author wishes to acknowledge the contributions of the many individuals who supported the completion of this research. The author owes the following groups and individuals a special thanks:

From Kansas State University - the Department of Electrical Engineering for its financial support; Dr. W. D. Curtis of the Department of Mathematics and Dr. D. R. Hummels of the Department of Electrical Engineering for serving on my graduate committee and for their advice and assistance; Dr. N. Ahmed, now with the Department of Electrical Engineering and Computer Engineering at the University of New Mexico, for serving as my major professor and for his invaluable support and guidance throughout my graduate studies.

From the Naval Underwater Systems Center - Dr. J. Beam for his general interest and support of this research; Dr. J. Ianniello for several valuable technical discussions, especially on the material of Section 3.4; Dr. G. C. Carter, for serving on my graduate committee and for his significant time, effort, technical guidance and constant encouragement during this research effort.

Reviewed and Approved: 19 June 1984

L. Freeman

L. Freeman
Head, Surface Ship Sonar Department

The author of this report is located at the
New London Laboratory, Naval Underwater Systems Center,
New London, Connecticut 06320.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR 7203	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* ANALYSIS OF TIME DELAY ESTIMATOR PERFORMANCE		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Kent N. Scarbrough		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Underwater Systems Center New London Laboratory New London, Connecticut 06320		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 733P80 B12611
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 19 June 1984
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Coherent Processing
Correlator Performance Estimate
Cramer-Rao Lower Bound
Doppler Compensation
Incoherent Processing
Performance Bounds
Time Delay Estimation
Ziv-Zakai Lower Bound

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Time delay estimation (TDE) is an area of active research with applications in a wide variety of fields. An important concern in the TDE problem is that of predicting the performance of the TDE methods employed. The Cramer-Rao lower bound (CRLB) is commonly used as a performance standard for time delay estimators. Part of the appeal of the CRLB is that, for cases of practical interest, it can be shown that the maximum likelihood (ML) estimate can be made arbitrarily close to the CRLB for sufficiently long observation times. If the observation time is

20. (Cont'd)

not "sufficiently long" to ensure that the estimation error is small, estimator performance is dominated by large estimation errors due to anomalous estimates, and actual performance can be much worse than predicted by the CRLB. This report presents a detailed analysis of time delay estimator performance.

By way of background, the generalized cross correlation (GCC) approach for TDE is discussed and it is shown that many of the commonly used TDE methods, including the ML estimator, are related through the GCC method. In recent work, a correlator performance estimate (CPE) and a modified Ziv-Zakai lower bound (ZZLB) have been developed, which predict performance more accurately than the CRLB for large estimation error conditions. Derivations of the CRLB, the CPE and the ZZLB are given, and the behavior of these performance estimates is investigated for the case of flat, low-pass signal and noise power spectra. The CPE and the ZZLB are seen to be characterized by a threshold signal-to-noise ratio (SNR). Above this threshold SNR, the CPE and the ZZLB coincide with the CRLB, while below the threshold, the CPE and the ZZLB deviate from the CRLB and predict much poorer performance than the CRLB. Further, the threshold SNR is shown to be approximately inversely proportional to the square root of the coherent processing time.

This behavior has significant implications for TDE signal processing techniques. In particular, significant performance gains can be realized by implementing a coherent processor as opposed to an incoherent processing algorithm. Simulation results are presented to corroborate these performance predictions. For the case of a time-varying time delay, these observations point to the necessity of pre-processing the received signals to compensate for the relative time compression. This compensation is required in order to implement a coherent processor and results in a more complex structure for the time delay estimator. A relatively simple compensation technique is described, and preliminary simulation results are presented to demonstrate the effectiveness of this technique.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES AND FIGURES

Table Page

Figure

LIST OF TABLES AND FIGURES (continued)

LIST OF SYMBOLS

B	bandwidth, highest frequency in low-pass signal and noise power spectra (Hz)
B_e	effective bandwidth (Hz)
c	speed of sound in the ocean (meters/second)
$C_{xy}(f)$	magnitude-squared coherence between $x(t)$ and $y(t)$
D	true time delay
D_o	maximum possible time delay
$E[\cdot]$	expectation operator
$\exp(\cdot)$	exponential function, also $e^{(\cdot)}$
f	frequency (Hz)
$G[\cdot]$	"valley filling" function or operator
j	$\sqrt{-1}$
L	distance between sensors (meters)
ln	natural logarithm (base e)
log	base 10 logarithm
$n_i(t)$	additive noise waveform at sensor i, i=1,2
$p(x)$	probability density function (pdf) of x
$p(x\|y)$	conditional pdf of x given y
$P[A]$	probability of anomaly
$P(C)$	probability of the event C
$P(C\|D)$	conditional probability of the event C given the event D
$P_e(\tau_1,\tau_2)$	minimum probability of error for the binary decision problem
$Q(x)$	the Q function, related to the error function, erf(x)
$s(t)$	source signal waveform
$R_{xx}(\cdot)$	auto correlation function for x(t)

LIST OF SYMBOLS (continued)

$R_{xy}(\cdot)$	cross correlation function for x(t) and y(t)
$R_i(f)$	Fourier transform for $r_i(t)$, i=1,2
$r_i(t)$	received signal waveform at sensor i, i=1,2
ΔR	difference in path lengths between the source and two sensors
t	time (seconds)
T	observation time (seconds)
T_k	kth subinterval of T
T_c	signal correlation time
W(f)	weighting function for GCC processor
$W_\phi(f)$	phase weighting function for GCC processor
α	relative attenuation parameter
β_i	time compression at sensor i, i=1,2
β	relative time compression, (β_2/β_1)
$\gamma_{xy}(f)$	complex coherence function for x(t) and y(t)
π	pi
τ	hypothesized time delay value (seconds)
θ	approximate bearing angle
Δ	separation between hypothesized delay values (seconds)
σ^2	variance
σ	standard deviation
$\phi(f)$	phase function
$\partial/\partial\tau$	partial derivative with respect to τ
$\partial^2/\partial\tau^2$	second partial derivative with respect to τ
∞	infinity
$\hat{}$	estimate, e.g., $\hat{D}$ is an estimate of D
$\star$	complex conjugate, e.g., $R^\star(f)$

LIST OF ABBREVIATIONS

AML	approximate maximum likelihood
CPE	correlation performance estimate
CRLB	Cramer-Rao lower bound
dB	decibel
FT	Fourier transform
FFT	fast Fourier transform
GCC	generalized cross correlation
Hz	Hertz
IFT	inverse Fourier transform
LMS	least mean square
ML	maximum likelihood
MSC	magnitude squared coherence
pdf	probability density function
PHAT	phase transform
SCC	standard cross correlation
SCOT	smoothed coherence transform
SNR	signal-to-noise ratio
TDE	time delay estimation
WP	Wiener processor
ZZLB	Ziv-Zakai lower bound

CHAPTER 1

INTRODUCTION

Time delay estimation (TDE) is an area of active research with applications in a wide variety of fields [1]. An important concern in any estimation problem is that of predicting the performance of the estimation methods employed. It is often possible to set a measure of performance by establishing bounds on performance. While performance bounds set an absolute standard or limit for performance, the usefulness of a given bound for predicting performance depends upon the tightness of the bound, i.e., how close the bound is to the greatest lower (or least upper) bound. This research conducts a comparative study of performance bounds for the time delay estimation problem. For the TDE problem, lower bounds on the variance of the time delay estimate can be computed. An important objective of this study is to gain a better understanding of the behavior of these bounds as functions of the signal-to-noise ratio, bandwidth, and observation time. In addition, the TDE performance bounds are compared to simulation results to determine their relative merit for predicting time delay estimator performance. The results of this study are found to have significant implications for TDE signal processing techniques.

1.1 TDE Background

In the basic time delay estimation problem, a signal and a delayed version of the signal, both of which may be corrupted by noise, are received at a sensor or sensors and the object is to estimate the delay value. This problem has found applications in seismic prospecting [2], bio-medical engineering [3], nuclear reactor engineering [4], and many other areas. Depending on the application, the signal may be electrical, acoustical, or even neutron flux fluctuations [4], and the receiver may be a single sensor, multiple sensors, or sensor arrays. The application which motivates this research is that of passive sonar source localization. The problem will be modelled as depicted in Figure 1.1.; an acoustic signal emanates from a point source, propagates through the ocean medium, and is received at two spatially separated sensors. Assuming a constant propagation speed of sound in the ocean, the time delay in the time of arrival of the signal at the two sensors is given by the difference in the path lengths between the source and the two sensors, divided by the speed of sound in the ocean.

For a given value of the time delay, it can be shown that the source must be located along a hyperbola as shown in Figure 1.2. Taking the baseline between the two sensors as the x-axis, with the origin at the mid-point between the sensors, the equation of the hyperbola is given by

$$\frac{x^2}{\Delta R^2} + \frac{y^2}{\Delta R^2 - L^2} = 1/4 \ , \qquad (1\text{-}1)$$

where L is the distance between sensors and ΔR is the difference in the path lengths between the source and the two sensors. Note that $\Delta R = D\,c$, where D is the time delay of the signal at sensor 1 relative to sensor 2 and c is the speed of sound in the ocean.

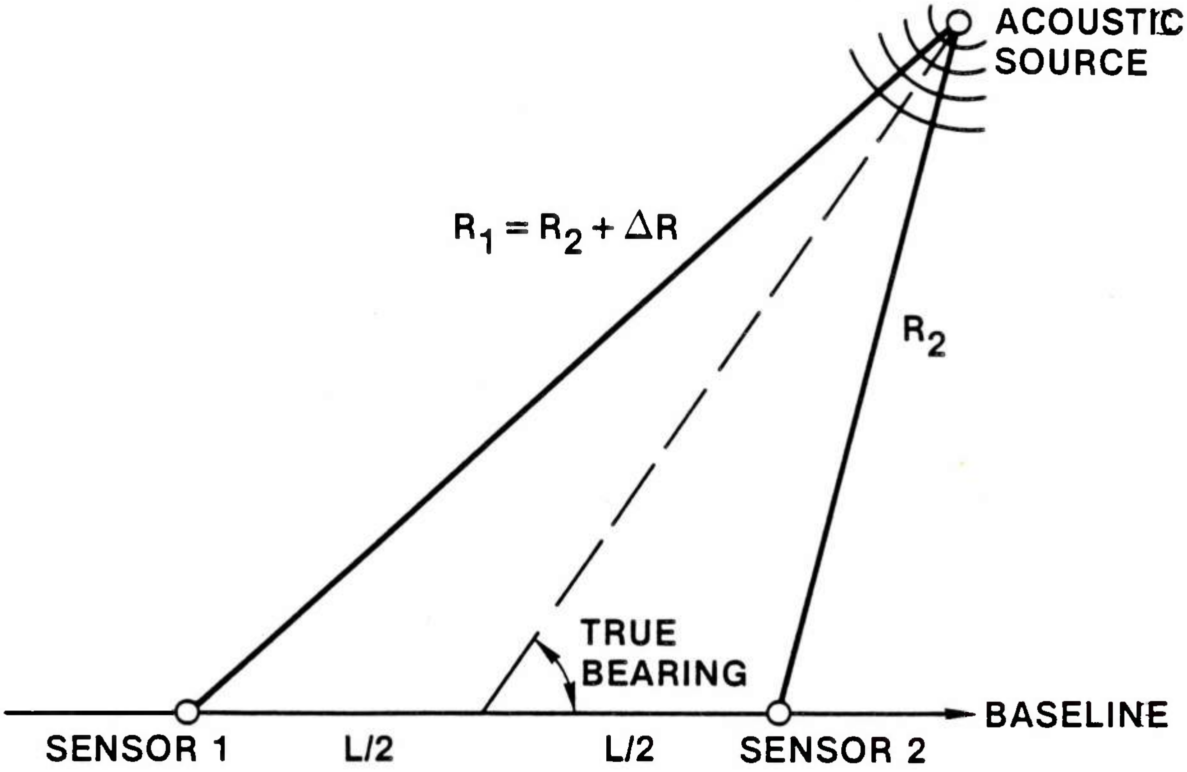


Figure 1.1 Physical Model of the TDE Problem

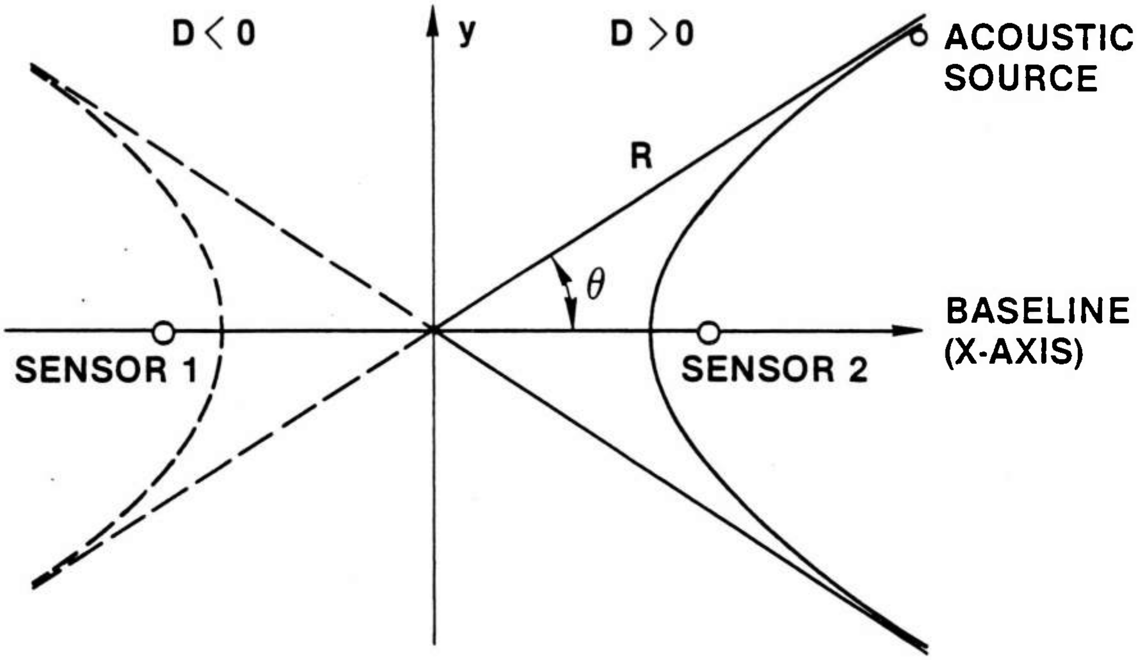


Figure 1.2 Bearing Angle Interpretation

The bearing of the source, relative to the midpoint of the baseline, is given to a good approximation by the bearing angle, θ, defined by the intersection of the hyperbolic asymptote with the baseline. It can be shown that

$$\cos \theta = \frac{\Delta R}{L} \left[1 + \frac{L^2 - \Delta R^2}{R^2} \right]^{1/2} \qquad (1\text{-}2)$$

where R is the distance to the source along the hyperbolic asymptote. For a distant point source ($R >> L$), θ is a very good approximation to the true bearing and (1-2) becomes

$$\theta \simeq \cos^{-1} \frac{\Delta R}{L} = \cos^{-1} \frac{D\ c}{L} \quad . \qquad (1\text{-}3)$$

Thus, if L and c are known and the time delay between the two sensors can be estimated, an estimate of the source bearing, relative to the position of the sensors, can be obtained from equation (1-3). For $D > o$, the source is closer to sensor 2 than to sensor 1 as in Figure 1.2. For $D < o$, the situation is reversed and the source is located along the corresponding hyperbola reflected through the y-axis. The maximum possible delay occurs when the source is in the endfire position (colinear with the sensors) and $D_{max} = L/c$. When the source is in the broadside position (along the y-axis in Figure 1.2) the time delay is zero. The time delay is known "a priori" to be limited by $-L/c \leq D \leq L/c$, or equivalently, $-L \leq \Delta R \leq L$.

Many techniques have been proposed for TDE including both frequency domain and time domain processors. Perhaps the simplest method for estimating the time delay is to compute the cross-correlation function between the signal and its delayed version. The argument for which the cross-correlation function attains its maximum value corresponds to the time delay estimate [5, pp. 64-65]. Knapp and Carter have shown that many of the commonly used TDE techniques are related through the Generalized Cross Correlation (GCC) approach [6]. The class of GCC processors are generally considered to be frequency domain processors and include: the Roth processor [7], the Eckart filter [8], the smoothed coherence transform (SCOT) [9], the phase transform (PHAT) [10], the Wiener processor [11], and the Hannan-Thomson or maximum likelihood processor [12, 6]. In this work, consideration is limited to the GCC approach for TDE which is discussed in some detail in Chapter 2. However, it should be noted that considerable work has been done on time domain processors for TDE. Many of these time domain techniques are based on Widrow's least mean square (LMS) algorithm [13]. Excellent discussions of adaptive time delay estimation are given in works by Youn [14], Youn and Ahmed [15], Chan, Riley, and Plant [16, 17], Etter and Stearns [18], and Feintuch, Bershad, and Reed [19].

1.2 TDE Performance

Several of the GCC processors, such as the maximum likelihood and Eckart estimators, are designed to be optimum with respect to some performance criteria. Others, such as the SCOT and PHAT, are intuitive or "ad hoc" techniques, developed to perform well for certain signal and noise spectra. Depending upon the performance criteria and the signal and noise spectra, a strong argument can be made for the optimality of each of the TDE processors. There is an obvious need for a performance standard against which the various TDE methods can be evaluated. The Cramer-Rao lower bound (CRLB) is commonly used as a performance standard [20-22]. The CRLB yields a lower bound on the variance of any unbiased time delay estimate as a function of several parameters (e.g. the signal and noise power spectra and the observation time). Part of the appeal of the CRLB is that under certain assumptions, there is a theorem which states that the maximum likelihood (ML) estimate can be made arbitrarily close to the CRLB for sufficiently long observation times [23, pp. 62-72]. However, the theorem does not specify how long the observation time must be. Thus, while the CRLB sets a lower bound on the variance of the time delay estimate, actual performance can be much worse for a given signal-to-noise ratio (SNR) and observation time. This is substantiated by the simulation results in [24, 25].

Several studies have been conducted to find a bound tighter than the CRLB, which would predict performance more accurately. Chow and Schultheiss have investigated a simplified version of the Barankin bound [26]. Ianniello has developed a correlator performance estimate (CPE) and has shown via simulation that, for the cross-correlation technique of TDE, the CPE yields a more accurate estimate of performance than the CRLB [27]. Weiss and Weinstein have proposed a modified version of the Ziv-Zakai lower bound (ZZLB) [28-30]. A comparison of the CPE and the ZZLB is reported by Ianniello, Weinstein, and Weiss in [31]. Further comparisons of the CPE, the ZZLB and the CRLB are presented in this dissertation. In particular, the behavior of these performance estimates is considered as a function of the observation time and SNR, and the implications of this behavior are discussed. Portions of this work, comparing the CRLB and the CPE, appear in [32].

1.3 Organization and Content

Before proceeding with the analysis of time delay estimator performance, the mathematical model for the TDE problem is introduced in Chapter 2. The GCC approach for TDE is discussed briefly and several specific estimators from the class of GCC processors are described. In Chapter 3, the shortcomings of the CRLB are pointed out, in order to motivate the need for better performance estimates such as the CPE and the ZZLB. Derivations of the CRLB, the CPE and the ZZLB are outlined and expressions for the variance of the delay estimate for specific signal and noise power spectra are given. Futher comparisons of these performance estimates are presented in Chapter 4, including comparison with simulation results. The threshold effect exhibited by the CPE and the ZZLB is shown to have important implications for TDE processing methods. In particular, significant performance gains may be obtained from increasing the coherent processing time as opposed to increasing the incoherent processing time. To this point, a stationary TDE model (no relative motion between source and sensors) has been assumed. In Chapter 5, the case of a time-varying time delay is considered. The observations of Chapter 4 point to the necessity of pre-processing the received signals to compensate for the relative motion before applying the GCC approach for TDE. One such compensation technique is described and preliminary simulation results are presented. Chapter 6 summarizes the conclusions of this research and offers some suggestions for future work.

CHAPTER 2

THE GCC APPROACH FOR TIME DELAY ESTIMATION

This chapter introduces the mathematical model and discusses the related assumptions for the time delay estimation problem considered in this work. Based on this model, the Generalized Cross Correlation (GCC) approach for estimating time delay is developed. Several specific techniques from the class of GCC estimators are formulated and the characteristics of these techniques are discussed briefly.

2.1 Mathematical Model

A commonly used model for a signal radiated from an acoustic source and received in the presence of additive noise at two spatially separated sensors is given by (2-1). Denoting the received signals as $r_1(t)$ and $r_2(t)$, let

$$r_1(t) = s(t) + n_1(t) \qquad (2\text{-}1a)$$

$$r_2(t) = \alpha s(t+D) + n_2(t),\ 0 \le t \le T \qquad (2\text{-}1b)$$

where s(t) represents the source signal, $n_1(t)$ and $n_2(t)$ are the additive noises, α is a relative attenuation parameter, and D is the time delay parameter to be estimated. The observation time, T, is assumed to be much larger than D. Additionally, it is assumed that s(t), $n_1(t)$, and $n_2(t)$ are zero mean, stationary, Gaussian random processes and that they are mutually uncorrelated. Also note that the model of (2-1) makes several implicit assumptions. In particular, the effects of multiple sources and multi-path arrivals are ignored. Since only the direct acoustic path is considered, the problem is essentially planar, with the source and the two sensors defining the plane of interest, as in Figure 1.1. The attenuation parameter, α, allows modelling of a relative attenuation in the signal strength due to the difference in path lengths between the source and sensors. In general, α will be frequency dependent rather than constant as in (2-1). Also the time delay, D, is assumed to be constant over the observation interval which requires that the source remain stationary relative to the sensors during this time interval. The problem of relative motion between source and sensors is discussed in Chapter 5. Finally, the speed of sound in the ocean is assumed constant, although it actually varies with depth (temperature and pressure).

The results of any analysis based on the simplified model of (2-1) must be applied cautiously to "real world" problems, such as the passive sonar problem which motivates this research. Nevertheless considerable insight into time delay estimator performance can be derived from analysis of this simple model. More detailed discussions of sonar signal processing considerations and effects of the ocean medium may be found in [33-35].

2.2 GCC Approach

An effective method of estimating the time delay, D, is to compute the cross correlation function between the received signals, $r_1(t)$ and $r_2(t)$,

$$R_{r_1r_2}(\tau) = E[r_1(t+\tau)\; r_2(t)] \tag{2-2}$$

where $E[\cdot]$ is the expectation operator. Substituting (2-1) into (2-2) yields

$$R_{r_1r_2}(\tau) = \alpha\; R_{ss}(\tau-D), \tag{2-3}$$

where $R_{ss}(\tau)$ is the auto-correlation function of the signal, $s(t)$. To obtain (2-3), the assumption that $s(t)$, $n_1(t)$, and $n_2(t)$ are mutually uncorrelated has been used. The cross correlation function can also be computed from the cross power spectrum, $G_{r_1r_2}(f)$, of the received signals, using the Fourier transform (FT) relationship

$$R_{r_1r_2}(\tau) = \int_{-\infty}^{\infty} G_{r_1r_2}(f)\; e^{j2\pi f\tau}\; df \tag{2-4a}$$

where, for the model of (2-1),

$$G_{r_1r_2}(f) = \alpha\; G_{ss}(f)\; e^{-j2\pi fD}\; . \tag{2-4b}$$

It can be seen from either (2-3) or (2-4) that $R_{r_1r_2}(\tau)$ attains its maximum value when τ=D. Thus, the argument, τ, which maximizes the cross correlation function is an estimate of the time delay.

In practice, the cross correlation function must be estimated from data acquired during a finite observation time, T. To improve the accuracy of the delay estimate, $\hat{D}$, it is usually desirable to prefilter the noisy received signals, $r_1(t)$ and $r_2(t)$, prior to computing the cross correlation function. For example, if the source signal is known to be bandlimited between f_1 and f_2, while the spectra of the additive noises extend beyond these limits, it would be reasonable to prefilter the received signals, passing only the frequency band between f_1 and f_2. This concept of prefiltering motivates the GCC approach for time delay estimation as depicted in Figure 2.1 [6].

The received signals, $r_1(t)$ and $r_2(t)$, are processed through filters having transfer functions, $H_1(f)$ and $H_2(f)$, respectively. The cross power spectrum between the filter outputs, $p_1(t)$ and $p_2(t)$, is given by [36, p. 353]

$$G_{p_1p_2}(f) = H_1(f)\ H_2^*(f)\ G_{r_1r_2}(f), \tag{2-5}$$

where * denotes the complex conjugate. The GCC function, $R^G_{r_1r_2}(\tau)$, is then obtained by taking the inverse Fourier transform (IFT) of $G_{p_1p_2}(f)$,

$$R^G_{r_1r_2}(\tau) = \int_{-\infty}^{\infty} W(f)\ G_{r_1r_2}(f)\ e^{j2\pi f\tau}\ df, \tag{2-6}$$

where $W(f) = H_1(f)^* H_2(f)$. Substituting (2-4b) in (2-6) yields

$$R^G_{r_1r_2}(\tau) = \int_{-\infty}^{\infty} \alpha\ W(f)\ G_{ss}(f)\ e^{-j2\pi fD}\ e^{j2\pi f\tau}\ df. \tag{2-7}$$

To ensure that $R^G_{r_1r_2}(\tau)$ has a maximum at τ=D, W(f) must be a real function, or equivalently, the prefilters, $H_1(f)$ and $H_2(f)$, must have identical phase characteristics.

While the realization of Figure 2.1 is useful conceptually, the approach suggested by (2-6) is often preferred. That is, the cross power spectrum,

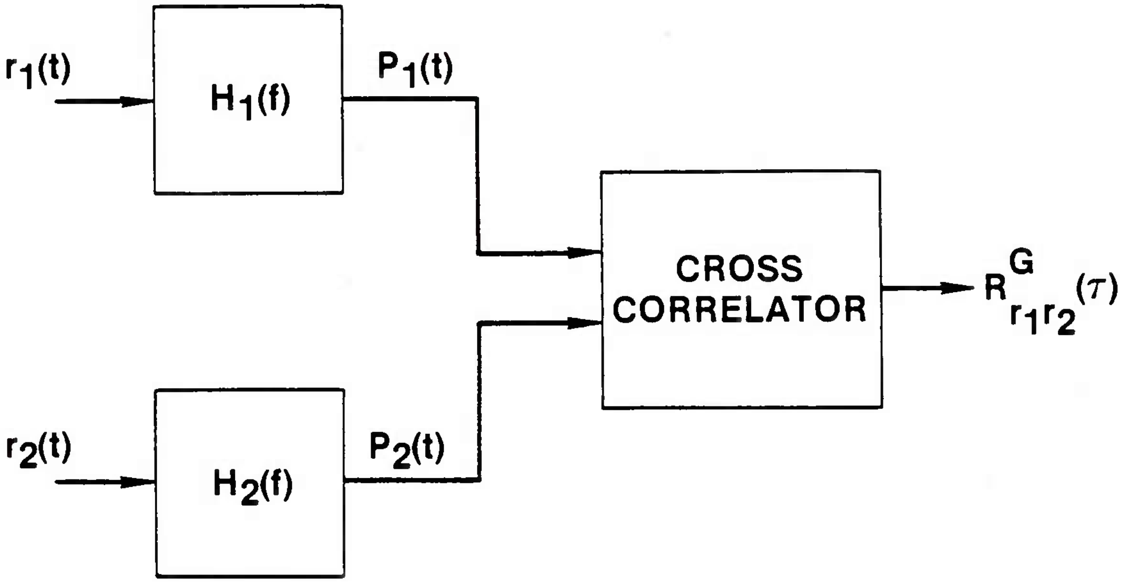


Figure 2.1 GCC Approach for TDE

$G_{r_1r_2}(f)$, is multiplied by a real weighting function, W(f), then the IFT of the result is computed to obtain the GCC function. Additional insight into the GCC approach may be gained by writing (2-6) in a slightly different form,

$$R^G_{r_1r_2}(\tau) = \int_{-\infty}^{\infty} W_\phi(f)\ e^{j\phi(f)}\ e^{j2\pi f\tau}\ df, \tag{2-8a}$$

where

$$W_\phi(f) = W(f)\left|G_{r_1r_2}(f)\right| = \alpha\ W(f)\ G_{ss}(f), \tag{2-8b}$$

and

$$e^{j\phi(f)} = G_{r_1r_2}(f)/\left|G_{r_1r_2}(f)\right| = e^{-j2\pi fD}\ . \tag{2-8c}$$

The derivative of the phase, $\phi(f)$, with respect to the frequency is a measure of the time delay and the GCC method can be thought of as applying a weighting function, $W_\phi(f)$, to the phase of the cross power spectrum [37]. Note that (2-8) is equivalent to (2-6).

A variety of weighting functions have been proposed to optimize performance with respect to certain criteria. As suggested previously, if a priori information about the signal and noise spectra exists, this information should be incorporated in the weighting function, W(f). Indeed many of the commonly used weightings are functions of the signal and noise power spectra. However, in most cases of practical interest, little or no a priori information is available. This necessitates the estimation of the cross power spectrum and the weighting function from the received signals. The pertinent spectra can be obtained using

Processor	$W(f) = H_1(f)\ H_2^*(f)$	$W_\phi(f) = \left\|G_{r_1r_2}(f)\right\| W(f)$
Standard Cross Correlation (SCC)	1	$G_{r_1r_2}(f)$
Roth	$\frac{1}{G_{r_1r_1}(f)}$	$\frac{G_{r_1r_2}(f)}{G_{r_1r_1}(f)}$
Smoothed Coherence Transform (SCOT)	$\frac{1}{\sqrt{G_{r_1r_1}(f)\ G_{r_2r_2}(f)}}$	$\sqrt{C_{r_1r_2}(f)}$
Wiener Processor (WP)	$C_{r_1r_2}(f)$	$C_{r_1r_2}(f)\left\|G_{r_1r_2}(f)\right\|$
Phase Transform (PHAT)	$\frac{1}{\left\|G_{r_1r_2}(f)\right\|}$	1
Maximum Likelihood (ML)	$\frac{C_{r_1r_2}(f)}{[1 - C_{r_1r_2}(f)]\left\|G_{r_1r_2}(f)\right\|}$	$\frac{C_{r_1r_2}(f)}{1 - C_{r_1r_2}(f)}$

Table 2.1 Weighting Functions for GCC Processors

standard spectral estimation techniques. Substitution of these estimates into (2-6) or (2-8a) enables an estimate of the GCC function, $\hat{R}^{G}_{r_1r_2}(\tau)$, to be computed. Inherent in the need to estimate the power spectra is a corresponding degradation in performance, the degree of which is dependent on the merit of the estimates. This point will be considered again in connection with the simulation results in Chapter 4. In the following discussion, specific techniques for time delay estimation are examined and it is shown how they are related through the GCC approach. The GCC weighting functions for these techniques are summarized in Table 2.1.

2.2.1 Standard Cross Correlation

The basis for the GCC approach is the standard cross correlation (SCC) method. As noted in (2-4a), the SCC function and the cross power spectrum form a FT pair. The SCC method can be thought of as applying a uniform weighting to the cross power spectrum prior to computing the IFT. Setting $W(f) = 1$ in (2-6), or equivalently, $W_{\phi}(f) = \left| G_{r_1r_2}(f) \right|$ in (2-8a), immediately yields the defining relationship for the SCC function, equation (2-4a).

2.2.2 The Roth Processor

One of the first modifications to the SCC method was proposed by Roth [7]. The Roth processor applies the weighting function

$$W(f) = \frac{1}{G_{r_1r_1}(f)} \tag{2-9a}$$

to obtain

$$R^{Roth}_{r_1r_2}(\tau) = \int_{-\infty}^{\infty} \frac{G_{r_1r_2}(f)}{G_{r_1r_1}(f)} e^{j2\pi f\tau}\, df. \tag{2-9b}$$

If $r_1(t)$ is the input to a linear filter which provides the minimum mean square error estimate of $r_2(t)$ (i.e. a Wiener filter), then $R^{Roth}_{r_1r_2}(\tau)$ is the impulse response of this filter. Thus, the transfer function of this optimum filter is simply $G_{r_1r_2}(f)/G_{r_1r_1}(f)$.

For the signal model under consideration,

$$G_{r_1r_1}(f) = G_{ss}(f) + G_{n_1n_1}(f) \ . \tag{2-10}$$

The Roth weighting function suppresses those frequency bands where the noise power, $G_{n_1n_1}(f)$, is large and the estimate $G_{r_1r_2}(f)$ is more likely to be in error. Further the Roth weighting tends to suppress those regions where $G_{ss}(f)$ is large, which has a whitening effect on the signal. This reduces the ambiguity that is introduced in the SCC function by strong tonals in the signal spectrum [9, 36].

2.2.3 The Smoothed Coherence Transform (SCOT)

Unless it is known that either $G_{n_1n_1}(f)$ or $G_{n_2n_2}(f)$ is the dominant noise process, there is no reason to choose the weighting function $W(f) = 1/G_{r_1r_1}(f)$ over $W(f) = 1/G_{r_2r_2}(f)$. The SCOT [9] avoids this uncertainty by selecting

$$W(f) = 1/\sqrt{G_{r_1r_1}(f)\, G_{r_2r_2}(f)} \;. \tag{2-11}$$

The SCOT is then given by

$$R^{SCOT}_{r_1r_2}(\tau) = \int_{-\infty}^{\infty} \gamma_{r_1r_2}(f)\, e^{j2\pi f\tau}\, df \tag{2-12a}$$

where

$$\gamma_{r_1r_2}(f) = \frac{G_{r_1r_2}(f)}{\left[G_{r_1r_1}(f)\, G_{r_2r_2}(f)\right]^{1/2}} \;. \tag{2-12b}$$

In (2-12b), $\gamma_{r_1r_2}(f)$ is the complex coherence between $r_1(t)$ and $r_2(t)$.

In terms of the GCC realization of Figure 2.1, the SCOT can be interpreted as employing two pre-whitening filters, $H_1(f) = 1/\sqrt{G_{r_1r_1}(f)}$ and $H_2(f) = 1/\sqrt{G_{r_2r_2}(f)}$, followed by a cross correlator. Thus the SCOT exhibits advantages similar to the Roth processor, suppressing frequency bands where the noise power is large and reducing the effect of strong tonals in the signal, while eliminating the ambiguity of the Roth processor as to which noise process is dominant. If $G_{n_1n_1}(f) = G_{n_2n_2}(f)$, the SCOT and the Roth technique are identical. Writing the SCOT, equation (2-12a), in the form of (2-8a) yields

$$R^{SCOT}_{r_1r_2}(\tau) = \int_{-\infty}^{\infty} C_{r_1r_2}(f)\, e^{-j2\pi fD}\, e^{j2\pi f\tau}\, df \tag{2-13}$$

where $C_{r_1r_2}(f)$ is the magnitude squared coherence (MSC), that is

$$C_{r_1r_2}(f) = \left|\gamma_{r_1r_2}(f)\right|^2 . \qquad (2\text{-}14)$$

The SCOT assigns greater weight to the phase (time delay) information in frequency bands where the coherence is high.

2.2.4 The Wiener Processor

Recently a new GCC processor, referred to as the Wiener processor (WP) by the authors, has been proposed in [11]. This processor uses two Wiener prefilters to obtain the minimum mean square estimates of s(t) from $r_1(t)$, and of $\alpha s(t-D)$ from $r_2(t)$ before computing the cross correlation. The prefilters are given by

$$H_1(f) = \frac{G_{ss}(f)}{G_{r_1r_1}(f)} = \frac{1}{\alpha}\,\frac{G_{r_1r_2}(f)}{G_{r_1r_1}(f)}\, e^{j2\pi fD} \qquad (2\text{-}15a)$$

and

$$H_2(f) = \frac{\alpha^2 G_{ss}(f)}{G_{r_2r_2}(f)} = \frac{\alpha\, G_{r_1r_2}(f)}{G_{r_2r_2}(f)}\, e^{j2\pi fD} \quad . \qquad (2\text{-}15b)$$

The corresponding weighting function is then the MSC,

$$W(f) = H_1(f)\, H_2^*(f) = C_{r_1r_2}(f) \ , \qquad (2\text{-}16a)$$

and the WP computes the GCC function,

$$R^{WP}_{r_1r_2}(\tau) = \int_{\infty}^{\infty} C_{r_1r_2}(f)\, G_{r_1r_2}(f)\, e^{j2\pi f\tau}\, df. \qquad (2\text{-}16b)$$

The WP suppresses the cross power spectrum in frequency regions where the coherence is low. Note that while the Wiener prefilters, $H_1(f)$ and $H_2(f)$, require knowledge of the attenuation parameter, α, the weighting function can be estimated directly from the received signals.

2.2.5 The Phase Transform (PHAT)

The PHAT is an "ad hoc" technique which uses the weighting [10]

$$W(f) = 1/\left|G_{r_1r_2}(f)\right| \tag{2-17a}$$

or

$$W_{\phi}(f) = 1. \tag{2-17b}$$

Thus the PHAT computes the IFT of the phase function,

$$R_{r_1r_2}^{PHAT}(\tau) = \int_{-\infty}^{\infty} e^{j\phi(f)}\ e^{j2\pi f\tau}\ df \tag{2-18a}$$

where for the model of (2-1)

$$\phi(f) = -2\pi fD \tag{2-18b}$$

so that, ideally

$$R_{r_1r_2}^{PHAT}\ (\tau) = \delta(\tau-D)\ . \tag{2-18c}$$

In practice, only an estimate, $\phi(f)$, of the phase function can be obtained, and $R_{r_1r_2}^{PHAT}(\tau)$ will not be an ideal delta function. The PHAT assigns equal weighting to the phase estimate throughout the frequency band, independent of SNR. Thus, the PHAT fails to suppress frequency bands where the SNR is relatively low and where the phase estimate is more likely to be in error. However, the PHAT effectively whitens the cross power spectrum and therefore eliminates the effect of strong tonals.

2.2.6 The Maximum Likelihood (ML) Estimator

The final weighting to be considered is that of the ML estimator [6]. This technique is identical to that proposed by Hannan and Thomson [12] and is often referred to as the HT processor. The ML designation will be used in this work. As shown in Appendix A, the ML estimator is obtained by applying the weighting function

$$W(f) = \frac{1}{\left|G_{r_1r_2}(f)\right|} \frac{C_{r_1r_2}(f)}{1 - C_{r_1r_2}(f)} \tag{2-19a}$$

thus

$$R^{ML}_{r_1r_2}(\tau) = \int_{-\infty}^{\infty} \frac{C_{r_1r_2}(f)}{1 - C_{r_1r_2}(f)} e^{j\phi(f)} e^{j2\pi f\tau} df. \tag{2-19b}$$

The ML estimator, like the SCOT, weights the phase relative to the MSC. However, $W^{SCOT}_{\phi}(f) = \sqrt{C_{r_1r_2}(f)}$ approaches unity when the coherence is large (note, $0 \leq C_{r_1r_2}(f) \leq 1$), whereas $W^{ML}_{\phi}(f) = C_{r_1r_2}(f)/(1 - C_{r_1r_2}(f))$ goes to infinity as the MSC approaches one. Compared to the SCOT, the ML estimator assigns much greater weight to the phase in frequency bands where the coherence is large. While the SCOT reduces the effect of strong tonals, the ML estimator tends to emphasize such tonals, where the coherence will be relatively large. In practice, the pertinent power spectra must be estimated and only an estimate of (2-19) can be obtained. Hence, only an approximate maximum likelihood (AML) estimator can be implemented.

CHAPTER 3

TIME DELAY ESTIMATOR PERFORMANCE BOUNDS

The Cramer-Rao lower bound (CRLB) is commonly used to set a bound on the variance of the time delay estimate. Use of the CRLB to predict time delay estimator performance is justified in that the ML estimator performance approaches CRLB performance for sufficiently long integration times, that is, the ML estimator is asymptotically efficient [23, p. 71]. In practice, the observation time is necessarily limited, and in many cases, the observation time will not be "sufficiently long" to attain CRLB performance. This leads to two important questions in the TDE problem. First, under what conditions is CRLB performance attainable, and second, what performance can be expected when conditions do not allow for CRLB performance. In this chapter, the CRLB, a correlator performance estimate (CPE) and a modified Ziv-Zakai lower bound (ZZLB) are investigated to find answers to these equations.

3.1 The Cramer-Rao Lower Bound (CRLB)

Consider any unbiased estimate, $\hat{D}(\underline{R})$, of the time delay D, where $\underline{R}$ is an observation vector with conditional probability density function $p(\underline{R}|\tau)$. Then the following inequality for the variance of the delay estimate holds [23, p. 66] or [38, p. 316],

$$\sigma_{\hat{D}}^2 = E[(\hat{D}(\underline{R}) - D)^2] \geq \left(-E\left[\frac{\partial^2 \ln p(\underline{R}|\tau)}{\partial\tau^2} \right] \right)^{-1} \Bigg|_{\tau=D} \qquad (3\text{-}1)$$

provided that $\partial p(\underline{R}|\tau)/\partial\tau$ and $\partial^2 p(\underline{R}|\tau)/\partial\tau^2$ exist and are absolutely integrable. From equation (A-8), (see Appendix A), it follows that

$$E\left[\frac{\partial^2 \ln p(\underline{R}|\tau)}{\partial\tau^2}\right] = E\left[\frac{\partial^2}{\partial\tau^2}(-1/2)\ J_1\right] \tag{3-2}$$

since the quantity c in (A-8c) is independent of τ. Substituting (A-10) into (3-2) and noting that the term J_2 is also independent of τ yields

$$E\left[\frac{\partial^2 \ln p(\underline{R}|\tau)}{\partial\tau^2}\right] = E\left[\frac{\partial^2}{\partial\tau^2}(1/2)\ J_3\right] \tag{3-3a}$$

$$= \frac{\partial^2}{\partial\tau^2}\int_{-\infty}^{\infty} E[R_1(f)\ R_2^*(f)]\ \frac{C_{r_1r_2}(f)\ e^{j2\pi f\tau}}{\left|G_{r_1r_2}(f)\right|\ \ (1 - C_{r_1r_2}(f))}\,df \tag{3-3b}$$

$$= T\int_{-\infty}^{\infty} \frac{\partial^2}{\partial\tau^2}\, e^{j2\pi f(\tau-D)}\ \frac{C_{r_1r_2}(f)}{1 - C_{r_1r_2}(f)}\,df \tag{3-3c}$$

To obtain (3-3c) from (3-3b), the relation $E[R_1(f)\ R_2^*(f)] = T\ G_{r_1r_2}(f)$ has been used. Substitution of (3-3c) into (3-1), after taking the indicated second partial derivative with respect to τ, results in the following expression for the CRLB,

$$\sigma^2_{\hat{D}} \geq \sigma^2_{CRLB} = \left[T\int_{-\infty}^{\infty}(2\pi f)^2\ \frac{C_{r_1r_2}(f)}{1 - C_{r_1r_2}(f)}\,df\right]^{-1}. \tag{3-4}$$

The CRLB depends upon the signal and noise power spectra, in the form of the magnitude squared coherence (MSC), and is inversely proportional to the observation time, T.

Now consider specific signal and noise power spectra,

$$G_{ss}(f) = \begin{cases} S_o/2 & , \quad o \leq |f| \leq B \\ o & , \quad \text{elsewhere} \end{cases} \tag{3-5a}$$

and

$$G_{nn}(f) = \begin{cases} N_o/2 & , \quad o \leq |f| \leq B \\ o & , \quad \text{elsewhere} \end{cases} \tag{3-5b}$$

where $G_{nn}(f) = G_{n_1n_1}(f) = G_{n_2n_2}(f)$. In this work, the signal-to-noise ratio defined as

$$\text{SNR} \triangleq \begin{cases} G_{ss}(f)/G_{nn}(f) & , \quad o \leq |f| \leq B \\ o & , \quad \text{elsewhere} \end{cases} . \tag{3-6}$$

Note that this is the SNR at the input of a single receiver. For these power spectra, the MSC is given by

$$C_{r_1r_2}(f) = \frac{G_{ss}(f)^2}{(G_{ss}(f) + G_{nn}(f))^2} \tag{3-7a}$$

$$= \frac{\text{SNR}^2}{(\text{SNR} + 1)^2} \tag{3-7b}$$

Thus (3-4) can be written in terms of the SNR and the CRLB becomes

$$\sigma^2_{CRLB} = \left[T \int_{-B}^{B} (2\pi f)^2 \; \frac{SNR^2}{2SNR + 1} \, df \right]^{-1} \tag{3-8a}$$

or

$$\sigma^2_{CRLB} = \frac{3}{8\pi^2} \cdot \frac{1}{B^3 T} \cdot \frac{2SNR + 1}{SNR^2} \tag{3-8b}$$

for the power spectra of (3-5).

As discussed in [21], the CRLB exhibits different behavior at high and low SNR. Specifically, it can be shown that,

$$2 \log \sigma_{CRLB} \simeq K + \log 2 - \log B^3 T - \log SNR, \quad SNR >> 1 \tag{3-9a}$$

and

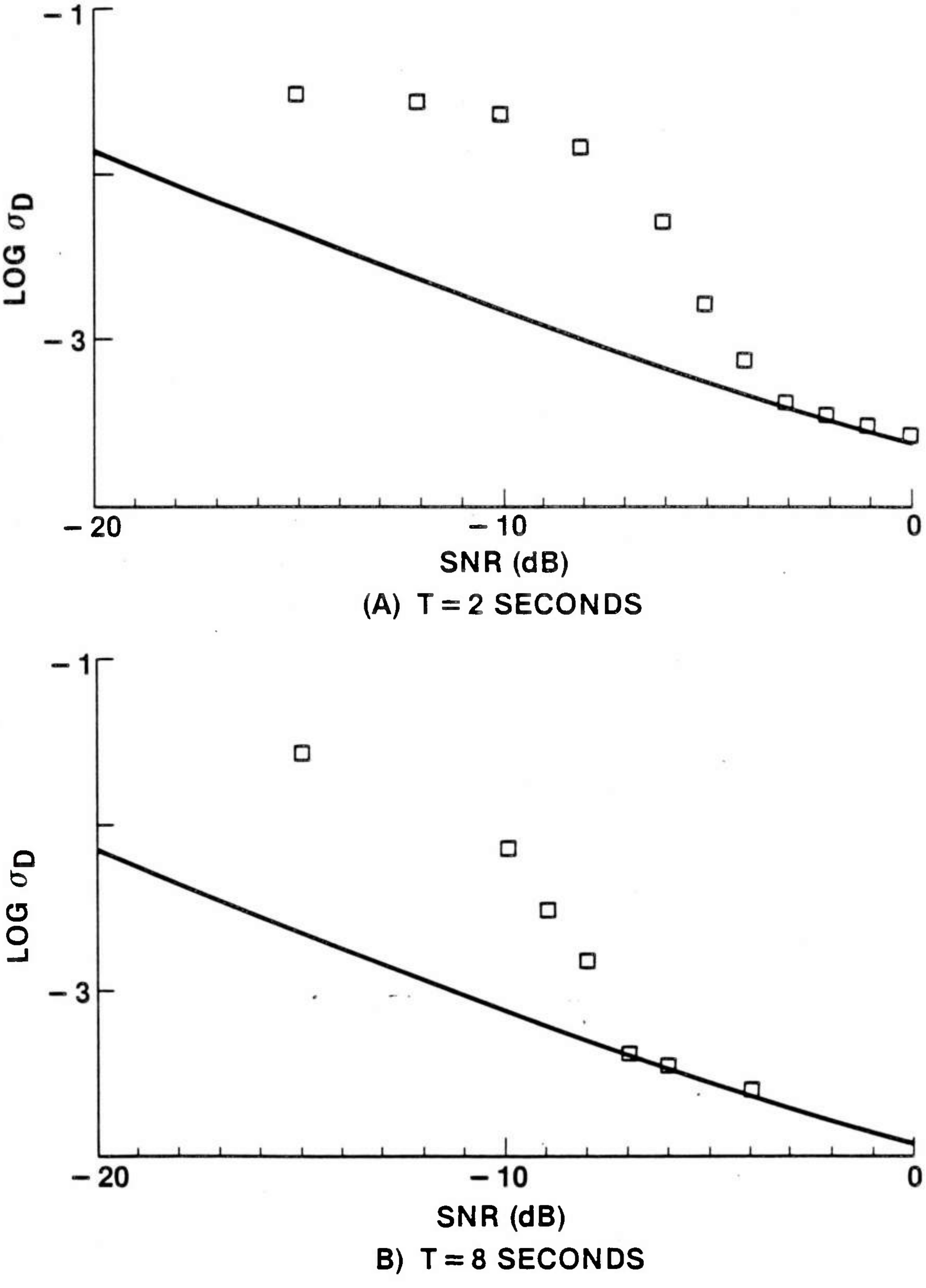
$$2 \log \sigma_{CRLB} \simeq K - \log B^3 T - 2 \log SNR, \quad SNR << 1 \tag{3-9b}$$

where $K = \log(3/8\pi^2)$. Thus, the high and low SNR approximations of $\log \sigma_{CRLB}$ are linear functions of log SNR, with the low SNR approximation having a slope twice that of the high SNR approximation. Further it is easy to show that the high and low SNR approximations intersect at log SNR = -log 2, that is, when the input SNR at a single receiver is -3 dB, independent of the observation time and signal bandwidth.

As discussed previously, the time delay is known a priori to be bounded by $-L/c \leq \tau \leq L/c$, where L is the sensor spacing and c is the propagation speed of sound in the ocean. Most TDE methods inherently utilize this knowledge to limit the range of the delay estimate. However, the CRLB does not incorporate this information to limit the variance of the delay estimate and $\sigma^2_{CRLB} \rightarrow \infty$ as SNR → ∞. Another shortcoming of greater significance is that when the observation time is not sufficiently long, the CRLB does not accurately predict time delay estimator performance. The explanation of this deficiency has to do with large estimation errors, which is the subject of the next section.

3.2 Large Estimation Errors

The ML estimate is a minimum variance estimate, that is, it attains the CRLB when the observation time is long enough. Additionally, when the estimation error is small, the variance of the ML estimate can be expected to be close to the CRLB. Therefore, the CRLB is a good estimator of TDE performance when the observation time is sufficiently long to ensure that the estimation error is small. To gain a better understanding of these relationships, a comparison of simulation results with the CRLB is shown in Figure 3.1. (See Appendix B for simulation details.) The signal and noise power spectra are given by (3-5) with B = 100 Hz, relative to a sampling frequency of 2048 Hz. In Figure 3.1(a), T = 2 seconds, and the simulation results are in good agreement with the CRLB for SNR $\geq$ -3 dB. Below -3 dB, the experimental variance deviates sharply from the CRLB. In Figure 3.1(b), the observation time has been increased to 8 seconds. Now the experimental variance agrees with the CRLB for SNR $\geq$ -7 dB, but again deviates from the CRLB below this SNR. Based on these results, increasing the observation time allows the estimator to maintain CRLB performance for lower SNR, but below some threshold SNR, estimator performance begins to degrade rapidly relative to the CRLB. Apparently, there is a fundamental trade-off between SNR and the observation or coherent processing time. For



(A) T = 2 SECONDS

B) T = 8 SECONDS

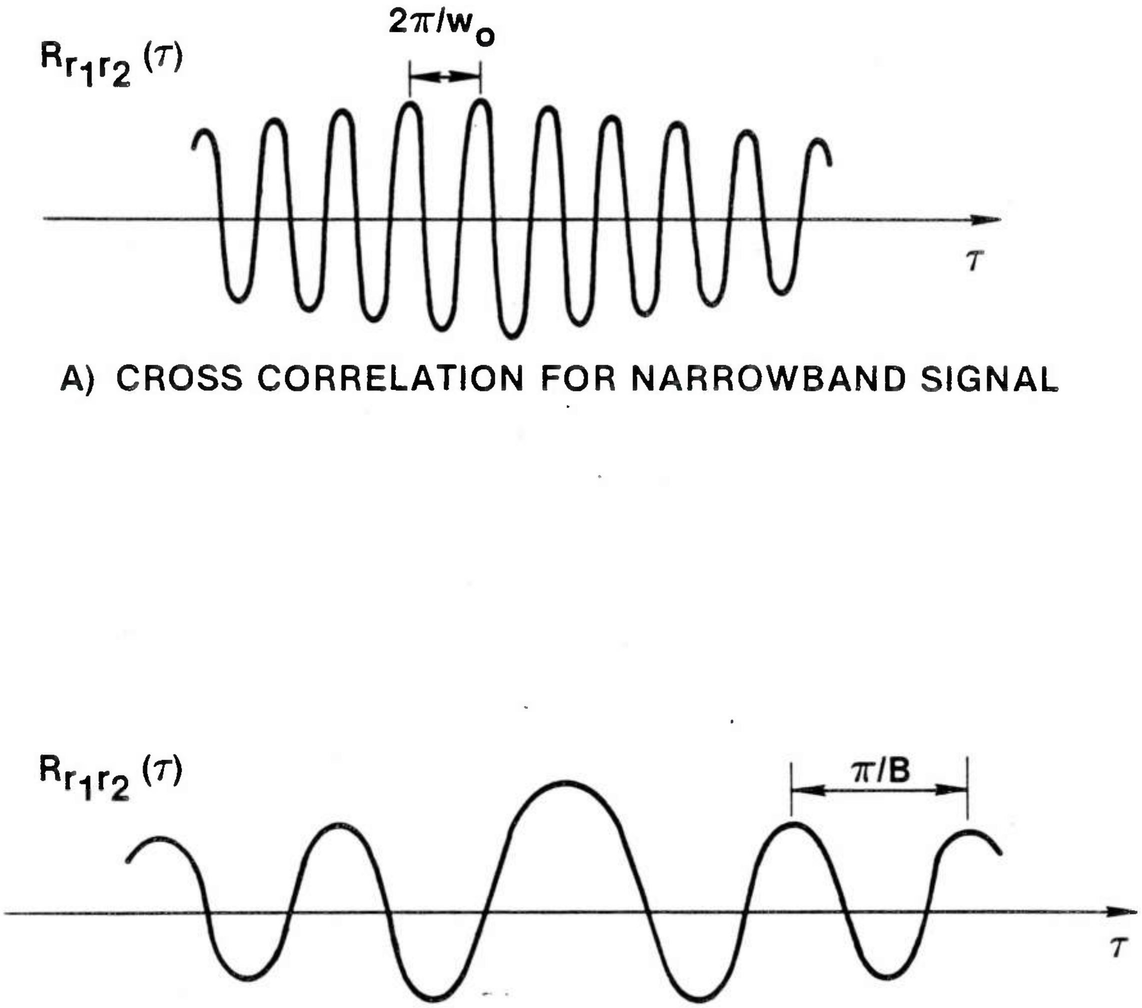
Figure 3.1 Comparison of CRLB and SCC Simulation Results

a given observation time, CRLB, or small estimation error, performance can be obtained above the threshold SNR. Below this SNR, large estimation errors must be taken into account and the CRLB is no longer applicable.

To understand the source of such large estimation errors, consider the cross correlation function for two types of signals. First consider a narrowband source such that the bandwidth, B, is small compared to the center frequency, f_0, ($B/f_0 << 1$). A typical cross correlation function for such a source signal in the absence of noise is depicted in Figure 3.2(a). The maximum occurs at the true delay, D, but adjacent peaks are nearly as large. To obtain CRLB performance, the processor must be able to resolve this ambiguity between peaks, which requires either a very large SNR or a very long observation time. A similar effect is observed for broadband signals. Consider a low-pass signal as in (3-5), Figure 3.2(b) shows a typical cross correlation function for this type of signal. In this case, adjacent peaks are more widely separated, but at low SNR the peak value of the estimated cross correlation function may occur at one of these side lobes, yielding an estimate of the time delay greatly different from the true delay. These large estimation errors are often referred to as ambiguities for narrowband signals and as anomalous estimates for broadband signals. In the remainder of this chapter, methods of accounting for such large estimation errors in the time delay estimate are investigated.

3.3 The Barankin Bound

As noted in [23, p. 72], the Barankin bound is a greatest lower bound. However, computation of the Barankin bound proves to be exceedingly



A) CROSS CORRELATION FOR NARROWBAND SIGNAL

B) CROSS CORRELATION FOR BROADBAND SIGNAL

Figure 3.2 Sources of Large Estimation Errors

difficult. Chow and Schultheiss have investigated a simplified version of the Barankin bound to address the problem of ambiguous estimates for narrow band signals [26]. While this procedure no longer yields a greatest lower bound, the resulting lower bound has been used with some success to address the ambiguity problem. In particular, the simplified Barankin bound exhibits a distinct threshold SNR. Above this SNR, the variance of the time delay estimate is characterized by the CRLB and below this SNR the Barankin bound exceeds the CRLB by a factor proportional to the square of the ratio of the center frequency to the bandwidth (i.e., f_o^2/B^2). However, the transition from the CRLB to the Barankin bound at the threshold SNR can not be clearly specified.

Also, like the CRLB, the Barankin bound ignores the a priori information about the range of possible delay values, and the variance goes to infinity, as the SNR goes to zero. While this approach succeeds in predicting the threshold phenomena, it is not the complete solution to the problem.

3.4 The Correlator Performance Estimate (CPE)

The CPE was developed by Ianniello [27] and provides an estimate of the performance of the SCC method of TDE (see Section 2.2.1). While this approach does not yield a lower bound on time delay estimator performance, it does accurately estimate the performance of the commonly used SCC method. The CPE computes the probability of an anomalous estimate, and based on this probability, the variance of the time delay estimate is determined. The analysis presented here follows that of Ianniello [27], which in turn is based upon an approximate analysis related to pulse position modulation systems [39, pp. 623-640].

The first step is to define an anomalous estimate more precisely. Consider the SCC function, $R_{r_1r_2}(\tau)$, of equation (2-3). Define the signal correlation time as

$$T_c \triangleq R_{ss}^{-1}(o)\int_{-\infty}^{\infty} R_{ss}(\tau)\, d\tau \quad . \tag{3-10}$$

The signal is highly correlated over the correlation time, T_c, and the correlation is relatively small beyond this value. If the range of possible delay values is limited by $\pm D_o$, there are approximately $M = 2D_o/T_c$ independent values of the SCC function. Let $R_m \triangleq R_{r_1r_2}(\tau_m)$ denote these M independent values, and assume the true delay is located at one of the τ_m, say $\tau_o = D$. An anomalous estimate occurs when $|\hat{D} - D| > T_c/2$, that is when the delay estimate is farther than $T_c/2$ from the true delay. The event A, defined as

$$A \triangleq [R_m > R_o, \text{ for at least one } \tau_m] \ , \tag{3-11}$$

is a reasonable approximation to what is meant by an anomalous estimate [38, p. 629]. Since the R_m are assumed to be independent, the probability of A can be formulated analogously to the probability of error in the communication of M equally likely signals, so that [39, p. 258]

$$P[\text{anomaly}] \cong P[A] = 1 - \int_{-\infty}^{\infty} p(R_o) \left[\int_{-\infty}^{R_o} p(R_m)\, dR_m \right]^{M-1} dR_o \qquad (3\text{-}12)$$

where $p(R_o)$ is the probability density function (pdf) of R_o, and $p(R_m)$ is the pdf for any R_m, which are assumed to have identical distributions.

Now expressions for the pdf of R_o and R_m are required. Assume the signal and noise power spectra are as in (3-5). Then the autocorrelation functions $R_{ss}(\tau)$ and $R_{nn}(\tau)$ can be written as

$$R_{ss}(\tau) = S_oB\ \rho(\tau) \qquad (3\text{-}13a)$$

and

$$R_{nn}(\tau) = N_oB\ \rho(\tau) \qquad (3\text{-}13a)$$

where

$$\rho(\tau) = \frac{R_{ss}(\tau)}{R_{ss}(o)} = \frac{R_{nn}(\tau)}{R_{nn}(o)} = \frac{\sin 2\pi\tau\ B}{2\pi\tau B} \qquad (3\text{-}13c)$$

is the normalized autocorrelation function of both the signal and noise. Note that $R_{ss}(o) = S_oB$ and $R_{nn}(o) = N_oB$ are the variances of the signal and noise, respectively. For a large observation time - bandwidth product ($BT >> 1$), the mean values and variances of R_o and R_m are given by [40, p. 183]

$$E[R_o] = S_oB\ , \quad E[R_m] = 0 \quad , \qquad (3\text{-}14a)$$

$$\sigma^2_{R_o} \cong [B^2(S_o + N_o)^2 + S_o^2]\ /K\ , \qquad (3\text{-}14b)$$

and

$$\sigma^2_{R_m} \cong B^2(S_o + N_o)^2/K \ , \qquad (3\text{-}14c)$$

where

$$K = \left[\frac{1}{T}\int_{-\infty}^{\infty} \rho^2(\tau)\ d\tau\right]^{-1} = 2BT \quad . \qquad (3\text{-}14d)$$

Assuming that R_o and R_m have Gaussian distributions, their pdf's are completely specified by their means and variances in (3-14). Substitution of the expressions for $p(R_o)$ and $p(R_m)$ into (3-12) yields.

$$P[A] = 1 - \int_{-\infty}^{\infty} \left(2\pi\sigma^2_{R_o}\right)^{-1/2} \exp\left[-(R_o - S_oB)^2/2\sigma^2_{R_o}\right]$$

$$\times\left[\int_{-\infty}^{R_o} \left(2\pi\sigma^2_{R_m}\right)^{-1/2} \exp\left[-R^2_m/2\sigma^2_{R_m}\right] dR_m\right]^{M-1} dR_o \quad . \qquad (3\text{-}15)$$

Making the change of variable $x = R_o/\sigma_{R_o}$ and $y = R_m/\sigma_{R_m}$, (3-15) can be written as

$$P[A] = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp[-(x - \overline{x})^2/2] \cdot \left[\int_{-\infty}^{\lambda x} \frac{1}{\sqrt{2\pi}} \exp[-y^2/2]\ dy\right]^{M-1} dx \qquad (3\text{-}16a)$$

where

$$\overline{x} = \frac{BS_o}{\sigma_{R_o}} = \frac{\sqrt{2BT}\ SNR}{\left[SNR^2 + (SNR + 1)^2\right]^{1/2}} \quad , \qquad (3\text{-}16b)$$

$$\lambda = \sigma_{R_o}/\sigma_{R_m} = \left[1 + \frac{SNR^2}{(SNR + 1)^2}\right]^{1/2} \quad , \qquad (3\text{-}16c)$$

and where SNR $\triangleq S_o/N_o$. Evaluating (3-10) yields $T_c = 1/(2B)$, thus, the number of independent values of the SCC function is $M = D_oB$. Equation (3-16a) must be evaluated numerically to obtain the probability of anomaly for a given set of parameters, B, T, D_o and SNR, (see Appendix C for computational details).

The variance of the time delay estimate can be computed as the probability of anomaly times the variance, given an anomaly, plus the probability of no anomaly times the variance, given no anomaly. If no anomaly occurs, the estimation error is small and the CRLB applies. When an anomaly does occur, it has been assumed that the anomalous estimate is equally likely to occur at any of the τ_m. Therefore, the variance, given no anomaly, is approximately that of a random variable uniformly distributed in the interval $[-D_o, D_o]$, or $D_o^2/3$. Then, the CPE for the variance of the time delay estimate is given by

$$\sigma^2_{CPE} = P[A]\, D_o^2/3 + (1 - P[A])\, \sigma^2_{CRLB} \quad . \qquad (3\text{-}17)$$

The CPE and the CRLB are compared in Figure 3.3. The CPE is characterized by three regions: 1) at low SNR, $P[A] \cong 1$ and the variance is limited by prior information about the range of possible delay values; 2) at intermediate SNR, there is a transition from the prior information limit to the CRLB, and 3) at high SNR, $P[A] \cong 0$ and the CPE coincides with the CRLB. The SNR at which the CPE begins to deviate from the CRLB is referred to as the threshold SNR, (SNR_{th} in Figure 3.3).

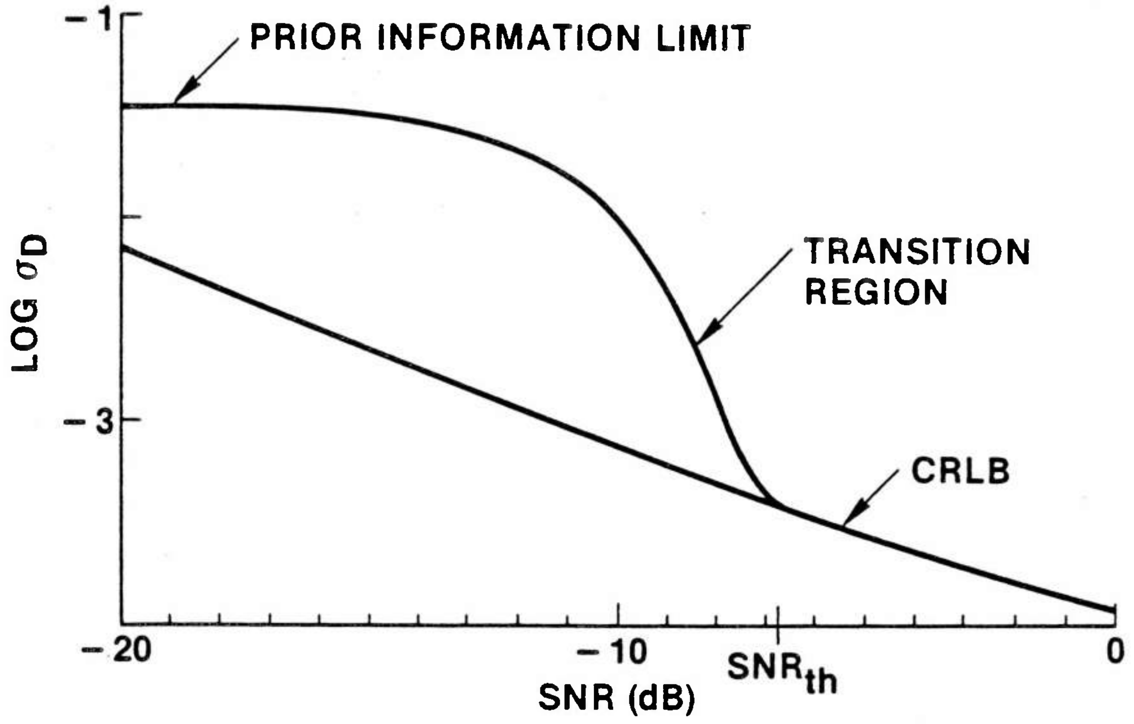
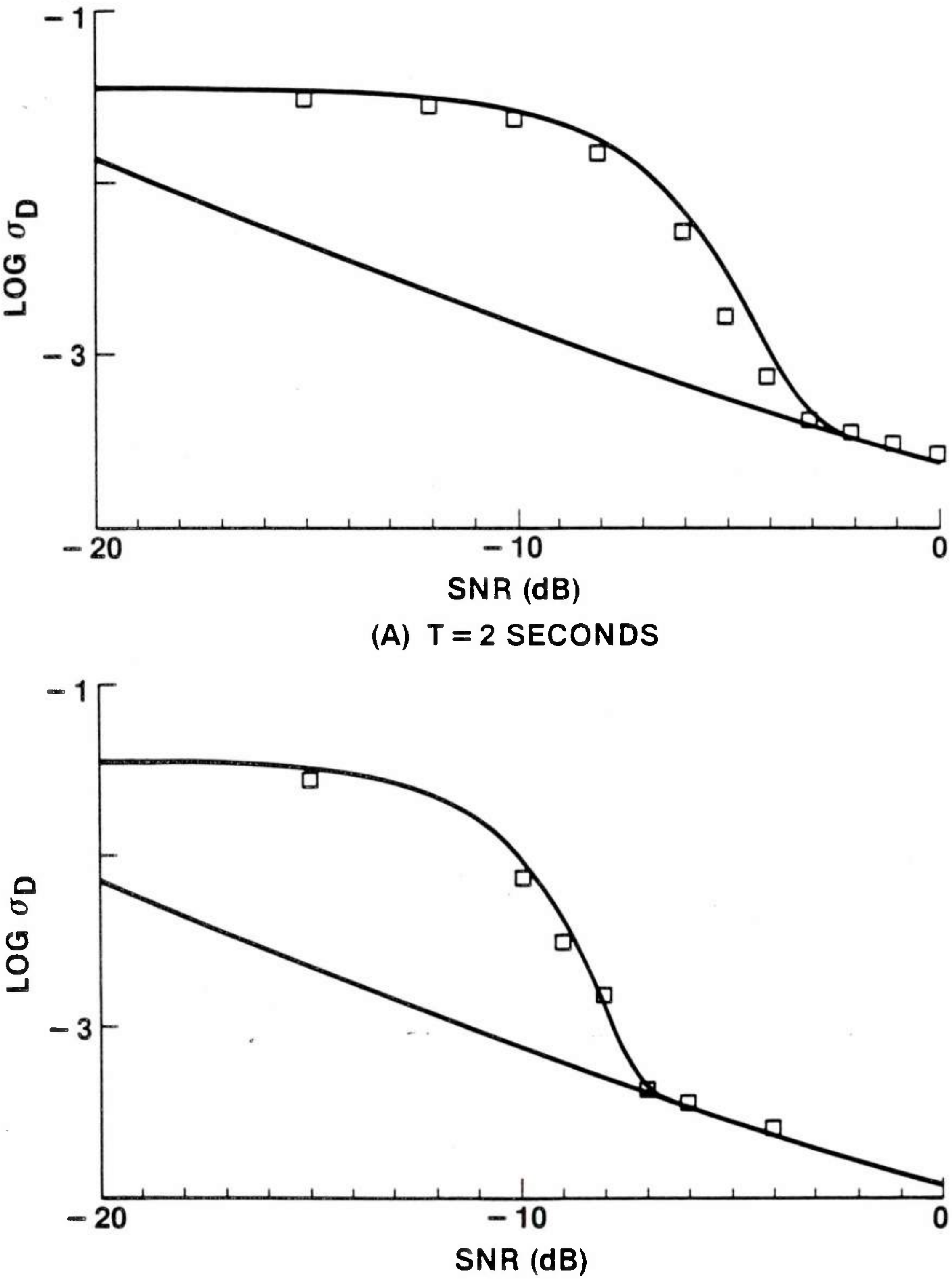


Figure 3.3 Comparison of CPE with CRLB



(A) T = 2 SECONDS

(B) T = 8 SECONDS

Figure 3.4 Comparison of CPE, CRLB and SCC Simulation Results

The CPE is compared with simulation results in Figure 3.4(a) for T = 2 seconds and in Figure 3.4(b) for T = 8 seconds. The theoretical variance is in close agreement with the experimental variance and predicts the threshold SNR within ≈ 1 dB. Although the CPE is based on an approximate analysis and several simplifying assumptions are made, Figure 3.4 shows that the CPE is, nevertheless, a very useful technique for predicting time delay estimator performance. However, the CPE was developed specifically for the SCC method of TDE and is only a performance estimate, not a performance bound. This leads one to question whether there may be a time delay estimator which can do better than the CPE predicts. Therefore, it is still of interest to find a lower bound for TDE, which is tighter than the CRLB. Note that if a lower bound could be found, which is close to the CPE, this would imply that the SCC method is nearly optimal.

3.5 The Ziv-Zakai Lower Bound (ZZLB)

The ZZLB was derived by Ziv and Zakai in [41], for the signal parameter estimation problem in communications. Improved versions of this bound were proposed independently by Chazan, Zakai and Ziv [42] and by Bellini and Tartar [43]. Recently, the ZZLB has been applied to the time delay estimation problems in works by Weiss, Weinstein and Ianniello [28–31]. In particular, the ZZLB for the variance of the time delay estimate is derived for the case of narrowband signals in [29] and for the case of low band signals in [30]. In this section the ZZLB is derived for the broadband signals of (3-5), following the procedure outlined in [30].

The ZZLB is based on the probability of error in deciding between two hypothesized values, say τ and $\tau + \Delta$, $(\Delta > 0)$, of the parameter to be estimated. Here the parameter of interest is again the time delay, D. Let $\hat{D}$ be an estimate of the true delay, obtained from an arbitrary estimation scheme. Now consider the following suboptimal decision procedure:

$$\text{if } |\hat{D} - \tau| < |\hat{D} - \tau - \Delta| \quad \text{, then } D = \tau,$$

$$\text{if } |\hat{D} - \tau| \geq |\hat{D} - \tau - \Delta| \quad \text{, then } D = \tau + \Delta. \tag{3-18}$$

That is, the decision rule selects τ, if the estimate is closer to τ than to $\tau + \Delta$, and selects $\tau + \Delta$, if $\hat{D}$ is closer to $\tau + \Delta$ than to τ. Note that the optimum decision rule would be based on the likelihood ratio test as discussed in [23, pp. 23-27]. The probability of error for the decision rule of (3-18) is given by,

$$P[\tau]\ P[\text{deciding } \tau + \Delta | \tau\ ] + P[\tau + \Delta]\ P[\text{deciding } \tau\ |\ \tau + \Delta] \tag{3-19}$$

where $P[\tau]$ denotes the probability of τ occuring and $P[\text{deciding } \tau + \Delta | \tau\ ]$ is the conditional probability of deciding that $\tau + \Delta$ is true, given that τ is true. Assuming that τ and $\tau + \Delta$ are equally likely to occur, (3-19) can be written as

$$(1/2)\ P[\hat{D} - \tau > \Delta/2 | \tau\ ] + (1/2)\ P[\hat{D} - \tau - \Delta \leq - \Delta/2\ | \tau + \Delta] \tag{3-20}$$

Defining the estimation error as $\epsilon \triangleq \hat{D} - D$, the minimum attainable probability of error for this binary decision problem, $P_e(\tau, \tau + \Delta)$, must be less than or equal to the probability of error for the suboptimum decision rule of (3-18), thus

$$P_e(\tau, \tau + \Delta) \leq (1/2)\, P[\epsilon > \Delta/2 | \tau\,] + (1/2)\, P[\epsilon \leq -\Delta/2 \mid \tau + \Delta]. \quad (3\text{-}21)$$

The range of possible delay values is again assumed to be bounded by $\pm D_0$, so that τ and Δ must satisfy the condition that $\tau,\ \tau + \Delta\ \epsilon\ [-D_0, D_0]$, or equivalently,

$$-D_o \leq \tau \leq D_o - \Delta, \quad o \leq \Delta \leq 2\, D_o. \qquad (3\text{-}22)$$

Integrating (3-21) with respect to τ over the interval $[-D_o, D_o - \Delta]$ yields

$$\int_{-D_o}^{D_o-\Delta} P_e(\tau, \tau + \Delta)\, d\tau$$

$$\leq (1/2) \int_{-D_o}^{D_o-\Delta} \Big[P[\epsilon > \Delta/2 | \tau\,] + P[\epsilon \leq - \Delta/2 |\ \tau + \Delta]\Big]\, d\tau$$

$$= (1/2) \int_{-D_o}^{D_o-\Delta} P[\epsilon > \Delta/2 | \tau\,]\, d\tau + (1/2) \int_{-D_o+\Delta}^{D_o} P[\epsilon \leq -\Delta/2 | \tau\,]\, d\tau$$

$$\leq (1/2) \int_{-D_o}^{D_o} P[\,|\epsilon| > \Delta/2 | \tau\,]\, d\tau \qquad (3\text{-}23)$$

Now let F(x) represent the average of $P[|\varepsilon| \geq x|\tau]$, where τ is uniformly distributed over $[-D_o, D_o]$, then

$$F(x) \triangleq \frac{1}{2D_o} \int_{-D_o}^{D_o} P[|\varepsilon| \geq x|\tau]\, d\tau$$

$$= \frac{1}{2D_o} P[|\varepsilon| \geq x] \qquad (3\text{-}24a)$$

and from (3-23)

$$\int_{-D_o}^{D_o-\Delta} P_e(\tau, \tau+\Delta)\, d\tau \leq D_o F(\Delta/2) \quad . \qquad (3\text{-}24b)$$

Multiplying both sides of (3-24b) by Δ/D_o and integrating with respect to Δ over $[o, 2D_o]$ yields

$$\frac{1}{D_o}\int_o^{2D_o} \Delta \int_{-D_o}^{D_o-\Delta} P_e(\tau, \tau+\Delta)\, d\tau\, d\Delta \leq \int_o^{2D_o} \Delta F(\Delta/2)\, d\Delta$$

$$= 4\int_o^{D_o} x\, F(x)\, dx \quad , \; x = \Delta/2$$

$$\leq 4\int_o^{2D_o} x\, F(x)\, dx$$

$$= 2x^2\, F(x)\Big|_o^{2D_o^+} - 2\int_o^{2D_o} x^2 dF(x) \quad . \qquad (3\text{-}25)$$

Since $F(x) = P[|\varepsilon| \geq x]$, it can always be assumed that $F(2D_o^+) = 0$. Note if $F(2D_o^+) \neq 0$, this implies that there is a non-zero probability that the estimation error $|\varepsilon| > 2D_o$, or $\hat{D}$ is more than $2D_o$ from the true delay, D. However, it is known a priori that $D \in [-D_o, D_o]$. Thus if $F(2D_o^+) \neq 0$ the estimate can be improved by an obvious modification (e.g., let $\hat{D} = 0$ if $|\hat{D}| > D_o$). Also note that the mean square error, or for an unbiased estimator, the variance of the time delay estimate is given by

$$\sigma_{\hat{D}}^2 = -\int_0^{2D_o} x^2 \, dF(x) \, . \tag{3-26}$$

The ZZLB of [42] is obtained by substituting (3-26) into (3-25) and solving for σ_D^2,

$$\sigma_{\hat{D}}^2 \geq \frac{1}{2D_o} \int_0^{2D_o} \Delta \int_{-D_o}^{D_o-\Delta} P_e(\tau, \tau + \Delta) \, d\tau \, d\Delta \quad . \tag{3-27}$$

In [43], Bellini and Tartara note that F(x) is a non-increasing function of x. Therefore, a tighter lower bound for the right hand side of (3-24b) is given by

$$G\left[\int_{-D_o}^{D_o-\Delta} P_e(\tau, \tau + \Delta) \, d\tau\right] \leq D_o \, F(\Delta/2) \tag{3-28}$$

where $G[\cdot]$ is a non-increasing function of Δ, obtained by filling in the valleys, if any exist, of the bracketed function. This "valley filling" process is illustrated in Figure 3.5. Proceeding as before, the modified ZZLB is given by

$$\sigma_{\hat{D}}^2 \geq \frac{1}{2D_o} \int_o^{2D_o} \Delta \quad G \left[\int_{-D_o}^{D_o-\Delta} P_e(\tau, \ \tau + \Delta) d\tau \right] d\Delta. \qquad (3\text{-}29)$$

In general, it is not possible to obtain a closed form expression for $P_e(\tau, \ \tau + \Delta)$. For the TDE model of (2-1), the Chernoff bound can be used to obtain a good approximation of $P_e(\tau, \ \tau + \Delta)$. The calculation of this approximation is given in Appendix D. Under a large observation time-bandwidth product (BT >> 1), it is found that

$$P_e(\tau, \ \tau + \Delta) \approx P_e(\Delta) \ , \text{ independent of } \tau \qquad (3\text{-}30a)$$

and

$$P_e(\Delta) = \exp[a(\Delta) + b(\Delta)]\cdot Q(\sqrt{2b(\Delta)}) \quad , \qquad (3\text{-}30b)$$

where

$$a(\Delta) = -T\int_0^\infty \ln\left[1 + \gamma(f, \ \Delta)\right] df \quad , \qquad (3\text{-}30c)$$

$$b(\Delta) = T\int_0^\infty \gamma(f, \ \Delta)/(1 + \gamma(f, \ \Delta)) \ df \quad , \qquad (3\text{-}30d)$$

$$\gamma(f, \ \Delta) = \frac{G_{ss}^2(f) \ \sin^2\pi f\Delta}{[G_{n_1n_1}(f) + G_{n_2n_2}(f)] \ G_{ss}(f) + G_{n_1n_1}(f) \ G_{n_2n_2}(f)} \ , \qquad (3\text{-}30e)$$

and

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-y^2/2} \ dy \ . \qquad (3\text{-}30f)$$

Additionally, it is shown in Appendix D, that $P_e(\Delta)$ is bounded by

$$P_e(\Delta) \geq \exp\left[\frac{-T}{2}\int_0^\infty \gamma^2(f,\Delta)\, df\right] \cdot Q\left[\left(2T\int_0^\infty \gamma(f,\Delta)\, df\right)^{1/2}\right]. \tag{3-31}$$

Since $P_e(\Delta)$ is independent of τ, the expression for the ZZLB in (3-27) can be simplified to yield

$$\sigma_{\hat{D}}^2 \geq \frac{1}{2D_o}\int_0^{2D_o} \Delta\, G[(2D_o-\Delta)\, P_e(\Delta)]d\Delta\ . \tag{3-32}$$

Now an expression for the ZZLB is derived for the specific signal and noise spectra of (3-5). Equation (3-30e) can be written as

$$\gamma(f,\Delta) = \frac{SNR^2 \sin^2 \pi f\Delta}{2\ SNR + 1} \tag{3-33a}$$

$$= SNR' \sin^2 \pi f\Delta \tag{3-33b}$$

where

$$SNR' \triangleq \frac{SNR^2}{2SNR + 1} \tag{3-33c}$$

and where SNR is defined in (3-6). Substituting (3-33b) into (3-30c) and (3-30d) yields

$$a(\Delta) = -T\int_0^B \ln(1 + SNR' \sin^2 \pi f\Delta)\, df \tag{3-34a}$$

and

$$b(\Delta) = T\int_0^B SNR' \sin^2 \pi f\Delta/(1 + SNR' \sin^2 \pi f\Delta)\, df\ . \tag{3-34b}$$

Taking the derivative of a(Δ) and b(Δ) with respect to Δ, one obtains after simplifying,

$$\frac{da(\Delta)}{d\Delta} = a'(\Delta) = -T\int_0^B \frac{\alpha(f)}{\beta(f)}\, df \tag{3-35a}$$

$$\frac{db(\Delta)}{d\Delta} = b'(\Delta) = T\int_0^B \frac{\alpha(f)}{\beta(f)^2}\, df \tag{3-35b}$$

$$\frac{d}{d\Delta}\,[a(\Delta) + b(\Delta)] = T\int_0^B \frac{\alpha(f)}{\beta(f)} \left(\frac{1}{\beta(f)} - 1\right) df \tag{3-35c}$$

where

$$\alpha(f) = \pi f\ \mathrm{SNR}'\ \sin 2\pi f\Delta \tag{3-35d}$$

$$\beta(f) = 1 + \mathrm{SNR}'\ \sin^2 \pi f\Delta\quad . \tag{3-35e}$$

With a little thought, it can be seen that for $\Delta \in [0, 1/2B]$, $b'(\Delta) \geq 0$ with equality holding only for $\Delta = 0$. Then, since $b(\Delta) \geq 0$ for all Δ, b(Δ) is a positive monotonically increasing function of Δ. Thus, the term $Q(\sqrt{2b(\Delta)})$ of equation (3-30b) is monotonically decreasing for $\Delta \in [0, 1/2B]$. Similarly, $a'(\Delta) + b'(\Delta) \leq 0$ for all $\Delta \in [0, 1/2B]$ with equality holding only for $\Delta = 0$. From equation (D-9), $a(\Delta) + b(\Delta) \leq 0$ for all Δ, so that a(Δ) + b(Δ) is a negative monotonically decreasing function for $\Delta \in [0, 1/2B]$. Thus, the term exp[a(Δ) + b(Δ)] in (3-30b) is also monotonically decreasing. Therefore, for $\Delta \in [0, 1/2B]$, $P_e(\Delta)$ is a monotonically decreasing function, and the probability of error in correctly deciding between the hypothesized delays, τ and τ + Δ, decreases as the separation, Δ, increases. This is an intuitively pleasing observation.

Next consider the behavior of $P_e(\Delta)$ for $\Delta > 1/2B$. Using the change of variable $\Omega = f\Delta$ in (3-34a) and (3-34b) yields

$$a(\Delta) = \frac{-T}{\Delta} \int_0^{B\Delta} \ln(1 + \mathrm{SNR}' \sin^2\pi\Omega)\, d\Omega \tag{3-36a}$$

and

$$b(\Delta) = \frac{T}{\Delta} \int_0^{B\Delta} \mathrm{SNR}' \sin^2\pi\,\Omega/(1 + \mathrm{SNR}' \sin^2\pi\Omega)\, d\Omega. \tag{3-36b}$$

The integrands in these equation are functions of $\sin^2\pi\Omega$ which is periodic in Ω with a period of 1. Note also that $\sin^2\pi\Omega$ is an even function. Then defining $\Delta_n = n/2B$, where n is a positive integer, (3-36) becomes

$$a(\Delta_n) = -\frac{BT}{n} \int_{-n/2}^{n/2} \ln(1 + \mathrm{SNR}' \sin^2\pi\Omega)\, d\Omega \tag{3-37a}$$

and

$$b(\Delta_n) = \frac{BT}{n} \int_{-n/2}^{n/2} \mathrm{SNR}' \sin^2\pi\Omega\,/(1 + \mathrm{SNR}' \sin^2\pi\Omega)\, d\Omega. \tag{3-37b}$$

Now (3-37a) can be solved for $a(\Delta_n)$ as follows,

$$a(\Delta_n) = -BT \int_{-1/2}^{1/2} \ln(1 + \mathrm{SNR}' \sin^2\pi\Omega)\, d\Omega$$

$$= -BT \int_0^1 \ln(1 + \frac{\mathrm{SNR}'}{2} - \frac{\mathrm{SNR}'}{2} \cos\pi\Omega')\, d\Omega', \quad \Omega' = 2\Omega \tag{3-38}$$

This integral can be found in integral tables [44, p. 461, eq'n 709] and after some algebra

$$a(\Delta_n) = -2BT \ln[(1 + \sqrt{1 + \mathrm{SNR}'})/2] \ . \tag{3-39}$$

Similarly $b(\Delta_n)$ can be written as

$$b(\Delta_n) = 2BT \int_0^{1/2} \mathrm{SNR}' \sin^2 \pi\Omega \,/(1 + \mathrm{SNR}' \sin^2 \pi\Omega)\, d\Omega. \tag{3-40}$$

Again resorting to the integral tables [44, p. 432, eq'n 343] and a little algebra, it can be shown that

$$b(\Delta_n) = BT \frac{\sqrt{1 + \mathrm{SNR}'} - 1}{\sqrt{1 + \mathrm{SNR}'}} \quad . \tag{3-41}$$

Substituting (3-39) and (3-41) into the expression for $P_e(\Delta)$ in (3-30b) gives

$$\begin{aligned} P_e(\Delta_n) &= \exp\left\{-BT\left[2\ln\left(\frac{1 + \ 1 + \mathrm{SNR}'}{2}\right) - \frac{\sqrt{1 + \mathrm{SNR}'} - 1}{\sqrt{1 + \mathrm{SNR}'}}\right]\right\} \\ &\qquad \times Q\left[\left(2BT \frac{\sqrt{1 + \mathrm{SNR}'} - 1}{\sqrt{1 + \mathrm{SNR}'}}\right)^{1/2}\right] \\ &\triangleq P_e \end{aligned} \tag{3-42}$$

This observation leads to a bound for the quantity $G[(2D_o - \Delta)\ P_e(\Delta)]$ in (3-32). The procedure followed to obtain the bound is described pictorially in Figure 3.5. A possible curve for $P_e(\Delta)$ is sketched in Figure 3.5(a), and the corresponding curves for $(2D_o - \Delta)\ P_e(\Delta)$ and $G[(2D_o - \Delta)\ P_e(\Delta)]$ are sketched in Figure 3.5(b). Define the rectangular gate function $U(\Delta, \Delta_n)$ as

$$U(\Delta, \Delta_n) = \begin{cases} 1 \ , & \Delta_{n-1} \leq \Delta \leq \Delta_n \\ 0 \ , & \text{elsewhere} \ . \end{cases}$$

Then, as seen from Figure 3.5(c),

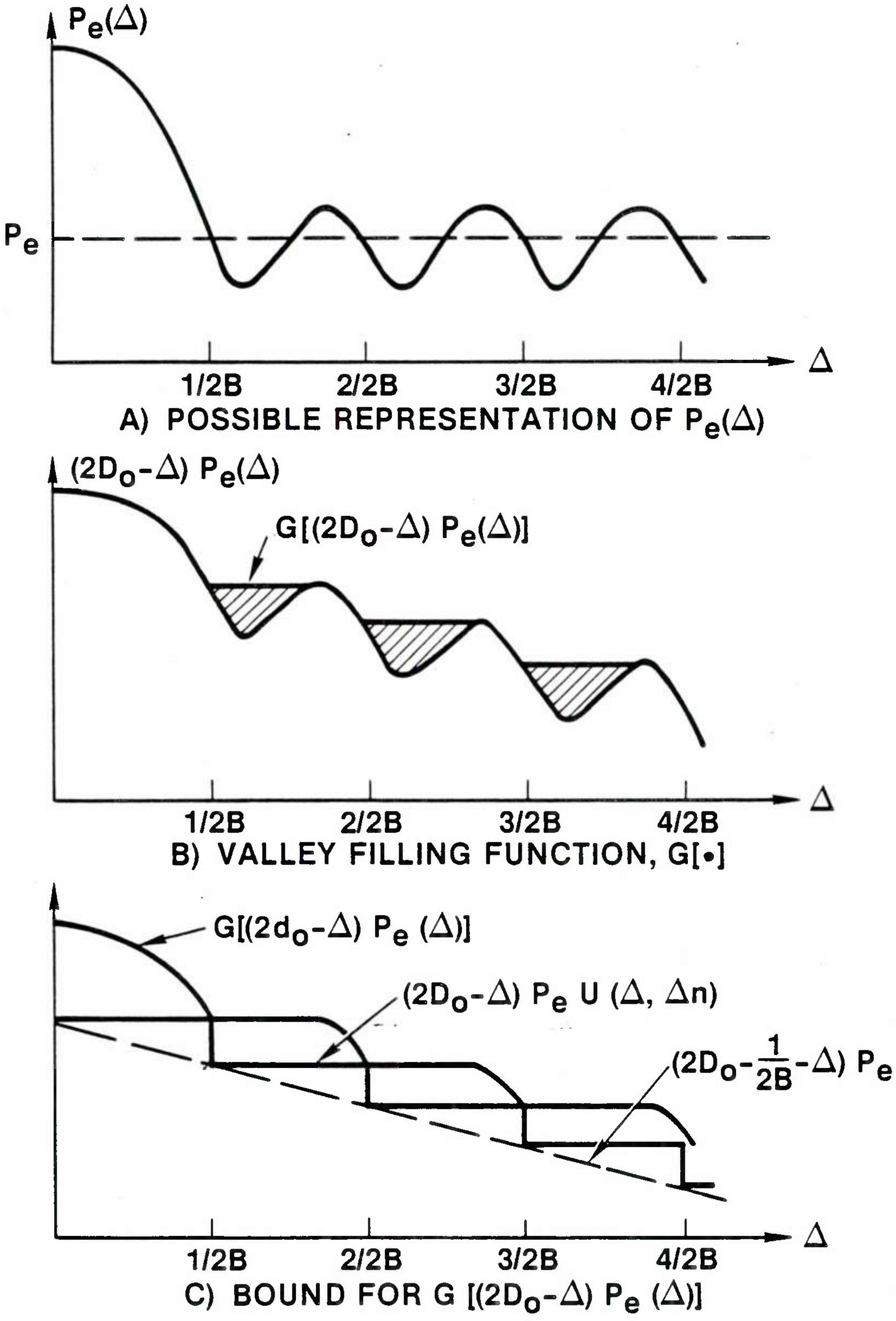
$$G[(2D_o - \Delta)\ P_e(\Delta)] \geq (2D_o - \Delta_n)\ P_e\ U(\Delta, \Delta_n) \qquad (3\text{-}43a)$$

$$\geq (2D_o - 1/(2B) - \Delta)\ P_e. \qquad (3\text{-}43b)$$

The expression on the right hand side of (3-43a) is represented by the decreasing step function in Figure 3.5(c), and the linear, dashed line represents the final expression in (3-43b).

This bound will prove to be sufficiently tight except near $\Delta = 0$, where the value of $P_e(\Delta)$ cannot be closely approximated by P_e. In this region, consider the bound in (3-31) for $P_e(\Delta)$,

$$P_e(\Delta) \geq \exp\left[-\frac{T}{2}\int_0^\infty \gamma^2(f,\Delta)\ df\right] \cdot Q\left[\left(2T\int_0^\infty \gamma(f,\Delta)\ d\Delta\right)^{1/2}\right]. \qquad (3\text{-}31)$$



A) POSSIBLE REPRESENTATION OF $P_e(\Delta)$

B) VALLEY FILLING FUNCTION, $G[\bullet]$

C) BOUND FOR $G\ [(2D_o - \Delta)\ P_e\ (\Delta)]$

Figure 3.5 Obtaining a Lower Bound for $G[(2D_o - \Delta)\ P_e(\Delta)]$

Now, simplifying the arguments of exp[·] and Q(·) in (3-31),

$$2T\int_0^\infty \gamma(f,\Delta)\, d\Delta \;=\; 2T\int_0^B \text{SNR}'\, \sin^2\pi f\Delta\, df$$

$$\leq 2T\int_0^B \text{SNR}'\, (\pi f\Delta)^2\, df$$

$$= \eta^2\, \Delta^2 \qquad (3\text{-}44a)$$

where

$$\eta^2 = 2\pi^2\, \text{SNR}'\, B^3 T/3 \quad . \qquad (3\text{-}44b)$$

Similarly,

$$-\frac{T}{2}\int_0^\infty \gamma^2(f,\Delta)\, df \;\geq\; -\frac{T}{2}\int_0^B \text{SNR}'^2\, (\pi f\Delta)^4\, df$$

$$= -\nu^4\Delta^4 \qquad (3\text{-}45a)$$

where

$$\nu^4 = \pi^4\, \text{SNR}'^2\, B^5\, T/10. \qquad (3\text{-}45b)$$

Substituting (3-44) and (3-45) into (3-31) yields

$$P_e(\Delta) \geq \exp(-\nu^4\Delta^4)\, Q(\eta\Delta), \quad \text{for small } \Delta\ . \qquad (3\text{-}46)$$

Then, from (3-43b) and (3-46), the function $G[(2D_0 - \Delta)\ P_e(\Delta)]$ is bounded by

$$G[(2D_o - \Delta)\ P_e(\Delta)] \geq \begin{cases} (2D_o - \Delta)\ \exp(-\nu^4\Delta^4)\ Q(\eta\Delta) & ,\ o \leq \Delta \leq z \\ (2D_o - \frac{1}{2B} - \Delta)\ P_e & ,\ z \leq \Delta \leq 2D_o - \frac{1}{2B} \\ o & ,\ 2D_o - \frac{1}{2B} \leq \Delta \leq 2D_o. \end{cases} \qquad (3\text{-}47)$$

The value for z is yet to be defined. Note that $(2D_o - 1/(2B) - \Delta) < o$ for $\Delta > 2D_o - 1/(2B)$. Therefore, the bound is set to zero in the third region in (3-47).

The choice of z is somewhat arbitrary, but compare the expressions for $P_e(z)$ in equations (3-31) and (3-42). Select z to satisfy the condition

$$\eta\ z = \sqrt{2b(\Delta_n)}$$

$$= \left[2BT\ \frac{\sqrt{1 + SNR'} - 1}{\sqrt{1 + SNR'}}\right]^{1/2} . \qquad (3\text{-}48)$$

Thus, the arguments for $Q(\cdot)$ are chosen so that Q is continuous between regions 1 and 2 in (3-47). Substituting (3-44b) into (3-48) and solving for z yields,

$$z = \frac{\sqrt{3}}{\pi\ B} \left[\frac{1 - (1 + SNR')^{-1/2}}{SNR'}\right]^{1/2} \qquad (3\text{-}49)$$

As SNR' → ∞ , z → 0, and as SNR' → 0 , application of L'Hospital's rule shows that

$$z \to \sqrt{3} \Big/ (\sqrt{2}\,\pi B) \simeq \frac{1}{2.5\ B}\ ;\quad \text{thus, } 0 \le z \le \frac{1}{2\ B}\ .$$

The expression for the ZZLB is obtained by substituting (3-47) into (3-32) to give

$$\sigma^2_{\hat{D}} \ge \frac{1}{2D_o}\left[\int_0^z \Delta\,(2D_o - \Delta)\,\exp(-\nu^4\Delta^4)\,Q(\eta\Delta)\,d\Delta \right.$$

$$\left. + \int_z^{2D_o - 1/(2B)} \Delta\,(2D_o - \frac{1}{2B} - \Delta)\,P_e\,d\Delta\right] \qquad (3\text{-}50)$$

The two integral terms on the right hand side of (3-50) will be considered separately. The first integral can be simplified as follows

$$\int_0^z \Delta\,(2D_o - \Delta)\,\exp(-\nu^4\Delta^4)\,Q(\eta\Delta)\,d\Delta$$

$$= \frac{1}{\eta^3}\int_0^{\eta z} y\,(2\eta D_o - y)\,\exp(-\nu^4 y^4/\eta^4)\,Q(y)\,dy,\ y = \eta\Delta. \qquad (3\text{-}51)$$

Now consider the term ν^4/η^4, using (3-44b) and (3-45b), yields

$$\nu^4/\eta^4 = \frac{9}{40BT} << 1 \qquad (3\text{-}52)$$

since BT >> 1 by assumption. Since the major contribution of the integral in (3-51) is for small values of y, the exponential term can be approximated by 1. Also note that over the range of the integral in (3-51)

$$2\eta D_o - y \leq \eta \; (2D_o - z) \approx 2\eta D_o \quad . \tag{3-53}$$

Then (3-51) becomes

$$\int_o^z \Delta \; (2D_o - \Delta) \; \exp(-\nu^4 \Delta^4) \; Q(\eta\Delta) \; d\Delta$$

$$\approx \frac{2D_o}{\eta^2} \int_o^{\eta z} y \; Q(y) \; dy \tag{3-54}$$

The second integral in (3-50) can be immediately evaluated to give

$$\int_z^{2D_o - 1/(2B)} \Delta \; (2D_o - 1/(2B) - \Delta) \; P_e \; d\Delta$$

$$= P_e \left[\left(2D_o - \frac{1}{2B} \right) \frac{\Delta^2}{2} - \frac{\Delta^3}{3} \right] \Bigg|_z^{2D_o - 1/(2B)}$$

$$\approx P_e \left[\; D_o \Delta^2 - \Delta^3/3 \right] \Bigg|_z^{2D_o} \quad , \; 2D_o >> \frac{1}{2B}$$

$$= P_e \; [4D_o^3/3 - D_o \; z^2 - z^3/3]$$

$$\approx 4P_e \; D_o^3/3 \quad , \; D_o >> z \tag{3-55}$$

Substituting (3-54) and (3-55) into (3-50) yields

$$\sigma_{\hat{D}}^2 \geq \frac{1}{\eta^2} \int_o^{\eta z} y \; Q(y) dy + 2P_e D_o^2/3 \quad . \tag{3-56}$$

Thus, a closed form expression for the Ziv-Zakai lower bound for the variance of the time delay is given by (3-56).

One final simplification can be made by evaluating the integral term in (3-56) as follows, using integration by parts

$$\int_{o}^{\eta z} y\ Q(y)\ dy = \left.\frac{y^2 Q(y)}{2}\right|_{o}^{\eta z} - \int_{o}^{\eta z} \frac{y^2 Q'(y)}{2}\ dy$$

$$= \eta^2 z^2\ Q(\eta z)/2 - (1/2)\int_{0}^{\eta z} y^2 Q'(y)\ dy \qquad (3\text{-}57)$$

Recall that

$$Q(y) = (2\pi)^{-1/2}\int_{y}^{\infty} e^{-\alpha^2/2}\ d\alpha\ . \qquad (3\text{-}58)$$

Using Leibniz rule yields

$$Q'(y) = -(2\pi)^{-1/2}\ e^{-y^2/2}\ . \qquad (3\text{-}59)$$

Then the integral term on the right hand side of (3-57) becomes

$$\frac{1}{2}\int_{0}^{\eta z} y^2 \phi'(y)\ dy = \frac{-1}{2\sqrt{2\pi}}\int_{0}^{\eta z} y\ .\ y\ e^{-y^2/2}\ dy \qquad (3\text{-}60a)$$

$$= \frac{1}{2\sqrt{2\pi}}\left[\eta z\ \exp(-\eta^2 z^2/2) - \int_{0}^{\eta z} e^{-y^2/2}\ dy\right] \qquad (3\text{-}60b)$$

$$= \frac{\eta z}{2\sqrt{2\pi}}\ \exp(-\eta^2 z^2/2) + \frac{1}{2}\ \phi(\eta z) - \frac{1}{4}\quad , \qquad (3\text{-}60c)$$

where (3-60b) is obtained from (3-60a) by integration by parts, and (3-60c) is obtained by re-writing the integral term in (3-60b) in terms of $\phi(y)$. Finally substituting (3-60) into (3-57) and (3-57) into (3-56), the final expression for the ZZLB is given by

$$\sigma_{\hat{D}}^2 \geq \frac{1}{2\eta^2}\left[[(\eta z)^2 - 1]\,\phi(\eta z) - \frac{\eta z}{\sqrt{2\pi}}\exp(-\eta^2 z^2/2) + \frac{1}{2}\right] + 2\,P_e\,D_o^2/3 \qquad (3\text{-}61)$$

where ηz is defined in (3-48), η^2 is given by (3-44b), and P_e is given by (3-42).

As SNR $\rightarrow 0$, $P_e \rightarrow 1/2$ (see (3-42)) and $\eta z \rightarrow 0$ (see (3-48)), and the ZZLB reduces to

$$\sigma_{ZZLB}^2 = D_o^2/3 \quad . \qquad (3\text{-}62)$$

This is the same expression obtained from the CPE for low SNR, when P[anomaly] = 1. At low SNR, the variance is limited by the a priori limit on the range of possible delay values. At high SNR, $P_e \rightarrow 0$ and $\eta z \rightarrow \sqrt{2BT}$, and (3-61) becomes

$$\sigma_{ZZLB}^2 = \frac{1}{4\eta^2} \quad . \qquad (3\text{-}63)$$

Substituting for η^2 from (3-44b) and expressing SNR' in terms of SNR using (3-33c) yields

$$\sigma_{ZZLB}^2 = \frac{3}{8\pi^2} \cdot \frac{1}{B^3T} \cdot \frac{1 + 2SNR}{SNR}$$

$$= \sigma_{CRLB}^2 \quad . \qquad (3\text{-}64)$$

At high SNR, the ZZLB coincides with the CRLB. Thus, the ZZLB exhibits behavior similar to that of the CPE. The ZZLB, CPE and CRLB are compared in Figure 3.6. The ZZLB is characterized by a prior information limit, a transition region, and the CRLB at low, intermediate and high SNR, respectively. The only difference between the CPE and ZZLB is that the ZZLB is somewhat smaller than the CPE in the transition region and predicts a threshold SNR that is somewhat lower than that predicted by the CPE. This is not surprising since the ZZLB is a lower bound and the CPE is only a performance estimate. A comparison of simulation results for the SCC method of TDE with the ZZLB is shown in Figure 3.7(a) for T = 2 seconds and in Figure 3.7(b) for T = 8 seconds. Note, these are the same simulation results as in Figures 3.1 and 3.4. The simulation results and the ZZLB are seen to be in good agreement, with the ZZLB predicting the threshold SNR extremely well.

A listing of a computer program which calculates the CRLB, the CPE and the ZZLB is given at the end of Appendix C. Also included are listings of subroutines which evaluate the probability of anomaly for the CPE and the Q function required for both the CPE and the ZZLB.

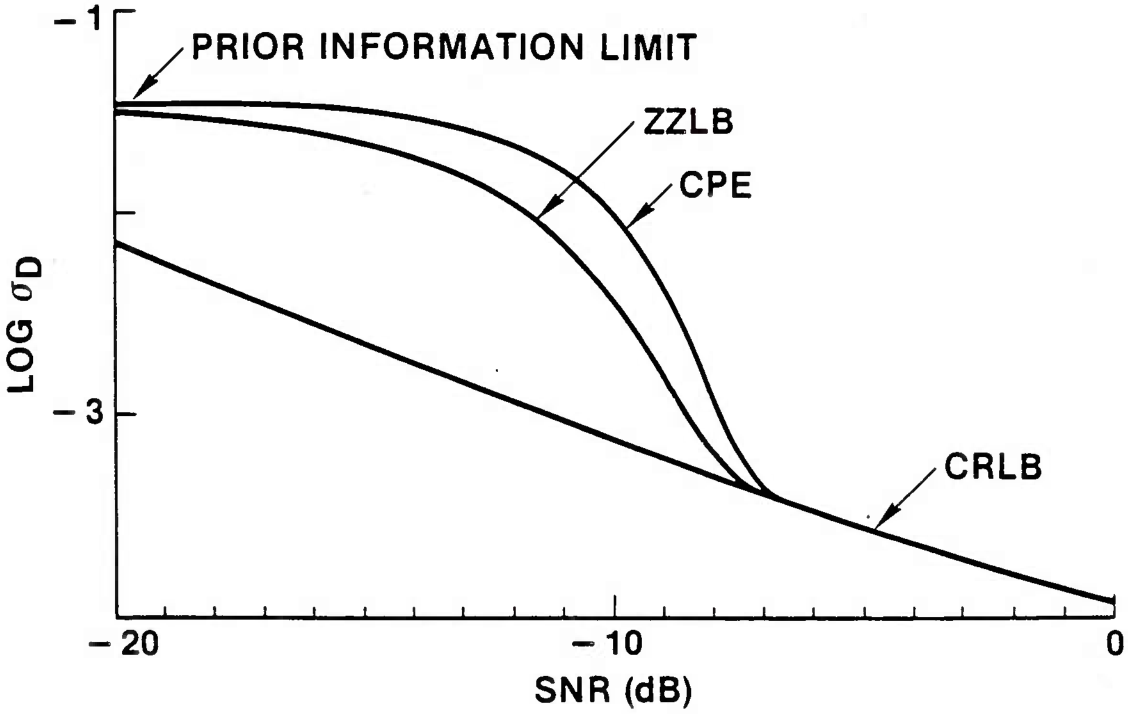
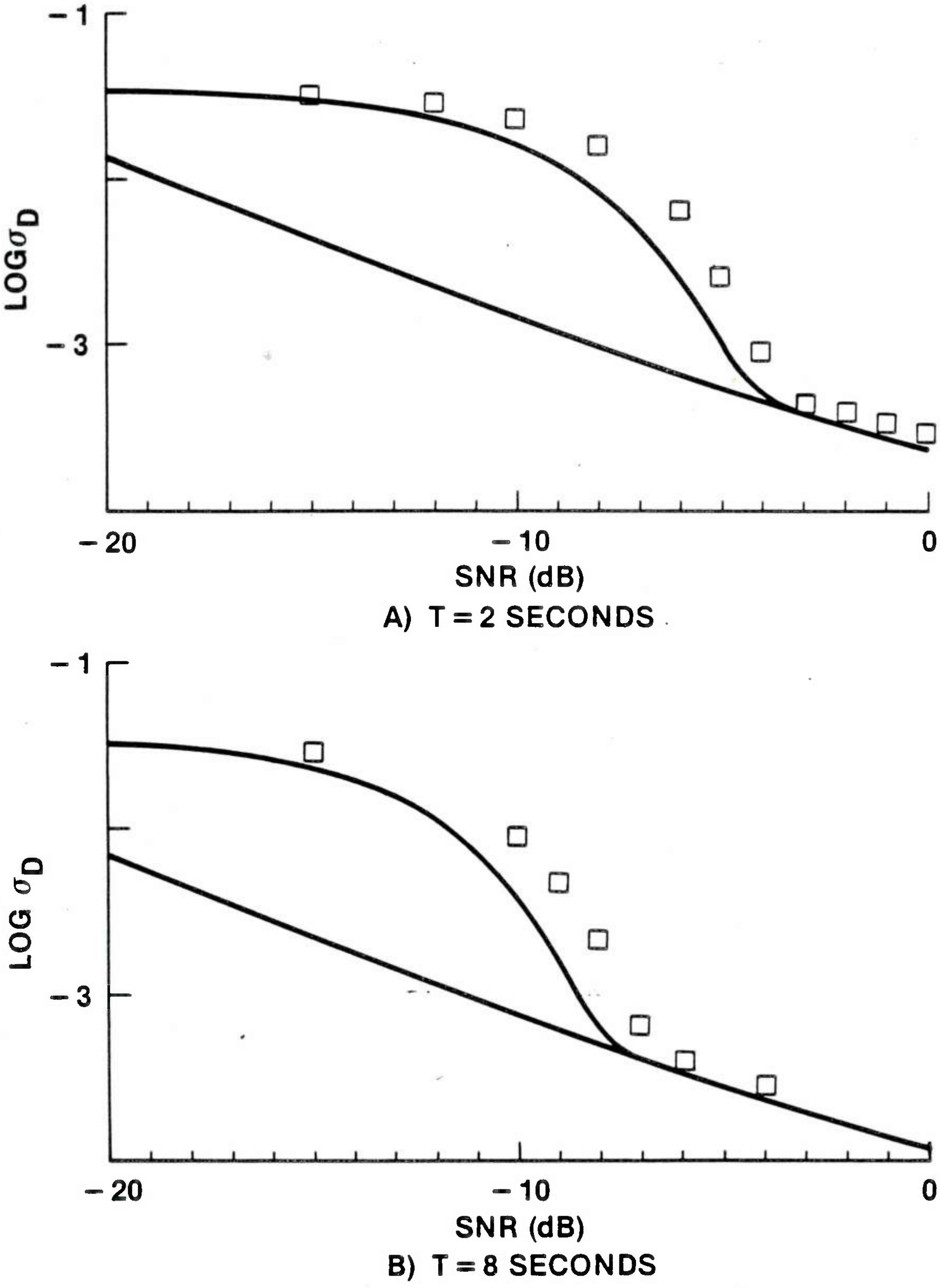


Figure 3.6 Comparison of ZZLB with CRLB and CPE



A) T = 2 SECONDS

B) T = 8 SECONDS

Figure 3.7 Comparison of ZZLB, CRLB and SCC Simulation Results

CHAPTER 4

THRESHOLD EFFECT CONSIDERATIONS FOR TDE

The CPE and the ZZLB both predict a threshold effect in the performance of TDE methods. Above a threshold SNR, time delay estimator performance is characterized by the CRLB, while below the threshold, performance degrades rapidly relative to the CRLB. Simulation results for the SCC method corroborate these predictions by the CPE and the ZZLB, as seen in Chapter 3. In this chapter, the threshold effect is investigated in greater detail and the implications of this effect are considered as related to coherent and incoherent processing techniques of TDE. Also, additional simulation results are presented to further substantiate the theoretical predictions and to allow comparison of several different GCC methods.

4.1 Behavior of the Threshold SNR

The effect on the ZZLB of varying the observation time, while keeping B and D_0 fixed, is shown in Figure 4.1. The threshold SNR decreases as the observation, or coherent processing time, T, is increased. However, the value of the variance of the time delay estimate (or equivalently, log σ_D, as in Figure 4.1) at which the ZZLB begins to deviate from the CRLB remains essentially constant, independent of the observation time. To attain CRLB performance at SNR below the threshold SNR for a given observation time, requires that the observation time be increased sufficiently to lower the threshold SNR below the desired operating SNR.

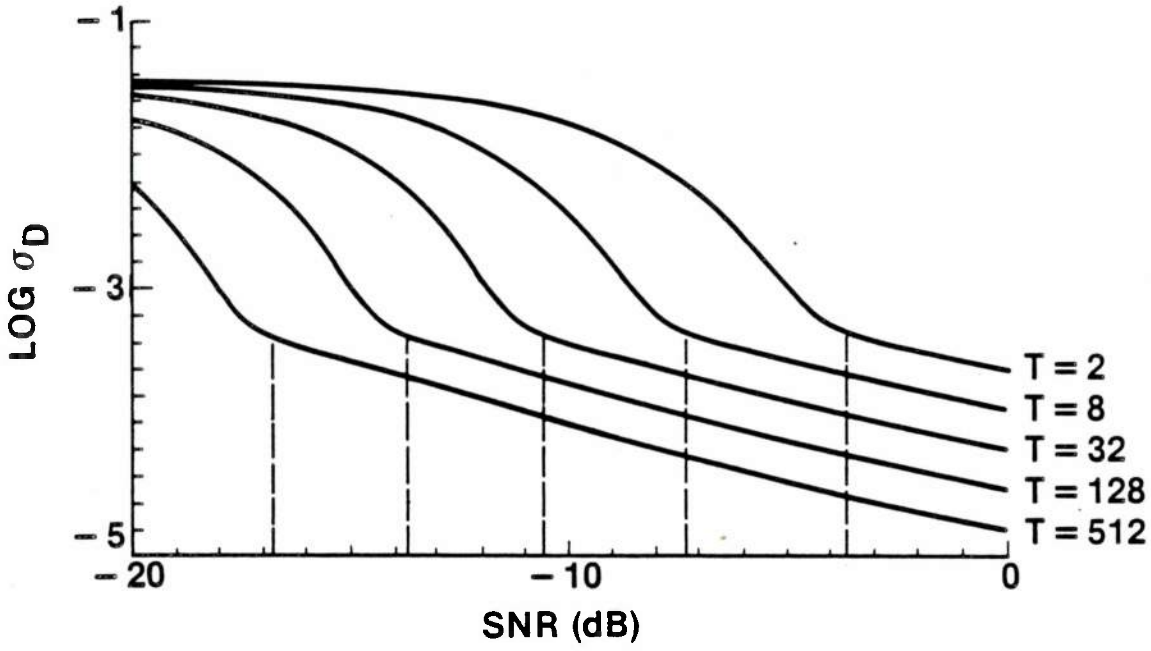


Figure 4.1 Effect of Varying Observation Time on ZZLB

(B = 100 Hz, D_o = 1/16 second, T = 2, 8, 32, 128, 512 seconds)

Consider the CRLB and the ZZLB as functions of the SNR, with all other parameters held constant. Below the threshold SNR, the ZZLB quickly becomes an order of magnitude larger than the CRLB. Thus, as suggested in [31], the threshold SNR can be approximated by the SNR which satisfies the condition

$$\sigma^2_{ZZLB} = 2\sigma^2_{CRLB} , \tag{4-1}$$

where σ^2_{ZZLB} and σ^2_{CRLB} are given by (3-61) and (3-64), respectively. The particular SNR for which (4-1) holds is given by

$$SNR'_{th} \simeq \frac{1}{BT} \left\{ F_+^{-1} \left[\left(\frac{3}{4\pi BD_o} \right)^2 \right] \right\}^2 \tag{4-2a}$$

where

$$F(x) = x^2 Q(x) \tag{4-2b}$$

and recall that

$$Q(x) = (2\pi)^{-1/2} \int_x^{\infty} e^{-y^2/2} \, dy. \tag{4-2c}$$

In (4-2a), $F_+^{-1}(\cdot)$ denotes the larger of the two solutions for the inverse of F(x). Also, recall the two definitions of signal-to-noise ratio used in this work,

$$SNR = \frac{G_{ss}(f)}{G_{nn}(f)} , \quad 0 \le |f| \le B \tag{4-3a}$$

and

$$SNR' = \frac{SNR^2}{2SNR + 1} . \tag{4-3b}$$

Note that (4-2a) is expressed in terms of SNR'. For SNR << 1, SNR' ≈ SNR^2 and for large BT product, the threshold SNR becomes very small as seen in Figure 4.1. Thus, for large BT, (4-2a) can be written in terms of SNR as

$$\mathrm{SNR}_{th} \approx \frac{1}{\sqrt{BT}} F_{+}^{-1}\left[\left(\frac{3}{4\pi BD_o}\right)^2\right] \quad . \tag{4-4}$$

Given that B and D_o are fixed and that BT is large, SNR_{th} is approximately inversely proportional to $\sqrt{T}$. Moreover, $F_{+}^{-1}(\cdot)$ is a relatively weak function of its argument (i.e., an order of magnitude change in the argument produces a comparatively small change in $F_{+}^{-1}(\cdot)$. Therefore, to a good approximation, SNR_{th} is inversely proportional to $\sqrt{BT}$, for large BT. Analytical considerations leading to (4-2a), given (4-1), are given in Appendix E.

The behavior of the CPE due to increasing observation time is essentially the same as that of the ZZLB as illustrated in Figure 4.2. Nuttall has shown in [45] that the threshold SNR for the CPE can be approximated by

$$\mathrm{SNR}_{th} = \frac{x}{\sqrt{1 - x^2/2} - x} \tag{4-5a}$$

where

$$x = \frac{1}{\sqrt{BT}} Q^{-1}\left(\frac{P[A]}{4D_o B - 1}\right) \tag{4-5b}$$

and where $Q^{-1}(\cdot)$ denotes the inverse of $Q(\cdot)$. For large BT, x is small and (4-5) becomes

$$\mathrm{SNR}_{th} \approx \frac{1}{\sqrt{BT}} Q^{-1}\left(\frac{P[A]}{4D_o B - 1}\right) \quad . \tag{4-6}$$

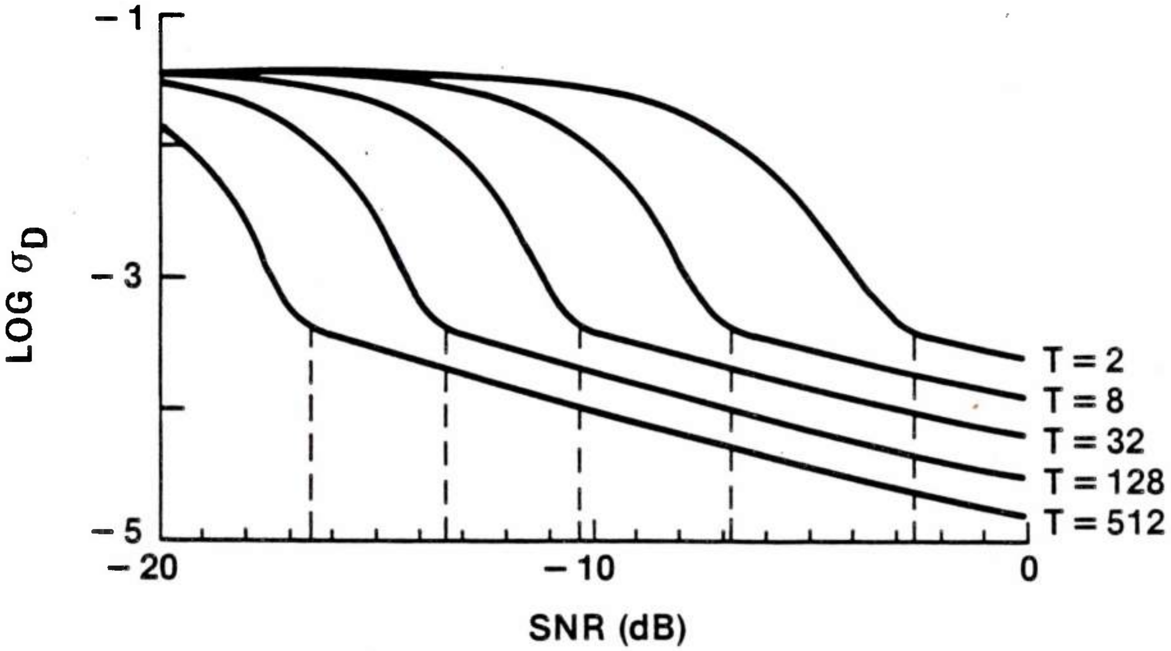


Figure 4.2 Effect of Varying Observation Time on CPE

(B = 100 Hz, D_0 = 1/16 second, T = 2, 8, 32, 128, 512 seconds)

Note the similarity between (4-4) and (4-6). The function $Q^{-1}(\cdot)$ is also a relatively weak function of its argument and the threshold SNR is again approximately inversely proportional to $\sqrt{BT}$. However, to obtain SNR_{th} from (4-5) or (4-6), the probability of anomaly, P[A], for which the CPE begins to deviate significantly from the CRLB must be known. A value of $P[A] = 5 \times 10^{-6}$ has been found empirically to give good agreement between the threshold SNR computed via (4-5) and that observed in Figure 4.2.

Values of the threshold SNR for the ZZLB, equation (4-2), and the CPE, equation (4-5), for various observation times are given in Table 4.1. Note that these values are in terms of SNR in dB. Equation (4-4b) can be used to obtain the corresponding values in terms of SNR'.

T (seconds)	CPE SNR_{th}(dB)	ZZLB SNR_{th}(dB)
2	-2.3	-4.3
8	-6.6	-8.0
32	-10.1	-11.3
128	-13.3	-14.6
512	16.4	-17.7

Table 4.1 Calculated Threshold SNR Values for CPE and ZZLB ($B = 100$ Hz, $D_o = 1/16$ sec., $P[A] = 5 \times 10^{-6}$ for CPE)

4.2 Implications of the Threshold Effect

The threshold effect has particular significance relative to the issue of coherent versus incoherent signal processing for TDE. First, observe that the parameter, T, referred to as the observation time in the expressions for the CRLB, the CPE, and the ZZLB, is actually the coherent processing time. If the observation time is T seconds, the performance predictions of Chapter 3 assume that the time delay estimate is obtained by coherently processing all T seconds of data coherently, resulting in a single estimate of the time delay.

An alternative method, which is attractive in many TDE applications, divides the T seconds of data into N sections, each of length T/N seconds. The N sections are processed individually to obtain N time delay estimates, one estimate for each T/N second data section. The N delay estimates are then averaged to obtain a final time delay estimate after T seconds. This alternative method is referred to as incoherent processing. For the incoherent processor, the coherent processing time is reduced to T/N seconds for each of the N sections, although the total observation time is still T seconds.

Above the threshold SNR, time delay estimator performance is characterized by the CRLB, and σ^2_{CRLB} is inversely proportional to the coherent processing time. Now consider the performance of the incoherent processor. For SNR greater than $SNR_{T/N}$, the threshold SNR for a coherent processing time of T/N, the variance of the time delay estimates obtained from the T/N second data sections is proportional to N/T. The incoherent processor then averages the N time delay estimates, which reduces the variance by a factor of N, assuming the N estimates are independent [36, p. 245]. Thus, for SNR > $SNR_{T/N}$, the variance of the time delay estimate for the incoherent processor is proportional to $(N/T)\cdot(1/N) = 1/T$. This is the same performance attained by coherently processing all T seconds of data to obtain a single time delay estimate. However, for SNR < $SNR_{T/N}$, the performance of the incoherent processor degrades rapidly, while the coherent processor maintains CRLB performance over the range $SNR_T < SNR < SNR_{T/N}$, where SNR_T represents the threshold SNR for a coherent processing time of T seconds. For this range of SNR, the coherent processor exhibits significant performance gains compared to the incoherent processor.

This effect is illustrated in Figure 4.3 for the case of T = 8 seconds and N = 4 sections (B = 100 Hz and $D_0 = 1/16$ second). The coherent processor operates on data sections 8 seconds long, that is, the coherent processing time is 8 seconds. The incoherent method processes the data in 4 sections, each 2 seconds long, so that the total observation time is again 8 seconds. The predicted performances for the coherent and incoherent techniques are shown by the curves labelled ZZLB (T = 8, coherent) and ZZLB (T = 8, incoherent), respectively. The third curve, labelled ZZLB (T = 2, coherent), is the ZZLB for a coherent processing time of 2 seconds. The variance of the incoherent method is reduced by a factor of 4 relative to the performance of a coherent processor for T = 2, due to the averaging of the 4 estimates. This appears as a constant offset between the curves ZZLB (T = 2, coherent) and ZZLB (T = 8, incoherent) on the log σ_D vs SNR (dB) plot of Figure 4.3. For SNR > SNR_2, the

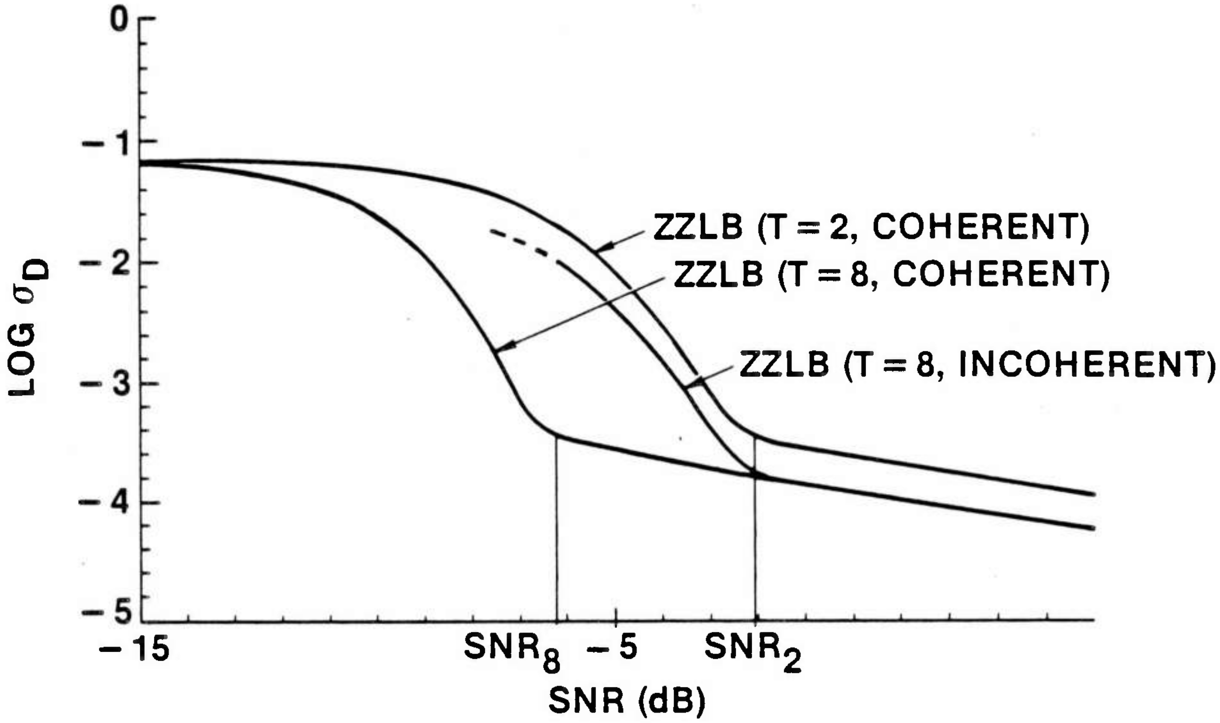


Figure 4.3 Comparison of Coherent and Incoherent Processor Performance (T = 8 seconds, N = 4 sections)

performance of the coherent and incoherent methods is identical and coincides with the CRLB for $\sqrt{T} = 8$. For SNR < SNR_2, the variance of the incoherent processor increases rapidly, while the coherent processor continues to attain CRLB performance until the SNR falls below SNR_8. In this example, the incoherent processor begins to deviate from the CRLB at $SNR_2 = -2$ dB, while the coherent processor maintains CRLB performance for SNR approximately 4 dB lower than SNR_2 (i.e., for SNR > $SNR_8 \approx -6$ dB).

In general, for large BT the threshold SNR is approximately proportional to $1/\sqrt{T}$, as noted previously. Thus, quadrupling the coherent processing time lowers the threshold SNR by a factor of 2, or by 3 dB. Similarly a factor of 64 increase in the coherent processing time will lower the threshold SNR by approximately 9 dB, see Table 4.1 and Figure 4.1. Therefore, coherent processing provides much better estimates of the time delay relative to incoherent processing for SNR below the threshold for the incoherent processor, $SNR_{T/N}$, but above the threshold for the coherent processor, SNR_T. For the signals considered here and for large BT,

$$SNR_T \approx SNR_{T/N} /\sqrt{N} \tag{4-7a}$$

or, in dB,

$$SNR_T \text{ (dB)} \approx SNR_{T/N} \text{ (dB)} - 5 \log N. \tag{4-7b}$$

For example, if a 100 second data segment were available, and the incoherent processor broke the data into 100 sections, each 1 second long, the incoherent processor would suffer a ~ 10 dB loss in the threshold SNR relative to the coherent processor. Performance gains can also be realized by increasing the signal bandwidth for a fixed observation time, as seen from equations (4-4) and (4-6). In many applications, and in the passive sonar problem in particular, the bandwidth of the source cannot be altered to improve performance. However, these results indicate that the processor should be designed to operate over the full bandwidth of the signal, when the signal and noise spectra are as defined in (3-5). Finally, note that to ensure the effective independence of the N estimates averaged by the incoherent processor, the length of each data section must be much greater than the delay value (T/N >> D). If the N estimates are not independent, the incoherent processor suffers a further degradation in performance relative to the coherent processor.

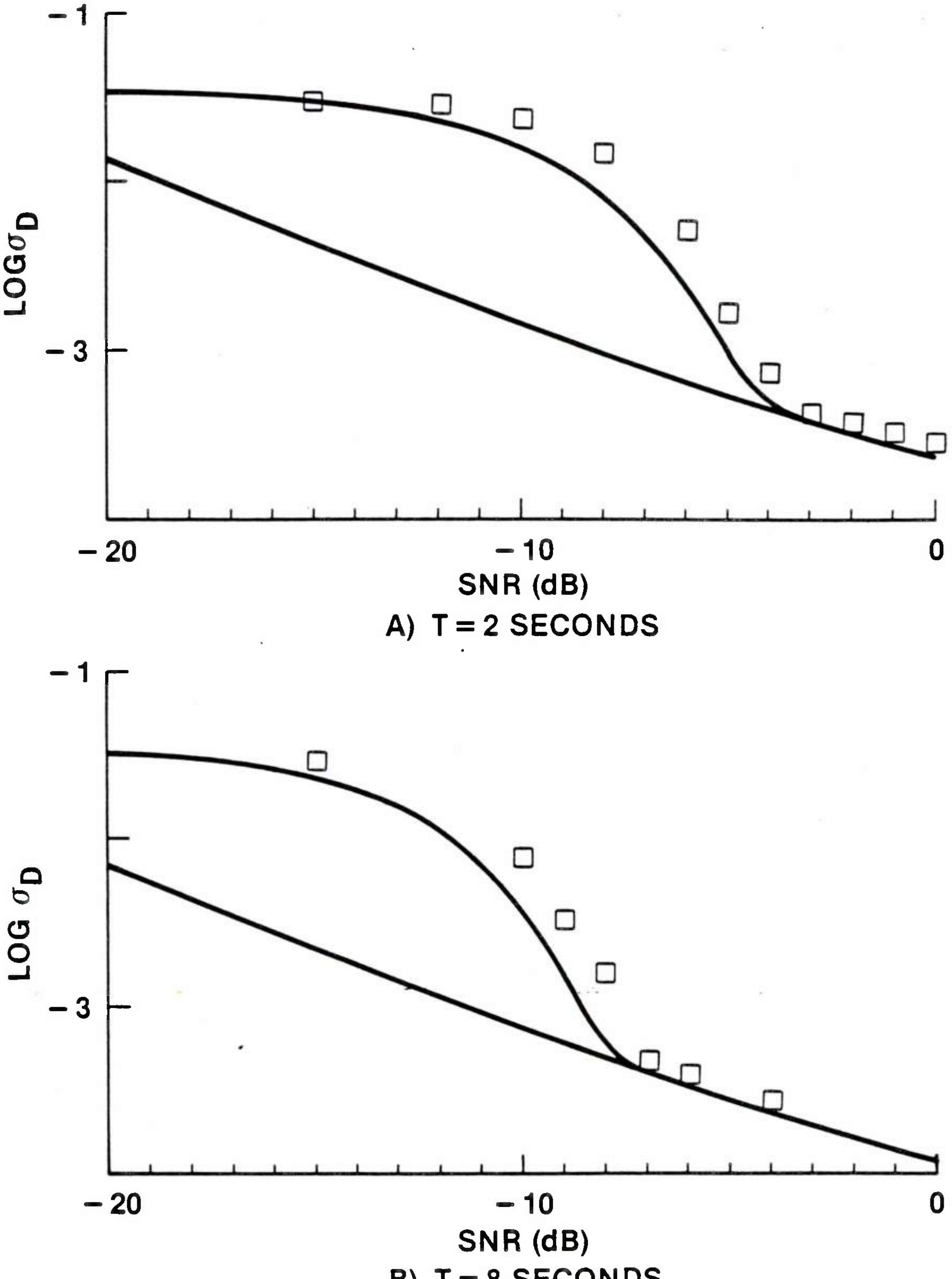
The performance gains, obtained by increasing the coherent processing time, are achieved at the expense of increased complexity in the coherent processing algorithm. As the coherent processing time is increased, it is usually necessary to compensate for the effect of time variations of the time delay (e.g., due to source motion). This requires a substantially more complicated processor than the standard GCC approach for the case of a fixed time delay as described in Chapter 2. These concerns will be discussed in Chapter 5. Before proceeding with this topic, some additional simulation results pertaining to the stationary time delay case are presented to further corroborate the performance predictions of the ZZLB.

4.3 Simulation Results

Simulation results for the SCC, the Wiener processor, the SCOT, the PHAT, and the AML techniques for TDE are compared with the ZZLB and CRLB in Figures 4.4 - 4.8. Results are shown for T = 2 seconds and T = 8 seconds, with B = 100 Hz and D_0 = 1/16 second for a sampling frequency of 2048 Hz. Complete simulation details are given in Appendix B. For the signal and noise spectra used in this simulation, see (3-5), similar results are obtained for each of the processors, and close agreement between the ZZLB and experimental results is observed. However, note that above threshold SNR, the experimental variances for the SCOT, the PHAT, and the AML techniques actually fall below the CRLB.

This phenomena can be explained in terms of an effective bandwidth, B_e, as follows. The CRLB and ZZLB were computed for B = 100 Hz. As discussed in Appendix B, the signal and noise sequences were obtained by processing white noise sequences through a low pass Butterworth filter, which had a cut-off frequency of 100 Hz. Although a high order filter was used, the signal and noise spectra are not ideal and contain some power at frequencies above 100 Hz. Since the signal and noise spectra have the same shape, the SNR, or equivalently, the MSC will remain approximately constant in some small interval beyond B Hz, say $B \leq f \leq B_e$, until the signal power is diminished sufficiently so that the SNR is effectively zero.

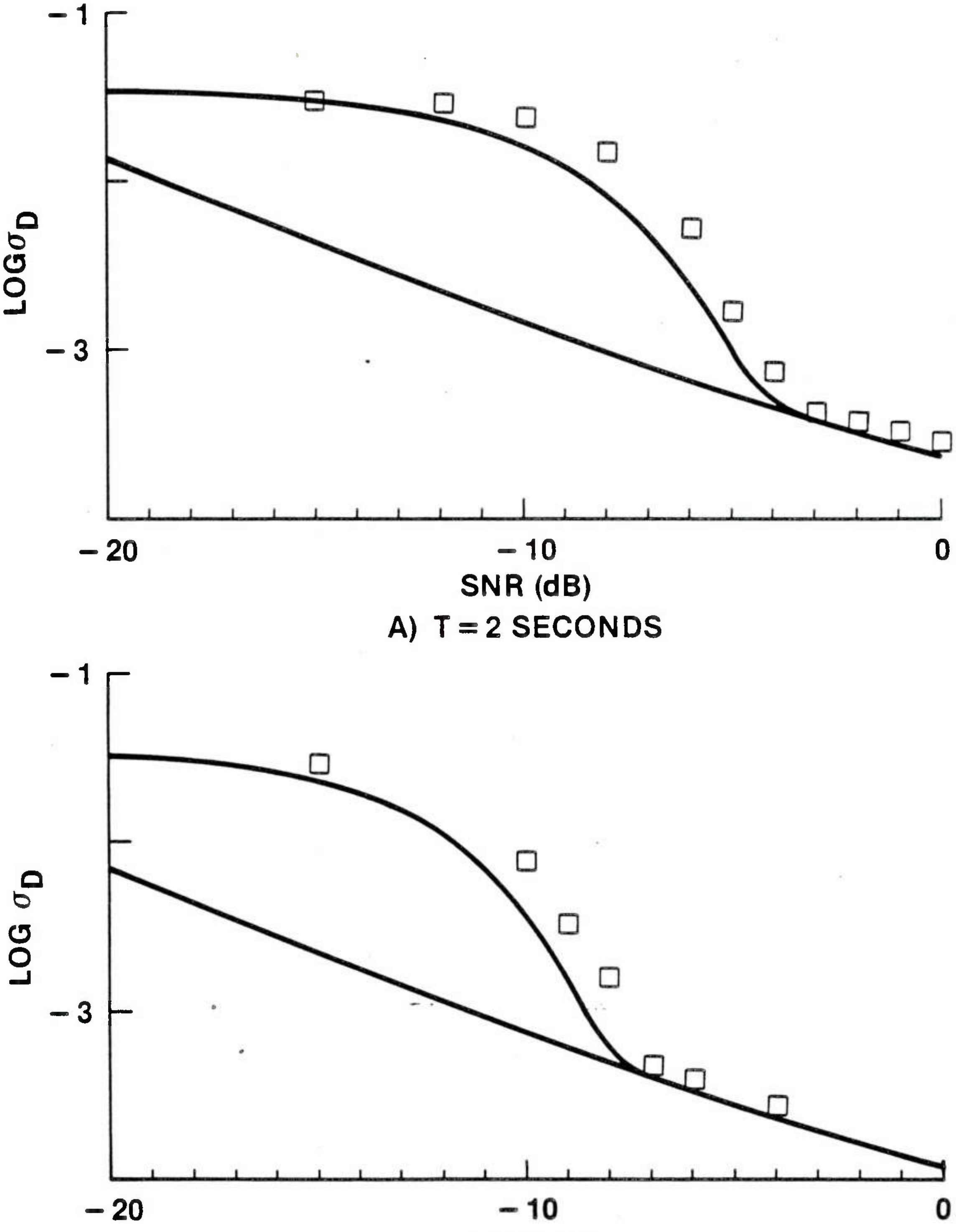
The phase weighting functions, $W_\phi(f)$, for the five processors considered here are shown in Figure 4.9. While the SCC and the MSC weighting functions go to zero above 100 Hz, the SCOT, the PHAT and the AML weighting functions do not fall to zero until $f \geq 130$ Hz. The latter three methods have an effective bandwidth of $B_e \approx 130$ Hz compared to B = 100 Hz. Since the SNR is approximately constant, the phase (time delay) information in this extended bandwidth is of the same quality as in the nominal bandwidth. From (3-9), it is easily shown that increasing the



A) T = 2 SECONDS

B) T = 8 SECONDS

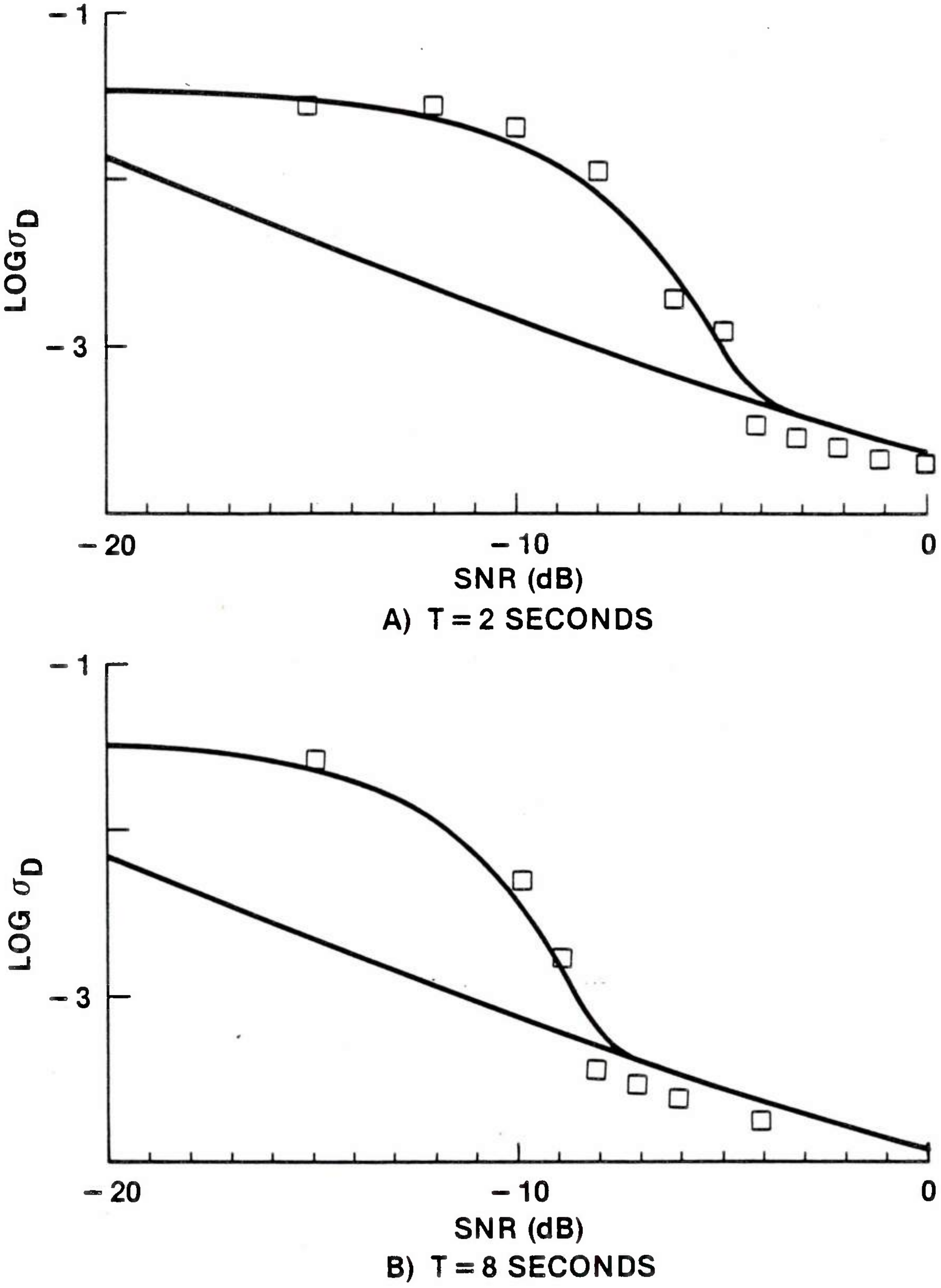
Figure 4.4 Comparison of SCC Results and ZZLB



A) T = 2 SECONDS

B) T = 8 SECONDS

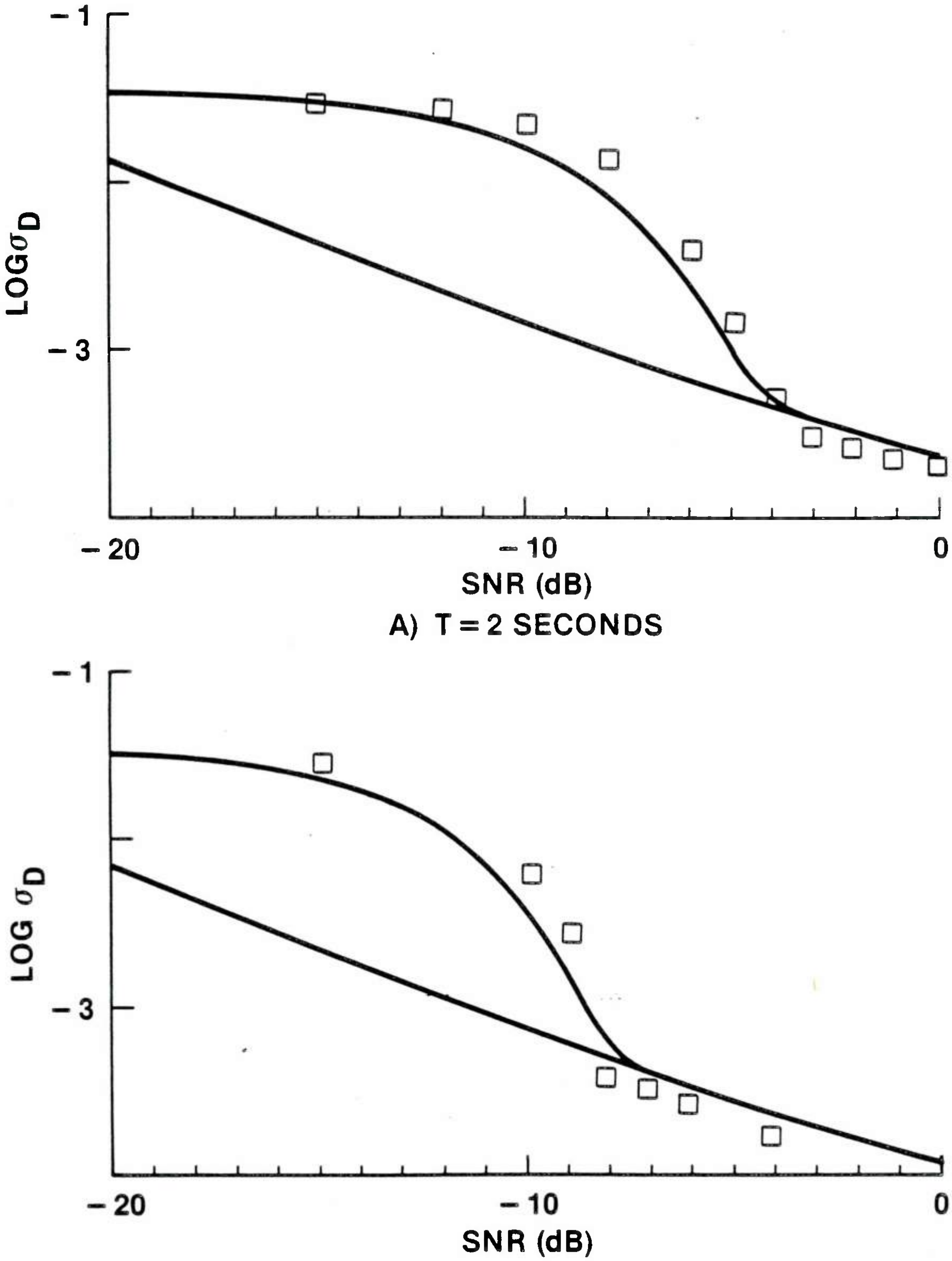
Figure 4.5 Comparison of Wiener Processor Results and ZZLB



A) T = 2 SECONDS

B) T = 8 SECONDS

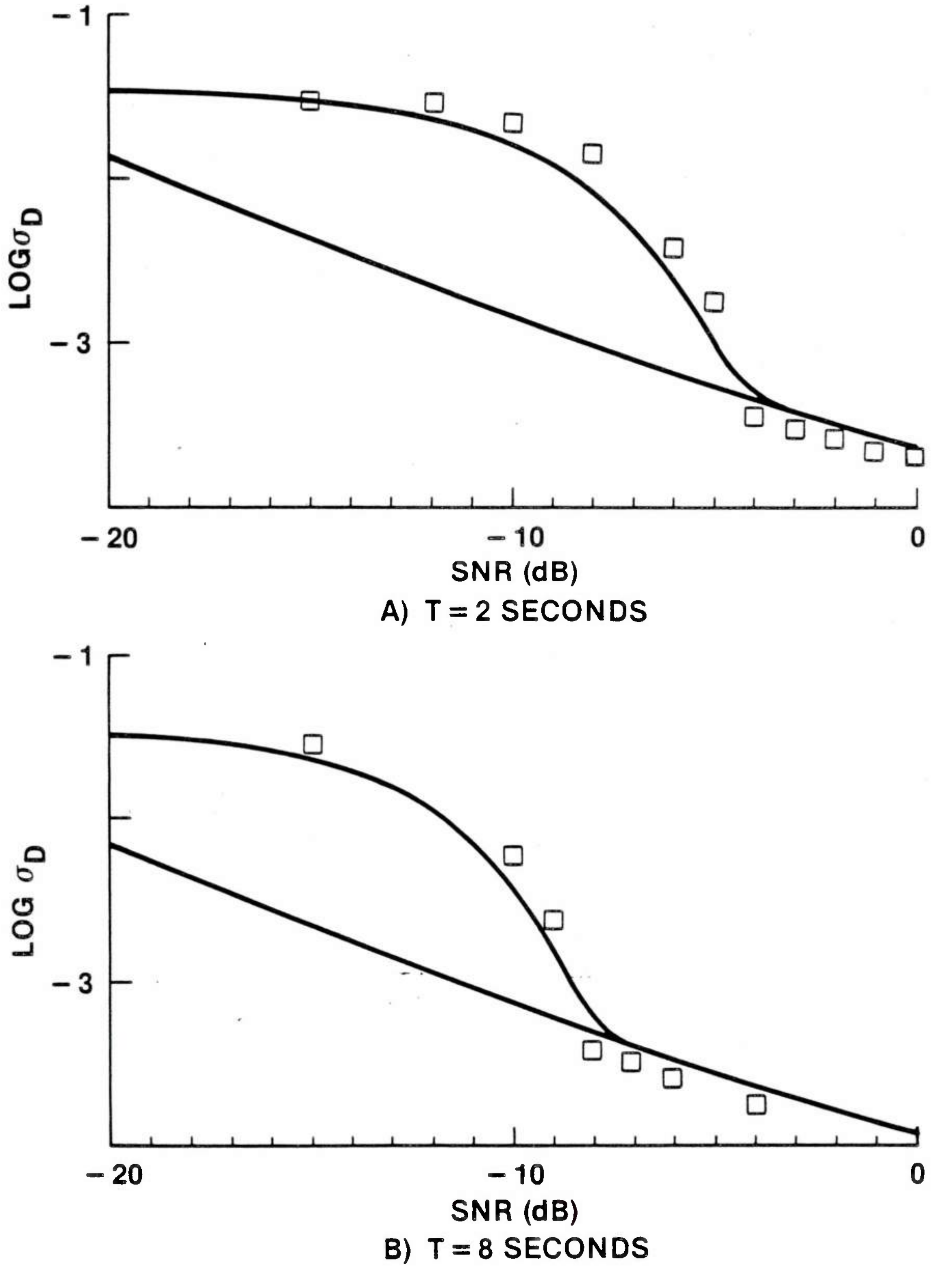
Figure 4.6 Comparison of SCOT Results and ZZLB



A) T = 2 SECONDS

B) T = 8 SECONDS

Figure 4.7 Comparison of PHAT Results and ZZLB

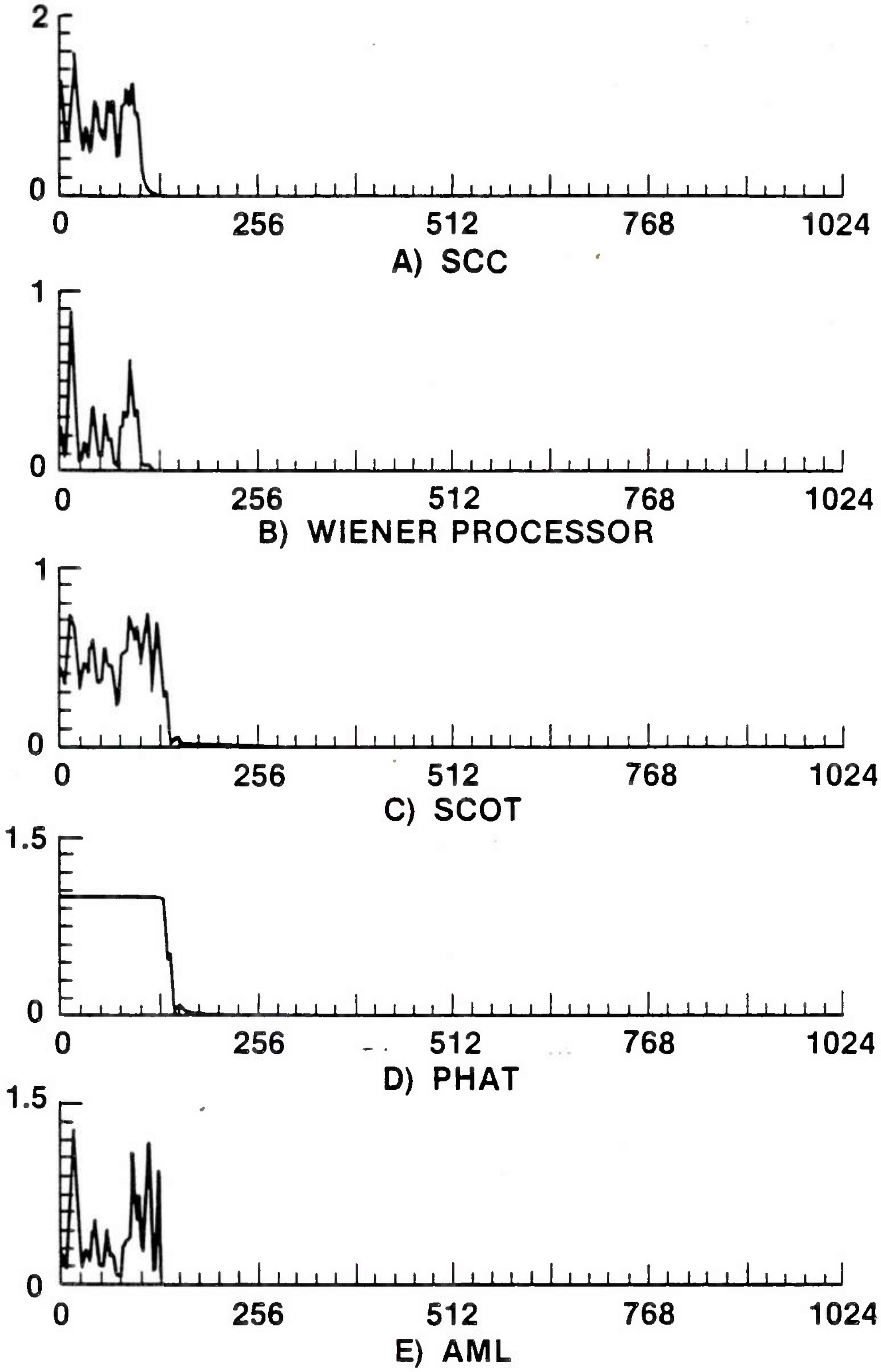


A) T = 2 SECONDS

B) T = 8 SECONDS

Figure 4.8 Comparison of AML Results and ZZLB

bandwidth by a factor of 1.3 has the effect of subtracting .17 from log σ_{CRLB}. Reducing the bound by this amount in Figures 4.6 - 4.8, reconciles the theoretical and experimental results above the threshold SNR. Also note that the threshold SNR will be slightly reduced due to the increase in the effective bandwidth; however, below the threshold SNR, this increase in bandwidth will have little effect on the ZZLB. This phenomena is an artifact of the technique used to generate the signal and noise sequences and does not have great practical significance. However, it does illustrate the whitening effect of the SCOT and the PHAT techniques, and that the AML method weights the phase according to the MSC, while the SCC and WP methods weight the phase relative to the magnitude of the cross power spectrum (i.e., the signal power), refer to Table 2.1.



A) SCC

B) WIENER PROCESSOR

C) SCOT

D) PHAT

E) AML

Figure 4.9 Phase Weighting Function, $W_{\phi}(f)$, vs. Frequency (Hz)

CHAPTER 5

ESTIMATION OF A TIME-VARYING TIME DELAY

It has been shown that significant performance gains are realized at low SNR by increasing the coherent processing time. For a constant or fixed time delay, increasing the coherent processing time requires little or no modification of the GCC processor structure. However, when the time delay varies with time, the received signals must be pre-processed to compensate for the changing time delay to avoid degradation in performance. The degree of degradation is dependent upon the amount of variation in the time delay, but even small rates of change in the delay can cause a significant decrease in performance [46, 47]. In this chapter, a simple model for a time-varying time delay is considered. Based on this model, the required changes in the GCC structure to implement an ML processor are discussed. A simplified compensation scheme is introduced and preliminary simulation results are presented to demonstrate the effectiveness of the compensation technique.

5.1 Time-Varying Time Delay Model

Consider the following modification of the model of (2-1),

$$r_1(t) = s'(\beta_1 t) + n_1(t) \tag{5-1a}$$

$$r_2(t) = \alpha s'(\beta_2 t+D)) + n_2(t), \quad 0 \leq t \leq T \tag{5-1b}$$

where $r_1(t)$ and $r_2(t)$ represent the signals received at two spatially separated sensors, $s'(t)$ is the source signal, $n_1(t)$ and $n_2(t)$ are additive noises, α is an attenuation parameter and D is the time delay of interest. The parameters β_1 and β_2 account for the time compressions in the source signal at the two sensors due to relative motion between the source and sensors. If the source is moving toward sensor 1 with a relative speed of v_1, then $\beta_1 = 1 + v_1/c$, where c is the propagation speed of the signal. For an acoustic signal in the ocean, $c >> v_1$, and β_1, $\beta_2 \approx 1$. It is assumed that β_1 and β_2 remain constant over the observation interval T. Now let $s(t) = s'(\beta_1 t)$ then (5-1) can be written as

$$r_1(t) = s(t) + n_1(t) \tag{5-2a}$$

$$r_2(t) = \alpha s(\beta(t+D)) + n_2(t) \tag{5-2b}$$

where $\beta = \beta_2/\beta_1$ is the relative time compression. There are now two parameters to be estimated, the time delay, D, and the relative time compression, β. It is again assumed that $s(t)$, $n_1(t)$, and $n_2(t)$ are sample functions from uncorrelated, zero mean, Gaussian random processes.

Note that (5-2b) can be written as

$$r_2(t) = s(t + d(t)) + n_2(t) \tag{5-3a}$$

where

$$d(t) = d't + d_0 \tag{5-3b}$$

and

$$d' = (\beta-1) \quad , \quad d_0 = \beta\, D \quad . \tag{5-3c}$$

In (5-3), d' represents the delay rate and d_0 represents the initial delay. In general, the time-varying time delay can be expressed as

$$d(t) = d_o + d't + d"t^2 + d"'t^3 + .. \quad , \tag{5-4}$$

where d_o, d', d" ... are constants. The model considered here assumes that higher order terms, such as the delay acceleration (d"), are negligible. This model is adequate for many applications, but is not suitable for situations where the higher order terms cannot be ignored.

5.2 The Maximum Likelihood Estimate

The cross correlation function between $r_1(t)$ and $r_2(t)$ of (5-1) is given by

$$R_{r_1r_2}(t+\tau, t) = \alpha R_{ss}((1 - \beta)t + \tau - \beta D) \tag{5-5a}$$

$$= \alpha R_{ss}(\tau - d't - d_o) \tag{5-5b}$$

While $r_1(t)$ and $r_2(t)$ are individually wide-sense stationary, they are no longer jointly wide-sense stationary, (i.e., $R_{r_1r_2}(t+\tau,t) \neq R_{r_1r_2}(\tau)$ for $t\varepsilon[0, T]$). The processes are only jointly stationary when $\beta = 1$, that is $\beta_1 = \beta_2$. However, the maximum likelihood estimator for the dynamic case of (5-2) can be obtained following a procedure similar to that used for the stationary case. The derivation of the ML estimator for β and D has been carried out by Knapp and Carter in [46] and is outlined in Appendix F for convenience. For cases of practical interest, the signal and noise spectra must be estimated and approximate ML estimates of β and D are obtained by maximizing the function J with respect to b and τ, where

$$J(b,\tau) = \int_{-\infty}^{\infty} R_1(f)\ R_2^*(bf)e^{j2\pi fb\tau}\ \hat{W}(f)df, \qquad (5\text{-}6a)$$

$$\hat{W}(f) = \frac{\hat{C}_{r_1r_3}(f)}{\left|\hat{G}_{r_1r_3}(f)\right|\ (1 - \hat{C}_{r_1r_3}(f))}\ , \qquad (5\text{-}6b)$$

and

$$r_3(t) = r_2(t/b)/\sqrt{b} \qquad (5\text{-}6c)$$

The values of b and τ which maximize J are the ML estimates of β and D, respectively. In (5-6a), $R_i(f)$ denotes the Fourier Transform of $r_i(t)$, and in (5-6c), $r_3(t)$ represents a time scaled version of $r_2(t)$, where the time scaling counteracts the effects of the relative time compression. Note that

$$R_3(f) = \sqrt{b}\ R_2(bf) \qquad (5\text{-}7a)$$

and (5-4a) can be written as

$$J(b,\tau) = \int_{-\infty}^{\infty} R_1(f)\ R_3^*(f)\ e^{j2\pi fb\tau}\ \hat{W}(f)df\quad . \qquad (5\text{-}7b)$$

The ML estimator can be implemented as illustrated in Figure 5.1. A value of b is selected, $r_2(t)$ is time scaled to obtain $r_3(t)$ corresponding to b, the FT of $r_1(t)$ and $r_3(t)$ is computed and W(f) is estimated, then $J(b,\tau)$ in (5-7b) is computed. The value of τ which maximizes J is the best estimate of D for the b selected. This process is repeated varying the value of b to generate an ambiguity surface for $J(b,\tau)$ in b and τ. The values of b and τ corresponding to the global peak of the ambiguity surface are the ML estimates of β and D. In practice, knowledge of the physical problem is used

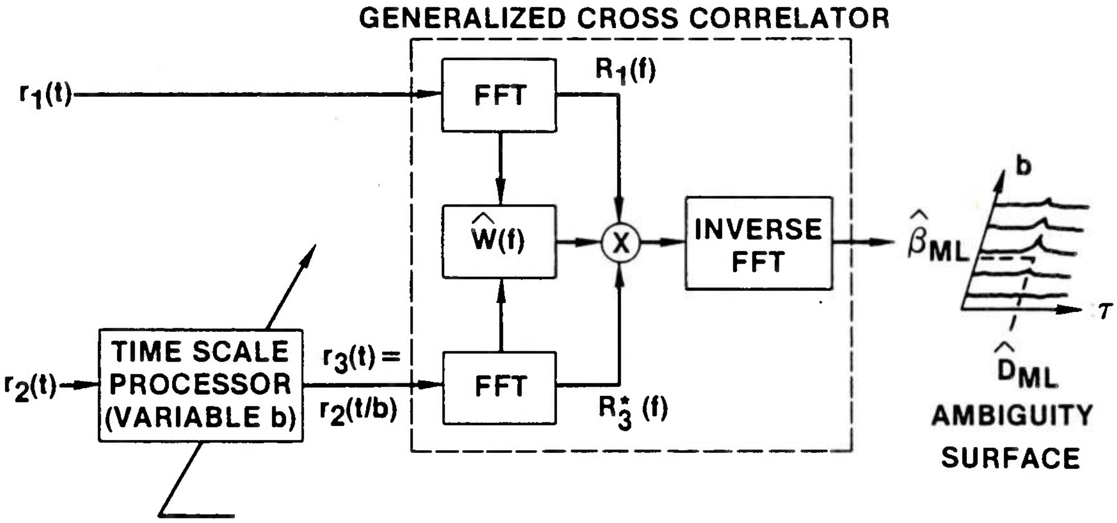


Figure 5.1 The ML Estimator for β, D

to determine a priori limits for β, and the number of compensations (trial b values) is chosen to provide the required resolution for β over this range. The ML estimator is computationally intensive, requiring $r_3(t)$ and $\hat{W}(f)$ to be recomputed for each trial value of b. Then, the function $J(b,\tau)$ must be computed for each b value and this set of functions must be searched to find the global peak. Additionally it is not immediately clear how to obtain $r_3(t)$ from $r_2(t)$. Conceptually, if $r_2(t)$ is recorded on analog tape, $r_3(t)$ can be obtained using a variable sampling rate. However, very fine adjustments of the nominal sampling rate are necessary and there are obvious synchronization problems. In many cases, only a sampled version, $r_2(k)$, of $r_2(t)$ will be available, and a high resolution interpolation process will be required to obtain $r_3(k) = r_2(k/b)$ from $r_2(k)$.

5.3 A Simplified Compensation Scheme

As discussed in Appendix B, a common technique for implementing the GCC processor is to segment the T second data record into M subintervals. It is often advantageous to use overlapping segments [48]; however, disjoint segments are assumed here to simplify the following discussion. The fast Fourier transform of each segment is computed and the pertinent power spectra are estimated for each segment. These M spectral estimates are then averaged in such a way that the signal components will sum coherently to obtain a final estimate of the cross power spectrum and the GCC weighting function. The inverse FFT of the product of these estimates yields the estimate of the GCC function.

For the stationary time delay case ($\beta = 1$), the M spectral estimates can simply be summed, and then divided by M to obtain the final spectral estimate. When $\beta \neq 1$, the received signals must be pre-processed to compensate for the relative time compression, or signal coherence will not be preserved when the spectral estimates are summed. The compensation method described here compensates for the relative time compression in the frequency domain and avoids the need to re-process $r_2(t)$ to obtain its time scaled version, $r_3(t)$, for each hypothesized value of β. This greatly reduces the computational load.

Consider the expression for $r_2(t)$ given in (5-3)

$$r_2(t) = s(t + d't + d_o) + n_2(t), \; 0 \leq t \leq T. \qquad (5\text{-}8)$$

Now partition T into M disjoint segments of length T/M. Define the kth interval as

$$T_k = \left\{ t \;\middle|\; (k-1)T/M \leq t \leq kT/M \right\} \qquad (5\text{-}9a)$$

and denote the midpoint of this interval as

$$t_k = (k-1/2)T/M. \qquad (5\text{-}9b)$$

Then the time delay at $t = t_k$ is given by $d_k + d_o$, where $d_k = d't_k$. The key step in the development of this compensation technique is to assume that the time-varying time delay in the interval T_k can be approximated by the value of the delay at the midpoint of the interval, that is

$$d't + d_o \simeq d_k + d_o \quad , \quad t \in T_k . \tag{5-10}$$

With this approximation, the signal model becomes

$$r_1(t) = s(t) + n_1(t) \tag{5-11a}$$

$$r_2(t) = \alpha\, s(t + d_k + d_o) + n_2(t) \tag{5-11b}$$

for $t \in T_k$, $k=1, \ldots, M$. Thus, $r_1(t)$ and $r_2(t)$ are again jointly stationary over each interval T_k, and the short term cross correlation function for this interval is given by

$$R_{r_1 r_2}(\tau) = \alpha\, R_{ss}(\tau - d_k - d_o). \tag{5-12}$$

Denoting the Fourier transform for $r_i(t)$, $t \in T_k$, as $R_i^k(f)$, where $i=1, 2$; the cross power spectrum for the kth data segment can be estimated as

$$\begin{aligned} \hat{G}^k_{r_1 r_2}(f) &= R_1^k(f)\, R_2^k(f)^* \\ &\simeq \alpha\, G_{ss}(f)\, e^{-j2\pi f(d_k + d_o)} \ . \end{aligned} \tag{5-13}$$

The cross power spectrum for the kth interval undergoes a phase rotation proportional to d_k. Compensating for this phase rotation, and then averaging the short term spectral estimates yields the following estimate of the cross power spectrum,

$$\hat{G}_{r_1 r_2}(f) = \frac{1}{M} \sum_{k=1}^{M} \hat{G}^k_{r_1 r_2}(f)\, e^{j2\pi f d_k} \tag{5-14a}$$

$$\simeq \alpha\, G_{ss}(f)\, e^{-j2\pi f d_o} \ . \tag{5-14b}$$

Multiplying (5-14) by the appropriate weighting function and computing the IFT of the product results in an estimate of the GCC function. In practice, d_k, or equivalently, $d' = \beta - 1$ is not known and it is necessary to compute (5-14) for several hypothesized values of d', say τ'_l , l = 1, 2, ..., N, then

$$G_{r_1r_2}(f, \tau'_l) = \frac{1}{M} \sum_{k=1}^{M} G_{r_1r_2}(f)\, e^{j2\pi f \tau'_l t_k} \quad . \qquad (5\text{-}15)$$

Computing the IFT of (5-15) for l = 1, ..., N results in an ambiguity surface which is a function of the hypothesized values for d' and d_o (or β and D). The position of the peak on this surface corresponds to the estimates for d' and d_o. This technique is illustrated in Figure 5.2.

5.4 Simulation Results

This compensation technique was implemented and preliminary simulation results comparing the SCC and SCOT methods have been obtained. Simulation details and a program listing is given in Appendix G. Briefly, the model in (5-2) and (5-3) was simulated for an assumed sampling frequency of 2048 Hz, T = 4 seconds = 8192 samples, and M' = 33 segments (50% overlap was used). Each segment was then 1/4 second or 512 samples long. The initial delay, d_o, was set to zero at the start of the simulation and the delay rate, d', was set at 2.5×10^{-4} samples/sampling interval. When the total delay became greater than 15 samples, the delay rate was set to -2.5×10^{-4} samples/sampling interval until the total delay decreased below -15 samples and the sign of d' was again changed and so on. Hypothesized delay rates, τ'_l, of 0, $\pm 2.5 \times 10^{-4}$, and $\pm 5 \times 10^{-4}$

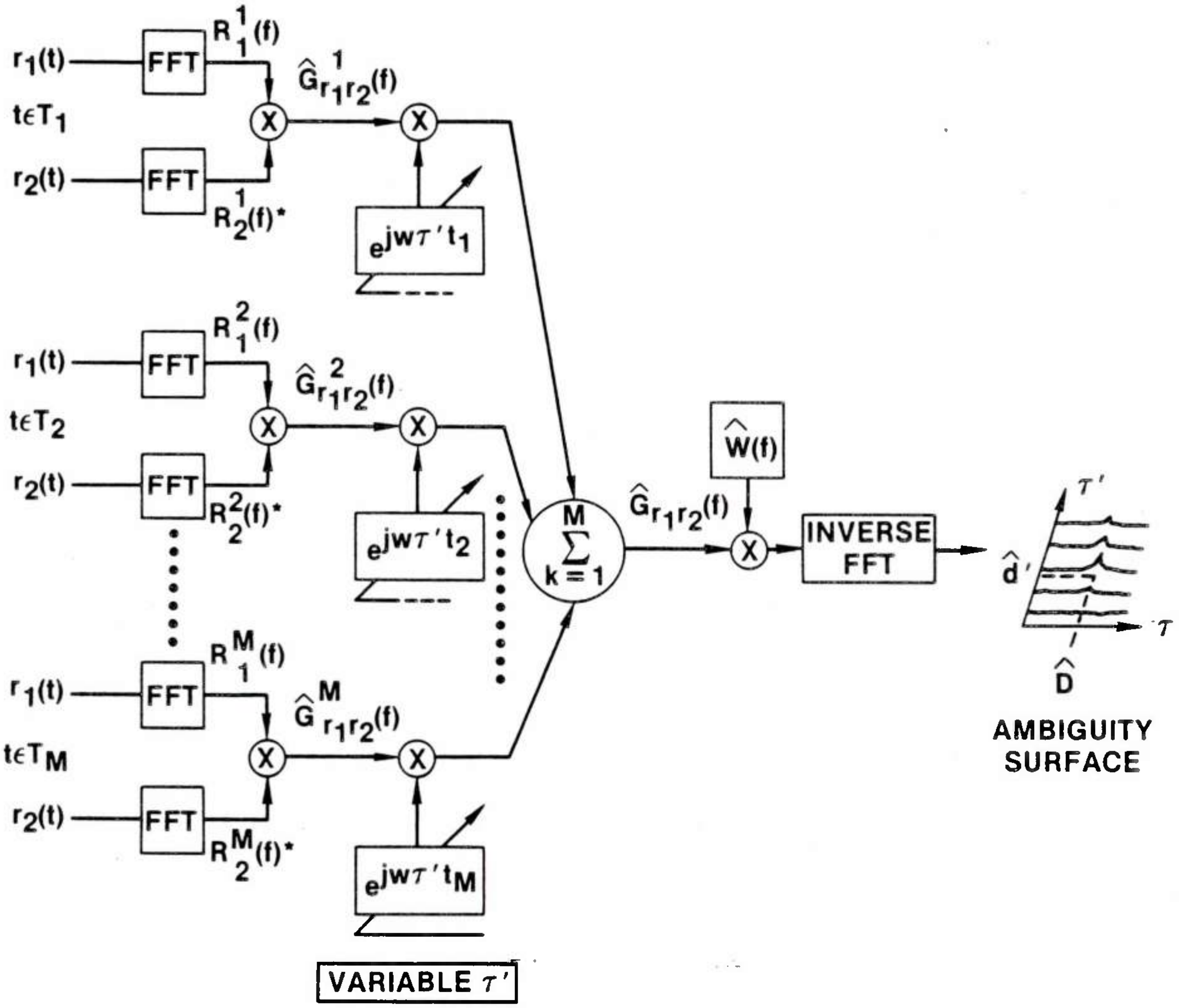
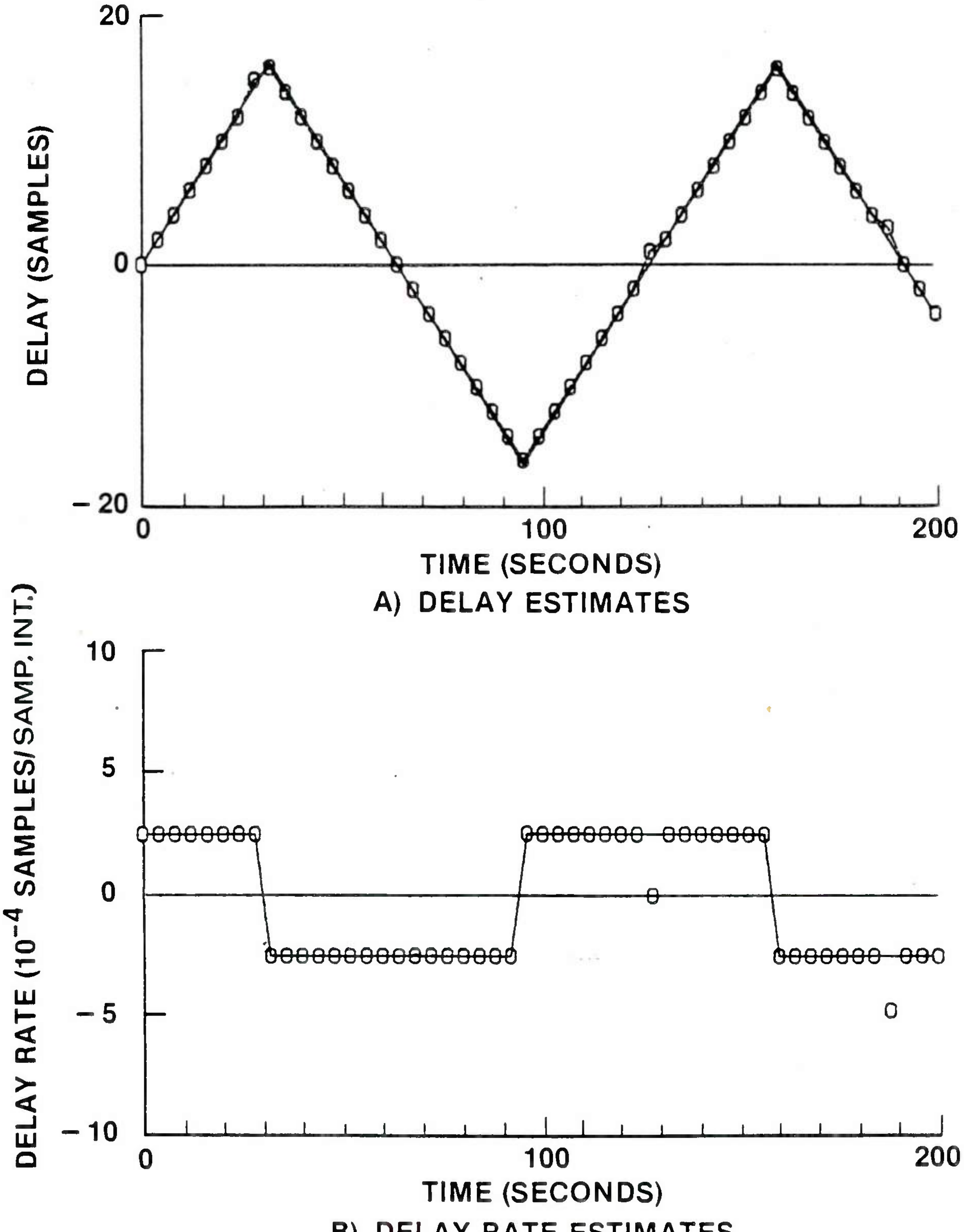


Figure 5.2 Frequency Domain Compensation Technique

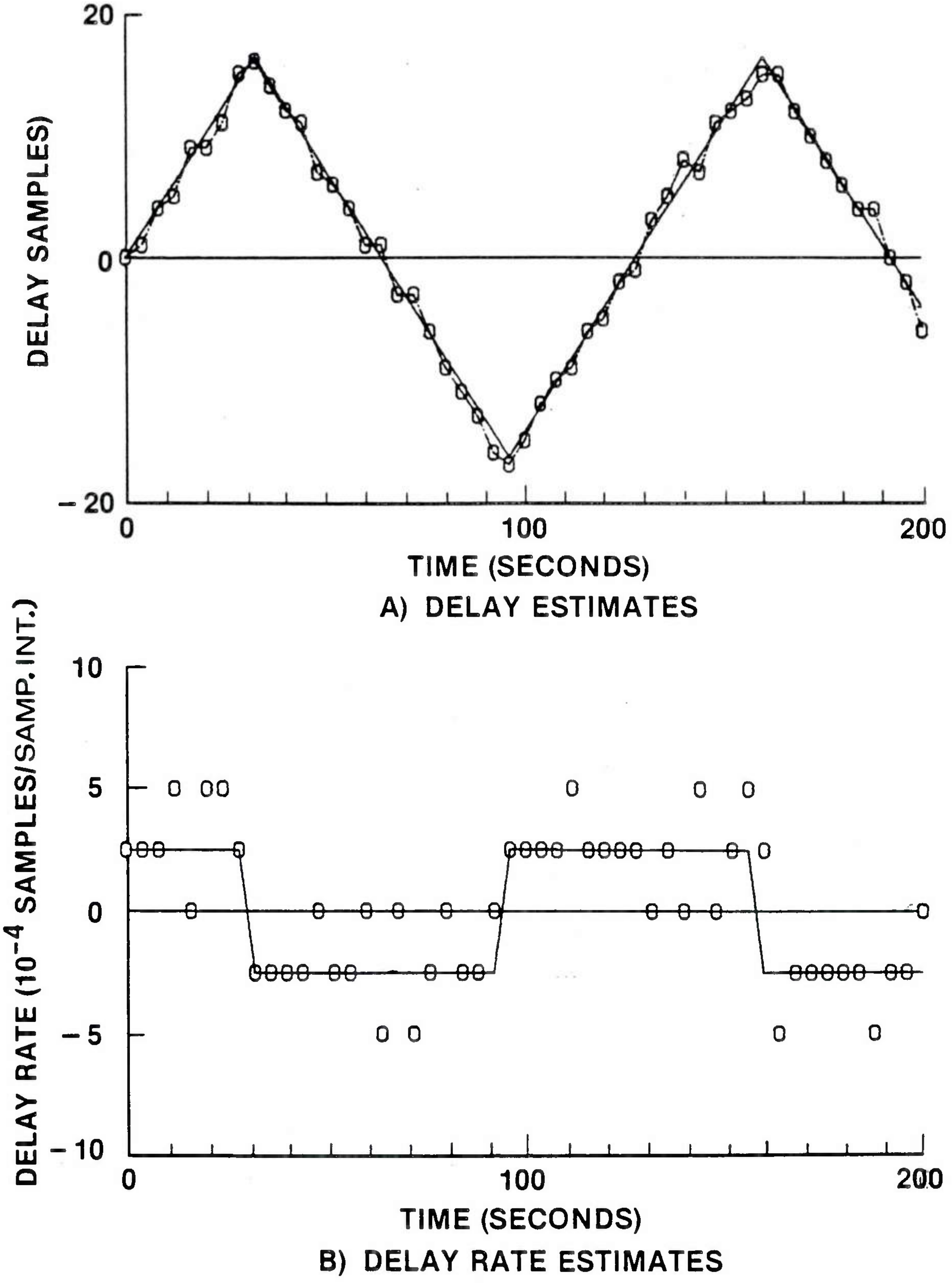
samples/sampling interval were used. The SNR was -3 dB. The simulation results for the SCOT and SCC techniques are displayed in Figures 5.3 and 5.4, respectively. In both cases, Figure (a) shows the estimated delay values (circles connected by dashed line) and the true delay values (solid line) as a function of time, and similarly, Figure (b) shows the estimated delay rate (circles) and the true delay rate (solid line) as a function of time. Both methods provided good estimates of the time delay with the SCOT predicting the delay value to within a fraction of a sample in every trial. The SCOT method did a much better job of estimating the delay rate than the SCC method. This is probably due to the narrower peak of the SCOT function compared to the SCC function, however, this effect has not been investigated thoroughly. Also note that in this preliminary simulation, no interpolation was performed to obtain the estimates of the delay and delay rate. The values of delay and delay rate corresponding to the global peak of the five discrete GCC functions obtained from the processor were used as the estimates.



A) DELAY ESTIMATES

B) DELAY RATE ESTIMATES

Figure 5.3 SCOT Simulation Results - Time-Varying Delay



A) DELAY ESTIMATES

B) DELAY RATE ESTIMATES

Figure 5.4 SCC Simulation Results - Time-Varying Delay

CHAPTER 6

SUMMARY AND RECOMMENDATIONS

This work has presented a detailed study of time delay estimator performance to obtain a better understanding of the fundamental limitations in the time delay estimation problem. To this end, an investigation of the behavior of the Cramer-Rao lower bound (CRLB) and the recently proposed correlator performance estimate (CPE) and Ziv-Zakai lower bound (ZZLB) was conducted. Derivations of these three performance estimates were outlined and expressions for the variance of the time delay estimate for specific signal and noise power spectra were obtained.

It was observed that, although the ML estimator is asymptotically efficient in attaining CRLB performance, actual performance can be much worse than that predicted by the CRLB for a given SNR and observation time. The CPE and the ZZLB were seen to be characterized by a threshold SNR. Above the threshold SNR, the CPE and the ZZLB coincide with the CRLB. Below the threshold, large estimation errors or anomalous estimates become dominant and the performance predicted by the CPE and the ZZLB degrades rapidly relative to the CRLB. As SNR goes to zero, the CPE and the ZZLB are further characterized by a prior information limit. That is, the predicted variance approaches a constant value which is a function of the maximum possible delay value. The threshold SNR was shown to be approximately inversely proportional to the square root of the observation (coherent processing) time. Simulation results were found to be in good agreement with the theoretical predictions.

This threshold effect was found to have important implications related to coherent versus incoherent processing considerations. A simple incoherent processor was considered which divides the T second observation interval into N sections, each of length T/N seconds. Each T/N second section is coherently processed separately to obtain N time delay estimates and finally the N estimates are averaged to yield a single estimate at the end of the observation interval. Compared to a coherent processor, which processes the entire T second observation interval coherently, the incoherent processor suffers a loss of 5 log N dB in the threshold SNR, that is, SNR_{COH} (dB) = SNR_{INCOH} (dB) - 5 log N. For example, if the incoherent processor divides 100 seconds of data into 100, 1 second sections, the coherent processor maintains CRLB performance for ~ 10 dB below the threshold SNR for the incoherent processor. Thus, significant performance gains can be obtained by increasing the coherent processing time.

However, increasing the coherent processing time requires a significantly more complex structure for time delay estimators, in the case of a time-varying time delay. The ML estimator for the time-varying delay case was seen to require that the received signals be pre-processed to compensate for the relative time compression between the two signals. In general, the required pre-processing is difficult to achieve from an implementation viewpoint, in addition to being computationally intensive. Therefore, a simplified compensation scheme was proposed which essentially computes short-term correlation functions so that the time delay can be considered constant over the correlation interval. The short-term correlation functions are then combined coherently to achieve a longer coherent processing time. Time domain (actually, time delay domain) compensation techniques which coherently sum the short-term correlation functions directly are discussed in [49,50]. The method described in this work is implemented in the frequency domain which is attractive in that it is easily incorporated into the structure of the GCC processor. Preliminary simulation results demonstrate that this compensation method is effective and allows the delay and delay rate to be accurately estimated. However, the method has only been tested for a limited set of conditions and further testing is required.

Future work based on this dissertation could take many directions. Some of these include:

1) Extension of the time delay model in (2-1) to account for multiple sources or multi-path interference and the development of the corresponding expression for the ZZLB. (The CRLB for the multi-sensor, multi-target problem is developed by Ng in [51].)

2) More extensive testing of the proposed compensation technique for the time-varying time delay model and comparison with time domain implementations.

3) Consideration of the relative importance of the fundamental limitations on time delay estimator performance discussed here (e.g., due to SNR and observation time) with the limitations imposed by uncertainty in sensor position.

These and other problems of equal importance provide the basis for a qreat deal of exciting research yet to be done in time delay estimation.

APPENDIX A

THE ML ESTIMATOR-FIXED TIME DELAY

The ML estimator for the case of a stationary or fixed time delay is derived by Knapp and Carter in [6]. The derivation is summarized in this appendix for convenience. The mathematical model is that of Section 2.1 and the same assumptions are made.

Suppose an observation vector, $\underline{R}$, which is dependent upon the parameter to be estimated, say τ, is given. For the TDE problem, the parameter of interest is the true time delay D. The ML estimate, $\hat{D}_{ML}$, is the value of τ which is most likely to have resulted in the occurrence of the observed vector, $\underline{R}$. More precisely, the ML estimate is given by $\hat{D}_{ML} = \tau_0$ such that

$$p(\underline{R} \mid \tau_0) \geq p(\underline{R} \mid \tau), \text{ for all allowable } \tau, \qquad \text{(A-1)}$$

where $p(\underline{R} \mid \tau)$ is the conditional probability density function of $\underline{R}$ given τ.

For purposes of analysis, consider periodic extensions of the received signals, $\tilde{r}_1(t)$ and $\tilde{r}_2(t)$, with period T, where T also represents the observation time. Then, for any integer m

$$\tilde{r}_i(t + mT) = r_i(t) \quad , \quad 0 \leq t \leq T, \; i = 1,2. \qquad \text{(A-2)}$$

The Fourier series coefficients of the received signals can be computed as

$$R_i(k) = \frac{1}{T}\int_0^T \tilde{r}_i(t)\ e^{-jkw_0t}\ dt \quad , \quad i = 1,\ 2 \tag{A-3a}$$

where

$$w_0 = 2\pi/T. \tag{A-3b}$$

The time waveforms can be reconstructed from the Fourier coefficients using the relationship

$$\tilde{r}_i(t) = \sum_{k=-\infty}^{\infty} R_i(k)\ e^{jkw_0t} \tag{A-4a}$$

$$= r_i(t) \quad , \quad 0 \leq t \leq T\ . \tag{A-4b}$$

In practice, the received signals are band limited so the number of non-zero Fourier coefficients will be finite, say $-N \leq k \leq N$. Thus, as seen in (A-4), $r_i(t)$ is characterized by the set of Fourier series coefficients $\left\{R_i(k),\ -N \leq k \leq N\right\}$.

Defining the vector $\underline{R}(k) = [R_1(k),\ R_2(k)]'$, where ' denotes transpose, the ML estimate of the time delay is obtained by maximizing the joint density function, $p(\underline{R}(-N),\ \underline{R}(-N+1),\ \ldots,\ \underline{R}(N) \mid \tau)$, with respect to the time delay variable τ. Observe that $R_i(k)$ is a linear transformation of $r_i(t)$, which is Gaussian, so $R_i(k)$ is also Gaussian for $i = 1,2$ and $-N \leq k \leq N$. Further, it can be shown that [36, p. 461]

$$E[R_1(k)\ R_2^*(l)] = \begin{cases} \frac{1}{T}\ G_{r_1r_2}(kw_0) & ,\ k = l \\ 0 & ,\ k \neq l\ . \end{cases} \tag{A-5}$$

It is also easily seen that $E[R_i(k)] = 0$, since $r_i(t)$ is zero mean. Therefore, the vectors, $\underline{R}(k)$, $-N \leq k \leq N$, are uncorrelated, zero mean, Gaussian random vectors. Let $\underline{R} = (\underline{R}(-N), \ldots, \underline{R}(N))$, then the joint density function can be written as

$$p(\underline{R} \mid \tau) = \prod_{k=-N}^{N} p(\underline{R}(k) \mid \tau) \qquad \text{(A-6a)}$$

where

$$p(\underline{R}(k) \mid \tau) = (2\pi \left| \underline{\Sigma}_{k|\tau} \right|^{1/2})^{-1} \exp\left[-\frac{1}{2} \underline{R}^{*'}(k) \underline{\Sigma}_{k|\tau}^{-1} \underline{R}(k) \right]. \qquad \text{(A-6b)}$$

In (A-6b), $\underline{\Sigma}_{k|\tau}$ denotes the covariance matrix of the vector $\underline{R}(k)$ given τ, and is defined as

$$\underline{\Sigma}_{k|\tau} = E\,[\underline{R}(k)\ \underline{R}^{*'}(k) | \tau] \qquad \text{(A-7a)}$$

$$= \frac{1}{T} \begin{bmatrix} G_{r_1r_1}(kw_o) & G_{r_1r_2}(kw_o) \\ G^{*}_{r_1r_2}(kw_o) & G_{r_2r_2}(kw_o) \end{bmatrix} \qquad \text{(A-7b)}$$

$$\triangleq \frac{1}{T}\, \underline{Q}\, (kw_o), \qquad \text{(A-7c)}$$

where $\underline{Q}$ is defined to be a spectral density matrix. The power spectra in (A-7b) are assumed known. The cross power spectra are dependent on the hypothesized time delay τ, hence, $\underline{\Sigma}_{k|\tau}$ is a function of τ. The determinant of $\underline{\Sigma}_{k|\tau}$, dropping the frequency argument of the power spectra, is

$$\left|\underline{\Sigma}_{k|\tau}\right| = (G_{ss} + G_{n_1n_1})(\alpha^2 G_{ss} + G_{n_2n_2})$$

$$- (\alpha G_{ss}\, e^{-jkw_o\tau})(\alpha G_{ss}\, e^{jkw_o\tau}), \qquad \text{(A-8)}$$

which is seen to be independent of the hypothesized delay τ. Substituting (A-6b) into (A-6a) yields

$$p(\underline{R} \mid \tau) = c \exp\left(\frac{-1}{2} J_1\right) \qquad \text{(A-8a)}$$

where

$$J_1 = \sum_{k=-N}^{N} \underline{R}^{*\prime}(k)\underline{\Sigma}_{k|\tau}^{-1} \underline{R}(k) \qquad \text{(A-8b)}$$

and

$$c = \prod_{k=-N}^{N} \left(2\pi\left|\underline{\Sigma}_{k|\tau}\right|^{1/2}\right)^{-1} . \qquad \text{(A-8c)}$$

Since c does not depend on τ, maximizing $p(\underline{R}|\tau)$ with respect to τ is equivalent to minimizing J_1, since $E[J_1] \geq 0$.

At this point, an approximation is made based on the relationship between the Fourier series and the Fourier transform (FT). For large T, it can be shown that [52, pp. 23-25].

$$J_1 \approx \int_{-\infty}^{\infty} \underline{R}^{*\prime}(f)\ \underline{Q}^{-1}(f)\ \underline{R}(f)\ df \qquad \text{(A-9)}$$

where $\underline{R}(f) = [R_1(f), R_2(f)]'$ and where $R_i(f)$ represents the FT of $r_i(t)$. Evaluating $\underline{Q}^{-1}(f)$ from (A-7) and substituting into (A-9), J_1 can be written as

$$J_1 = J_2 - J_3 \qquad \text{(A-10a)}$$

where

$$J_2 = \left[\frac{\left|R_1(f)\right|^2}{G_{r_1r_1}(f)} + \frac{\left|R_2(f)\right|^2}{G_{r_2r_2}(f)} \right] \cdot \frac{1}{1 - C_{r_1r_2}(f)} \, df \ , \qquad \text{(A-10b)}$$

$$J_3 = 2\int_{-\infty}^{\infty} R_1(f)\ R_2^{*}(f) \ \frac{\alpha\ G_{ss}(f)\ e^{j2\pi f\tau}}{G_{r_1r_1}(f)\ G_{r_2r_2}(f)\ (1 - C_{r_1r_2}(f))} \, df, \qquad \text{(A-10c)}$$

and recall

$$C_{r_1r_2}(f) = \frac{\left|G_{r_1r_2}(f)\right|^2}{G_{r_1r_1}(f)\ G_{r_2r_2}(f)} = \frac{(\alpha G_{ss}(f))^2}{G_{r_1r_1}(f)\ G_{r_2r_2}(f)} \ . \qquad \text{(A-10d)}$$

Now J_2 does not depend on τ, therefore, maximizing J_3 with respect to τ is equivalent to minimizing J_1, and yields the ML estimate. The information about the true delay, D, is contained in $R_1(f)$ and $R_2(f)$ which are obtained from the received signals, $r_1(t)$ and $r_2(t)$, over the observation interval T. The other terms in (A-10c) depend on the known power spectra with the hypothesized delay τ appearing in the exponential term. The product $R_1(f)\ R_2^{*}(f)$ can be viewed as T times an estimate of the cross power spectrum, $\hat{G}_{r_1r_2}(f)$.

Re-writing J_3 in terms of $G_{r_1r_2}(f)$ and $C_{r_1r_2}(f)$ and noting that multiplying J_3 by a positive constant does not affect the location of the peak, the ML estimate of the time delay is the value of τ at which

$$R^{ML}_{r_1 r_2}(\tau) = \int_{-\infty}^{\infty} \frac{\hat{G}_{r_1 r_2}(f)}{\left|G_{r_1 r_2}(f)\right|} \, \frac{C_{r_1 r_2}(f)}{1 - C_{r_1 r_2}(f)} \, e^{j2\pi f\tau} \, df \qquad \text{(A-11)}$$

attains its peak value. From (A-11), it is seen that the ML weighting function is indeed given by (2-19),

$$W(f) = \frac{1}{\left|G_{r_1 r_2}(f)\right|} \cdot \frac{C_{r_1 r_2}(f)}{1 - C_{r_1 r_2}(f)} \quad . \qquad \text{(A-12)}$$

APPENDIX B

SIMULATION DETAILS-FIXED TIME DELAY

A computer simulation was conducted to corroborate the theoretical TDE performance predictions and to compare the performance of several of the GCC processors. The simulation results are presented in Section 4.3. The simulation was implemented on a VAX 11-750.

The signal model of (2-1) was simulated as follows. Two independent, Gaussian, pseudo-random sequences having zero mean and unit variance were generated to represent the additive noise sequences, $n_1(t)$ and $n_2(t)$. A third independent Gaussian sequence, again with zero mean and unit variance, was generated to represent the broadband source signal, $s(t)$. The required Gaussian random sequences were obtained by first generating random sequences which were uniformly distributed on the interval (0, 1) using the technique in [53, p. S-11]. These uniformly distributed random sequences were then transformed to Gaussian random variables using the method in [54, p. 953]. Specifically, if U_1 and U_2 represent a pair of random variables uniformly distributed on (0, 1), the transformation

$$X_1 = (-2 \ln U_1)^{1/2} \cos(2\pi U_2) \tag{B-1a}$$

$$X_2 = (-2 \ln U_1)^{1/2} \sin(2\pi U_2) \tag{B-1b}$$

yield a pair (X_1, X_2) of independent Gaussian random variables with zero mean and unit variance.

The three Gaussian random sequences were then processed through a 12th order (six cascaded 2nd order sections) low-pass Butterworth filter with a cut-off frequency of 100 Hz, relative to an assumed sampling frequency of 2048 Hz. The noise sequences were scaled to obtain the desired SNR. To obtain $r_1(t)$, the signal sequence was simply added to one of the noise sequences, and to obtain $r_2(t)$, the signal sequence was delayed (D = 4 samples) and added to the second noise sequence.

The GCC function of the sequences, $r_1(t)$ and $r_2(t)$ was computed following the procedure given in [53, pp. 2.3-1 - 2.3-18]. The sequences were processed using 512 point data segments and 50% overlap. Thus, each data segment represents 1/4 second of data. To obtain a 2 second observation time, 17 overlapped segments must be processed. Similarly, an 8 second observation time requires 65 overlapped segments. Each 512 point segment was weighted with a Hamming window and then zero padded to 1024 points. The Fourier transform of each 1024 point segment, for both $r_1(t)$ and $r_2(t)$, was computed using a fast Fourier transform (FFT). The resulting Fourier coefficients were used to compute auto and cross-power spectral estimates for each segment, and these estimates were then averaged to obtain the final estimates of the spectra for the observation interval.

Using the spectral estimates, the GCC weighting functions were computed and multiplied by the estimate of the cross power spectrum. The inverse FFT of the result was computed to yield an estimate of the GCC functions. These functions were searched for their peak values for delay values over the range ± 128 samples. Thus, the value of D_0 in seconds is 128/2048 = 1/16 second. The time delay estimate is given by the delay value corresponding to the peak of the GCC function in the search interval. The simulation was conducted for two observation times, 2 and 8 seconds, and over a range of SNR values from 0 to -15 dB. A total of 2000 trials was conducted at each SNR to obtain the experimental time delay variances, which are displayed in Figures 4.4-4.8.

APPENDIX C

NUMERICAL EVALUATION OF P[A]

In this appendix, techniques for numerically evaluating the expression for P[A] in (3-16a) are discussed. Recall

$$P[A] = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-(x-\overline{x})^2/2} \left[\int_{-\infty}^{\lambda x} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy \right]^{M-1} dx, \quad (C-1)$$

where $\overline{x} = f(B, T, SNR)$ and $\lambda = g(SNR)$ are given in (3-16). This equation can be re-written in terms of the Q function to give

$$P[A] = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-(x-\overline{x})^2/2} [Q(-\lambda x)]^{M-1} dx \qquad (C-2a)$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-y^2/2} dy \quad , \qquad (C-2b)$$

and note that $Q(-\infty) = 1$, $Q(0) = 1/2$, and $Q(-x) = 1 - Q(x)$.

Consider the integral on the right hand side of (C-2a). Making the change of variable $u = (x-\overline{x})/ 2$ yields

$$1 - P[A] = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-u^2} Q(-\lambda(2u + \overline{x}))^{M-1} du. \qquad (C-3)$$

This expression for 1 - P[A] can be approximated using the Gauss-Hermite Quadrature formula [54, p. 890, eq. 25.4.46]. Consider a function G(x), then

$$\int_{-\infty}^{\infty} e^{-x^2} G(x)\, dx \approx \sum_{i=1}^{m} W_{m,i}\, G(x_{m,i}) \quad . \tag{C-4}$$

That is, the integral can be approximated by a weighted sum of the function G(x) evaluated for specific arguments. In (C-4), $x_{m,i}$ represents the ith zero of the mth order Hermite polynomial, and $W_{m,i}$ is the weight corresponding to $x_{m,i}$ and is a function of the mth order Hermite polynomial. If the zeroes are arranged in ascending order and the weights are arranged in corresponding order, then the following relations hold,

$$x_{m,i} = - x_{m,m+1-i} \quad , \; i=1,2,\ldots,m \tag{C-5a}$$

and

$$W_{m,i} = W_{m,m+1-i} \quad . \tag{C-5b}$$

The zeroes and weights are available in tables, for example see [55, Table 5]. Note that most tables make use of the property of (C-5) and only list the positive zeroes and their corresponding weights.

Applying the approximation of (C-4) to (C-3) yields

$$1 - P[A] \approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^{m} W_{m,i}\, [Q(-\lambda(\sqrt{2}\, x_{m,i} + \bar{x}))]^{M-1} \quad . \tag{C-6}$$

At the end of this appendix, a listing is given of the FORTRAN program BOUND.FOR, which computes the CRLB, the CPE, and the ZZLB as functions of SNR. Additionally, listings of the subroutines PROB.FOR and Q.FOR, which evaluate P[A] and the Q function, respectively, are included. The values of P[A] which are of greatest interest are of the order 10^{-4} or smaller. Thus, the approximation in (C-6) must be evaluated to 5 or more significant digits. This requires a large value for m (m = 60 in the subroutine) and the use of double precision calculations. An alternative method of evaluating P[A] is proposed by Nuttall in [45], which reduces the number of significant digits required for the calculation. These two techniques have been found to yield very similar values for P[A].

```
C***********************************************************************
C
C       CRLB - CPE - ZZLB ROUTINE
C
C       VAX-11 FORTRAN SOURCE FILENAME:         BOUND.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING    KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                    PROGRAMMER(S)
C       --------        ----                    -------------
C       00.0            JUNE 1983               K SCARBROUGH
C       01.0            JULY 1983               K SCARBROUGH
C***********************************************************************
C
C       PURPOSE
C
C               This program computes the Cramer-Rao lower bound (CRLB),
C               Ianniello's estimate of Correlator Performance (CPE),
C               and the Ziv-Zakai lower bound (ZZLB), for the case of
C               low passed gaussian signal & noise sequences.
C
C       ROUTINE(S) ACCESSED OR CALLED BY THIS ROUTINE
C
C               PROB, Q, HEADER, SGOPEN, SGTRAN
C
C***********************************************************************
C
C       INPUT PARAMETERS
C
C               SF    =  SAMPLING FREQUENCY
C               FREQ  =  CUTOFF FREQUENCY
C               TIME  =  OBSERVATION TIME
C               WIDTH =  CORRELATION WINDOW WIDTH
C               SNR1  =  MIN SNR VALUE(DB)
C               SNR2  =  MAX SNR VALUE(DB)
C               SSTEP =  SNR STEP SIZE
C               EPS   =  EPSILON VALUE
C               PSTEP =  STEP SIZE FOR PRINTING CRB/CPE VALUES
C
C***********************************************************************
C
C       NOTE :  The equations for the CPE and the ZZLB assume
C               that the signal has a low-passed (flat) power spectrum.
C
C***********************************************************************
C
      REAL  CRB(8002), CPE(8002), ZZB(8002), SNRSAV(4001), PASAV(4001)
      REAL  BW, CPELIM, CRBVAL, ZZBLIM, FREQ, EPS
      REAL  ETA2, ETAZ, ETAZ2, PA, PE, Q_ETAZ
      REAL  SNR, SNR1, SNR2, SSTEP, SNRDB, SNRFCT, SQSNR1
      REAL  TIME, ZBW, ZBWRD, ZZ1, ZZ2
      INTEGER  MEXP, NPT, NPT2, PSTEP, WIDTH, UNITP
      LOGICAL  EQUAL1, EQUAL2
      CHARACTER*40  CRBNAM, CPENAM, ZZBNAM
      DATA   PI /3.14592654/
C
C*************************************************************
```

```
C
C  INITIALIZE CONSTANTS
C
      NPT = 0
      EQUAL1 = .FALSE.
      EQUAL2 = .FALSE.
      TWOPI = 2. * PI
      SQTPI = SQRT(TWOPI)
      UNITP = 99
C
C***********************************************************
C
C  FORMAT STATEMENTS
C
    1 FORMAT(A)
    3 FORMAT(' ',A,F10.2,A)
    5 FORMAT(' ',A,I10,A)
    7 FORMAT(' ',A,A)
    9 FORMAT(' ',/,4X,'SNR(dB)',9X,'CRLB',12X,'CPE',11X,'ZZLB',/)
   11 FORMAT('1',//,4X,'SNR(dB)',6X,'PR(anomaly)',4X,'ZZB Term 1',
     &                                          5X,'ZZB Term 2',/)
C
C***********************************************************
C
C  ACCEPT INPUT PARAMETERS
C
      TYPE 1, '$SAMPLING FREQUENCY    : '
      READ *, FS
      TYPE 1, '$OBSERVATION TIME(sec): '
      READ *, TIME
      TYPE 1, '$CORR WINDOW(samples)  : '
      READ *, WIDTH
      TYPE 1, '$CUTOFF FREQ           : '
      READ *, FREQ
      TYPE 1, '$MIN SNR VALUE(dB)     : '
      READ *, SNR1
      TYPE 1, '$MAX SNR VALUE(dB)     : '
      READ *, SNR2
      TYPE 1, '$SNR STEPSIZE(dB)      : '
      READ *, SSTEP
      TYPE 1, '$EPSILON VALUE         : '
      READ *, EPS
      TYPE 1, '$PRINT STEPSIZE        : '
      READ *, PSTEP
C
C***********************************************************
C
C  COMPUTE PROGRAM VARIBLES
C
      BW = FREQ
      ZBW = BW * 2.
      D0 = WIDTH / FS
      CPELIM = (D0**2) / 12.
      ZZBLIM = 2. * CPELIM
      MEXP = IFIX(2. * D0 * BW)
      CRBVAL = 3.0 / (8.0 * (PI**2) * (BW**3) *TIME)
C
```

```
C************************************************************
C
C  COMPUTE CRLB, CPE & ZZLB AS FUNCTIONS OF SNR
C
        DO SNRDB = SNR1, SNR2, SSTEP
           NPT = NPT + 1
           JPT = 2 * NPT - 1
           IPT = 2 * NPT
C
           SNRSAV(NPT) = SNRDB
           SNR = 10.**(SNRDB / 10)
           SNRFCT = (SNR**2) / (1. + 2. * SNR)
           SQSNR1  = SQRT(SNRFCT + 1.)
C
C  COMPUTE CRLB
C
           CRBTMP = CRBVAL / SNRFCT
C
C  COMPUTE CPE
C
           PA = PROB(SNR,BW,TIME,MEXP)
           PASAV(NPT) = PA
           CPETMP = PA * CPELIM + (1. - PA) * CRBTMP
C
C  COMPUTE ZZLB
C
           ZTMP1 = LOG((1. + SQSNR1) / 2.)
           ZTMP2 = (SQSNR1 - 1.) / (2. * SQSNR1)
           ZTMP3 = EXP(-ZBW * TIME * (ZTMP1 - ZTMP2))
C
           ETAZ = SQRT(ZBW * TIME * (SQSNR1 - 1.) / SQSNR1)
           ETAZ2 = ETAZ**2
           Q_ETAZ = Q(ETAZ)
C
           ZTMPA = (ETAZ2 - 1) * Q_ETAZ
           ZTMPB = ETAZ * EXP(-ETAZ2 / 2.) / SQTPI
           ZTMPC = 0.50
C
           PE = Q_ETAZ * ZTMP3
           ZZ1 = PE * ZZBLIM
           ZZ2 = 2 * CRBTMP * (ZTMPA - ZTMPB + ZTMPC)
C
           ZZBTMP = ZZ1 + ZZ2
C
C  COMPUTE LOG10 OF STANDARD DEVIATION OF TIME DELAY ESTIMATE
C
           CRBTMP = LOG10(CRBTMP) / 2.
           CPETMP = LOG10(CPETMP) / 2.
           ZZBTMP = LOG10(ZZBTMP) / 2.
C
C  DETERMINE THRESHOLD SNR VALUES
C
           IF (.NOT.EQUAL1) THEN
                DIFF = CPETMP - CRBTMP
                IF (DIFF .LT. ABS(EPS * CRBTMP)) THEN
                    EQUAL1 = .TRUE.
                    INDEX1 = NPT
                    INDEX2 = IPT
                END IF
           END IF
C
           IF (.NOT.EQUAL2) THEN
                DIFF = ZZBTMP - CRBTMP
                IF (DIFF .LT. ABS(EPS * CRBTMP)) THEN
                    EQUAL2 = .TRUE.
                    INDEX3 = NPT
                    INDEX4 = IPT
                END IF
           END IF
C
C  STORE SNR VALUES & BOUND VALUES IN OUTPUT ARRAY
C   (SNR VALUES NEEDED FOR XY PLOTTING)
C
           CRB(JPT) = SNRSAV(NPT)
           CRB(IPT) = CRBTMP
           CPE(JPT) = SNRSAV(NPT)
           CPE(IPT) = CPETMP
           ZZB(JPT) = SNRSAV(NPT)
           ZZB(IPT) = ZZBTMP
C
        END DO
C
C*************************************************************
C
C  OPEN OUTPUT FILES, OUTPUT CRLB & CPE VALUES TO DISK
C
        NPT2 = NPT * 2
        CALL SGOPEN(1,'WRITE','CRB FILE : ',CRBNAM,'REAL',NPT2)
        CALL SGOPEN(2,'WRITE','CPE FILE : ',CPENAM,'REAL',NPT2)
        CALL SGOPEN(3,'WRITE','ZZB FILE : ',ZZBNAM,'REAL',NPT2)
        CALL SGTRAN(1,'WRITE','REAL',CRB,NPT2)
        CALL SGTRAN(2,'WRITE','REAL',CPE,NPT2)
        CALL SGTRAN(3,'WRITE','REAL',ZZB,NPT2)
C
C*************************************************************
C
C  CREATE LISTING FILE:  WRITE HEADER
C                        ECHO INPUT PARAMETERS
C                        OUTPUT CRB - CPE INTERCEPT VALUES
C                        OUTPUT CRB - ZZB INTERCEPT VALUES
C                        OUTPUT CRB, CPE & ZZB VALUES
C
        CALL HEADER (UNITP,'*** CRLB - CPE - ZZLB PROGRAM ***')
        WRITE(UNITP,3)  ' Sampling Frequency   : ', FS   , ' Hz'
        WRITE(UNITP,3)  ' Observation Time     : ', TIME , ' sec'
        WRITE(UNITP,5)  ' Correlation Window   : ', WIDTH, ' samples'
        WRITE(UNITP,5)  ' Ind Corr Values      : ', MEXP
        WRITE(UNITP,3)  ' Cutoff Freq          : ', FREQ , ' Hz'
        WRITE(UNITP,3)  ' Min SNR Value        : ', SNR1 , ' dB'
        WRITE(UNITP,3)  ' Max SNR Value        : ', SNR2 , ' dB'
        WRITE(UNITP,3)  ' SNR Stepsize         : ', SSTEP, ' dB'
        WRITE(UNITP,*)  ' Epsilon Value        : ', EPS
C
        WRITE(UNITP,*)
```

```
      WRITE(UNITP,3)  ' Threshold SNR(dB)    : ', SNRSAV(INDEX1),' dB'
      WRITE(UNITP,*)  ' Prob of Anomaly      : ', PASAV(INDEX1)
      WRITE(UNITP,*)  ' CPE Value            : ', CPE(INDEX2)
      WRITE(UNITP,*)  ' CRLB Value           : ', CRB(INDEX2)
C
      WRITE(UNITP,*)
      WRITE(UNITP,3)  ' Threshold SNR(dB)    : ', SNRSAV(INDEX3),' dB'
      WRITE(UNITP,*)  ' CPE Value            : ', CPE(INDEX4)
      WRITE(UNITP,*)  ' ZZLB Value           : ', CRB(INDEX4)
C
      WRITE(UNITP,*)
      WRITE(UNITP,7)  ' CRLB Filename        : ', CRBNAM
      WRITE(UNITP,7)  ' CPE Filename         : ', CPENAM
      WRITE(UNITP,7)  ' ZZLB Filename        : ', ZZBNAM
C
      WRITE(UNITP,9)
      DO IPT = 1, NPT2, PSTEP*2
         JPT = IPT + 1
         WRITE(UNITP,*) CRB(IPT), CRB(JPT), CPE(JPT), ZZB(JPT)
      END DO
C
      STOP
      END
```

```
C***********************************************************************
C
C       PROBABILITY OF ANOMALY ROUTINE
C
C       VAX-11 FORTRAN SOURCE FILENAME:         PROB.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING    KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                    PROGRAMMER(S)
C       --------        ----                    -------------
C       00.0            MAY 1983                KENT SCARBROUGH
C
C***********************************************************************
C
C       CALLING SEQUENCE
C
C               VALUE = PROB(SNR,BW,T,M)
C
C       PURPOSE
C
C               This routine calculates the probability of anomaly
C               for band-limited signal and noise sequences. The
C               signal & noise are assumed to have the same ideal
C               band(low)-passed spectral characteristics.
C
C       ROUTINE(S) ACCESSED OR CALLED BY THIS ROUTINE
C
C               Q
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               SNR     Signal-to-Noise ratio
C
C               BW      Bandwidth of signal and noise
C
C               T       Observation time (seconds)
C
C               M       Number of independent correlation estimates
C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               NONE
C
C***********************************************************************
C
C       NOTE 1: The equation for PR(anomaly) was derived following
C               Ianniello's paper in IEEE TRANS ASSP, Dec 1982.
C
C       NOTE 2: The formula for the Gauss-Hermite Quadrature rule
C               can be found in "Handbook of Mathematical Functions"
C               by Abramowitz & Stegun, Nat'l Bureau of Standards, 1964.
C               See equation 25.4.46, page 890.
C
C       NOTE 3: The weights and zeroes for the G-H Quad. rule were
C               taken from "Gaussian Quadrature Formulas" by Stroud
C               & Secrest, Prentice Hall, 1966. See Table 5.
C
```

```
C**********************************************************************
C
        REAL FUNCTION PROB(SNR,BW,T,M)
C
        DOUBLE PRECISION   WEIGHT(60), ZERO(60)
        LOGICAL  FIRST/.TRUE./
        INTEGER  M, I, IWT, JWT, NWT
        REAL  ALPHA, BETA, BW, P, PBAR, PTMP
        REAL  SNR, T, X
C
        DATA  SQRTPI/1.772453851/, SQRT2/1.414213562/
C
C**********************************************************************
C
        DATA (ZERO(I),I=1,30)/ 10.15910924618,9.520903677013,
     &          8.992398001405,8.520569284118,8.085188654249,
     &          7.675839937505,7.286276594396,6.912381532189,
     &          6.551259167063,6.200773557993,5.859290196394,
     &          5.525521086139,5.198426534576,4.877150077473,
     &          4.560973757936,4.249286435956,3.941560733926,
     &          3.637335876171,3.336204653548,3.037803338231,
     &          2.741803748070,2.447906902308,2.155837871230,
     &          1.865341531233,1.576179011975,1.288124674869,
     &          1.000963499561,0.714488781673,0.428500064221,
     &          0.142801238703 /
C
C**********************************************************************
C
C  NOTE: THE FIRST TWO (& LAST TWO) WEIGHTS HAVE BEEN SET TO
C        ZERO DUE TO THE DYNAMIC RANGE LIMITATIONS (E-38 to E+38)
C        OF FORTRAN 77 ON THE VAX WITHOUT THE FLOATING_G OPTION.
C        THE ACTUAL VALUE OF THE WEIGHTS SHOULD BE:
C
C               WEIGHT(1) = .110958724797E-44 = WEIGHT(60)
C               WEIGHT(2) = .243974758815E-39 = WEIGHT(59)
C
C
        DATA (WEIGHT(I),I=1,30) /   .000000000000E-00,.000000000000E-00,
     &            .377162672712E-35,.133255961176E-31,.171557314767E-28,
     &            .102940599717E-25,.334575695575E-23,.651256725750E-21,
     &            .815364047302E-19,.692324790958E-17,.415244410969E-15,
     &            .181662457626E-13,.594843051606E-12,.148895734906E-10,
     &            .2899359012808E-9,.4456822775226E-8,.5475554619277E-7,
     &            .5433516134205E-6,.4394286936267E-5,.2918741904156E-4,
     &            .1602773346818E-3,.7317735569655E-3,.2791324828953E-2,
     &            .8932178360308E-2,.2406127276611E-1,.5471897093218E-1,
     &            .1052987636977856,.1717761569188851,.2378689049586589,
     &            .2798531175228290  /
C
C**********************************************************************
C
        SAVE SQRT2, SQRTPI, FIRST, WEIGHT, ZERO
C
        NWT = 60
C
C************************************************************
C
```

```
C  ON FIRST CALL TO ROUTINE, INITIALIZE SECOND HALF OF
C  WEIGHTING AND ZERO ARRAYS
C
      IF (FIRST) THEN
         DO IWT = 1, NWT / 2
            JWT = NWT - IWT + 1
            WEIGHT(JWT) = WEIGHT(IWT)
            ZERO(JWT) = - ZERO(IWT)
         END DO
      END IF
C
C***********************************************************
C
C  COMPUTE CONSTANTS - FUNCTIONS OF SNR
C
      ALPHA = SQRT2 * SQRT(BW * T) * SNR
     &            / SQRT( SNR**2 + (1 + SNR)**2 )
      BETA  = SQRT( 1 + (SNR**2) / (1 + SNR)**2 )
C
C***********************************************************
C
C  USE GAUSS-HERMITE QUADRATURE TO COMPUTE P(anomaly)
C
      PBAR = 0.0
      DO IWT = 1, NWT
         X = BETA * (SQRT2 * ZERO(IWT) + ALPHA)
         P = 1.0 - Q(X)
         PTMP = WEIGHT(IWT) * (P**(M-1)) / SQRTPI
         PBAR = PBAR + PTMP
      END DO
      PROB = 1.0 - PBAR
C
      RETURN
      END
```

```
C************************************************************************
C
C       Q FUNCTION                              (UTILITY LIBRARY)
C
C       VAX-11 FORTRAN SOURCE FILENAME:         Q.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING    KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                    PROGRAMMER(S)
C       --------        ----                    -------------
C       00.0            MAR 31, 1983            F W RATCLIFFE
C       01.0            APR 08, 1983            K SCARBROUGH
C
C************************************************************************
C
C       CALLING SEQUENCE
C
C               VALUE = Q (X)
C
C       PURPOSE
C
C               This routine evaluates the function Q(x), that is,
C       the integral from x to + infinity of a Gaussian distribution
C       function for a zero mean, unit variance random variable.
C
C       ROUTINE(S) ACCESSED OR CALLED BY THIS ROUTINE
C
C               NONE
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               X       value at which to evaluate the error function
C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               NONE
C
C************************************************************************
C
C       NOTE 1: This routine uses two different approximations
C               to compute Q(x) depending on the value of x.
C               For  0 < X < 2.326 , see Note 2.
C               For      X > 2.326 , see Note 3.
C
C       NOTE 2: For values of X in the range 0 < X < 2.326,
C               (i.e., for  .5 > Q(x) > .01 ),  equation 26.2.17
C               on page 932 of reference 1 is used to evaluate Q(x).
C
C       NOTE 3: For values of X > 2.326 (i.e., for Q(x) < .01 )
C               equation 13 on page 641 of reference 2 is used
C               to evaluate Q(x).
C
C       NOTE 4: These approximations insure that the error in the
C               value of Q(x) is less than 0.001 % .
C
C       NOTE 5: This routine works for positive and negative values
C               of X.  Note that Q(-x) = 1 - Q(x).
```

```
C
C       REFERENCE 1:    Abramowitz and Stegun, HANDBOOK OF MATHEMATICAL
C                       FUNCTIONS, National Bureau of Standards, 1964.
C
C       REFERENCE 2:    P. O. Borjesson, C. E. Sundberg, SIMPLE
C                       APPROXIMATIONS OF THE ERROR FUNCTION Q(x) FOR
C                       COMMUNICATION APPLICATIONS, IEEE Transactions
C                       on Communications, March 1979, pp 639-643.
C
C***********************************************************************
C
        REAL FUNCTION Q (X)
C
        DOUBLE PRECISION F(5), P, T, ZEXP, QTEMP, SQT2PI
        REAL  A, B, X, XPOS, DEN
C
        DATA  SQT2PI /2.506628275/, P /0.2316419/
        DATA  F /0.319381530, -.356563782, 1.781477937, -1.821255978,
     &          1.330274429/
        DATA  A /0.279/, B /7.123/
C
C*********************************
C
C  Initialize Variables
C
C
        QTEMP = 0.0
        XPOS = ABS(X)
        ZEXP = EXP(-(X**2)/2.)/SQT2PI
C
C*********************************************
C
C  Determine range of X, and compute Q(x)
C  using the appropriate approximation
C
C
        IF  (XPOS.LE.2.326)  THEN
                T = 1. / (1. + P * XPOS)
                DO I = 1, 5
                    QTEMP = QTEMP + F(I) * (T**I)
                END DO
                QTEMP = QTEMP * ZEXP
        ELSE
                DEN = (1. - A) * XPOS + A * SQRT(B + XPOS**2)
                QTEMP = ZEXP / DEN
        END IF
C
C******************************************
C
C  Check for negative values of X, and
C  adjust Q(x) if necessary
C
C
        IF  (X.GE.0.)  THEN
                Q = QTEMP
        ELSE
                Q = 1.0 - QTEMP
```

```
      END IF
C
      RETURN
      END
```

APPENDIX D

DERIVATION OF EXPRESSION FOR $P_e(\tau, \tau+\Delta)$

The derivation of the expression for $P_e(\tau, \tau+\Delta)$, equation (3-30), is given in [29] and is included in this appendix for completeness. Also considerations leading to the bound for $P_e(\Delta)$ in (3-31) are given.

Recall $P_e(\tau, \tau+\Delta)$ represents the minimum attainable probability of error for the binary decision problem of deciding which of the two hypothesized delay values, τ or $\tau+\Delta$, is correct. For large BT, $P_e(\tau, \tau+\Delta)$ can be approximated by the Chernoff formula [23, p. 125, eq. (484)]

$$P_e(\tau, \tau+\Delta) \approx \frac{1}{2} \exp[l(q_o) + q_o^2\, l''/2]\; Q[q_o \sqrt{l''(q_o)}]$$

$$+ \frac{1}{2} \exp[l(q_o) + (1 - q_o)^2\, l''(q_o)/2]\; Q[(1 - q_o) \sqrt{l''(q_o)}] \qquad \text{(D-1a)}$$

where

$$l(q) = \ln \int_{-\infty}^{\infty} [p(\underline{R} \mid \tau)]^q\, [p(\underline{R} \mid \tau+\Delta)]^{1-q}\, d\underline{R} \qquad \text{(D-1b)}$$

and where q_o is the value for which $l'(q) = 0$ (note, $l'(q) = dl/dq$ and $l''(q) = d^2l/dq^2$). $\underline{R}$ is the observation vector made up of the Fourier coefficients of the received signals, $r_1(t)$ and $r_2(t)$, and $p(\underline{R}|\tau)$ is the conditional pdf of $\underline{R}$ given, as discussed in Appendix A.

Substituting the expressions for $p(\underline{R}|\tau)$ and $P(\underline{R}|\tau+\Delta)$ from (A-6) into (D-1b) and carrying out the integration yields

$$l(q) = -\frac{1}{2}\sum_{k}\Bigg\{q \ln\left[\det\underline{\Sigma}_{k|\tau}\right] + (1-q)\ln\left[\det\underline{\Sigma}_{k|\tau+\Delta}\right] + \ln\left[\det\left(q\,\underline{\Sigma}_{k|\tau}^{-1} + (1-q)\,\underline{\Sigma}_{k|\tau+\Delta}^{-1}\right)\right]\Bigg\} \qquad \text{(D-2)}$$

As noted in (A-8), $\det\underline{\Sigma}_{k|\tau}$ is independent of τ. Recall from (A-7) that

$$\underline{\Sigma}_{k|\tau} = \begin{bmatrix} G_{ss}(kw_0) + G_{nn}(kw_0) & \alpha G_{ss}(kw_0)\, e^{-jkw_0\tau} \\ \alpha G_{ss}(kw_0)\, e^{jkw_0\tau} & \alpha^2 G_{ss}(kw_0) + G_{nn}(kw_0) \end{bmatrix} \qquad \text{(D-3a)}$$

thus

$$\underline{\Sigma}_{k|\tau}^{-1} = \frac{1}{\det\underline{\Sigma}_{k|\tau}}\begin{bmatrix} \alpha^2 G_{ss}(kw_0) + G_{nn}(kw_0) & -\alpha G_{ss}(kw_0)\, e^{-jkw_0\tau} \\ -\alpha\, G_{ss}(kw_0)\, e^{-jkw_0\tau} & G_{ss}(kw_0) + G_{nn}(kw_0) \end{bmatrix}. \qquad \text{(D-3b)}$$

Substituting (D-3) into (D-2) and simplifying yields

$$l(q) = -\frac{1}{2}\sum_{k} \ln\left[1 + 4q(1-q)\,\gamma(kf_0,\Delta)\right] , \qquad \text{(D-4)}$$

where $\gamma(f,\Delta)$ is given in (3-30e). Then taking the first and second derivatives of $l(q)$ with respect to q gives

$$l'(q) = -\sum_{k}\frac{2\,(1-2q)\,\gamma(kf_0,\Delta)}{1 + 4q(1-q)\,\gamma(kf_0,\Delta)} \qquad \text{(D-5a)}$$

and

$$l''(q) = \sum_k \left[\frac{4\ \gamma(kf_o,\Delta)}{1 + 4q(1-q)\ \gamma(kf_o,\Delta)} + \frac{8(1-2q)^2\ \gamma^2(kf_o,\Delta)}{[1 + 4q(1-q)\ \gamma(kf_o,\Delta)]^2}\right] . \qquad \text{(D-5b)}$$

Note that $l'(q_o) = 0$ when $q_o = 1/2$. For large BT, the summations in (D-4) and (D-5) can be approximated as integrals using the large T approximation as in Appendix A. Evaluating $l(q)$ and $l''(q)$ at $q_o = 1/2$ yields

$$l(1/2) \simeq -\frac{T}{2}\int_{-\infty}^{\infty} \ln\,[1 + \gamma(f,\Delta)]\ df \qquad \text{(D-6a)}$$

and

$$l''(1/2) \simeq 4T\int_{-\infty}^{\infty} \frac{\gamma(f,\Delta)}{1 + \gamma(f,\Delta)}\ df \quad . \qquad \text{(D-6b)}$$

Substituting $q_o = 1/2$ and (D-6) into (D-1a) and simplifying yields the expression for $P_e(\tau,\ \tau+\Delta)$ given in (3-30), namely

$$P_e(\tau,\ \tau+\Delta) = P_e(\Delta) \qquad \text{(D-7a)}$$

where

$$P_e(\Delta) = \exp(a(\Delta) + b(\Delta))\ Q(\sqrt{2b(\Delta)}), \qquad \text{(D-7b)}$$

and where

$$a(\Delta) = -\ T\int_0^{\infty} \ln\ (1 + \gamma(f,\Delta))\ df \quad , \qquad \text{(D-7c)}$$

$$b(\Delta) = T\int_0^{\infty} \gamma(f,\Delta)/(1 + \gamma(f,\Delta))\ df \quad , \qquad \text{(D-7d)}$$

and

$$\gamma(f,\Delta) = \frac{G_{ss}(f)^2\ \sin^2\pi f\Delta}{[G_{n_1n_1}(f) + G_{n_2n_2}(f)]\ G_{ss}(f) + G_{n_1n_1}(f)\ G_{n_2n_2}(f)} \quad . \qquad \text{(D-7e)}$$

Now to obtain the bound for $P_e(\Delta)$ given in (3-31), let $\gamma = \gamma(f,\Delta)$ and note that $0 \leq \gamma/(1+\gamma) \leq 1$. Also, note that

$$\ln(1+\gamma) = -\ln\left(1 - \frac{\gamma}{1+\gamma}\right) = \sum_{n=1}^{\infty} \frac{1}{n} \left(\frac{\gamma}{1+\gamma}\right)^n \tag{D-8a}$$

thus,

$$0 \leq \ln(1+\gamma) - \frac{\gamma}{1+\gamma} = \sum_{n=2}^{\infty} \frac{1}{n} \left(\frac{\gamma}{1+\gamma}\right)^n \tag{D-8b}$$

$$= \frac{1}{2} \left(\frac{\gamma}{1+\gamma}\right)^2 \sum_{n=0}^{\infty} \frac{2}{n+2} \left(\frac{\gamma}{1+\gamma}\right)^n$$

$$\leq \frac{1}{2} \left(\frac{\gamma}{1+\gamma}\right)^2 \sum_{n=0}^{\infty} \left(\frac{\gamma}{1+\gamma}\right)^n$$

$$= \frac{1}{2} \frac{\gamma^2}{1+\gamma}, \leq \frac{1}{2}\gamma^2 \quad . \tag{D-8c}$$

From (D-8b) it is easily seen that

$$0 \geq a(\Delta) + b(\Delta)$$

$$= -T\int_0^{\infty} \left[\ln(1 + \gamma(f,\Delta)) - \frac{\gamma(f,\Delta)}{1 + \gamma(f,\Delta)}\right] df$$

$$\geq -\frac{T}{2}\int_0^{\infty} \gamma^2(f, \Delta)\, df \quad . \tag{D-9}$$

Also, since $\gamma/(1+\gamma) \leq \gamma$,

$$0 \leq b(\Delta) \leq T \int_0^\infty \gamma(f,\Delta)\, df \quad . \tag{D-10}$$

Substituting (D-9) and (D-10) into (D-7b), immediately yields (3-31),

$$P_e(\Delta) \geq \exp\left[\frac{-T}{2}\int_0^\infty \gamma^2(f,\Delta)\, df\right] \cdot Q\left[\left(2T\int_0^\infty \gamma(f,\Delta)\, df\right)^{1/2}\right] \quad . \tag{D-11}$$

APPENDIX E

APPROXIMATION FOR THE THRESHOLD SNR

The approximations leading to the expression for the threshold SNR of (4-2) are given in this appendix. The initial assumption is that the threshold SNR can be closely approximated by the SNR which satisfies the condition

$$\sigma^2_{ZZLB} = 2\sigma^2_{CRLB} \; . \tag{E-1}$$

From (3-61)

$$\sigma^2_{ZZLB} = \frac{1}{2\eta^2}\left[[(\eta z)^2 - 1]\, Q(\eta z) - \frac{\eta z}{\sqrt{2\pi}}\, e^{-(\eta z)^2/2} + \frac{1}{2}\right]$$

$$+ \; 2P_e \, D_o^2/3 \tag{E-2a}$$

where

$$\eta z = \left[2BT \; \frac{\sqrt{1 + SNR'} - 1}{\sqrt{1 + SNR'}}\right]^{1/2} \; , \tag{E-2b}$$

and

$$P_e = \exp\left\{-BT\left[2 \ln\left(\frac{1 + \sqrt{1 + SNR'}}{2}\right) - \frac{\sqrt{1 + SNR'} - 1}{\sqrt{1 + SNR'}}\right]\right\} Q(\eta z) . \tag{E-2c}$$

For large BT, the threshold SNR is small, then assume that SNR' << 1. This assumption allows the following approximations to be made,

$$\frac{1 + \sqrt{1 + SNR'}}{2} = \frac{1 + \sqrt{1 + SNR'}}{2}\left[\frac{1 + \sqrt{1 + SNR'}}{1 + \sqrt{1 + SNR'}}\right]$$

$$= 1 + \frac{SNR'}{2(1 + \sqrt{1 + SNR'})}$$

$$\simeq 1 + \frac{SNR'}{4} \qquad \text{(E-3a)}$$

and similarly,

$$\frac{\sqrt{1 + SNR'} - 1}{\sqrt{1 + SNR'}} = \frac{\sqrt{1 + SNR'} - 1}{\sqrt{1 + SNR'}}\left[\frac{\sqrt{1 + SNR'} + 1}{\sqrt{1 + SNR'} + 1}\right]$$

$$= \frac{SNR'}{1 + SNR' + \sqrt{1 + SNR'}}$$

$$\simeq \frac{SNR'}{2} \quad . \qquad \text{(E-3b)}$$

Substituting (E-3a) and (E-3b) into (E-2a) and (E-2b) yields

$$\eta z \simeq \sqrt{BT\ SNR'} \qquad \text{(E-4a)}$$

and

$$P_e \simeq \exp\left\{-BT\ [2\ \ln(1 + SNR'/4) - SNR'/2]\right\}\ Q(\sqrt{BT\ SNR'}). \qquad \text{(E-4b)}$$

Note that ln(1 + x) ≃ x for small x, thus

$$P_e \simeq Q(\sqrt{BT\ SNR'}). \qquad \text{(E-5)}$$

Further suppose that BT SNR' is large enough so that

$$e^{-(\eta z)^2/2} \simeq e^{-(BT\ SNR')/2} \simeq 0. \qquad \text{(E-6)}$$

Substitution of (E-4)-(E-6) into (E-2a) yields

$$\sigma^2_{ZZLB} \simeq \frac{1}{2\eta^2} \; [(BT\ SNR' - 1)\ Q(\sqrt{BT\ SNR'}) + \tfrac{1}{2}]$$

$$+ \; 2\ Q(\sqrt{BT\ SNR'})\ D_o^2/3 \quad . \qquad \text{(E-7)}$$

Recall from (3-63) and (3-64) that

$$\sigma^2_{CRLB} = \frac{1}{4\eta^2} \qquad \text{(E-8a)}$$

where

$$\eta^2 = 2\pi^2\ SNR'\ B^3T/3 \quad . \qquad \text{(E-8b)}$$

Substituting (E-7) and (E-8a) into (E-1) and simplifying yields

$$(BT\ SNR' - 1)\ Q(\sqrt{BT\ SNR'}) + 4Q(\sqrt{BT\ SNR'})\ D_0^2\ \eta^2/3 \simeq \frac{1}{2} \qquad \text{(E-9a)}$$

and substituting for η from (E-8b) gives

$$BT\ SNR'\ Q(\sqrt{BT\ SNR'})\ [1 - 1/(BT - SNR') + 8\pi^2B^2D_0^2/9] \simeq \frac{1}{2} \quad . \qquad \text{(E-9b)}$$

In the derivation of the ZZLB it was assumed that $D_o >> 1/2B$, thus, $BD_o >> 1$ and (E-9) becomes

$$BT\ SNR'\ Q(\sqrt{BT\ SNR'}) \simeq \frac{9}{16\ \pi^2B^2D_o^2} \quad . \qquad \text{(E-10)}$$

Finally solving (E-10) for SNR' yields the approximate expression for SNR'_{th} of (4-2),

$$SNR'_{th} \simeq \frac{1}{\sqrt{BT}} \left[F_+^{-1} \left[\left(\frac{3}{4\pi\ BD_o} \right)^2 \right] \right]^2 \quad , \qquad \text{(E-11a)}$$

where

$$F(x) = x^2 Q(x), \qquad \text{(E-11b)}$$

and $F_+^{-1}(\cdot)$ denotes the larger of the two solutions of the inverse of $F(\cdot)$.

APPENDIX F

THE ML ESTIMATOR - TIME VARYING TIME DELAY

This appendix briefly summarizes the derivation given by Knapp and Carter in [46] for the ML estimator for the time-varying TDE problem of (5-1). The procedure is very similar to that used to derive the ML estimator for the case of a fixed time delay as discussed in Appendix A, and similar notation is used here. The ML estimate for the parameters β_1, β_2 and D is obtained by maximizing the conditional pdf $p(\underline{R} \mid b_1, b_2, \tau)$, where b_1, b_2, and τ represent hypothesized values for β_1, β_2, and D, respectively. The conditional pdf can again be expressed as

$$p(\underline{R} \mid b_1, b_2, \tau) = c \exp(-\tfrac{1}{2} J) \tag{F-2a}$$

where

$$J = \sum_{k=-N}^{N} \underline{R}^{*\prime}(k) \; \underline{\Sigma}_{k|b,\tau}^{-1} \; \underline{R}(k) \tag{F-2b}$$

and

$$c = \prod_{k=-N}^{N} \left(2\pi \left| \underline{\Sigma}_{k|b,\tau} \right|^{1/2}\right)^{-1} . \tag{F-2c}$$

However, for the signal of (5-1)

$$\underline{\Sigma}_{k|b,\tau} \triangleq E\left[\underline{R}(k)\, \underline{R}^{*\prime}(k) \,\middle|\, b_1, b_2, \tau\right]$$

$$= \frac{1}{T}\begin{bmatrix} \frac{1}{b_1} G_{ss}\left(\frac{kw_o}{b_1}\right) + G_{n_1n_1}\left(\frac{kw_o}{b_1}\right) & \frac{\alpha}{\max(b_1, b_2)} G_{ss}\left(\frac{kw_o}{b_1}\right) e^{-jkw_o b\tau} \\ \frac{\alpha}{\max(b_1, b_2)} G_{ss}\left(\frac{kw_o}{b_1}\right) e^{jkw_o b\tau} & \frac{\alpha^2}{b_2} G_{ss}\left(\frac{kw_o}{b_1}\right) + G_{n_2n_2}(bkw_o) \end{bmatrix}$$

$$\overset{\Delta}{=} \frac{1}{T} \underline{Q} (kw_o) \ , \qquad \text{(F-3)}$$

where $b = b_2 / b_1$.

In practice, the power spectral matrix $\underline{Q}$ must be estimated from the data. Since $b_1, b_2 \simeq 1$, $\underline{Q}$ may be replaced by

$$\underline{Q}(f) \simeq \hat{\underline{Q}}(f) = \begin{bmatrix} \hat{G}_{r_1r_1}(f) & \hat{G}_{r_1r_3}(f) \\ \hat{G}^*_{r_1r_3}(f) & \hat{G}_{r_3r_3}(f) \end{bmatrix} \qquad \text{(F-4a)}$$

where

$$r_3(t) \overset{\Delta}{=} r_2(t/b)/\ b \qquad \text{(F-4b)}$$

and hence

$$\hat{G}_{r_1r_3}(f) \simeq \frac{\alpha}{b_1 b_2} \ G_{ss}\left(\frac{f}{b_1}\right) e^{-j2\pi f b\tau} \quad . \qquad \text{(F-4c)}$$

That is, $r_3(t)$ represents a time scaled version of $r_2(t)$ where the time scaling compensates for the relative time compression between $r_1(t)$ and $r_2(t)$. Then, the quantity c in (F-2) is independent of τ and is approximately independent of b_1 and b_2 (since b_1, $b_2 \simeq 1$), so that maximizing $p(\underline{R} \mid b_1, b_2, \tau)$ is equivalent to minimizing J in (F-2b).

For large T,

$$J = \frac{1}{T} \int_{-\infty}^{\infty} \underline{R}^{*\prime}(f)\ \underline{Q}(f)^{-1}\ \underline{R}(f)\ \ df. \tag{F-3}$$

By substituting for $\underline{Q}(f)^{-1}$ and manipulating (F-3), it can be shown that minimizing J is equivalent to maximizing

$$J'(b, \tau) = \int_{-\infty}^{\infty} R_1(f)\ R_2^{\star}(\beta f)\ W(f)\ e^{j2\pi f b \tau}\ df \quad , \tag{F-4a}$$

or equivalently,

$$J''(b, \tau) = \int_{-\infty}^{\infty} R_1(f)\ R_3^{\star}(f)\ W(f)\ e^{j2\pi f b \tau}\ df \quad , \tag{F-4b}$$

where

$$\hat{W}(f) = \frac{\hat{C}_{r_1 r_3}(f)}{\left|\hat{G}_{r_1 r_3}(f)\right| \quad (1 - \hat{\underline{C}}_{r_1 r_3}(f))} \quad . \tag{F-4c}$$

As noted in Chapter 5, the ML estimates of $\beta = \beta_2/\beta_1$ and D correspond to the values of b and τ for which J' (or J") is a maximum.

APPENDIX G

SIMULATION DETAILS - TIME-VARYING TIME DELAY

A computer simulation routine implementing the compensation technique described in section 5.3 has been written and preliminary simulation results have been obtained as discussed in section 5.4. The signal and noise sequences were generated in the same manner as described in Appendix B. However, in this case, a time shifted version of the signal sequence was generated using the technique proposed by Youn [14, 56]. Program listings for the simulation main routine, DOPPLER, and the subroutine SHIFT, which generates the time shifted signal are included in this appendix. The portions of the main routine which specifically relate to the compensation technique have been boxed. Otherwise the processing is essentially identical to that for the case of a fixed time delay. The functions and subroutines called by the main routine are given in the header of the program listing, however, the subroutine listings (other than SHIFT) have not been included. Variables are also documented in the listings. However, it is worth noting that a maximum value on the delay value (MAXDEL) is set in the program. When the time delay exceeds this value in absolute value, the sign of time delay rate is changed (D_RATE=-D_RATE).

```
C***********************************************************************
C
C       DOPPLER COMPENSATION SIMULATION ROUTINE
C
C       VAX-11 FORTRAN SOURCE FILENAME:         DOPPLER.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING    KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                    PROGRAMMER(S)
C       --------        ----                    -------------
C       01.0            JULY 1983               KENT SCARBROUGH
C
C***********************************************************************
C
C       PURPOSE
C
C               This routine implements a simulation to test
C               a frequency domain technique to compensate for
C               the "doppler" shift which occurs due to relative
C               motion between source and sensors in the time
C               delay estimation problem.
C
C       ROUTINE(S) CALLED BY THIS ROUTINE
C
C               BUTTER, FFTCX, FORD2, GCC, GWNSEQ, HAMMING, HEADER,
C               PEAK, SGOPEN, SGTRAN, REPLY, RSEED, RVINIT, SHIFT,
C               SSEED, WAML, WMSC, WOPT, WPHT, WSCC, WSCT
C
C       OTHER ROUTINE(S) ACCESSED BY THIS ROUTINE
C
C               FFS, FFT842, LPDES, HPDES, BPDES, BSDES, COEFF, GAUSS
C
C***********************************************************************
C***********************************************************************
C
C
        REAL    R1(32*1024), R2(32*1024), SG1(32*1024+61), SG2(32*1024)
        REAL    R1W(75), R2W(75), SG1W(75), SG2W(75), PAST(75)
        REAL    WORKX(2050), WORKY(2050)
        REAL    RXX(2048), RYY(2048), WINDOW(1024), WEIGHT(1025)
        REAL    GXX(1025), GYY(1025), GXYR(1025), GXYI(1025)
        REAL    GXYRSV(1025,21), GXYISV(1025,21)
        REAL    TDELAY(500), TDRATE(500)
        REAL    SCCTAU( 500), SCTTAU( 500), AMLTAU( 500)
        REAL    MSCTAU( 500), OPTTAU( 500), PHTTAU( 500)
        REAL    SCCTDT( 500), SCTTDT( 500), AMLTDT( 500)
        REAL    MSCTDT( 500), OPTTDT( 500), PHTTDT( 500)
        REAL    SCCPK, SCTPK, AMLPK, MSCPK, OPTPK, PHTPK
        REAL    DELDOT, DEL_F, DEL_T, D_RATE, DMIN, FNORM, FS
        REAL    NSFRQL, NSFRQU, SGFRQL, SGFRQU, PKVAL, SCALE
        REAL    SNR1, SNR2, SSTP, TIME, TDEL1
        INTEGER  ADELAY, MAXDEL, FSECT, NSECT, IPNT, LPNT
        INTEGER  PKLOC, NCMP, NTRL, NPTS, NPTP1, NPTM2, NPTD2
        INTEGER  NFFT, NSEG, NSIN, TSEG, TPTS, WIDTH, UNITP
        INTEGER  NSEED, SEED(38)
        LOGICAL  SCC, SCOT, AML, MSC, OPT, PHAT, RDSEED, SVSEED
        LOGICAL  FILTER, RESET, NOSET
```

```
      CHARACTER    FTYPE*2, SNRDB*3, SNRTMP*2, FCTN*40, TNAM*40
C
C
C***********************************************************************
C
C  INITIALIZATION
C
      DATA  TDEL1  /       0/, MAXDEL /      15/, DELDOT / 2.5E-4/
      DATA  D_RATE / 2.5E-4/, FS     /  2048./, FTYPE  /    'LP'/
      DATA  FSECT  /       2/, NSFRQL /     0.0/, NSFRQU /   100./
      DATA  NCMP   /       2/, NTRL   /      51/, NPTS   /     512/
      DATA  SGFRQL /     0.0/, SGFRQU /    100./, SNR1   /    -3.0/
      DATA  SNR2   /    -4.0/, SSTP   /    -3.0/, TIME   /     4.0/
      DATA  NWT    /      51/, WIDTH  /     256/
C
      DATA  SCC    / .TRUE./, SCOT   / .TRUE./, AML    /.FALSE./
      DATA  MSC    /.FALSE./, OPT    /.FALSE./, PHAT   /.FALSE./
C
      DATA  FILTER / .TRUE./, RESET  / .TRUE./, NOSET  /.FALSE./
      DATA  RDSEED /.FALSE./, SVSEED /.FALSE./, NSEED  /      38/
C
C***************************************************
C
C       FORMAT STATEMENTS
C
    1 FORMAT(A)
    3 FORMAT(A,I10,A)
    4 FORMAT(A,F10.2,A)
    5 FORMAT(A,8X,A)
    7 FORMAT(I2.2)
C
C*************************************************************
C
C       CALCULATE CONSTANTS - INITIALIZE PROGRAM VARIABLES
C
      FNYQ = FS / 2.
      NPTP1 = NPTS + 1
      NPTM2 = NPTS * 2
      NPTD2 = NPTS / 2
      NFFT = NPTM2
      TPTS = IFIX(TIME * FS)
      NSEG = TPTS / NPTS
      TSEG = 2 * NSEG - 1
C
      DEL_F = FNYQ / FLOAT(NPTS)
      DEL_T = TIME / FLOAT(TSEG)
      TWOPI = 2.0 * 3.141593
      NCMP21 = NCMP * 2 + 1
      NWT2 = (NWT - 1) / 2
C
      IPNT = NPTS - WIDTH/2 + 1
      LPNT = NPTS + WIDTH/2
      DMIN = FLOAT(IPNT - NPTS)
C
      IF  ((FTYPE.EQ.'LP').OR.(FTYPE.EQ.'HP'))  THEN
          NSECT = 2*FSECT
      ELSE IF (FTYPE.EQ.'BP')  THEN
```

```
            FSECT = FSECT / 2
            NSECT = 4*FSECT
         ELSE
            FILTER = .FALSE.
         END IF
         NSAV = 75
C
         IF ((NSFRQU.EQ.SGFRQU).AND.(NSFRQL.EQ.SGFRQL))  RESET = .FALSE.
C
C  COMPUTE HAMMING WINDOW
C
         CALL HAMMING(WINDOW,NPTS,WPOWER)
         FNORM = TSEG * NPTS * WPOWER
C
C  INITALIZE RANDOM NUMBER GENERATOR
C
         IF (RDSEED) THEN
            CALL SGOPEN(98,'READ','NOPROMPT','SEED.GCC','INTEGER',NSEED)
            CALL SGTRAN(98,'READ','INTEGER',SEED,NSEED)
            CALL RSEED(SEED(1),SEED(7))
         END IF
C
         UNITP = 99
C
C************************************************************
C
C  ECHO CHECK OF PARAMETERS
C
         CALL HEADER(UNITP,'*** MOVING SOURCE SIMULATION ***')
C
         WRITE(UNITP,3)  ' Number of Trials      : ', NTRL
         WRITE(UNITP,4)  ' Sampling Frequency    : ', FS   , ' Hz     '
         WRITE(UNITP,4)  ' Observation Time      : ', TIME , ' seconds'
         WRITE(UNITP,3)  ' Segment Length        : ', NPTS , ' points '
         WRITE(UNITP,3)  ' FFT Size              : ', NFFT , ' points '
         WRITE(UNITP,3)  ' Correlation Window    : ', WIDTH, ' points '
C
         WRITE(UNITP,*)
         WRITE(UNITP,5)  ' Filter Type           : ', FTYPE
         WRITE(UNITP,3)  ' Filter Order          : ', NSECT
         WRITE(UNITP,4)  ' Lower Cutoff (noise)  : ', NSFRQL , ' Hz'
         WRITE(UNITP,4)  ' Upper Cutoff (noise)  : ', NSFRQU , ' Hz'
         WRITE(UNITP,4)  ' Lower Cutoff (signal) : ', SGFRQL , ' Hz'
         WRITE(UNITP,4)  ' Upper Cutoff (signal) : ', SGFRQU , ' Hz'
         WRITE(UNITP,4)  ' First SNR Value(dB)   : ', SNR1   , ' dB'
         WRITE(UNITP,4)  ' Last SNR Value(dB)    : ', SNR2   , ' dB'
         WRITE(UNITP,4)  ' SNR Stepsize(dB)      : ', SSTP   , ' dB'
C
         TEMP1 = D_RATE * 1.0E4
         TEMP2 = DELDOT * 1.0E4
         WRITE(UNITP,*)
         WRITE(UNITP,3)  ' Number of Comp.       : ', NCMP
         WRITE(UNITP,4)  ' Initial Delay Value   : ', TDEL1
         WRITE(UNITP,3)  ' Maximum Delay Value   : ', MAXDEL
         WRITE(UNITP,4)  ' Initial Delay Rate    : ', TEMP1, ' E-4'
         WRITE(UNITP,4)  ' Assumed Rate Stepsize: ', TEMP2, ' E-4'
         WRITE(UNITP,3)  ' No. of Filter Coeffs  : ', NWT
```

```
C
        WRITE(UNITP,*)
        IF (RDSEED) WRITE(UNITP,*)
     &    'Initial Seed Values Read From File: SEED.GCC'
        IF (SVSEED) WRITE(UNITP,*)
     &    'Final Seed Values Output To File: SEED.GCC'
C
        WRITE(UNITP,*)
C
C***********************************************************************
C***********************************************************************
C
C  MAIN LOOP :  EXECUTED ONCE FOR EACH SNR VALUE
C
        DO SNR = SNR1, SNR2, SSTP
C
            ASNR = 10.**(SNR/10.)
            SCALE = 1. / SQRT(ASNR)
C
            ISNR = IFIX(SNR)
            ITMP = ABS(ISNR)
            WRITE(SNRTMP,7) ITMP
            IF (ISNR.LT.0) THEN
                SNRDB = 'N' // SNRTMP
            ELSE
                SNRDB = '0' // SNRTMP
            END IF
C
C*************************************************************
C*************************************************************
C
C  SECONDARY LOOP :  EXECUTED ONCE FOR EACH TRIAL
C
            DO ITRIAL = 1, NTRL
C
C
C***************************************************
C***************************************************
C
C       GENERATE RECEIVED SIGNAL SEQUENCES
C
C           GENERATE WHITE,GAUSSIAN NOISE SEQUENCES
C
                CALL GWNSEQ(R1,TPTS,0.0,1.0)
                CALL GWNSEQ(R2,TPTS,0.0,1.0)
                IF (ITRIAL.EQ.1)
     &                  CALL GWNSEQ(PAST,NWT2,0.0,1.0)
                CALL GWNSEQ(SG1,TPTS,0.0,1.0)
C
C***************************************************
C
C           FILTER SEQUENCES TO OBTAIN DESIRED SPECTRA
C
C
                IF (FILTER)  THEN
                    CALL BUTTER(R1,TPTS,R1W,NSFRQL,NSFRQU,FS,
     &                                                  FTYPE,FSECT,RESET)
```

```
          CALL BUTTER(R2,TPTS,R2W,NSFRQL,NSFRQU,FS,
   &                                FTYPE,FSECT,NOSET)
          IF (ITRIAL.EQ.1) THEN
            CALL BUTTER(PAST,NWT2,SG1W,SGFRQL,SGFRQU,FS,
   &                                FTYPE,FSECT,RESET)
            CALL BUTTER(SG1,TPTS,SG1W,SGFRQL,SGFRQU,FS,
   &                                FTYPE,FSECT,NOSET)
          ELSE
            CALL BUTTER(SG1,TPTS,SG1W,SGFRQL,SGFRQU,FS,
   &                                FTYPE,FSECT,RESET)
          END IF
        END IF
C***********************************************
C
C     SHIFT SIGNAL SEQUENCE, APPEND PAST VALUES
C     AND SAVE 'NEW' PAST VLUES
C
        DO IPT = TPTS, 1, -1
          JPT = IPT + NWT2
          SG1(JPT) = SG1(IPT)
        END DO
C
        DO IPT = 1, NWT2
          JPT = TPTS + IPT
          SG1(IPT) = PAST(IPT)
          PAST(IPT) = SG1(JPT)
        END DO
C
C***********************************************
C
C     GENERATE TIME SHIFTED VERSION OF SIGNAL SEQUENCE
C
        TDELAY(ITRIAL) = TDEL1
        TDRATE(ITRIAL) = D_RATE
C
        CALL SHIFT(SG1,SG2,TPTS,NWT,D_RATE,TDEL1)
C
        IF (ABS(TDEL1) .GT. FLOAT(MAXDEL))  D_RATE = -D_RATE
C
C***********************************************
C
C     SCALE NOISE SEQ TO OBTAIN DESIRED SNR,
C     ADD SIGNAL TO NOISE SEQUENCES
C
C
          DO IPT = 1, TPTS
            R1(IPT) = SCALE * R1(IPT) + SG1(IPT)
            R2(IPT) = SCALE * R2(IPT) + SG2(IPT)
          END DO
C
C***********************************************
C***********************************************
C
C   IMPLEMENT GCC PROCESSOR
C
C     ZERO SUMMING ARRAYS & WORK ARRAYS
C
```

```
                CALL RVINIT(GXX,NPTP1,0.0)
                CALL RVINIT(GYY,NPTP1,0.0)
                CALL RVINIT(GXYR,NPTP1,0.0)
                CALL RVINIT(GXYI,NPTP1,0.0)
                CALL RVINIT(GXYRSV,1025 * 21,0.0)
                CALL RVINIT(GXYISV,1025 * 21,0.0)
                CALL RVINIT(WORKX,NPTM2+2,0.0)
                CALL RVINIT(WORKY,NPTM2+2,0.0)
C
                SCCPK = 0.0
                SCTPK = 0.0
                AMLPK = 0.0
                MSCPK = 0.0
                OPTPK = 0.0
                PHTPK = 0.0
C****************************************************
C
C           LOOP EXECUTED ONCE FOR EACH DATA SEGMENT
C
                DO ISEG = 1, TSEG
C
C****************************************************
C
C           WEIGHT DATA ARRAYS WITH HAMMING WINDOW
C           AND PAD WITH ZEROES (DOUBLE LENGTH)
C
                    DO IPT = 1, NPTS
                        IND1 = (ISEG - 1) * NPTD2 + IPT
                        RXX(IPT) = R1(IND1) * WINDOW(IPT)
                        RYY(IPT) = R2(IND1) * WINDOW(IPT)
                        IND2 = NPTS + IPT
                        RXX(IND2) = 0.0
                        RYY(IND2) = 0.0
                    END DO
C
C****************************************************
C
C           COMPUTE FORWARD FFT AND AVERAGE THE
C           PERIODOGRAM AUTO-SPECTRAL ESTIMATES
C
                    CALL FFTCX(RXX,RYY,NFFT,'FORWARD','NONORM')
                    CALL FORD2(RXX,RYY,NFFT,WORKX,WORKY)
C
                    GXX(1) = GXX(1) + RXX(1) * RXX(1)
                    GYY(1) = GYY(1) + RYY(1) * RYY(1)
                    GXYR(1) = GXYR(1) + RXX(1) * RYY(1)
                    DO IPT = 2, NPTS
                        JPT = 2 * (IPT-1)
                        JP1 = JPT + 1
                        GXX(IPT) = GXX(IPT) + RXX(JPT)**2 + RXX(JP1)**2
                        GYY(IPT) = GYY(IPT) + RYY(JPT)**2 + RYY(JP1)**2
                        GXYR(IPT) = RXX(JPT) * RYY(JPT)
     &                                             + RXX(JP1) * RYY(JP1)
                        GXYI(IPT) = RXX(JPT) * RYY(JP1)
     &                                             - RXX(JP1) * RYY(JPT)
                    END DO
                    GXX(NPTP1) = GXX(NPTP1) + RXX(NPTM2) * RXX(NPTM2)
```

```
                   GYY(NPTP1) = GYY(NPTP1) + RYY(NPTM2) * RYY(NPTM2)
                   GXYR(NPTP1) = GXYR(NPTP1) + RXX(NPTM2) * RYY(NPTM2)
C
C*************************************************
C
C          COMPENSATE FOR TIME SHIFT BETWEEN DATA SEGMENTS
C          & AVERAGE COMPENSATED CROSS-SPECTRAL ESTIMATES
C
                   DO JCMP = 1, NCMP21
                       ICMP = JCMP - (NCMP + 1)
                       TAUDOT = FLOAT(ICMP) * DELDOT
                       RSEG =  FLOAT(ISEG) - .5
                       ARGTMP = TWOPI* RSEG * DEL_T * TAUDOT
                       GXYRSV(1,JCMP) = GXYRSV(1,JCMP) + GXYR(1)
                       GXYISV(1,JCMP) = 0.0
                       DO IPT = 2, NPTS
                           FREQ = FLOAT(IPT - 1) * DEL_F
                           ARG = FREQ * ARGTMP
                           JPT = 2 * (IPT-1)
                           JP1 = JPT + 1
                           GXYRSV(IPT,JCMP) = GXYRSV(IPT,JCMP) +
     &                      GXYR(IPT) * COS(ARG) - GXYI(IPT) * SIN(ARG)
                           GXYISV(IPT,JCMP) = GXYISV(IPT,JCMP) +
     &                      GXYR(IPT) * SIN(ARG) + GXYI(IPT) * COS(ARG)
                       END DO
                       ARG = FNYQ * ARGTMP
                       GXYRSV(NPTP1,JCMP) = GXYRSV(NPTP1,JCMP) +
     &                                           GXYR(NPTP1) * COS(ARG)
                       GXYISV(NPTP1,JCMP) = 0.0
                   END DO
C
C*************************************************
C
C          GO BACK FOR NEXT DATA SEGMENT
C
               END DO
C
C*************************************************
C*************************************************
C
C          LOOP EXECUTED ONCE FOR EACH COMPENSATION VALUE
C
C
               DO JCMP = 1, NCMP21
                   ICMP = JCMP - (NCMP + 1)
                   TAUDOT = FLOAT(ICMP) * DELDOT
C
                   DO IPT = 1, NPTP1
                       GXYR(IPT) = GXYRSV(IPT,JCMP)
                       GXYI(IPT) = GXYISV(IPT,JCMP)
                   END DO
C
C*************************************************
C
C       NORMALIZE SPECTRAL ESTIMATES TO ACCOUNT FOR:
C           1) POWER REDUCTION DUE TO WINDOWING
C           2) SUMMING TSEG ESTIMATES
```

```
C           3) NUMBER OF POINTS/SEGMENT
C       NOTE:  NO NORMALIZATION IS DONE IN FFT'S
C
                   IF (JCMP .EQ. 1) THEN
                       DO IPT = 1, NPTP1
                           GXX(IPT) = GXX(IPT) / FNORM
                           GYY(IPT) = GYY(IPT) / FNORM
                       END DO
                   END IF
C
                   DO IPT = 1, NPTP1
                       GXYR(IPT) = GXYR(IPT) / FNORM
                       GXYI(IPT) = GXYI(IPT) / FNORM
                   END DO
C
C**************************************************
C
C           COMPUTE GCC FUNCTIONS FOR EACH COMPENSATION VALUE
C           DETERMINE GLOBAL PEAK IN TAU-TAUDOT COORDINATES
C
C
                   IF (SCC) THEN
                       CALL RVINIT(WEIGHT,NPTP1,1.0)
                       CALL GCC(WORKX,WORKY,NFFT,WEIGHT,GXYR,GXYI)
                       CALL PEAK(WORKX,IPNT,LPNT,PKVAL,PKLOC)
                       IF (SCCPK .LT. PKVAL)  THEN
                           SCCPK = PKVAL
                           IND = PKLOC - IPNT + 1
                           SCCTAU(ITRIAL) = FLOAT(IND - WIDTH/2)
                           SCCTDT(ITRIAL) = TAUDOT
                       END IF
                   END IF
C
                   IF (SCOT) THEN
                       CALL WSCT(WEIGHT,NPTP1,GXX,GYY)
                       CALL GCC(WORKX,WORKY,NFFT,WEIGHT,GXYR,GXYI)
                       CALL PEAK(WORKX,IPNT,LPNT,PKVAL,PKLOC)
                       IF (SCTPK .LT. PKVAL)  THEN
                           SCTPK = PKVAL
                           IND = PKLOC - IPNT + 1
                           SCTTAU(ITRIAL) = FLOAT(IND - WIDTH/2)
                           SCTTDT(ITRIAL) = TAUDOT
                       END IF
                   END IF
C
                   IF (AML) THEN
                       CALL WAML(WEIGHT,NPTP1,GXYR,GXYI,GXX,GYY)
                       CALL GCC(WORKX,WORKY,NFFT,WEIGHT,GXYR,GXYI)
                       CALL PEAK(WORKX,IPNT,LPNT,PKVAL,PKLOC)
                       IF (AMLPK .LT. PKVAL)  THEN
                           AMLPK = PKVAL
                           IND = PKLOC - IPNT + 1
                           AMLTAU(ITRIAL) = FLOAT(IND - WIDTH/2)
                           AMLTDT(ITRIAL) = TAUDOT
                       END IF
                   END IF
C
```

```
                    IF (MSC) THEN
                        CALL WMSC(WEIGHT,NPTP1,GXYR,GXYI,GXX,GYY)
                        CALL GCC(WORKX,WORKY,NFFT,WEIGHT,GXYR,GXYI)
                        CALL PEAK(WORKX,IPNT,LPNT,PKVAL,PKLOC)
                        IF (MSCPK .LT. PKVAL)  THEN
                            MSCPK = PKVAL
                            IND = PKLOC - IPNT + 1
                            MSCTAU(ITRIAL) = FLOAT(IND - WIDTH/2)
                            MSCTDT(ITRIAL) = TAUDOT
                        END IF
                    END IF
C
                    IF (OPT) THEN
                        CALL WOPT(WEIGHT,NPTP1,FRQL,FRQU,FS,FTYPE,NSECT)
                        CALL GCC(WORKX,WORKY,NFFT,WEIGHT,GXYR,GXYI)
                        CALL PEAK(WORKX,IPNT,LPNT,PKVAL,PKLOC)
                        IF (OPTPK .LT. PKVAL)  THEN
                            OPTPK = PKVAL
                            IND = PKLOC - IPNT + 1
                            OPTTAU(ITRIAL) = FLOAT(IND - WIDTH/2)
                            OPTTDT(ITRIAL) = TAUDOT
                        END IF
                    END IF
C
                    IF (PHAT) THEN
                        CALL WPHT(WEIGHT,NPTP1,GXYR,GXYI)
                        CALL GCC(WORKX,WORKY,NFFT,WEIGHT,GXYR,GXYI)
                        CALL PEAK(WORKX,IPNT,LPNT,PKVAL,PKLOC)
                        IF (PHTPK .LT. PKVAL)  THEN
                            PHTPK = PKVAL
                            IND = PKLOC - IPNT + 1
                            PHTTAU(ITRIAL) = FLOAT(IND - WIDTH/2)
                            PHTTDT(ITRIAL) = TAUDOT
                        END IF
                    END IF
C
                    IF (OUTPUT)  CLOSE(1)
C
C***************************************************
C
C       GO BACK FOR NEXT DOPPLER COMPENSATION
C
                END DO
C
C***************************************************
C***************************************************
C
C       GO BACK FOR NEXT TRIAL
C
C
            END DO
C
C*************************************************************
C*************************************************************
C
C       OUTPUT ESTIMATES OF TAU & TAUDOT
C
```

```
C
            IF (SCC) THEN
                TNAM = 'SCCTAU.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',SCCTAU,NTRL)
                TNAM = 'SCCTDT.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',SCCTDT,NTRL)
            END IF
C
            IF (SCOT) THEN
                TNAM = 'SCTTAU.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',SCTTAU,NTRL)
                TNAM = 'SCTTDT.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',SCTTDT,NTRL)
            END IF
C
            IF (AML) THEN
                TNAM = 'AMLTAU.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',AMLTAU,NTRL)
                TNAM = 'AMLTDT.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',AMLTDT,NTRL)
            END IF
C
            IF (MSC) THEN
                TNAM = 'MSCTAU.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',MSCTAU,NTRL)
                TNAM = 'MSCTDT.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',MSCTDT,NTRL)
            END IF
C
            IF (OPT) THEN
                TNAM = 'OPTTAU.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',OPTTAU,NTRL)
                TNAM = 'OPTTDT.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',OPTTDT,NTRL)
            END IF
C
            IF (PHAT) THEN
                TNAM = 'PHTTAU.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',PHTTAU,NTRL)
                TNAM = 'PHTTDT.' // SNRDB
                CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
                CALL SGTRAN(1,'WRITE','REAL',PHTTDT,NTRL)
            END IF
C
            TNAM = 'TDELAY.' // SNRDB
            CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
            CALL SGTRAN(1,'WRITE','REAL',TDELAY,NTRL)
            TNAM = 'TDRATE.' // SNRDB
            CALL SGOPEN(1,'WRITE','NOPROMPT',TNAM,'REAL',NTRL)
            CALL SGTRAN(1,'WRITE','REAL',TDRATE,NTRL)
            CLOSE(1)
C
C*************************************************************
C
C  GO BACK FOR NEXT SNR
C
        END DO
C
C***********************************************************************
C***********************************************************************
C
C  SAVE SEED VALUES ON DISK IF DESIRED
C
        IF (SVSEED)  THEN
            CALL SSEED(SEED(1),SEED(7))
            CALL SGOPEN(98,'WRITE','NOPROMPT','SEED.GCC','INTEGER',38)
            CALL SGTRAN(98,'WRITE','INTEGER',SEED,38)
        END IF
C
        STOP
        END
```

```
C***********************************************************************
C
C       TIME SHIFTED SEQUENCE GENERATOR
C
C       VAX-11 FORTRAN SOURCE FILENAME:         SHIFT.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING    KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                    PROGRAMMER(S)
C       --------        ----                    -------------
C       00.0            JULY, 1983              K SCARBROUGH
C
C***********************************************************************
C
C       CALLING SEQUENCE
C
C               CALL SHIFT(X1,X2,NPTS,NWT,DDOT,DELAY)
C
C       PURPOSE
C
C               This routine generates a time shifted version of a
C               sequence.  The time shift is linear with rate DDOT
C               for the length of the sequence.  The delay rate
C               can be changed on subsequent calls to the routine,
C               however, the time delay is computed by this routine
C               and is continuous between calls.
C
C       EMBEDDED ROUTINE(S) CALLED BY THIS ROUTINE
C
C               COEFF
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               X1     -  Input sequence to be time-shifted
C
C               NPTS   -  Number of points in input sequence
C
C               NWT    -  Number of coefficients for 'Shift-Filter'
C                         (NWT is assumed to be odd, and .LE. 61)
C
C               DDOT   -  Delay rate
C
C               DELAY  -  Initial time delay
C                         (used only on first call to subroutine)
C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               X2     -  Time-shifted version of X1
C
C               DELAY  -  Value of time delay after NPTS
C
C***********************************************************************
C
        SUBROUTINE SHIFT(X1,X2,NPTS,NWT,DDOT,DELAY)
C
        REAL  X1(*), X2(*), XTMP(61), WT(61), PAST(61)
        LOGICAL  FIRST /.TRUE./
```

```
      SAVE FIRST, PAST, TDEL
C
C*************************************************************
C
C  INITIALIZE PARAMETERS
C
      NWT2 = (NWT - 1) / 2
      IF (FIRST)  THEN
          FIRST = .FALSE.
          TDEL = DELAY
          CALL RVINIT(PAST,NWT2,0.0)
      END IF
C
      DO IPT = 1, NWT2
          JPT = IPT + NWT2
          XTMP(IPT) = PAST(IPT)
          XTMP(JPT) = X1(IPT)
      END DO
C
C*************************************************************
C
C  GENERATE TIME SHIFTED SEQUENCE
C
      DO IPT = 1, NPTS
C
C***************************************************
C
C        CALCULATE 'TIME-SHIFT' FILTER COEFFICIENTS
C
          CALL COEFF(WT,NWT,TDEL)
          TDEL = TDEL + DDOT
C
C***************************************************
C
C        IMPLEMENT 'TIME-SHIFT' FILTER
C
          KPT = IPT + NWT2
          XTMP(NWT) = X1(KPT)
          X2(IPT) = 0.0
          DO IWT = 1, NWT
              X2(IPT) = X2(IPT) + WT(IWT) * XTMP(IWT)
          END DO
C
C***************************************************
C
C        SHIFT VALUES IN XTMP ARRAY
C
          DO IWT = 1, NWT - 1
              XTMP(IWT) = XTMP(IWT + 1)
          END DO
C
      END DO
C
C*************************************************************
C
C  SAVE LAST 'NWT2' VALUES OF X1 ARRAY
C
```

```
        DO IPT = 1, NWT2
            JPT = IPT + NPTS - NWT2
            PAST(IPT) = X1(JPT) .
        END DO
C
C*************************************************************
C
C  RETURN TIME-SHIFTED SEQUENCE AND
C  CURRENT DELAY VALUE TO MAIN ROUTINE
C
        DELAY = TDEL
C
        RETURN
        END
```

```
C***********************************************************************
C
C  SUBROUTINE COEFF(WT,NWT,TDEL)
C
C       This routine computes the coefficients for a filter
C       which generates a time shifted version of a given
C       sampled sequence.
C
C       NOTE :  It is assumed that NWT will be odd.
C
C***********************************************************************
C
        SUBROUTINE COEFF(WT,NWT,TDEL)
C
        REAL  WT(*)
C
        PI = 3.141593
        NWT2 = (NWT - 1) / 2
C
        DO IWT = 1, NWT
            JWT = NWT - IWT
            ARG = (FLOAT(JWT - NWT2) - TDEL) * PI
            IF (ARG.NE.0.0) THEN
                WT(IWT) = SIN(ARG) / ARG
            ELSE
                WT(IWT) = 1.0
            END IF
        END DO
C
        RETURN
        END
```

REFERENCES

[1] G. C. Carter, Ed., Special Issue on Time Delay Estimation, IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, no. 3, June 1981.

[2] L. C. Wood and S. Treitel, "Seismic signal processing," Proc. IEEE, vol. 63, pp. 649-661, Apr 1975.

[3] M. A. Rodriquez, R. H. Williams and T. J. Carlow, "Signal delay and waveform estimation using unwrapped phase averaging," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, no. 3, pp. 508-513, June 1981.

[4] Lj. Kostic, "Local steam transit time estimation in a boiling water reactor," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, no. 3, pp. 555-560, June 1981.

[5] J. S. Bendat and A. G. Piersol, Engineering Applications of Correlation and Spectral Analysis. New York: Wiley, 1980.

[6] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 320-327, Aug 1976.

[7] P. R. Roth, "Effective measurements using digital signal analysis," IEEE Spectrum, vol. 8, pp. 62-70, Apr 1971.

[8] C. Eckart, "Optimal rectifier systems for detection of steady signals," Scripps Inst. Oceanography, Marine Physical Lab., Rep. SIO 12692, SIO Ref. 52-11, 1952.

[9] G. C. Carter, A. H. Nuttall, and P. G. Cable, "The smoothed coherence transform," Proc. IEEE, vol. 61, pp. 1497-1498, Oct 1973.

[10] -----, "The smoothed coherence transform (SCOT)," Naval Underwater Systems Center, New London, CT, Tech. Memo. TL-159-72, 8 Aug 1972.

[11] A. Hero and S. Schwartz, "A new generalized cross-correlator," to appear in IEEE Trans. Acoust., Speech, Signal Processing, 1984.

[12] E. J. Hannan and P. J. Thomson, "Estimating group delay," Biometrika, vol. 60, no. 2, pp. 241-253, 1973.

[13] B. Widrow et al, "Adaptive noise cancelling: Principles and applications," Proc. IEEE, vol. 63, no. 12, pp. 1692-1716, Dec 1975.

[14] D. H. Youn, "A class of adaptive methods for estimating coherence and time delay functions," Ph.D. dissertation, Dept. Electrical Eng., Kansas State Univ., Manhattan, KS, 1982.

[15] D. H. Youn and N. Ahmed, "Time delay estimation via coherence; An adaptive approach," to appear in IEEE Trans. Acoust., Speech, Signal Processing, 1984.

[16] Y. T. Chan, J. M. Riley and J. B. Plant, "A parameter estimation approach to time delay estimation and signal detection," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-28, no. 1, pp. 8-15, Feb 1980.

[17] ----, "Modeling of time delay and its application to estimation of nonstationary delays," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, no. 3, pp. 577-581, June 1981.

[18] D. M. Etter and S. D. Stearns, "Adaptive estimation of time delays in sampled data systems," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, no. 3, pp. 582-587, June 1981.

[19] P. L. Feintuch, N. J. Bershad, and F. A. Reed, "Time delay estimation using the LMS adaptive filter - Dynamic behavior," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, no. 3, pp. 571-576, June 1981. (See also F. A. Reed, P. L. Feintuch, and N. J. Bershad, "Time delay estimation using the LMS adaptive filter - Static behavior," same issue, pp. 561-570)

[20] W. R. Hahn, "Optimum signal processing for passive sonar range and bearing estimation," JASA, vol. 58, no. 1, pp. 201-207, July 1975.

[21] A. H. Quazi, "An overview of the time delay estimate in active and passive systems for target localization," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, no. 3, pp. 534-540, June 1981.

[22] R. L. Kirlin and J. N. Bradley, "Delay estimation simulations and a normalized comparison of published results," IEEE Trans. Acoust., Speech, Signal Processing (Corr.), vol. ASSP-30, no. 3, pp. 580-511, June 1982.

[23] H. L. Van Trees, Detection, Estimation, and Modulation Theory - Part I. New York: Wiley, 1968.

[24] K. Scarbrough, N. Ahmed, and G. C. Carter, "On the simulation of a class of time delay estimation algorithms," IEEE Trans. Acoust., Speech, Signal Processing (Corr.), vol. ASSP-29, no. 3, pp. 534-540, June 1981.

[25] J. C. Hassab and R. E. Boucher, "A quantitative study of optimum and suboptimum filters in the generalized correlator," in Proc. 1979 IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 124-127, 1979.

[26] S. K. Chow and P. M. Schultheiss, "Delay estimation using narrow-band processes," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, no. 3, pp. 478-484, June 1981.

[27] J. P. Ianniello, "Time delay estimation via cross-correlation in the presence of large estimation errors," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-30, no. 6, pp. 998-1003, Dec 1982.

[28] A. Weiss and E. Weinstein, "Composite bound on the attainable mean-square error in passive time delay estimation from ambiquity prone signals," IEEE Trans. Info. Theory (Corr.), vol. IT-28, no. 6, pp. 977-979, Nov 1982.

[29] ----, "Fundamental limitations in passive time delay estimation - Part I: Narrow-band systems," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, no. 2, pp. 472-485, Apr 1983.

[30] E. Weinstein, "Performance analysis of time delay estimators," Woods Hole Oceanographic Institute, Woods Hole, Mass., WHOI Technical Report, 1983.

[31] J. P. Ianniello, E. Weinstein, and A. Weiss, "Comparison of the Ziv-Zakai lower bound on time delay estimation with correlator performance," in Proc. 1983 Int. Conf. Acoust., Speech, Signal Processing, pp. 875-878, 1983.

[32] K. Scarbrough, R. Tremblay, and G. C. Carter, "Performance predictions for coherent and incoherent processing techniques of time delay estimation," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, no. 5, pp. 1191-1196, Oct 1983.

[33] W. C. Knight, R. G. Pridham, and S. M. Kay, "Digital signal processing for sonar," Proc. IEEE, vol. 69, no. 11, pp. 1451-1506, Nov 1981.

[34] A. V. Oppenheim, Ed., Applications of Digital Signal Processing. Englewood Cliffs, NJ: Prentice Hall, 1978 (especially Ch. 6 - Sonar Signal Processing by A. B. Baggeroer).

[35] R. J. Urick, Principles of Underwater Sound. New York: McGraw-Hill, 1983.

[36] A. Papoulis, Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill, 1965.

[37] K. Scarbrough, N. Ahmed, and G. C. Carter, "Comparison of two methods for time delay estimation of sinusoids," Proc. IEEE (Corr.), vol. 70, no. 1, pp. 90-92, Jan 1982; see also K. Scarbrough, N. Ahmed, D. H. Youn, and G. C. Carter, "On the SCOT and Roth algorithms for time delay estimation," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 371-374, 1982.

[38] A. M. Mood, F. A. Graybill, and D. C. Goes, Introduction to the Theory of Statistics. New York: McGraw Hill, 1974.

[39] J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering. New York: Wiley, 1965.

[40] J. S. Bendat and A. G. Piersol, Random Data: Analysis and Measurement Procedures. New York: Wiley-interscience, 1971.

[41] J. Ziv and M. Zakai, "Some lower bounds on signal parameter estimation," IEEE Trans. Inform./Theory, vol. IT-15, no. 3, pp. 386-391, May 1969.

[42] D. Chazan, M. Zakai, and J. Ziv, "Improved lower bounds on signal parameter estimation," IEEE Trans. Commun., vol. IT-23, pp. 90-93, Jan. 1975.

[43] S. Bellini and G. Tartara, "Bounds on error in signal parameter estimation," IEEE Trans. Commun., vol. COMM-22, pp. 340-342, March 1974.

[44] S. M. Selby, Ed., CRC Standard Mathematical Tables, Seventeenth Edition. The Chemical Rubber Co., 1969.

[45] A. H. Nutall, "Threshold signal-to-noise ratio for time delay estimation via cross-correlation," Naval Underwater Systems Center, New London, CT, NUSC Tech. Memo. 831065, April 1983.

[46] C. H. Knapp and G. C. Carter, "Estimation of time delay in the presence of source or receiver motion," J. Acoust. Soc. Am., vol. 61, no. 6, pp. 1545-1549, June 1977.

[47] W. B. Adams, J. P. Kuhn, and W. P. Whyland, "Correlator compensation requirements for passive time-delay estimation with moving source or receivers," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-28, no. 2, pp. 158-168, April 1980.

[48] G. C. Carter and A. H. Nuttall, "On the weighted overlapped segment-averaging method for power spectral estimation," Proc. IEEE, vol. 68, no. 10, pp. 1352-1354, Oct. 1980.

[49] J. P. Kuhn, S. E. Robinson, and D. W. Winfield, "Time delay estimation techniques for moving sources," in Proc. 24th Midwest Symp. Circuits and Syst., pp. 69-75, June 1981.

[50] J. W. Betz, "Comparison of the deskewed short-time correlator and the maximum likelihood correlator," to appear in IEEE Trans. Acoust., Speech, Signal Processing, April 1984.

[51] L. Ng, "Optimum multi-sensor, multi-target localization and tracking," Ph.D. dissertation, Univ. of Connecticut, Storrs, CT, 1983. (Also, NUSC Tech. Rep. 6931, Naval Underwater Systems Center, New London, CT, June 1983.)

[52] G. M. Jenkins and D. G. Watts, Spectral Analysis and its Applications. San Francisco, CA: Holden Day, 1968.

[53] Digital Signal Processing Committee, Ed., Programs for Digital Signal Processing. New York: IEEE Press, 1979.

[54] M. Abramowitz and I. A. Stegun, Ed., Handbook of Mathematical Functions, Tenth Printing. U.S. Government Printing Office, Washington, D.C., 1972.

[55] A. H. Stroud and D. Secrest, Gaussian Quadrature Formulas, Englewood Cliffs, NJ: Prentice Hall, 1966.

[56] D. H. Youn, N. Ahmed, and G. C. Carter, "A method for generating a class of time-delayed signals," in Proc. 1981 IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 1257-1260, 1981.

INITIAL DISTRI8UTION LIST

Addressee	No. of Copies
NAVSEASYSCOM, SEA-63D (R. Cockerill, W. Williams)	2
NAVPGSCOL	1
DTIC	12
NOSC, Code 6565 (Library)	1
Dr. N. Ahmed, Dept. EECE, Univ. of New Mexico, Albuquerque, New Mexico 87131	2
Prof. Y. T. Chan, Dept. of Electrical Engineering, Royal Military College, Kingston, Ontario, Canada K7L ZW3	1
Dr. J. A. Presley, Jr., ORINCON, 3366 N. Torrey Pines Ct., La Jolla, CA 92037	1
R. D. Trueblood, ORINCON, 3366 N. Torrey Pines Ct., La Jolla, CA 92037	1
J. 8etz, RCA Automated Systems, 8ox 588, Burlington, MA 01803	1
Dr. L. Ludeman, Dept. of Electrical Engineering, New Mexico State Univ., Las Cr[illegible] NM 88003	1
Roger Tremblay, 613 Fourth Ave. [illegible]ttle, WA 98119	1
Dr. J. Devore, Dept. Electrical [illegible]ng, Kansas State Univ., Manhattan, KS 66502	1