

AD-A173 396

EFFICIENT RECURSIVE BATCH TIME DELAY DIFFERENCE
ESTIMATION IN THE PRESENCE OF (U) NAVAL UNDERWATER SYSTEMS
CENTER NEW LONDON CT NEW LONDON LAB. R A LATOURETTE

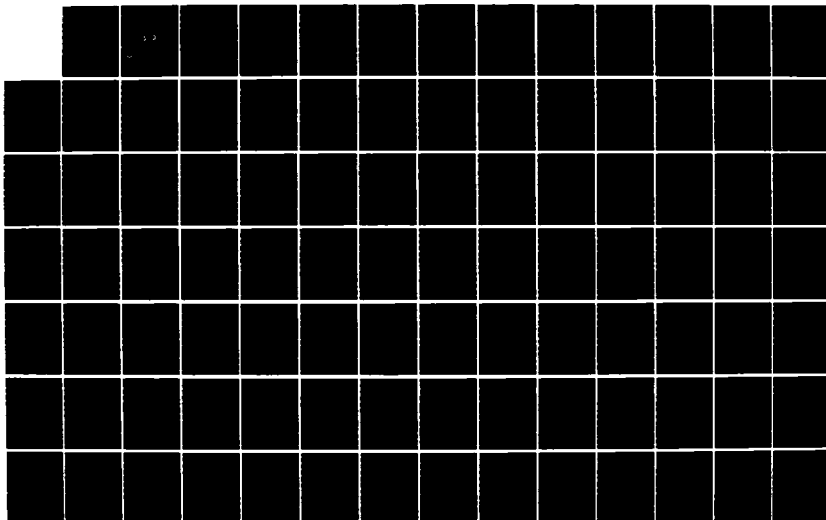
1/4

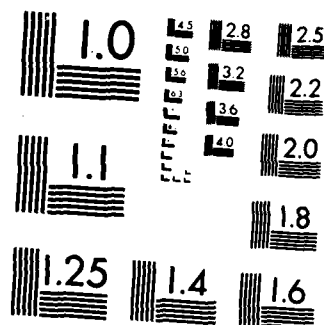
UNCLASSIFIED

17 JUL 86 NUSC-TR-7743

F/G 17/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A173 396

10

**NUSC Technical Report 7743
17 July 1986**

Efficient Recursive Batch Time Delay Difference Estimation in the Presence of Target Motion

**Robert A. LaTourette
Submarine Sonar Department**

**DTIC
ELECTE
OCT 23 1986**
S D

DTIC FILE COPY



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

Approved for public release, distribution unlimited.

Preface

This study was prepared as a dissertation in partial fulfillment of the degree of Doctor of Philosophy in Applied Mathematical Sciences at the University of Rhode Island. It was prepared under NUSC Project No. A46418. The Principal Investigator is Patricia Dean (Code 3291). The Sponsoring Activity is NAVSEA, Program Manager CAPT J. Wilson (NAVSEA PMS 418).

The author appreciates the technical inputs provided by Dr. Lawrence Ng from the Lawrence Livermore National Laboratory; Professor Pan Tai Liu from the University of Rhode Island; Dr. N. Owsley, Messrs. R. Garber, S. Kessler from the Naval Underwater Systems Center; and Mr. H. Jarvis and Dr. C. Wenk from Analysis & Technology, Inc.

The Technical Reviewer for this report was Dr. G. C. Carter, (Code 3331.)

Reviewed and Approved: 17 July 1986

F. J. Kingsbury

F. J. Kingsbury
Head, Submarine Sonar Department

W. A. Von Winkle

W. A. Von Winkle
Associate Technical Director
for Technology

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TR 7743			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Naval Underwater Systems Center		6b. OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State, and ZIP Code). New London Laboratory New London, CT 06320			7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION NAVSEA		8b. OFFICE SYMBOL (if applicable) PMS 418	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code) Washington, D.C. 20362			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
					WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) EFFICIENT RECURSIVE BATCH TIME DELAY DIFFERENCE ESTIMATION IN THE PRESENCE OF TARGET MOTION					
12. PERSONAL AUTHOR(S) Robert A. LaTourette					
13a. TYPE OF REPORT		13b. TIME COVERED FROM TO		14. DATE OF REPORT (Year, Month, Day) 1986 July 17	
				15. PAGE COUNT 310	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Time Delay Estimation, Target Motion, MAP, Recursive Batch, LMSF		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
The optimum batch process for estimating time delay difference parameters of a single moving target from a time delay history of two sensor measurements is investigated. The derived process simultaneously entails (1) maximizing the product of the measured cross correlation function with the expected cross correlation function (MISMATCH function) over the block of observations and (2) minimizing the squared error of the estimated time delay difference trajectory with respect to the expected model of the time delay difference trajectory. Two major numerical algorithms were derived to implement the derived process. The first was based on discrete dynamic programming techniques. The second on continuous gradient search techniques. Both algorithms were developed to be efficient recursive batch processors with good initialization characteristics. In addition, a target power spectrum estimate was derived.					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Robert A. LaTourette			22b. TELEPHONE (Include Area Code) (203) 440-5337		22c. OFFICE SYMBOL Code 3213

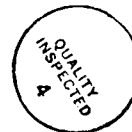
19. ABSTRACT (Cont'd.)

Simulation and theoretical studies were conducted to determine the optimum parameter selection for the derived algorithms. The optimum ranges for algorithm parameters were determined subject to the target dynamic state and the signal to noise ratio (SNR).

Monte Carlo simulations were conducted to evaluate relative algorithm performance with respect to a standard reference and various target dynamic and spectral conditions. The standard reference was selected as the Cramer-Rao Lower Bound (CRLB) for coherent integration of a stationary target without a priori information. The gradient search algorithm was found to have superior steady state performance characteristics. The gradient search algorithm performance was found to exceed that of the CRLB reference due to the successful inclusion of a priori information. Moderate mismatches resulted in minimum degradation in algorithm performance. Both derived algorithms had good initialization properties.

TABLE OF CONTENTS

	Page
LIST OF FIGURES	iii
LIST OF TABLES	ix
LIST OF ACRONYMS	x
TERMS DEFINED BY ENGLISH SYMBOLS	xi
TERMS DEFINED BY GREEK SYMBOLS	xix
CHAPTER 1 -- INTRODUCTION	1-1
1.1 Background	1-1
1.2 Technical Objectives	1-4
1.3 Previous Work	1-6
1.4 Report Organization	1-8
1.5 Statement of Contribution and Executive Summary	1-8
CHAPTER 2 -- PROBLEM FORMULATION	2-1
2.1 Introduction	2-1
2.2 Description of the Observation Scenario	2-1
2.3 Description of Input Sensor Measurements	2-2
2.4 Statement of Problem	2-5
CHAPTER 3 -- THEORETICAL DEVELOPMENT OF MAP ESTIMATORS	3-1
3.1 Introduction	3-1
3.2 The MAP Equation	3-2
3.3 The Maximum Likelihood Estimator for Problem 1	3-3
3.4 The Maximum Likelihood Estimator for Problem 2	3-19
3.5 Least Mean Square Fit Approximation	3-34
3.6 Spectral Parameter Estimation	3-40
3.7 Derivation of the A Priori Knowledge Penalty Function	3-45
3.8 Summary	3-51
CHAPTER 4 -- PRACTICAL IMPLEMENTATION OF THE MAP ESTIMATORS	4-1
4.1 Introduction	4-1
4.2 Dynamic Programming Approach	4-2
4.2.1 Basic Concept	4-3
4.2.2 Recursive Modification	4-9
4.2.3 Initialization	4-18
4.3 Gradient Search Approach	4-24
4.3.1 Basic Concept	4-24
4.3.2 Recursive Update	4-33



A-1

ility Codes
and/or
Special

TABLE OF CONTENTS (Cont'd)

	<u>Page</u>
4.3.3 Initialization	4-35
4.3.4 Sum of Squared Second Differences Penalty Function	4-37
4.4 Related Recursive Batch Algorithms	4-37
4.4.1 Sequential Algorithm	4-38
4.4.2 Spectral Estimation Algorithms	4-38
CHAPTER 5 -- OPTIMUM PARAMETER SELECTION	5-1
5.1 Test Method	5-2
5.2 The Proportionality Constant K	5-5
5.3 The Data Block Size M and Integration Time T	5-12
CHAPTER 6 -- COMPARATIVE ALGORITHM PERFORMANCE	6-1
6.1 Introduction	6-1
6.2 Semi Static Performance of the Gradient Search Algorithm	6-2
6.3 Semi-Static Performance Comparison Between the Gradient Search Algorithms and Other Related MAP Algorithms	6-11
6.4 Gradient Search Algorithm Dynamic Performance	6-18
6.5 Gradient Search Algorithm Performance With Spectral Mismatch	6-24
6.6 Initialization Performance	6-30
CHAPTER 7 -- SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS	7-1
7.1 Future Work	7-10
REFERENCES	R-1
APPENDIX A -- ALTERNATIVE DERIVATION OF THE MISMATCH FUNCTION	A-1
APPENDIX B -- EVALUATION OF MISMATCH FUNCTION INTEGRALS	B-1
APPENDIX C -- RECURSIVE LMSF ERROR ACCUMULATION EXPERIMENT	C-1
APPENDIX D -- THEORETICAL LOWER RMS ERROR LIMIT FOR THE DYNAMIC PROGRAMMING ALGORITHM DESCRETIZED POINTS	D-1
BIBLIOGRAPHY	B8-1

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
2-1	Observation Scenario	2-2
3-1	Linear Model for the Signal Power Spectrum	3-41
4-1	Partial Discretized Data Block	4-4
5-1	Correlation Function vs Normalized Lag Points	5-4
5-2	Target Time Delay Difference Trajectory vs Time	5-6
5-3	RMS Time Delay Difference Error vs Proportionality Constant K, Dynamic Programming SSR (M = 10, T = sec, N = 31)	5-8
5-4	RMS Time Delay Difference Error vs Proportionality Constant K, Gradient Search Algorithm (M = 10, T = 5 sec, N = 31)	5-11
5-5	RMS Time Delay Difference Error vs Block Size, Static Scenario, Dynamic Programming (SSR) (T = 5 sec, K = 0, N = 31, SNR = -15 dB)	5-14
5-6	RMS Time Delay Difference Error vs Block Size, Static Scenario, Dynamic Programming (SSR) (T = 5 sec, K = 0.5, N = 31)	5-16
5-7	RMS Time Delay Difference Error vs Block Size, Dynamic Scenario, Dynamic Programming (SSR) (T = 5 sec, K = 0, N = 31)	5-18
5-8	RMS Time Delay Difference Error vs Block Size, Dynamic Scenario, Dynamic Programming (SSR) (T = 1 sec, K = 0, N = 31)	5-19
5-9	RMS Time Delay Difference Error vs Block Size, Dynamic Scenario, Dynamic Programming (SSR) (T = 5 sec, K = 0.5, N = 31)	5-21
5-10	RMS Time Delay Difference Error vs Proportionality Constant K, Dynamic Program SSR (M = 20, T = 5 sec, N = 31)	5-23
5-11	RMS Time Delay Difference Error vs Proportionality Constant K, Dynamic Program SSR (M = 30, T = 5 sec, N = 31)	5-24

LIST OF FIGURES (Cont'd)

<u>Figure</u>	<u>Page</u>
5-12 RMS Time Delay Difference Error vs Proportionality Constant K, Dynamic Program SSR (SNR = -15 dB, T = 5 sec, N = 31)	5-25
5-13 RMS Time Delay Difference Error vs Block Size, Dynamic Scenario, Dynamic Programming (SSR) (T = 5 sec, Optimal K, N = 31)	5-26
5-14 RMS Time Delay Difference Error vs Block Size, Dynamic Scenario, Dynamic Programming (SSR) (T = 1 sec, K = 0.5, N = 31)	5-28
5-15 RMS Time Delay Difference Error vs Block Size, Dynamic Scenario, Dynamic Programming (SSR) (T = 1 sec, Optimal K, N = 31)	5-29
5-16 RMS Time Delay Difference Error vs Block Size, Gradient Search, Static Geometry (T = 5 sec, K = 0, N = 9)	5-32
5-17 RMS Time Delay Difference Error vs Block Size, Gradient Search, Static Geometry (T = 5 sec, K = 0.5, N = 9)	5-33
5-18 RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Geometry (T = 5 seconds, K = 0, N = 9)	5-35
5-19 RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Scenario (T = 5 sec, K = 0.5, N = 9)	5-37
5-20 RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Scenario (T = 5 sec, Optimal K, N = 9)	5-39
5-21 RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Scenario (T = 1 sec, K = 0, N = 9)	5-41
5-22 RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Scenario (T = 1 sec, K = 0.5, N = 9)	5-43

LIST OF FIGURES (Cont'd)

<u>Figure</u>		<u>Page</u>
5-23	RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Scenario ($T = 1$ sec, Optimal K , $N = 9$)	5-44
5-24	Theoretical Time Delay Difference Standard Deviation vs Number of Lag Points (Independent Noise)	5-55
5-25	Theoretical Time Delay Difference Standard Deviation vs Number of Lag Points (Correlated Noise)	5-56
5-26	Theoretical Time Delay Difference Standard Deviation vs Number of Lag Points (Correlated Noise with Pre- whitening)	5-57
5-27	RMS Time Delay Difference Error vs Number of Lag Points, Gradient Search, Dynamic Scenario ($T = 5$, $K = 0$)	5-60
5-28	RMS Time Delay Difference Error vs Number of Lag Points, Gradient Search with Prewhitening, Dynamic Scenario ($T = 5$, $K = 0$)	5-61
5-29	RMS Time Delay Difference Error vs Number of Lag Points, Dynamic Program, Dynamic Scenario ($T = 5$, $K = 0$)	5-62
5-30	Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = 5 dB, Correlated Noise, $T = 5$ sec)	5-65
5-31	Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = 5 dB, Uncorrelated Noise, $T = 5$ sec)	5-67
5-32	Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = -15 dB, Correlated Noise, $T = 5$ sec)	5-69
5-33	Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = 5 dB, Correlated Noise, $T = 1$ sec)	5-70
5-34	Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = 5 dB, Correlated Noise, $N = 9$)	5-72

LIST OF FIGURES (Cont'd)

<u>Figure</u>		<u>Page</u>
5-35	Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = -15 dB, Correlated Noise, N = 9)	5-73
5-36	RMS Time Delay Difference Error vs Averaging Time (SNR = 5 dB, N = 3, Dynamic Scenario, Gradient Search)	5-75
5-37	RMS Time Delay Difference Error vs Time Delay Difference Rate (SNR = 5 dB, T = 5, Gradient Search)	5-77
5-38	RMS Time Delay Difference Error vs Number of Lag Points (SNR = -15 dB, T = 10 sec, 4 x Nominal Time Delay Difference Rate)	5-78
5-39	Theoretical Time Delay Difference Standard Deviation vs Lag Point Spacing (SNR = 5 dB, Correlated Noise, N = 9, T = 5)	5-81
5-40	Theoretical Time Delay Difference Standard Deviation vs Lag Point Spacing (SNR = 5 dB, Correlated Noise, N = 3, T = 5)	5-83
5-41	Theoretical Time Delay Difference Standard Deviation vs Lag Point Spacing (SNR = 5 dB, Uncorrelated Noise, N = 9, T = 5)	5-84
5-42	RMS Time Delay Difference Error vs Lag Point Spacing, Gradient Search (Correlated Noise, N = 9, T = 5, K = 0)	5-86
5-43	RMS Time Delay Difference Error vs Lag Point Spacing, Dynamic Programming (Correlated Noise, N = 9, T = 5, K = 0)	5-88
6-1	RMS Time Delay Difference Error vs SNR, Gradient Search (K = 1.5, N = 9, M = 10)	6-6
6-2	RMS Time Delay Difference Error vs SNR, Gradient Search (T = 5, N = 9, M = 10)	6-9
6-3	Standard Deviation in SNR Error vs SNR, Gradient Search (K = 1.5, T = 5, N = 9, M = 10)	6-12

LIST OF FIGURES (Cont'd)

<u>Figure</u>	<u>Page</u>
6-4 RMS Time Delay Difference Error vs SNR	6-14
6-5 RMS Time Delay Difference Error vs SNR	6-16
6-6 RMS Time Delay Difference Error vs SNR, Gradient Search (N = 9, M = 10, K = 1.5, T = 5 sec)	6-20
6-7 RMS Time Delay Difference Rate Error vs SNR	6-22
6-8 RMS Time Delay Difference Error vs SNR, Gradient Search (K = 1.5, N = 9, M = 10, T = 5 sec)	6-25
6-9 RMS Spectral Parameter γ vs SNR, Gradient Search (K = 1.5, N = 9, M = 10, T = 5 sec)	6-28
6-10 Time Delay Difference Error vs Time Gradient Search (K = 1.5, SNR = -18)	6-35
6-11 Time Delay Difference Error vs Time Gradient Search (K = 0.5, SNR = -18)	6-36
6-12 Time Delay Difference Error vs Time Gradient Search (K = 0.1, SNR = -18)	6-37
6-13 Time Delay Difference Error vs Time Gradient Search (K = 0.1, through 1.5, SNR = -18)	6-40
C-1 Absolute Error vs Iteration Number (Block Size = 10, Noise Variance = 0.01)	C-6
C-2 Percentage Error vs Iteration Number (Block Size = 10, Noise Variance = 0.01)	C-7
C-3 Absolute Error vs Iteration Number (Block Size = 10, Noise Variance = 0.1)	C-8
C-4 Percentage Error vs Iteration Number (Block Size = 10, Noise Variance = 0.1)	C-9
C-5 Absolute Error vs Iteration Number (Block Size = 10, Noise Variance = 1)	C-10
C-6 Percentage Error vs Iteration Number (Block Size = 10, Noise Variance = 1)	C-11

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
C-7	Absolute Error vs Iteration Number (Block Size = 5, Noise Variance = 0.1)	C-12
C-8	Percentage Error vs Iteration Number (Block Size = 5, Noise Variance = 0.1)	C-13

LIST OF TABLES

<u>Table</u>	<u>Page</u>
6-1 Dynamic Programming (SSR) Nominal Target Dynamics . . .	6-32
6-2 Dynamic Programming (SSR) Double Nominal Target Dynamics	6-32
6-3 Gradient Search Nominal Target Dynamics	6-42
6-4 Gradient Search Double Nominal Target Dynamics	6-42

LIST OF ACRONYMS

ARMA Auto recursive moving average

CRLB Cramer Rao Lower Bound

FIR Finite impulse response (filter)

IIR Infinite impulse response

LMSF Least mean square fit

MAP Maximum a posteriori (estimator)

ML Maximum likelihood (estimator)

SNR Signal-to-noise ratio

SSD Sum of square differences

SSR Sum of squared residuals

TERMS DEFINED BY ENGLISH SYMBOLS

a	Component of parameter vector $\underline{A}$
$\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}, \tilde{e}$	Parameter for the estimate of the slope parameter γ
$\underline{A}$	Parameter vector
A_s	Total signal power
A_n	Normalized received signal power
$\tilde{A}_n$	Modified normalized received signal power
B	Highest continuous frequency
B_i^j	Binomial coefficient
B_w	Target spectrum bandwidth $2\pi(f_2 - f_1)$
$c_{f_1}(\omega_k), c_{f_2}(\omega_k)$	Fourier coefficients at frequency ω_k at sensors 1 and 2
c_i	Components of independent noise or whitened mismatch vector $\underline{C}$

TERMS DEFINED BY ENGLISH SYMBOLS (Continued)

$\tilde{c}_i$	Components of the mismatch vector $\tilde{\underline{C}}$
$\underline{C}$	Vector of expected discrete cross-correlation lag points (independent noise or prewhitened)
$\tilde{\underline{C}}$	Vector of discrete expected cross-correlation lag points (discrete mismatch function)
$\underline{C}_F(\omega_k)$	Vector of Fourier coefficients at frequency ω_k
d_1, d_2	Components of vectors $\underline{D}_1$ and $\underline{D}_2$
$\tilde{d}_i^1, \tilde{d}_i^2$	Components of vectors $\tilde{\underline{D}}^1$ and $\tilde{\underline{D}}^2$
D_p	Dipole spacing
$\underline{D}_1, \underline{D}_2$	Vector of piecewise linear time delay parameters at sensors 1 and 2
$\underline{\tilde{D}}^1, \underline{\tilde{D}}^2$	Vector of quadratic time delay parameter at sensors 1 and 2
$E[\quad]$	Expectation operator

— Expectation operator

TERMS DEFINED BY ENGLISH SYMBOLS (Continued)

f_c	Center frequency of target power spectrum (Hz)
<u>FIT</u>	A vector of the accumulated ML function over time delay difference trajectories
f_1, f_2	Upper and lower bandpass cut-off frequencies
$G_s(f)$	Band-limited power spectrum of target signal
<u>G</u>	Gradient vector
$G_{n_1}(f), G_{n_2}(f)$	Band-limited power spectrums of received noise
$h_{i,j}$	Components of the H matrix
h_k, j_k	Gaussian distribution parameters for frequency ω_k
H	Constant least squares parameter matrix
I_1, I_2	Non-stationary Fourier coefficient modification factors
K	Constant of proportionality between ML and a priori information terms in the MAP function

TERMS DEFINED BY ENGLISH SYMBOLS (Continued)

$l(i)$	Component of least square fit parameter
$\underline{L}$	Vector of least square fit parameters
L_g	Length of the sensor array
L	Lower triangular matrix factor of matrix W
m	Number of correlation measurements in T seconds
M	Number of measurements in a data block
M	Lower triangular unity matrix
N	Index of largest frequency component
N_p	Number of subdivisions per lag point
$NPMAX$	Maximum allowable discretized points change allowed for the optimum time delay difference trajectory during one measurement cycle
NPT	Total number of discretized points in search interval
N_s	Total noise power received at each sensor point

TERMS DEFINED BY ENGLISH SYMBOLS (Continued)

$N_1(\omega_k), N_2(\omega_k)$	Fourier coefficient of received noise for frequency ω_k
o	Component of vector $\underline{0}$
$\underline{0}$	Observation vector
p	Order of a polynomial equation
P_t	Total target power for sloped power spectrum
$Q_{1,2}$	Non-stationary cross-power spectrum modification factors
Q_1, Q_2	Non-stationary auto-power spectrum modification factors
r_i	Lag point components of vector $\underline{R}$
$\tilde{r}_i$	Lag point components of vector $\tilde{\underline{R}}$
$\underline{R}$	Vector of discrete cross-correlation lag points (independent or whitened noise)
$\tilde{\underline{R}}$	Vector of discrete measured cross-correlation lag points

TERMS DEFINED BY ENGLISH SYMBOLS (Continued)

$R(t)$	Measured cross-correlation function between sensors 1 and 2
$\bar{R}(t)$	Expected or target-only cross-correlation function between sensors 1 and 2 (mismatch function)
$s(t)$	Coherent signal waveform from target
SNR_{AVG}	Average SNR over a block of data
<u>SSD</u>	A vector of the accumulated Sum of Square Differences of time delay difference trajectory
<u>SSR</u>	A vector of the accumulated Sum of Squared Residuals of time delay difference trajectories
t	Time
τ	Constant time interval between lag points
T	Observation interval for each sensor measurement
T	Matrix of optimum time delay difference trajectories for each of NPT discretized ending points

TERMS DEFINED BY ENGLISH SYMBOLS (Continued)

u_i	Lag point component of vector $\underline{u}$
$\underline{u}$	Flat spectrum component of discrete mismatch function
U	Upper triangular matrix factor of matrix W
v_i	Lag point component of vector $\underline{v}$
$\underline{v}$	Slope spectrum component of discrete mismatch function
w	Sum of the normalized MAP function over a block of data
W	Constant noise correlation matrix between lag points
$x_1(t)$	Received signal at sensor 1
$x_2(t)$	Received signal at sensor 2
$x_1(\omega), x_2(\omega)$	Fourier transform of received signals $x_1(t)$ and $x_2(t)$
$\underline{y}$	Vector of received signal Fourier coefficients

TERMS DEFINED BY ENGLISH SYMBOLS (Continued)

$z_{i,j}$ Element of the matrix z

$z(\omega_k)$ Covariance matrix at frequency ω_k

TERMS DEFINED BY GREEK SYMBOLS

$\alpha_1(\omega_k), \alpha_2(\omega_k)$ Fourier coefficients of target signal for stationary target at frequency ω_k

$\tilde{\alpha}_1(\omega_k), \tilde{\alpha}_2(\omega_k)$ Fourier coefficients of target signal for non-stationary target at frequency ω_k

β_1, β_2 Piecewise linear time delay parameters

$\tilde{\beta}_1, \tilde{\beta}_2$ Quadratic time delay parameters

γ Spectral slope parameter

$\Delta\beta$ Slope at a given piecewise linear segment of the time delay difference trajectory

$\underline{\Delta\beta T}$ Vector of time delay differences slope parameters

$\hat{\Delta\dot{d}}_i$ Estimated time delay difference rate at time $\tilde{t}_i$

$\hat{\Delta\ddot{d}}_i$ Estimated time delay difference acceleration at time $\tilde{t}_i$

$\underline{\Delta D}$ Vector difference of $\underline{D}_2 - \underline{D}_1$

TERMS DEFINED BY GREEK SYMBOLS (Continued)

$\underline{\Delta \tilde{D}}$	Vector difference of $\underline{\tilde{D}}^2 - \underline{\tilde{D}}^1$
Δr_i	Subinterval measured cross-correlation function
$\overline{\Delta r_i}$	Subinterval expected cross-correlation function
Δs	Piecewise linear time delay trajectory slope parameters
Δt	Smallest time interval used for practical correlation measurement
$\Delta \omega$	Incremental bandwidth of Fourier coefficients
$n_1(t)$	Incoherent noise waveform received at sensor 1
$n_2(t)$	Incoherent noise waveform received at sensor 2
$\theta, \tilde{\theta}$	Equivalent time delay differences
λ	Wave length
τ	Total observation interval for block of received measurements

TERMS DEFINED BY GREEK SYMBOLS (Continued)

$\phi_1(t), \phi_2(t)$	Signal delay as seen by sensors 1 and 2
$\tilde{\phi}_1, \tilde{\phi}_2$	Alternative time delays
ω_c	Target spectrum center frequency ($\pi(f_2 - f_1)$)
ω_k	kth frequency components ($\omega_k = 2\pi f_k$)

EFFICIENT RECURSIVE BATCH TIME DELAY DIFFERENCE ESTIMATION IN THE PRESENCE OF TARGET MOTION

CHAPTER 1

INTRODUCTION

In this chapter we shall define the problem addressed in this study along with our associated technical objectives. We shall give background information on our chosen problem, including a synopsis of relevant previous works. We shall state clearly the organization of the material presented in this study. Finally, we shall make a statement of contribution along with an executive summary of our major results.

1.1 BACKGROUND

We are concerned with the problem of estimating time delay difference parameters between time histories of broad spectrum energy observed at two spatially separated sensors. The observations at each sensor consist of measurements made on a combination of time-varying fields of coherent broad spectrum energy radiating from a single moving target and incoherent local broad spectrum noise energy. Each measurement consists of T second observations of the received time varying energy field.

There are numerous estimators and associated algorithms designed to estimate the time delay difference between two sensors due to a stationary target. However, we are primarily concerned with the more difficult problem of estimating time delay difference parameters for significant target motion at low Signal-power to Noise-power Ratio (SNR). In the real world, targets of interest are generally in motion and initial detection of targets is generally made at low SNR.

The above problem has two significant phases. The first, and by definition the most critical, is the initialization of the estimation process on target until a steady state condition is achieved. The second is the continuing steady state recursive estimation of the target time delay difference parameters for the duration of the problem. In the past, practical estimation algorithms have placed most emphasis on the later steady state estimation process with nominal concern for the initialization problem.

The traditional approach to the above estimation procedure is to use recursive adaptive (digital) filtering techniques. These filter estimators combine the current measurement with the previous cycle filter states to obtain the current estimate. Since the previous filter states are a function of all previous measurements, these filters are called Infinite Impulse Response (IIR) filters. The detail design of these filter algorithms represents a compromise between parameter estimate variance, dynamic parameter following capability, and transient characteristics.

Our approach to the above problem shall be to simultaneously operate on a finite block of measured data represented by M consecutive T -second observations. This type of estimator is called a batch estimator. If the finite block of M measurements is sequentially updated with new measurements every T seconds while discarding the oldest measurement, the estimator is called a recursive batch estimator. In the parallel field of digital filtering, the equivalent filter type is called a Finite Impulse Response (FIR) filter.

We have chosen a batch estimator for several reasons. First, maintaining a history of all input measurements up to the block size will yield the potential for good initialization algorithms. Second, a maintained history of data measurements and their associated time delay difference estimates lend themselves naturally to a priori information assumptions on the target time delay difference trajectory; and finally individual measurements in the measurement block can respond quickly to target dynamics while maintaining the total observation time of the entire block.

The advantages of a recursive batch estimator are two: first, a single measurement update results in new time delay difference parameter updates; and second, time delay difference parameter estimates are based solely on local measurements.

Our objective is to investigate and design optimal recursive batch estimators for the two-sensor non-stationary target time delay estimation problem. Our estimation algorithm should have good initialization

properties and be capable of estimating significant target motion at low SNR.

The practical problem which motivates our study is from the field of underwater acoustics. We want to estimate the bearing and bearing rate (equivalent to time delay difference and time delay difference rate) from passive spatially separated measurements of a broad spectrum noise field due to a moving target. However, the recursive batch techniques we shall examine are applicable to any estimate of time delay difference due to measurements of broad spectrum signal energy from spatially separated point sensors. In addition, the recursive batch algorithm development will have general application to similarly based algorithms.

Practical problems require practical realizable solutions. Therefore, our work will strive to apply practical considerations to theoretical work at every opportunity. In addition, a major objective of this study is to develop, analyze, and test the ground work for candidate estimation algorithms based on our theoretical results.

1.2 TECHNICAL OBJECTIVES

Major technical objectives of this study will involve three related areas: (1) theoretical development, (2) numerical study, and (3) validation via computer simulation.

Our principal theoretical development will be to derive a Maximum A Posteriori (MAP) estimator for the non-stationary time delay estimation problem. A key step in this process is to develop suitable models for the non-stationary time delay process. From our derived models a Maximum Likelihood (ML) estimator and a priori information function will be derived to complete our MAP estimator. The MAP estimator will be extended to estimate time delay difference parameter trajectories over a block of input data. The time delay difference parameters to be estimated include the following:

1. Time delay difference trajectory
2. Time delay difference rate trajectory
3. Time delay difference acceleration trajectory.

Other related parameters to be estimated are:

1. Average received SNR
2. A spectral shape parameter.

Our principal numerical effort will be to develop an efficient practical algorithm to implement our MAP estimators over a block of input data measurements. The practical algorithms will take on two distinct forms. The first form will be algorithms using discrete dynamic programming techniques. The second form will be algorithms using gradient search techniques. In both forms our algorithms will stress practical application.

Our first experimental objective will be to determine the performance characteristics of our major algorithms as a function of algorithmic parameter selection. This work will yield insight into intelligent algorithmic parameter selection.

Our second experimental objective is to study relative algorithm performance and performance against standard algorithms and theoretical bounds.

1.3 PREVIOUS WORK

The problem of optimum time delay estimation under the assumption of a stationary single target has been thoroughly studied by many investigators. Their work is adequately summarized in a paper by C. H. Knapp and G. C. Carter¹ on the generalized correlation methods for estimation of time delay. This work is of basic interest to us since our general approach to time delay estimation is also using correlation techniques.

The problem of optimum time delay estimation under the assumption of a single non-stationary target has received less attention. However, some papers have been written in this area. Notable among these is a paper by Knapp and Carter² on extending the generalized cross correlation techniques to the case of a moving target. Their concept is to time compress or expand one or both of the two received spacially separated waveforms to compensate for target motion and then perform a standard generalized cross-correlation technique for estimating the

time delay. Although the Knapp and Carter work² leads to a significantly different implementation strategy than our work, we shall make extensive use of their general approach to calculate our ML estimator. In addition, Y. T. Chan, J. M. Riley, and J. B. Plant³ have done work where the non-stationary time delay process is viewed as a FIR filter, and Friedlander⁴ has done work in combined spectral and time delay estimation using an Auto Recursive Moving Average (ARMA) model. The time delay estimation portion involves a moving average or FIR filter representation of the unknown non-stationary time delay. We have made less use of the above work in the development of our estimators. However, the FIR model for the non-stationary time delay process has a parallel in our development of a 'MISMATCH' function.

L. C. Ng⁵ in his PhD dissertation addresses the problem of multi-target, multisensor tracking of a non-stationary target. The batch processing approach and general time delay trajectory modeling used by Ng is similar to our own development. We shall expand on the work of Ng for the two-sensor one-moving-target scenario. Our most significant development in this area shall be the improved derivation of the 'MISMATCH' function and its redefinition for use in a practical estimator.

The general approach for practical time delay estimation involves maximizing a MAP function over a block of data. An important practical implementation of the above approach was inspired by the dynamic programming technique employed by J. J. Wolcin^{6,7,8} and refined to a

recursive batch process by R. A. LaTourette, S. G. Greineder, and J. J. Wolcin.⁹

1.4 REPORT ORGANIZATION

The organization of this report basically follows our technical objectives. Chapter 1 defines our problem and summarizes our contribution and results. Chapter 2 discusses our assumed models. Chapter 3 develops the theoretical MAP estimator. Chapter 4 describes our practical algorithm development. Chapter 5 gives the results of our empirical algorithm parameter investigation. Chapter 6 presents the results of our relative algorithm performance studies. Finally, Chapter 7 summarizes our work and suggests future work.

1.5 STATEMENT OF CONTRIBUTION AND EXECUTIVE SUMMARY

We have derived an approximate MAP estimator for the time delay difference parameters of a moving acoustical target as seen by two sensors. The target is assumed to radiate flat broad spectrum Gaussian noise. The ML portion of the MAP estimator was determined to be the set of parameters which maximized the product of the measured normalized cross-correlation lag points between point sensors and the normalized expected cross-correlation lag points (mismatch function).

For the case of linear time delay motion, we have calculated the correct analytical form of the mismatch function and the appropriate normalization factor. To the best of our knowledge, this is the first

correct derivation of the mismatch function to appear in the literature.

The derivation of the mismatch function was expanded to include the more general case of a linear target power spectrum. From this work we were able to develop an estimator for a parameter related to the slope of the unknown target power spectrum.

For our ML estimator we have assumed that the received target time delay trajectory is linear over our measurement interval and continuous (piecewise linear) between measurements. However, over a block of M contiguous measurements we have assumed that the received target time delay difference trajectory is a quadratic function of time. From the later assumption we have derived an a priori information penalty function based upon the sum of the squared residuals between the candidate ML time delay difference estimates and their best quadratic least mean square fit (LMSF). The role of the LMSF a priori information penalty function is to enhance the local ML estimate with information obtained over the whole block of M measurements. We noted that using the LMSF a priori information penalty function simultaneously with the ML estimate was different and superior to post refining the ML estimates with a quadratic LMSF. The quadratic LMSF penalty function also yields a procedure for obtaining refined estimates of time delay difference, time delay difference rate, and time delay difference acceleration. The optimum relative weight K between the ML estimate and the a priori information was determined to have a complex functional dependency including algorithm implementation.

A major emphasis of our study was to derive practical recursive block estimation algorithms based on our theoretical work. Therefore, our implementation problem can be defined as follows: we wish to determine the time delay difference parameters which maximizes our derived MAP function over a block of M consecutive measurements. We further require that the estimates of time delay difference parameters be recursively updated as the oldest data measurement is removed from the data block and a new measurement is added to the data block.

For the above objective, we have derived an efficient recursive dynamic programming algorithm and a recursive gradient search algorithm.

The basic dynamic programming algorithm is based upon a similar algorithm developed by Wolcin^{6,7,8} for frequency estimation. We claim credit for introducing the recursive modification to the algorithm, modifying the algorithm for the time domain application, and increasing algorithm efficiency.

The recursive modification for the dynamic programming algorithm involved recursively updating the sum of square residuals of a quadratic LMSF. For this purpose we employed an algorithm developed by Ng¹⁰ for efficiently updating recursive least squares parameters. In studying Ng's algorithm, we discovered a numerical instability and implemented a numerical technique to eliminate this instability.

The gradient search algorithm is based upon calculating the gradient vector of the MAP function over the block of input measurements with respect to the unknown time delay difference initial offset and rates. We have analytically calculated the required gradient vector. The algorithm uses the previous update parameter estimates to determine excellent initial parameter estimates for the current measurement update. To the best of our knowledge, this is the first numerical search technique to be applied to this type of problem.

For both the dynamic programming and gradient search algorithms we have recognized and have placed special emphasis on improving initial acquisition properties.

The following is a top level list of major results obtained from simulation experiments. A more detailed list can be found in Chapter 7.

1. The gradient search algorithm was found to have superior steady state performance characteristics with respect to other related recursive batch algorithms and the Cramer-Rao Lower Bound (CRLB) of a coherent integrator with equivalent averaging time but no a priori information. The gradient search algorithm obtains its advantage over the CRLB by the addition of a priori information.

2. The gradient search algorithm yields meaningful estimates of both time delay difference and time delay difference rate. The time delay difference acceleration estimate was not meaningful.
3. The gradient search algorithm only yields a valid target power spectral shape parameter estimate for SNR level greater than or equal to -10 dB.
4. Moderate target power spectral modeling errors have little affect on the performance of the gradient search algorithm.
5. Optimal parameter selection can be determined for the gradient search algorithm. However, the optimum parameters are often dependent on the target motion characteristics.
6. The dynamic programming algorithm has excellent initialization properties. With moderate target dynamics, the dynamic program algorithm can successfully initialize within a few dB of steady state performance thresholds. The gradient search algorithm initialization performance is slightly degraded from the dynamic programming algorithm. The gradient search algorithm initialization performance is still very good, but there is potential for improvement.

CHAPTER 2

PROBLEM FORMULATION

2.1 INTRODUCTION

In this section we shall discuss the assumed observation scenario, mathematically describe the resulting input sensor measurements, and formulate the general batch Maximum A Posteriori (MAP) time delay difference estimation problem. Since a major objective of this study is to realize practical estimators, practical considerations shall be noted and incorporated throughout.

2.2 DESCRIPTION OF THE OBSERVATION SCENARIO

Let us assume a sensor array which can be represented as two sensors in space separated by a distance D_p (Dipole Spacing). Let us further assume that a distant target has relative motion with respect to the received sensor array. The target is at a sufficient range from the sensor array that the received wavefronts can be considered plane waves. Figure 2-1 depicts the scenario below. The following equation states the condition for which our plane wave assumption is valid.

$$\text{RANGE} \gg \frac{L_g^2}{\lambda} \text{ (Steinberg)}^{11} \quad (2-1)$$

L_g = Array or aperture length

λ = Wave length of the highest frequency.

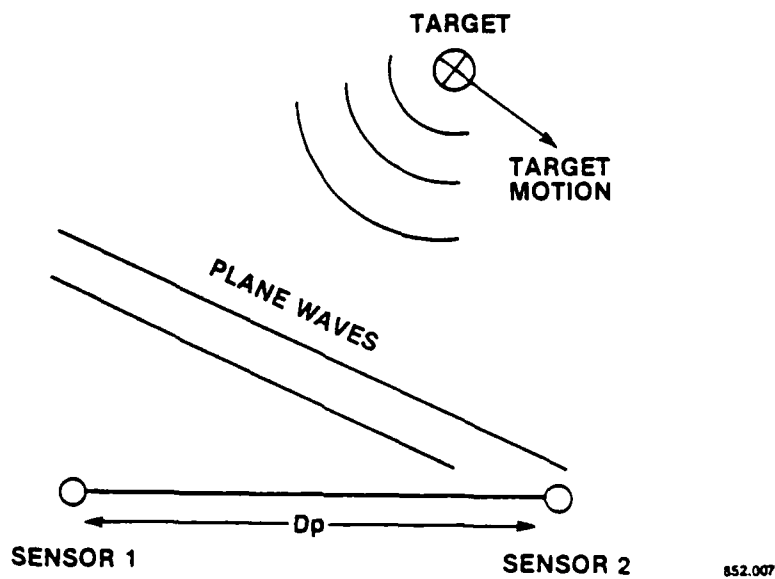


Figure 2-1. Observation Scenario

In practice, if the target is a large object such as a ship, or to a lesser extent an aircraft, the motion of the target will be continuous and slowly changing over an extended period of time. We can assume that the resulting relative bearing trajectory and the corresponding time delay trajectories can be adequately modeled by a low-order polynomial over the time interval of interest.

2.3 DESCRIPTION OF INPUT SENSOR MEASUREMENTS

From Figure 2-1 the received signal at both sensor points can be modeled as shown below.

$$x_1(t) = s(t + \phi_1(t)) + n_1(t)$$

$$x_2(t) = s(t + \phi_2(t)) + n_2(t)$$

(2-2)

where $s(t)$ is the signal waveform and $n_1(t)$ and $n_2(t)$ are the noise fields at sensor 1 and sensor 2, respectively.

We shall assume that $s(t)$, $n_1(t)$, $n_2(t)$ are zero mean, Gaussian and mutually uncorrelated random processes with known band-limited flat* power spectrums; i.e.,

$$G_s(f) = \begin{cases} \frac{A_s}{2(f_2 - f_1)}, & f_1 \leq f \leq f_2 \text{ or } -f_2 \leq f \leq -f_1 \\ 0, & \text{otherwise} \end{cases} \quad (2-3)$$

$$G_{n_1}(f) = G_{n_2}(f) = \begin{cases} \frac{N_s}{2(f_2 - f_1)}, & f_1 \leq f \leq f_2 \text{ or } -f_2 \leq f \leq -f_1 \\ 0, & \text{otherwise} \end{cases} \quad (2-4)$$

where

A_s = Total signal power

N_s = Total noise power

f_1 = lower bandpass cutoff frequency

f_2 = upper bandpass cutoff frequency.

* Although we assume that both the signal and noise spectrums are flat, in practice, this assumption will rarely be true. The normal practice is to prewhiten or flatten the anticipated noise spectrum. The nominal assumption is that the signal spectrum will have the same shape as the noise spectrum and thus our prefilter will be optimal. Later in the analysis we shall loosen the requirement that the filtered signal spectrum will be flat and develop an algorithm to estimate a signal spectrum parameter. However, the primary development shall be with a flat signal and noise spectrum assumption. Therefore, emphasis must be placed on analyzing performance with non-flat spectrums.

In the above definition there are two implied assumptions. The first is that the nominal receive signal power is identical at sensors 1 and 2. The second is that the noise processes and the signal process have identical bandpasses. In practice, both assumptions are not unreasonable. In general, we would expect identical signal power at both receive sensors. Also for most practical applications the received signals are processed through some bandpass filter with a smaller bandwidth than either the signal process or the noise process. Therefore, the cutoff frequencies of the input bandpass process can be considered as the limiting frequencies f_1 and f_2 .

$\phi_1(t)$ and $\phi_2(t)$ represent the received signal delay as seen by sensor 1 and sensor 2, respectively, due to the moving target as a function of time.

We shall assume that over the total observation interval (τ) that $\phi_1(t)$ and $\phi_2(t)$ can be adequately modeled as a low order polynomial or specifically as a quadratic function; i.e.,

$$\begin{aligned}\phi_1(t) &= \tilde{d}_2^1 t^2 + \tilde{d}_1^1 t + \tilde{d}_0^1 & 0 \leq t \leq \tau \\ \phi_2(t) &= \tilde{d}_2^2 t^2 + \tilde{d}_1^2 t + \tilde{d}_0^2 & 0 \leq t \leq \tau\end{aligned}\tag{2-5}$$

where the $\tilde{d}$'s are coefficients of a polynomial representing time delay as a function of time.

We shall further assume that if the total observation interval τ is subdivided into M equal subdivisions each of length T , that $\phi_1(t)$ and $\phi_2(t)$ can also be adequately modeled as a piecewise linear function defined below for $\phi_1(t)$.

$$\phi_1^k(t) = \phi_1^{k-1}(t_{k-1}) + \Delta s_1^k(t - t_{k-1}), \quad t_{k-1} \leq t \leq t_k, \\ k = 1, 2, 3, \dots M \quad (2-6)$$

$$\phi_1^0(0) = d_0^1 \quad (2-6A)$$

where now we define

k = subinterval number

d_0^1 = initial or reference time delay at sensor 1

$$\Delta s_k^1 = (d_k^1 - d_{k-1}^1)/T \quad (2-6B)$$

d_k^1 = the time delay at sensor 1 at time t_k .

Note that for this model the d 's now represent actual time delays at given time instances.

2.4 STATEMENT OF PROBLEM

Our objective is to investigate optimal and suboptimal estimators for the time delay difference between two-point sensors for a single moving target as a function of time. The estimators we will examine shall operate on M input measurements, each of T second duration, from the two-point sensor array. The measurements will be from the current time back to $M \cdot T$ or τ seconds in the past. This type of process is known as a batch process and our estimators will be known as batch

estimators. The batch process has the advantage of receiving updates, processing data, and making decisions at a T second rate while maintaining a total observation time of $M \cdot T$ or τ seconds. A primary objective for our estimators will be to enhance the above advantage.

In general, a batch estimator yields estimates for any point within the processed block of data. The most accurate estimate⁵ is at an interior point of the input block, while the least accurate estimate is at the end. For our study we shall be interested in the first or most recent estimate and the middle estimate.

A second requirement for our approach is that they are recursive estimators (i.e., for every new sensor measurement (T seconds), there will be new estimator outputs). This requires that our input block of M sensor measurements acts as a sliding window. When a new sensor measurement is received, it is added to the block of input data while the oldest sensor measurement is dropped from the block of input data. New estimates will be calculated after each update. For the case of the middle time frame estimate, this process will result in a time history of estimates each separated by T seconds in time and the most recent estimate delayed by one half the block length. The most recent time frame estimate will also result in a time history of estimates each separated by T seconds, but the most recent estimate will be nominally at current time. Obviously, with the appropriate time shift of one half the data block length, these two time histories will coincide. We would expect that the variance of the middle time frame estimates would be less than the most recent time frame estimates.

As mentioned in Chapter 1, we will approach the above problem using MAP techniques. The MAP approach was chosen to take advantage of a priori knowledge of the characteristics of the target time delay difference trajectory. This analysis will yield the form of the input sensor measurements and optimum or near optimum processing strategy. These results will be directed towards practical considerations to yield a basis for practical estimators.

CHAPTER 3

THEORETICAL DEVELOPMENT OF MAP ESTIMATORS

3.1 INTRODUCTION

In this chapter we shall examine the two-sensor, single moving target time delay difference estimation problem outlined in Chapter 2. Our objective is to determine a form for the input measurements and derive an optimum or near optimum processing strategy. A key concept is that we are looking at two problems in one. On one hand we would like to view our problem as making a single measurement over a large interval of time τ . For this problem we assume that the unknown time delay difference trajectory is quadratic in time. This approach has the advantage of long integration time of the observed measurements. We shall refer to the above as Problem 1. A second view is that our measurements consist of M intervals of duration T seconds, where $M \cdot T = \tau$. The total time delay difference trajectory is assumed to be piecewise linear with each of the M subintervals being linear. This approach has the advantage of being more adaptive (T second update rate) to the time delay difference trajectory dynamics and has a simpler model (linear) than Problem 1. We shall call this Problem 2. Ideally, we would like to simultaneously gain the advantages of long observation time of the measurement data of Problem 1 and the adaptability of Problem 2.

3.2 THE MAP EQUATION

Our general approach shall be to apply Maximum A Posteriori (MAP) estimation techniques as outlined by Van Trees.¹² The MAP estimate is defined as the set of parameters $\hat{a}(\underline{O})$ which maximizes the a posteriori density function $P_{a/o}(\underline{A}/\underline{O})$ of a random parameter vector $\underline{A}$ conditioned on observations or measurements $\underline{O}$.

The standard procedure is to rewrite $P_{a/o}(\underline{A}/\underline{O})$ in terms of the measurements $\underline{O}$ conditioned on the unknown parameter vector $\underline{A}$ (maximum likelihood estimate) and a priori knowledge. By Bayes rule, we have the following.

$$P_{a/o}(\underline{A}/\underline{O}) = \frac{P_{o/a}(\underline{O}/\underline{A}) P_a(\underline{A})}{P_o(\underline{O})} \quad (3-1)$$

Taking the logarithm of Equation 3-1 which will not alter the location of the maximum yields the following result:

$$\ln P_{a/o}(\underline{A}/\underline{O}) = \ln P_{o/a}(\underline{O}/\underline{A}) + \ln P_a(\underline{A}) - \ln P_o(\underline{O}) \quad (3-2)$$

Since the measurement space $\underline{O}$ does not depend on the parameter vector $\underline{A}$, the final MAP equation becomes the following:

$$\underset{\underline{A}}{\text{MAX}} [\ln P_{o/a}(\underline{O}/\underline{A}) + \ln P_a(\underline{A})] \quad (3-3)$$

Equation 3-3 contains two components. The first $[\ln P_{o/a}(\underline{Q}/\underline{A})]$ represents a Maximum Likelihood (ML) estimate of the parameter vector $\underline{A}$ from the observed measured data $\underline{Q}$. The second component $[\ln P_a(\underline{A})]$ represents a priori knowledge of the unknown parameter vector $\underline{A}$.

In the following sections our goal will be to develop expressions representing the ML and a priori portions of the MAP estimator.

3.3 THE MAXIMUM LIKELIHOOD ESTIMATOR FOR PROBLEM 1

Our objective is to find the form of the ML estimator for the time delay difference trajectory between two-point sensors for a single moving target. We assume that we observe the sensor measurements for a period of time τ and that the received signal time delay as seen by each sensor can be represented by a quadratic function in time. Therefore, one can describe the time delay trajectory at sensors 1 and 2, respectively, as the parameter vectors $\underline{\tilde{D}}^1(\tilde{d}_2^1, \tilde{d}_1^1, \tilde{d}_0^1)$ and $\underline{\tilde{D}}^2(\tilde{d}_2^2, \tilde{d}_1^2, \tilde{d}_0^2)$ which represent coefficients to a quadratic equation in time. The time delay difference trajectory $\underline{\Delta D}(\Delta \tilde{d}_2, \Delta \tilde{d}_1, \Delta \tilde{d}_0)$ is also quadratic in time where

$$\begin{aligned}\Delta \tilde{d}_2 &= \tilde{d}_2^2 - \tilde{d}_2^1 \\ \Delta \tilde{d}_1 &= \tilde{d}_1^2 - \tilde{d}_1^1 \\ \Delta \tilde{d}_0 &= \tilde{d}_0^2 - \tilde{d}_0^1\end{aligned}\tag{3-4}$$

Equation 2-5 for the time delay difference $\Delta\phi(\underline{\tilde{D}}, t)$ is repeated below for convenience.

$$\Delta\phi(\underline{\tilde{D}}, t) = \Delta\tilde{d}_2 t^2 + \Delta\tilde{d}_1 t + \Delta\tilde{d}_0 . \quad (3-5)$$

Our approach to initially estimate the form of the ML estimator for Problem 1 will follow the lines developed by Knapp and Carter.²

We have from Chapter 2 and above the following models for the signal received at sensor 1 and sensor 2, respectively.

$$\begin{aligned} x_1(t) &= s(t + \phi_1(\underline{\tilde{D}}^1, t)) + \eta_1(t) \\ x_2(t) &= s(t + \phi_2(\underline{\tilde{D}}^2, t)) + \eta_2(t) \end{aligned} \quad (3-6)$$

where $s(t)$ and $\eta(t)$ are described in Chapter 2.

The Knapp and Carter approach is to represent the observed signal waveforms $[x_1(t), x_2(t)]$ as a set of Fourier coefficients $[c_{f_1}(\omega_{-N}), c_{f_1}(\omega_{-N+1}) \dots c_{f_1}(\omega_N), c_{f_2}(\omega_{-N}), c_{f_2}(\omega_{-N+1}) \dots c_{f_2}(\omega_N)]$. These Fourier coefficients are assumed to be independent zero mean Gaussian random processes. This assumption assumes a sufficient Bandwidth Time (BT) product (i.e., $BT > 8$).⁵ The procedure for calculating the ML estimate is to calculate the joint conditional Gaussian density function of the Fourier coefficients conditioned, at least initially, on the unknown parameter vectors $\underline{\tilde{D}}^1, \underline{\tilde{D}}^2$. The ML estimate is the vectors

$\hat{\underline{D}}^1, \hat{\underline{D}}^2$ which maximize the log of the conditional Gaussian density function. However, it is important to note that our primary interest is to estimate the unknown time delay difference parameter vector $\hat{\underline{\Delta D}}$.

Proceeding along the above line of reasoning, we can represent the kth frequency component of $x_1(t)$ and $x_2(t)$ as the following:

$$c_{f_1}(\omega_k) = \frac{1}{\tau} \int_0^{\tau} x_1(t) e^{-j\omega_k t} dt \quad (3-7)$$

$$c_{f_2}(\omega_k) = \frac{1}{\tau} \int_0^{\tau} x_2(t) e^{-j\omega_k t} dt$$

where $\omega_k = k \Delta\omega$ and $\Delta\omega = 2\pi/\tau$.

We can now represent our observation vector $[x_1(t), x_2(t)]^T$ over the interval τ by a vector $\underline{Y}^T$ of their Fourier coefficients given below:

$$\underline{Y}^T = [\underline{C}_f^T(\omega_{-N}), \underline{C}_f^T(\omega_{-N+1}) \dots, \underline{C}_f^T(\omega_0) \dots \underline{C}_f^T(\omega_N)] \quad (3-8)$$

$$\text{where } \underline{C}_f^T(\omega_k) = [c_{f_1}(\omega_k), c_{f_2}(\omega_k)] \quad (3-9)$$

Our objective is to calculate the probability density of $\underline{Y}$ conditioned on the parameter vectors $\underline{\tilde{D}}^1$ and $\underline{\tilde{D}}^2$ which we shall designate as $P(\underline{Y}/\underline{\tilde{D}}^1, \underline{\tilde{D}}^2)$. Knapp and Carter² give this conditional density function as the following.

$$P(\underline{Y}/\underline{\tilde{D}}^1, \underline{\tilde{D}}^2) = \prod_{k=-N}^N h_k e^{-J_k^{1/2}} \quad (3-10)$$

where

$$h_k = [2\pi |E[\underline{C}_f(\omega_k) \underline{C}_f^{*T}(\omega_k)]|]^{-1} \quad (3-10A)$$

and

$$J_k = \underline{C}_f^{*T}(\omega_k) [E[\underline{C}_f(\omega_k) \underline{C}_f^{*T}(\omega_k)]]^{-1} \underline{C}_f(\omega_k) \quad (3-10B)$$

Note E represents the expectation operation.

The above equations reflect the assumption that the $\underline{C}_{f1}(\omega_k)$'s are independent zero mean Gaussian random variables. Therefore,

$$E[\underline{C}_f(\omega_k) \underline{C}_f^{*T}(\omega_q)] = 0 \text{ for } \omega_k \neq \omega_q \quad (3-11)$$

and the only quantity remaining to be calculated is the following covariance matrix:

$$E[\underline{C}_f(\omega_k) \underline{C}_f^{*T}(\omega_k)] \quad (3-12)$$

We shall now roughly determine the form of the covariance matrix of Equation 3-12. This calculation will allow us to obtain the general form of a practical approximate ML estimator. In later analysis we shall re-examine this calculation in more accurate detail with a different theoretical model. This new analysis will yield the precise form of the 'mismatch function'.

Our approach to calculate the covariance matrix (Equation 3-12) shall be similar to that adopted by Ng.⁵

From Equations 3-6 and 3-7, we have that

$$c_{f_1}(\omega_k) = \frac{1}{\tau} \int_0^{\tau} [s(t, \phi_1(\underline{\theta}^1, t)) + n_1(t)] e^{-j\omega_k t} dt \quad (3-13)$$

The above expression can be rewritten as a signal component $\tilde{\alpha}_1(\omega_k)$ and a noise component $N_1(\omega_k)$.

$$c_{f_1}(\omega_k) = \tilde{\alpha}_1(\omega_k) + N_1(\omega_k) \quad (3-14)$$

$N_1(\omega_k)$ is the Fourier coefficient at frequency $\omega_k = k \Delta\omega$ for the noise process as seen at sensor 1.

$\tilde{\alpha}_1(\omega_k)$ represents the Fourier coefficient at $\omega_k = k \Delta\omega$ for the non-stationary signal process as seen at sensor 1.

$$\tilde{\alpha}_1(\omega_k) = \frac{1}{\tau} \int_0^{\tau} s(t, \phi_1(\underline{\theta}^1, t)) e^{-j\omega_k t} dt \quad (3-15)$$

We would like to be able to express $\tilde{\alpha}_1(\omega_k)$, the Fourier coefficient for the non-stationary target, in terms of $\alpha_1(\omega_k)$, the Fourier coefficient for a stationary target, which we have assumed knowledge of.

The stationary Fourier coefficient for sensor 1 is given below.

$$\alpha_1(\omega_k) = \frac{1}{\tau} \int_0^{\tau} s(t) e^{-j\omega_k t} dt, \quad (3-16)$$

and the stationary band-limited signal $s(t)$ can be written in terms of the Fourier coefficients as the following:

$$s(t) \approx \sum_{\ell=-N}^N \alpha_1(\omega_{\ell}) e^{j\omega_{\ell} t}. \quad (3-17)$$

In Equation 3-15, make the substitution $\gamma = t + \phi_1(\bar{D}^1, t)$, yielding:

$$\tilde{\alpha}_1(\omega_k) = \frac{1}{\tau} \int_{\phi_1(\bar{D}^1, \tau)}^{\tau + \phi_1(\bar{D}^1, \tau)} s(\gamma) e^{-j\omega_k \tilde{\phi}_1(\bar{D}^1, \gamma)} \frac{d\gamma}{1 + \dot{\phi}_1(\bar{D}^1, \gamma)} \quad (3-18)$$

where

$$\tilde{\phi}_1(\bar{D}^1, \gamma) = \frac{-\tilde{\beta}_1 + \sqrt{\tilde{\beta}_1^2 - 4 \tilde{d}_2^1 (\tilde{d}_0^1 - \gamma)}}{2 \tilde{d}_2^1} \quad (3-18A)$$

and

$$\tilde{\beta}_1 = 1 + \tilde{d}_1^1. \quad (3-18B)$$

Substituting Equation 3-17 into Equation 3-18 and exchanging summation with integration yields:

$$\tilde{\alpha}_1(\omega_k) = \sum_{\ell=-N}^N \alpha_1(\omega_\ell) I_1(k, \ell, \tau, \underline{\tilde{D}}^1) \quad (3-19)$$

where

$$I_1 = \frac{1}{\tau} \int_{\phi_1(\underline{\tilde{D}}^1, \tau)}^{\tau + \phi_1(\underline{\tilde{D}}^1, \tau)} e^{j[\gamma \omega_\ell - \omega_k \tilde{\phi}_1(\underline{\tilde{D}}^1, \gamma)]} \frac{d\gamma}{1 + \phi_1(\underline{\tilde{D}}^1, \gamma)} \quad (3-19A)$$

We can repeat the above formulation for sensor 2 to yield the following:

$$\tilde{\alpha}_2(\omega_k) = \sum_{\ell=-N}^N \alpha_2(\omega_\ell) I_2(k, \ell, \tau, \underline{\tilde{D}}^2) \quad (3-20)$$

where

$$I_2 = \frac{1}{\tau} \int_{\phi_2(\underline{\tilde{D}}^2, \tau)}^{\tau + \phi_2(\underline{\tilde{D}}^2, \tau)} e^{j[\gamma \omega_\ell - \omega_k \tilde{\phi}_2(\underline{\tilde{D}}^2, \gamma)]} \frac{d\gamma}{1 + \phi_2(\underline{\tilde{D}}^2, \gamma)} \quad (3-20A)$$

We now have the equations to symbolically calculate the four elements of the covariance matrix of Equation 3-12.

The (1,1) element using Equations 3-12, 3-9, and 3-14 can be written as the following:

$$E[c_{f_1}(\omega_k) c_{f_1}^*(\omega_k)] = E[\tilde{\alpha}_1(\omega_k) \tilde{\alpha}_1^*(\omega_k)] + E[N_1(\omega_k) N_1^*(\omega_k)] \quad (3-21)$$

Note that due to our uncorrelated processes assumption,

$$E[\tilde{\alpha}_1(\omega_k) N_1^*(\omega_k)] = E[N_1(\omega_k) \tilde{\alpha}_1^*(\omega_k)] = 0 \quad (3-22)$$

Letting $\overline{\quad}$ represent the expectation operation and substituting Equation 3-19 into Equation 3-21 yields the following.

$$E[c_{f_1}(\omega_k) c_{f_1}^*(\omega_k)] = \sum_{\ell=-N}^N \overline{(\alpha_1(\omega_\ell) \alpha_1^*(\omega_\ell) I_1 I_1^*)} + \overline{N_1(\omega_k) N_1^*(\omega_k)} \quad (3-23)$$

If we now let τ effectively approach infinity while maintaining ω_k constant, we can rewrite Equation 3-23 as an integral. Note the following relationships.

$$\omega_\ell = \frac{2\pi\ell}{\tau} \rightarrow \omega \quad (3-24A)$$

$$B = \omega_N = \frac{2\pi N}{\tau} \quad (3-24B)$$

$$-B = \omega_{-N} = \frac{-2\pi N}{\tau} \quad (3-24C)$$

$$\lim_{\tau \rightarrow \infty} \overline{\tau \alpha_1(\omega_\ell) \alpha_1^*(\omega_\ell)} = G_S(\omega_\ell) \quad (3-24D)$$

$$\lim_{\tau \rightarrow \infty} \overline{\tau N_1(\omega_k) N_1^*(\omega_k)} = G_{n_1}(\omega_k) \quad (3-24E)$$

where $G_s(\omega)$ is the signal power spectrum described in Chapter 2 and $G_{\eta_1}(\omega)$ is the noise power spectrum as seen at sensor 1 (also described in Chapter 2).

With the above in mind, Equation 3-23 can be written as the following.

$$E[c_{f_1}(\omega_k), c_{f_1}^*(\omega_k)] = \frac{1}{4\pi} \int_{-B}^B G_s(\omega) \lim_{\tau \rightarrow \infty} (I_1(\omega_k, \omega, \tau, \tilde{D}^1)) d\omega + \frac{1}{\tau} G_{\eta_1}(\omega_k) \quad (3-25)$$

Certainly if $G_s(\omega)$ is a constant as we initially assumed in Chapter 2, we can rewrite Equation 3-25 as the following. Note that we make the above assumption for convenience at this point. Later we shall relax this restriction.

$$E[c_{f_1}(\omega_k) c_{f_1}^*(\omega_k)] = G_s(\omega_k) Q_1(\tau, \tilde{D}^1) + \frac{G_{\eta_1}(\omega_k)}{\tau} \quad (3-26)$$

where

$$Q_1(\omega_k, \tau, \tilde{D}^1) = \frac{1}{4\pi} \int_{-B}^B \lim_{\tau \rightarrow \infty} (I_1(\omega_k, \omega, \tau, \tilde{D}^1) I_1^*(\omega_k, \omega, \tau, \tilde{D}^1)) d\omega \quad (3-26A)$$

One can view the function $Q(\cdot)$ as a distortion of the autopower spectrum for a stationary target due to target motion. The distortion will be in terms of spectral spreading or contraction and magnitude

distortion. For reasonable target motion, we would expect the auto-power spectrum distortion to be small.

Two points are important to note at this time. The first is that, at low SNR where the noise power will dominate, small signal power losses due to motion will be impossible to separate from other causes of signal power loss. The second point is that sensor 1 autopower spectrum depends only on parameters $\underline{\tilde{D}}^1$. Our primary interest is to maximize with respect to the difference parameters $\underline{\Delta\tilde{D}} = \underline{\tilde{D}}^2 - \underline{\tilde{D}}^1$.

We can use identical arguments to obtain an expression for $E[c_{f_2}(\omega_k), c_{f_2}^*(\omega_k)]$ or element (2,2) which is written below.

$$E[c_{f_2}(\omega_k), c_{f_2}^*(\omega_k)] = G_s(\omega_k) Q_2(\omega_k, \tau, \underline{\tilde{D}}^2) + G_{\eta_2}(\omega_k) \quad (3-27)$$

Using similar arguments as above and noting that $E[N_1(\omega) N_2(\omega)] = 0$, we can write an expression for $E[c_{f_1}(\omega_k) c_{f_2}^*(\omega_k)]$ or element (1,2) which is shown below.

$$E[c_{f_1}(\omega_k) c_{f_2}^*(\omega_k)] = \frac{1}{4\pi} \int_{-B}^B G_s(\omega_k) \lim_{\tau \rightarrow \infty} (I_1(\omega, \omega_k, \tau, \underline{\tilde{D}}^1) \cdot I_2^*(\omega, \omega_k, \tau, \underline{\tilde{D}}^2)) d\omega \quad (3-28)$$

Again for convenience assuming that $G_s(\omega_k)$ is a constant, we have

$$E[c_{f_1}(\omega_k), c_{f_2}^*(\omega_k)] = G_s(\omega_k) Q_{1,2}(\omega_k, \tau, \underline{\tilde{D}}^2, \underline{\tilde{D}}^1) \quad (3-29)$$

where

$$Q_{1,2}(\omega_k, \tau, \underline{\tilde{D}}^1, \underline{\tilde{D}}^2) = \frac{1}{4\pi} \int_{-B}^B \lim_{\tau \rightarrow \infty} (I_1(\omega, \omega_k, \tau, \underline{\tilde{D}}^1) I_2^*(\omega, \omega_k, \tau, \underline{\tilde{D}}^2)) d\omega . \quad (3-29A)$$

Finally, we can write $E[c_{f_2}(\omega_k), c_{f_1}^*(\omega_k)]$ or element (2,1) as the following:

$$E[c_{f_2}(\omega_k), c_{f_1}^*(\omega_k)] = G_S(\omega_k) Q_{1,2}^*(\omega_k, \tau, \underline{\tilde{D}}^2, \underline{\tilde{D}}^1) . \quad (3-30)$$

Summarizing Equations (3-26), (3-27), (3-29), and (3-30), we have the following matrix:

$$\begin{aligned} Z(\omega_k) &= \tau E[\underline{C}_f(\omega_k) \underline{C}_f^*(\omega_k)] \\ &= \begin{bmatrix} G_S(\omega_k) \tau Q_1(\omega_k, \tau, \underline{\tilde{D}}^1) + G_{\eta_1}(\omega_k) & G_S(\omega_k) \tau Q_{1,2}(\omega_k, \tau, \underline{\tilde{D}}^1, \underline{\tilde{D}}^2) \\ G_S(\omega_k) \tau Q_{1,2}^*(\omega_k, \tau, \underline{\tilde{D}}^1, \underline{\tilde{D}}^2) & G_S(\omega_k) \tau Q_2(\omega_k, \tau, \underline{\tilde{D}}^2) + G_{\eta_2}(\omega_k) \end{bmatrix} \end{aligned} \quad (3-31)$$

Looking ahead, we can write an expression for the inverse of matrix $Z(\omega_k)$.

$$z(\omega_k)^{-1} = \frac{1}{|z(\omega_k)|} \cdot$$

$$= \begin{bmatrix} G_s(\omega_k) \tau Q_2(\omega_k, \tau, \underline{\tilde{D}}^2) + G_{n_2}(\omega_k) & -G_s(\omega_k) \tau Q_{1,2}(\omega_k, \tau, \underline{\tilde{D}}^1, \underline{\tilde{D}}^2) \\ -G_s(\omega_k) \tau Q_{1,2}^*(\omega_k, \tau, \underline{\tilde{D}}^1, \underline{\tilde{D}}^2) & G_s(\omega_k) \tau Q_1(\omega_k, \tau, \underline{\tilde{D}}^1) + G_{n_1}(\omega_k) \end{bmatrix}$$

(3-32)

where the determinant $|z(\omega_k)|$ can be written as:

$$|z(\omega_k)| = G_s(\omega_k)^2 \tau^2 [Q_1(\omega_k, \tau, \underline{\tilde{D}}^1) Q_2(\omega_k, \tau, \underline{\tilde{D}}^2) - Q_{1,2}(\omega_k, \tau, \underline{\tilde{D}}^1, \underline{\tilde{D}}^2) \cdot$$

$$Q_{1,2}^*(\omega_k, \tau, \underline{\tilde{D}}^1, \underline{\tilde{D}}^2)] + G_s(\omega_k) \tau [G_{n_2}(\omega_k) Q_1(\omega_k, \tau, \underline{\tilde{D}}^1)$$

$$+ G_{n_1}(\omega_k) Q_2(\omega_k, \tau, \underline{\tilde{D}}^2)] + G_{n_1}(\omega_k) G_{n_2}(\omega_k) \cdot \quad (3-32A)$$

We can now express the conditional density function $P(\underline{Y}/\underline{\tilde{D}}^1, \underline{\tilde{D}}^2)$ (Equation 3-10) by substituting Equations 3-32 and 3-32A into Equation 3-10 via Equations 3-10A and 3-10B.

$$P(\underline{Y}/\underline{\tilde{D}}^1, \underline{\tilde{D}}^2) = \prod_{k=-N}^N [2\pi |z(\omega_k)|]^{-1} \exp\{[-\underline{C}_f^{*T}(\omega_k) z^{-1}(\omega_k) \underline{C}_f(\omega_k)]^{1/2}\}$$

(3-33)

The standard procedure is to take the natural log of the conditional density function (Equation 3-33). We shall also let τ effectively approach infinity while maintaining $\omega_k = 2\pi k/\tau$ as a constant.

Note the following limits:

$$\omega_k = \frac{2\pi k}{\tau} \rightarrow \omega \quad (3-34A)$$

$$\omega_N = \frac{2\pi N}{\tau} \rightarrow B \quad (3-34B)$$

$$\omega_{-N} = \frac{-2\pi N}{\tau} \rightarrow B \quad (3-34C)$$

$$\lim_{\tau \rightarrow \infty} \tau c_{f_1}(\omega_k) c_{f_1}^*(\omega_k) = X_1(\omega) \quad (3-34D)$$

$$\lim_{\tau \rightarrow \infty} \tau c_{f_2}(\omega_k) c_{f_2}^*(\omega_k) = X_2(\omega) \quad (3-34E)$$

where $X_1(\omega)$ is the Fourier transform of received signal $x_1(t)$ at sensor 1 and $X_2(\omega)$ is the Fourier transform of received signal $x_2(t)$ at sensor 2.

The ML estimator are the parameters $\hat{\underline{D}}^1, \hat{\underline{D}}^2$ which maximize the following expression resulting from taking the natural log of Equation 3-32A and converting summation to their appropriate integrals.

$$\left(-\frac{\tau}{2} \int_{-B}^B \ln\left(\frac{2\pi}{\tau}\right) d\omega \right) \quad \left(-\frac{\tau}{4} \int_{-B}^B \ln|z(\omega)| d\omega \right)$$

(1)

(2)

$$\left(-\frac{1}{4\pi} \int_{-B}^B \frac{[|x_1(\omega)|^2 z_{2,2}(\omega) + |x_2(\omega)|^2 z_{1,1}(\omega)] d\omega}{|z(\omega)|} \right)$$

(3)

$$\left(+\frac{1}{2\pi} \int_{-B}^B x_1(\omega) x_2^*(\omega) \left[\frac{z_{1,2}(\omega) + z_{2,1}^*(\omega)}{2|z(\omega)|} \right] d\omega \right)$$

(3-35)

(4)

where $z_{i,j}$ are the elements of matrix $z(\omega)$ given in Equation 3-31.

Note expression 3-35 is symbolically the same as that arrived at by Knapp and Carter.²

Let us now examine each of the four terms in expression 3-35.

The first term is independent of $\tilde{D}^1$ and $\tilde{D}^2$ and therefore can be ignored.

The second term is a function of the determinant $|z(\omega)|$. This determinant as can be seen from Equation 3-32A is a function of $\tilde{D}^1$ and $\tilde{D}^2$ but is also a function of the noise power spectrums $G_{\eta_1}(\omega)$ and

$G_{n_2}(\omega)$. Although it is not clear at this point, for reasonable target motion, the primary effect of non-zero $\tilde{D}^1$ and $\tilde{D}^2$ is a small distortion in magnitude of the received auto and cross-power spectrums. Certainly at low SNR, variations in term 2 due to $\tilde{D}^1$ and $\tilde{D}^2$ will be unobservable even if a significant dependence exists. Note that for no target motion [$\tilde{D}^1 = \tilde{D}^2 = 0$] and our flat signal and noise power spectrum assumption that $|Z(\omega)|$ is a constant for all frequencies of interest. For now, we shall ignore term 2 and examine term 3.

Term 3 is dependent on $\tilde{D}^1$ and $\tilde{D}^2$ but is also dependent on the noise power spectrum $G_{n_1}(\omega)$ and $G_{n_2}(\omega)$. Examining term 3, it symbolically appears to be the sum of cross-matching the measured auto power spectrums for sensors 1 and 2 against what can be interpreted as the expected auto power spectrums for sensors 1 and 2. For small expected auto power spectrum magnitude variations, term 3 will be weakly dependent on $\tilde{D}^1$ and $\tilde{D}^2$ and certainly unobservable at low SNR. Therefore, we shall ignore term 3.

Term 4 is dependent on $\tilde{D}^1$ and $\tilde{D}^2$ and is not significantly dependent on the noise power spectral densities $G_{n_1}(\omega)$ and $G_{n_2}(\omega)$. In fact, as we shall see later, term 4 is primarily dependent on the difference parameter $\Delta\tilde{D}$, which is exactly what we want. Therefore, the estimate $\tilde{D}^1$ and $\tilde{D}^2$ which maximizes term 4 shall be our primary interest. Knapp and Carter² reached a similar conclusion for a first approximation.

As noted earlier, the determinant $|Z(\omega)|$ is weakly dependent on $\tilde{D}^1$ and $\tilde{D}^2$ and this dependency is certainly unobservable at low SNR. In addition, our flat noise and power spectrum assumptions makes $|Z(\omega)|$ effectively constant as a function of ω . For this reason, we shall write the ML estimator for term 4 without $|Z(\omega)|$.

$$\underset{\tilde{D}^1, \tilde{D}^2}{\text{MAX}} \frac{1}{2\pi} \int_{-B}^B x_1(\omega) x_2^*(\omega) \left[\frac{z_{1,2}(\omega) + z_{1,2}^*(\omega)}{2} \right] d\omega \quad (3-36)$$

Using Parseval's theorem, we can view Equation 3-36 as equivalent to maximizing with respect to $\tilde{D}^1$ and $\tilde{D}^2$ the integral of the product of the measured cross-correlation function with the expected cross-correlation function.

$$\underset{\tilde{D}_1, \tilde{D}_2}{\text{MAX}} \int_{-\infty}^{\infty} R(\tilde{t}) \bar{R}(\tilde{t}) d\tilde{t} \quad (3-37)$$

where $R(\tilde{t})$ is the measured cross-correlation function between sensors 1 and 2; and $\bar{R}(\tilde{t})$, the expected correlation function, is given by the following inverse Fourier transform.

$$\bar{R}(\tilde{t}) = \frac{1}{2\pi} \int_{-B}^B \frac{z_{1,2}(\omega) + z_{1,2}^*(\omega)}{2} e^{j\omega\tilde{t}} d\omega \quad (3-38)$$

Equation 3-38 shall now be referred to as the mismatch function.

We now have a general form for a practical approximate ML estimator for the unknown target time delay difference trajectory. In the next

section, we shall examine a more pliable target motion model and obtain a precise expression for the mismatch function.

3.4 THE MAXIMUM LIKELIHOOD ESTIMATOR FOR PROBLEM 2

From a dynamic and a practical point of view, it is not desirable to wait an entire τ seconds to complete our measurements and make decisions. Let us assume that we actually make M measurements of T seconds each over the τ second interval [i.e., $M T = \tau$]. We shall also assume that the unknown time delay difference trajectory is a piecewise linear function as described in Equation 2-6, with a linear segment for each of the M time segments. The unknown parameter vectors $\underline{D}^1$ and $\underline{D}^2$ now represent the time delay components d_k^1, d_k^2 at the end points of each of the $k = 1, 2, 3, \dots, M$ piecewise linear segments. In addition, the initial time delays d_0^1, d_0^2 may be considered part of $\underline{D}^1$ and $\underline{D}^2$. As an example, vector $\underline{D}^1$ components are shown below.

$$\underline{D}^{1T} = [d_0^1, d_1^1, d_2^1, \dots, d_M^1] \quad . \quad (3-39)$$

We will now obtain an approximation of Equation 3-37 with our new time delay trajectory model. Since the cross correlation is a linear process, the measured cross-correlation function $R(\tilde{t})$ for the entire interval τ can be viewed as the linear sum of the M subintervals measurements Δr_i

$$R(\tilde{t}) = \sum_{i=1}^M \Delta r_i(\tilde{t}) \quad . \quad (3-40)$$

For the same reason the expected cross-correlation function (mismatch function) can be viewed as the sum of each of the M subinterval expected cross-correlation functions $\overline{\Delta r_i}$.

$$\overline{R}(\tilde{t}) = \sum_{i=1}^M \overline{\Delta r_i}(\tilde{t}) \quad . \quad (3-41)$$

Substituting Equations 3-40 and 3-41 into 3-37 yields the following result:

$$\underset{\underline{D}^1, \underline{D}^2}{\text{MAX}} \sum_{i=1}^M \int_{-\infty}^{\infty} \Delta r_i(\tilde{t}) \overline{\Delta r_i}(\tilde{t}) d\tilde{t} + 2 \sum_{i=1}^M \sum_{j=i+1}^M \int_{-\infty}^{\infty} \Delta r_i(\tilde{t}) \Delta r_j(\tilde{t}) d\tilde{t} \quad . \quad (3-42)$$

Time segments Δr_i and Δr_j for $i \neq j$ are separated by at least T seconds. We shall assume that the contribution to the overall correlation function due to these terms is minimal and thus Equation 3-42 can be approximated by:

$$\underset{\underline{D}^1, \underline{D}^2}{\text{MAX}} \sum_{i=1}^M \int_{-\infty}^{\infty} \Delta r_i(\tilde{t}) \overline{\Delta r_i}(\tilde{t}) d\tilde{t} \quad . \quad (3-43)$$

Note that the contribution of $\int_{-\infty}^{\infty} \Delta r_{i-1}(\tilde{t}) \overline{\Delta r_i}(\tilde{t}) d\tilde{t}$ will certainly be greater than $\int_{-\infty}^{\infty} \Delta r_{i-2}(\tilde{t}) \overline{\Delta r_i}(\tilde{t}) d\tilde{t}$ which will be greater than $\int_{-\infty}^{\infty} \Delta r_{i-3}(\tilde{t}) \overline{\Delta r_i}(\tilde{t}) d\tilde{t}$ for most actual target trajectories. This assumption will be of use later when determining a suitable a priori information penalty function.

Equation 3-43 is attractive. The integral in Equation 3-43 can be replaced by a summation yielding a form conducive for batch techniques. In addition, the expected cross correlation or mismatch function $\overline{\Delta r}_i(\tilde{t})$ is now defined for a single piecewise linear segment. We shall take advantage of this development to derive the detail form of the mismatch function.

Let us now rewrite our receive signal model to reflect the above assumptions. For the first segment we have the following:

$$\begin{aligned} x_1(t) &= s(t + \phi_1(\underline{D}^1, t)) + \eta_1(t) \\ x_2(t) &= s(t + \phi_2(\underline{D}^2, t)) + \eta_2(t) \end{aligned} \quad (3-44)$$

where

$$\phi_1(\underline{D}^1, t) = d_0^1 + \Delta s_1^1 t, \quad 0 \leq t \leq T \quad (3-44A)$$

$$\phi_2(\underline{D}^2, t) = d_0^2 + \Delta s_1^2 t, \quad 0 \leq t \leq T \quad (3-44B)$$

and

$$\text{MAX}|\Delta s_1^1| \text{ and } \text{MAX}|\Delta s_1^2| \ll 1 \quad (3-44C)$$

Although not required, we shall assume for convenience that the initial received time delays d_0^1 and d_0^2 are known for the first subinterval. For all subsequent subintervals, the initial time delays d_i^1 and d_i^2 will be determined from the previous subinterval (i.e., $d_i^1 = d_{i-1}^1 + \Delta s_{i-1}^1 T$ and $d_i^2 = d_{i-1}^2 + \Delta s_{i-1}^2 T$). Since all the piecewise linear segment are identical, we shall proceed with our analysis using the first segment as a model.

The first step in calculating the precise form of the mismatch function is to calculate the precise form of the non-stationary Fourier coefficients $\tilde{\alpha}_1(1)$ and $\tilde{\alpha}_1(2)$ for a linear moving target.

Let us define:

$$\beta_1 = 1 + \Delta s_1^1 \approx 1 \quad (3-45A)$$

$$\beta_2 = 1 + \Delta s_1^2 \approx 1 \quad (3-45B)$$

and note that

$$\tilde{\phi}_1(\underline{D}^1, \gamma) = \frac{\gamma - d_0^1}{\beta_1} \quad (3-45C)$$

$$\tilde{\phi}_2(\underline{D}^2, \gamma) = \frac{\gamma - d_0^2}{\beta_2} \quad (3-45D)$$

$$T \equiv \tau \quad (3-45E)$$

We can now substitute Equations 3-45A through 3-45E into the expression for I_1 (Equation 3-19A) yielding the following result.

$$I_1 = \frac{1}{\beta_1 T} \int_{d_0^1}^{\beta_1 T + d_0^1} e^{j[\gamma \omega_k - \omega_\ell(\gamma - d_0^1)/\beta_1]} d\gamma \quad (3-46)$$

Evaluating the integral in Equation 3-46 and with the appropriate algebraic manipulation yields the following expression for I_1 .

$$I_1 = e^{j\omega_\ell d_0^1} \text{Sinc}[(\beta_1 \omega_\ell - \omega_k)(T/2)] e^{j(\beta_1 \omega_\ell - \omega_k)(T/2)} \quad (3-47)$$

Note $\text{Sinc}(x)$ is defined as $\sin(x)/x$.

Similar manipulation can be performed yielding an identical form for I_2 .

$$I_2 = e^{j\omega_\ell d_0^2} \text{Sinc}[(\beta_2 \omega_\ell - \omega_k)(T/2)] e^{j(\beta_2 \omega_\ell - \omega_k)(T/2)} \quad (3-48)$$

Equations 3-19 and 3-20 are still valid for the non-stationary Fourier coefficients $\tilde{\alpha}_1(1)$ and $\tilde{\alpha}_1(2)$, respectively, but now use the above expressions for I_1 and I_2 which are based on the linear time delay trajectory model.

The next step in our calculation is to derive a precise expression for the covariance matrix $E[\underline{c}_f(\omega_k), \underline{c}_f^{*T}(\omega_k)]$ (Equation 3-12). The (1,1) element in this matrix is given by Equation 3-25. If we substitute Equation 3-47 into Equation 3-25 and noting that $T \equiv \tau$ we have the following.

$$E[c_{f_1}(\omega_k), c_{f_1}^{*}(\omega_k)] = \frac{1}{4\pi} \int_{-B}^B G_s(\omega) \lim_{\tau \rightarrow \infty} (\text{Sinc}^2[(\beta_1 \omega - \omega_k) \frac{T}{2}]) d\omega$$

$$+ G_{n_1}(\omega_k)/T \quad . \quad (3-49)$$

Now let $\tilde{\omega} = \frac{\beta_1}{2} \omega$ and $d\tilde{\omega} = \frac{\beta_1}{2} d\omega$ and substitute into Equation 3-49

$$E[c_{f_1}(\omega_k), c_{f_1}^{*}(\omega_k)] = \frac{1}{2\pi\beta_1} \int_{-B\beta/2}^{B\beta/2} G_s(\frac{2\tilde{\omega}}{\beta_1}) \lim_{\tau \rightarrow \infty} (\text{Sinc}^2[(\tilde{\omega} - \frac{\omega_k}{2})T]) d\tilde{\omega}$$

$$+ \frac{G_{n_1}(\omega_k)}{T} \quad . \quad (3-50)$$

As we let T approach infinity, the limit of the $\text{Sinc}^2(\cdot)$ approaches a delta function² as shown below.

$$\lim_{\tau \rightarrow \infty} (\text{Sinc}^2[(\tilde{\omega} - \frac{\omega_k}{2})T]) \rightarrow \frac{2\pi}{T} \text{Delta}(\tilde{\omega} - \frac{\omega_k}{2}) \quad . \quad (3-51)$$

If we now assume that $G_s(\tilde{\omega})$ is relatively constant over the interval $[\frac{\omega_k}{2} - \frac{2\pi}{T}, \frac{\omega_k}{2} + \frac{2\pi}{T}]$, we can substitute Equation 3-51 into Equation 3-50 and evaluate the integral to yield the following.

$$E[c_{f_1}(\omega_k), c_{f_1}^*(\omega_k)] = \frac{1}{\beta_1 T} G_s\left(\frac{\omega_k}{\beta_1}\right) + \frac{1}{T} G_{n_1}(\omega_k), \quad -B \leq \omega_k \leq B \quad (3-52)$$

The above result is pleasing. It states that the doppler shift due to linear target motion results in either an expansion or contraction of the stationary target auto power spectrum with an appropriate scale factor to maintain constant total power. This result is in agreement with Knapp and Carter.²

Certainly for reasonable signal power spectrums passed through a prefilter, the slight level change $(1/\beta_1)$ in the received signal power spectrum will be completely unobservable with respect to other natural variations in received signal power levels or noise power levels. This result supports our decision to ignore term 2 in expression 3-35.

An equation similar to Equation 3-52 can be derived for the (2,2) element of the covariance matrix $E[\underline{C}_f(\omega_k), \underline{C}_f^{*T}(\omega_k)]$ which is given below.

$$E[c_{f_2}(\omega_k), c_{f_2}^*(\omega_k)] = \frac{1}{\beta_2 T} G_s\left(\frac{\omega_k}{\beta_2}\right) + \frac{1}{T} G_{n_2}(\omega_k), \quad -B \leq \omega_k \leq B \quad (3-53)$$

The critical (1,2) element of covariance matrix $E[\underline{C}_f(\omega_k), \underline{C}_f^{*T}(\omega_k)]$ is more difficult to analyze. Substituting Equations 3-47 and 3-48 into Equation 3-28 yields the following key integral.

$$E[c_{f_1}(\omega_k), c_{f_2}^*(\omega_k)] = \frac{1}{4\pi} \int_{-B}^B G_s(\omega) \lim_{T \rightarrow \infty} \text{Sinc}[(\beta_1 \omega - \omega_k) \frac{T}{2}] \cdot \text{Sinc}[(\beta_2 \omega - \omega_k) \frac{T}{2}] e^{j\omega(\Delta d_0 + \Delta B(T/2))} d\omega \quad (3-54)$$

Note $\Delta d_0 = d_0^1 - d_0^2$ and $\Delta B = B_1 - B_2$.

Ng¹³ developed an excellent direct analytical approximation to the integral of Equation 3-54 which is shown in slightly modified form in Appendix A. This direct analytical result shall be shown after the following intuitive approach which was initially used to approximate the integral of Equation 3-54.

The key concept comes from examining the method actually used in practice to calculate a correlation function (correlogram) over a period of time T . The actual technique is to average consecutive sums of m smaller correlograms each of duration Δt seconds. (In practice, Δt is determined to obtain a sufficient bandwidth time product; however, for reasonable bandwidth, Δt can be quite small.) Therefore we have:

$$m \Delta t = T \quad (3-55)$$

The integrand of Equation 3-54 represents a process observed over T seconds. Let us redefine the integrand of Equation 3-54 as the average of the sum of m processes observed consecutively every Δt seconds. We shall assume that for each process the time delay between sensors is constant for the given Δt second interval. This assumption implies that for each subinterval:

$$\beta_1 = \beta_2 = \text{for convenience } \text{MAX}[\beta_1, \beta_2] = \beta \quad (3-56)$$

Note: At this point there is no firm guidance on what to choose for the constant value of β although it is obvious that $\text{Min}(\beta_1, \beta_2) \leq \beta \leq \text{MAX}(\beta_1, \beta_2)$. We choose Equation 3-56 to be compatible with the result obtained by Ng.¹³ For our assumptions on β_1 and β_2 the above is a mute point.

However, the constant delay for each Δt subinterval is given by:

$$\text{delay}_i = \Delta d_0 + i \Delta t \Delta \beta, \quad i = 1, 2, 3 \dots m \quad (3-57)$$

This model represents the linear time delay difference trajectory by a staircase approximation. The validity of this model is dependent on $\beta_1 \approx \beta_2 \approx 1$ and sufficient bandwidth.

Also note that $G_S(\omega)$ in Equation 3-54 is defined by the following limit:

$$G_S(\omega) = \lim_{T \rightarrow \infty} T \alpha_1(\omega) \alpha_2^*(\omega) \quad . \quad (3-58A)$$

However, since our observation interval is now Δt seconds, $G_S(\omega)$ should be redefined as the following:

$$G_S(\omega) = \lim_{\Delta T \rightarrow \infty} \Delta \tau \alpha_1(\omega) \alpha_2^*(\omega) \quad . \quad (3-58B)$$

In order that the limiting process of the integral of Equation 3-54 remains unchanged, and noting Equation 3-55, we have:

$$G_S(\omega) \rightarrow \frac{G_S(\omega)}{m} \quad . \quad (3-58C)$$

With the above assumptions, the integral of Equation 3-54 can be rewritten as the following:

$$\frac{1}{4\pi} \int_{-B}^B \frac{G_S(\omega)}{m} \lim_{\Delta T \rightarrow \infty} \left(\text{Sinc}^2 \left[(\beta \omega - \omega_k) \frac{\Delta t}{2} \right] \frac{1}{m} \sum_{i=1}^m e^{j\omega(\Delta d_0 + i \Delta t \Delta \beta)} \right) d\omega \quad . \quad (3-59)$$

Note that the $\text{Sinc}^2(\cdot)$ functions in Equation 3-54 now have identical arguments which will remain unchanged for all m subintervals.

We shall now let $\tilde{\omega} = \beta \omega/2$ and $d\tilde{\omega} = \beta/2 d\omega$. Substituting into Equation 3-59 yields:

$$\frac{1}{2\pi\beta} \int_{-\beta B/2}^{\beta B/2} \frac{G_s(\frac{2\tilde{\omega}}{\beta})}{m} \lim_{\Delta t \rightarrow \infty} \left(\text{Sinc}^2[(\tilde{\omega} - \frac{\omega_k}{2})\Delta t] \frac{1}{m} \sum_{i=1}^m \right. \\ \left. e^{j(2\tilde{\omega}/\beta)(\Delta d_0 + i\Delta t\Delta\beta)} \right) d\tilde{\omega} \quad (3-60)$$

Now if we let Δt become relatively large, the exponential terms are not unduly affected but the $\text{Sinc}^2(\cdot)$ term approaches the following limit.

$$\lim_{\Delta t \rightarrow \infty} \text{Sinc}^2[(\tilde{\omega} - \frac{\omega_k}{2})\Delta t] \rightarrow \frac{2\pi}{\Delta t} \text{Delta}(\tilde{\omega} - \frac{\omega_k}{2}) \quad (3-61)$$

For the exponential terms, if we now let m become large, we have by the definition of an integral and Equation 3-55 the following limit:

$$\begin{aligned}
\lim_{m \rightarrow \infty} \frac{1}{m} \sum_{i=1}^m e^{j(2\tilde{\omega}/B)(\Delta d_0 + (i/m)\Delta B T)} &\rightarrow \int_0^1 e^{j(2\tilde{\omega}/B)(\Delta d_0 + x\Delta B T)} dx \\
&= \text{Sinc}\left[\frac{\tilde{\omega}\Delta B T}{B}\right] e^{j(2\tilde{\omega}/B)(\Delta d_0 + (\Delta B/2)T)} \quad (3-62)
\end{aligned}$$

Substituting Equations 3-61 and 3-62 into Equation 3-60 we have the following result.

$$\frac{1}{BT} \int_{-BB}^{BB} G_s\left(\frac{2\tilde{\omega}}{B}\right) \text{Delta}\left(\tilde{\omega} - \frac{\omega_k}{2}\right) \text{Sinc}\left[\frac{\tilde{\omega}\Delta B T}{B}\right] e^{j(2\tilde{\omega}/B)(\Delta d_0 + (\Delta B/2)T)} d\tilde{\omega} \quad (3-63)$$

If we now assume that $G_s(\tilde{\omega})$ is relatively constant over the interval $[\frac{\omega_k}{2} - \frac{2\pi}{\Delta T}, \frac{\omega_k}{2} + \frac{2\pi}{\Delta T}]$, Equation 3-63 evaluates to:

$$\begin{aligned}
E[c_{f_1}(\omega_k), c_{f_2}^*(\omega_k)] &= \frac{1}{BT} G_s\left(\frac{\omega_k}{B}\right) \text{Sinc}\left[\frac{\omega_k \Delta B T}{2B}\right] e^{j(\omega_k/B)(\Delta d_0 + (\Delta B/2)T)}, \\
- BB &\leq \omega_k \leq BB \quad (3-64)
\end{aligned}$$

The direct evaluation of Equation 3-54 by Ng¹³ yields the identical result with similar assumptions. (See Appendix A.)

Equation 3-64 yields intuitively pleasing results. Similar to the case for the received auto power spectrum, Equation 3-64 predicts a frequency expansion or contraction with an appropriate scale factor to maintain constant total power. This effect is primarily due to doppler and is inversely proportional to $\beta = \text{Max}(\beta_1, \beta_2)$. Equation 3-64 also predicts a loss of coherency primarily due to tangential target motion represented by $\Delta\beta$. Finally, Equation 3-64 predicts an effective time delay shift to the midpoint of the observation interval represented by $\Delta d_0 + \frac{\Delta\beta}{2} T$. The $\text{Sinc}(\cdot)$ factor in Equation 3-64 is the significant difference between our result and the result obtained by Knapp and Carter².

Similar to past analysis, we have that $\beta \approx 1$ and thus Equation 3-64 can be rewritten to a much more convenient form.

$$E[c_{f_1}(\omega_k), c_{f_2}^*(\omega_k)] = \frac{G_s(\omega_k)}{T} \text{Sinc}[\omega_k \Delta\beta T/2] e^{j\omega_k(\Delta d_0 + (\Delta\beta/2)T)} \quad (3-65)$$

Assuming Δd_0 is known, Equation 3-65 is only dependent on one parameter, $\Delta\beta$, or equivalently, $\Delta\beta T$.

Equation 3-65 via Equation 3-31 can be substituted into Equation 3-38 to yield an expression for the mismatch function.

$$\bar{R}(\tilde{t}) = \frac{1}{2\pi} \int_{-B}^B G_s(\omega) \text{Sinc}\left(\omega \frac{\Delta B}{2} T\right) \cos\left[\omega(\tilde{t} + [\Delta d_0 + \frac{\Delta B}{2} T])\right] d\omega \quad (3-66)$$

We shall now make use of the flat band-limited assumptions on $G_s(\omega)$ given in Equation 2-3 and repeated below for convenience.

$$G_s(f) = \begin{cases} \frac{A_s}{2(f_2 - f_1)}, & f_1 \leq f \leq f_2 \text{ or } -f_2 \leq f \leq -f_1 \\ 0, & \text{elsewhere.} \end{cases} \quad (3-67)$$

Let $\omega_1 = 2\pi f_1$ and $\omega_2 = 2\pi f_2$

$$B\omega = \omega_2 - \omega_1 \text{ and } \omega_c = \frac{\omega_1 + \omega_2}{2}$$

and for now assume that the total received signal power A_s is 1.

Substituting the above assumptions on $G_s(\omega)$ into Equation 3-66 yields the following after some algebra.

$$\bar{R}(\tilde{t}) = \frac{1}{B\omega} \int_{\omega_c - B\omega/2}^{\omega_c + B\omega/2} \text{Sinc}\left(\omega \frac{\Delta B}{2} T\right) \cos\left[\omega(\tilde{t} + (\Delta d_0 + \frac{\Delta B}{2} T))\right] d\omega \quad (3-68)$$

This integral is analytically approximated in Appendix 3, yielding an analytical equation for the mismatch function which depends on the following:

1. $\tilde{t}$: the correlation lag point.
2. Δd_0 : The initial time delay difference between the two sensors.
3. $\Delta \delta T$: The total change in the time delay difference between the two sensors during the observation time interval.

Since Δd_0 , T are assumed known, then the only quantity to maximize Equation 3-37 is $\Delta \delta$ or, more conveniently, $\Delta \delta T$.

At this point we shall summarize what we have derived. If we assume that the unknown signal time delay difference trajectory is linear over an observation interval of T seconds with a known initial time delay difference, we can determine an approximate ML estimate for the unknown slope of the time delay difference trajectory by maximizing the product between the measured cross correlation function and the expected cross-correlation (mismatch) function with respect to the unknown slope. Also, we have determined that an approximate ML estimator for an unknown piecewise linear time delay difference trajectory (Problem 2) is to maximize the sum of the individual but continuous linear segment correlations with respect to their unknown slopes. In addition, we have derived an approximate analytical equation for the expected cross-correlation function (mismatch function) assuming a

linear time delay difference trajectory, relatively slow target motion, and sufficient signal bandwidth.

3.5 LEAST MEAN SQUARE FIT APPROXIMATION

At this point it is productive to reformulate our estimator in a similar but more practical form. Our input measurements over an interval of time T will take on the form of normalized noisy discrete cross-correlation lag points designated by the vector $\tilde{\mathbf{R}}$, where the components $\tilde{r}_i$ of the vector $\tilde{\mathbf{R}}$ are measurements of the cross correlation at discrete times $\tilde{t}_i = i \text{ tau}$, $i = -N, -N + 1, \dots, 0, \dots, N$ and tau is a constant time delay. The normalization factor shall be an estimate of the total received power at either sensor 1 or 2 (signal power (A_s) + noise power (N_s)). Therefore at $\tilde{t}_0 = 0$,

$$\tilde{r}_0 \approx A_s / (A_s + N_s) \quad (3-69)$$

and for stationary conditions (no target motion) we have

$$\lim_{T \rightarrow \infty} \tilde{r}_0 = A_s / (A_s + N_s) \quad (3-69A)$$

In addition, we shall at least initially assume that the noise component of the lag measurements $\tilde{r}_i$'s is correlated and has a known constant correlation matrix $\mathbf{W}$. This assumption corresponds to filtering and over-sampling our original noise processes $\eta_1(t)$ and $\eta_2(t)$ (Chapter 2). We shall show that the correlated noise problem can be transformed into an independent noise problem.

Our new estimator, shown below, will minimize the squared error between the measured cross-correlation lag data $\tilde{\mathbf{R}}$ and a normalized expected cross-correlation lag data $A \cdot \tilde{\mathbf{C}}$ with respect to the unknown time delay difference slope parameter $\Delta\beta T$ and the normalization factor A_n .

$$\min_{\Delta\beta T, A} [\tilde{\mathbf{R}} - A_n \tilde{\mathbf{C}}]^T \mathbf{W}^{-1} [\tilde{\mathbf{R}} - A_n \tilde{\mathbf{C}}] \quad (3-70)$$

where the components $\tilde{C}_i$ of the vector $\tilde{\mathbf{C}}$ are the mismatch Equation 3-68 evaluated at times $\tilde{t}_i = i \text{ tau}$, $i = -N, -N + 1, \dots, 0, \dots N$.

Note that the unknown normalization factor A_n is an estimate of the signal power divided by the total receive power.

$$\hat{A}_n = \frac{A_s}{A_s + N_s} \quad (3-71)$$

If $\mathbf{W}$ is the identity matrix, our measured data is uncorrelated and we have a straightforward least squares problem. If $\mathbf{W}$ is, as we assume, a constant symmetric positive definite matrix, we can transform Equation 3-70 into a straightforward least squares problem. This technique is called pre-whitening and is described by Whalen¹⁴ and briefly outlined below.

If W is positive definite, we can always¹⁵ factor it into lower (L) and upper (U) triangular matrices which are transposes of each other ($L^T = U$).^{*} These triangular matrices can be easily inverted to yield the following equation

$$W^{-1} = L^{-1T} L^{-1} . \quad (3-72)$$

If Equation 3-72 is substituted into Equation 3-70 along with the following substitutions,

$$\underline{R} = L^{-1} \tilde{\underline{R}} \quad (3-73)$$

$$\underline{C} = L^{-1} \tilde{\underline{C}} , \quad (3-74)$$

we arrive at the following simple least squares problem.

$$\underset{\Delta B^T, A_n}{\text{MIN}} [\underline{R} - A_n \underline{C}]^T [\underline{R} - A_n \underline{C}] . \quad (3-75)$$

^{*} The factorization process is a form of a Choleski decomposition.¹⁵ In practice, a small positive factor must be added to the main diagonal terms in order to ensure numerical positive definiteness and proper factorization.

The effect of Equation 3-73 is to uncorrelate or whiten the noise components of the measurement vector $\underline{\tilde{R}}$. Since the whitening process of Equation 3-73 affects the underlying measured signal cross-correlation, Equation 3-74 corrects the expected signal cross-correlation model.

From this point onwards, we shall talk to Equation 3-75 and its assumed assumption of uncorrelated noise. If correlated noise is present, we have the option of implementing Equations 3-73 and 3-74 to transform the problem to the case of uncorrelated noise.

To minimize Equation 3-75 with respect to A_n , we shall take the partial derivative of Equation 3-69 with respect to A_n and set the result equal to 0. The resulting estimate for A_n is given below.

$$\hat{A}_n = \frac{\underline{\tilde{C}}^T \underline{\tilde{R}}}{\underline{\tilde{C}}^T \underline{\tilde{C}}} \quad (3-76)$$

Substituting Equation 3-76 into Equation 3-75 yields after expansion:

$$\underset{\Delta BT}{\text{MIN}} \quad \underline{\tilde{R}}^T \underline{\tilde{R}} - \frac{(\underline{\tilde{C}}^T \underline{\tilde{R}})^2}{\underline{\tilde{C}}^T \underline{\tilde{C}}} \quad (3-77)$$

Only the second term in Equation 3-77 depends on the parameter ΔBT . Unfortunately, the solution is not equivalent to maximizing the lag product of the matching function $\underline{\tilde{C}}$ with the measured cross-correlation data $\underline{\tilde{R}}$. This is true because the dot product $\underline{\tilde{C}}^T \underline{\tilde{C}}$ is not constant for all choices of ΔBT .

However, we can modify the definition of our matching function to the following.

$$\text{Matching function} = \frac{\tilde{A}_n \underline{C}}{\sqrt{\underline{C}^T \underline{C}}} \quad (3-78)$$

where

$$\tilde{A}_n = A_n \sqrt{\underline{C}^T \underline{C}} \quad (3-78A)$$

Equation 3-75 can be rewritten to the following form.

$$\text{MIN}_{\Delta BT, \tilde{A}_n} [\underline{R} - \tilde{A}_n \underline{C} / \sqrt{\underline{C}^T \underline{C}}]^T [\underline{R} - \tilde{A}_n \underline{C} / \sqrt{\underline{C}^T \underline{C}}] \quad (3-79)$$

If we take the derivative of Equation 3-79 and set the result equal to zero, we obtain the following expression for the estimate of $\tilde{A}_n$.

$$\hat{\tilde{A}}_n = \frac{\underline{C}^T \underline{R}}{\sqrt{\underline{C}^T \underline{C}}} \quad (3-80)$$

Substituting Equation 3-80 into Equation 3-79 and expanding yields the following result.

$$\text{MIN}_{\Delta BT} \underline{R}^T \underline{R} - \frac{(\underline{C}^T \underline{R})^2}{\underline{C}^T \underline{C}} \quad (3-81)$$

This is the same expression as Equation 3-77, but now the second term can be rewritten as the following.

$$\underset{\Delta \underline{B} \underline{T}}{\text{MAX}} \left[\frac{\underline{C}^T \underline{R}}{\sqrt{\underline{C}^T \underline{C}}} \right]^2 \quad (3-82)$$

Maximizing Equation 3-82 is equivalent* to maximizing the following expression.

$$\underset{\Delta \underline{B} \underline{T}}{\text{MAX}} \frac{\underline{C}^T \underline{R}}{\sqrt{\underline{C}^T \underline{C}}} \quad (3-83)$$

Equation 3-83 is exactly what we want; maximizing the lag product between normalized matching function and the normalized measured cross-correlation.

This result can easily be extended to the case of a piecewise linear time delay difference trajectory over a block of M data segments. With Equation 3-43 in mind, define the measured cross-correlation vector for time segment j as $\underline{R}_j$. $\underline{\Delta \underline{B} \underline{T}}$ now becomes a vector with a component for each unknown piecewise linear slope. The approximate ML estimator can now be written as:

$$\underset{\underline{\Delta \underline{B} \underline{T}}}{\text{MAX}} \sum_{j=1}^M \frac{\underline{R}_j^T \underline{C}}{\sqrt{\underline{C}^T \underline{C}}} \quad (3-84)$$

Note that $\underline{C}$ has an implied dependency on the time segment j since each time segment has a unique slope parameter $\Delta \beta_j$; i.e., $\underline{C}(\Delta \beta_j T)$.

* Note the above expression disregards negative maximums, but these are not of interest.

Equation 3-84 can be rewritten in an equivalent form.

$$\underset{\Delta \underline{\beta}^T}{\text{MAX}} \sum_{j=1}^M \hat{\underline{A}}_{n_j} = \underset{\Delta \underline{\beta}^T}{\text{MAX}} \sum_{j=1}^M \hat{\underline{A}}_{n_j} \sqrt{\underline{C}^T \underline{C}} \quad (3-85)$$

where $\hat{\underline{A}}_{n_j}$ is the estimated power normalization factor for the j th segment. With Equation 3-71 in mind, the SNR estimate for the j th interval is given by:

$$\hat{\text{SNR}}_j = \frac{\underset{\Delta \underline{\beta}^T}{\text{MAX}} \left[\hat{\underline{A}}_{n_j} / \sqrt{\underline{C}^T \underline{C}} \right]}{1 - \underset{\Delta \underline{\beta}^T}{\text{MAX}} \left[\hat{\underline{A}}_{n_j} / \sqrt{\underline{C}^T \underline{C}} \right]} \quad (3-86)$$

The average SNR estimate for the entire block of data is given by:

$$\hat{\text{SNR}}_{\text{AVG}} = \frac{1}{M} \sum_{j=1}^M \frac{\underset{\Delta \underline{\beta}^T}{\text{MAX}} \left[\hat{\underline{A}}_{n_j} / \sqrt{\underline{C}^T \underline{C}} \right]}{1 - \underset{\Delta \underline{\beta}^T}{\text{MAX}} \left[\hat{\underline{A}}_{n_j} / \sqrt{\underline{C}^T \underline{C}} \right]} \quad (3-87)$$

In summary, Equations 3-84, 3-85, and 3-87 are the key equations for the practical ML estimation portion of the MAP estimator (Equation 3-3). Equations 3-84, 3-85, and 3-87 assume a piecewise linear target time delay difference trajectory.

3.6 SPECTRAL PARAMETER ESTIMATION

In the previous sections we have assumed that the signal power spectrum $G_s(\omega)$ can be modeled by a flat bandpass spectrum given by Equation 2-3 and repeated below for convenience.

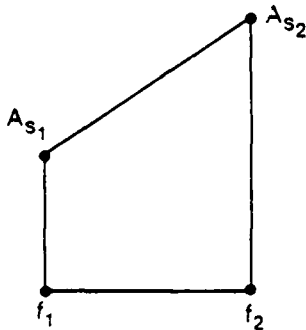
$$G_s(\omega) = \frac{A_s}{2[f_2 - f_1]}, \quad f_1 \leq f \leq f_2 ; -f_2 \leq f \leq -f_1$$

$$0, \quad \text{otherwise.} \quad (3-88)$$

Note $\omega = 2\pi f$.

It would not be a serious violation of previous work if we let the signal power spectrum $G_s(\omega)$ have one more degree of freedom by allowing an unknown spectral slope parameter γ .

Let us model the signal power spectrum $G_s(\omega)$ by a linear model shown in Figure 3-1 for a one-sided spectrum.



$$\text{Total power} = P_t = \frac{A_{s2} + A_{s1}}{2} [f_2 - f_1] \quad (3-89)$$

Figure 3-1. Linear Model for the Signal Power Spectrum

$$\text{Let us define } f_c = \frac{f_2 + f_1}{2}. \quad (3-90)$$

Then the one-sided power spectrum $G_s(\cdot)$ can be defined for any frequency as the following.

$$G_s(f) = \frac{A_{s_2} + A_{s_1}}{2} + \frac{A_{s_2} - A_{s_1}}{f_2 - f_1} [f - f_c], \quad f_1 \leq f \leq f_2$$

$$0, \quad \text{otherwise.} \quad (3-91)$$

If we normalize $G_s(f)$ by the total power given in Equation 3-89, we have

$$G_s(f) = G_s(f)/P_t = \frac{1}{f_2 - f_1} + \frac{A_{s_2} - A_{s_1}}{A_{s_2} + A_{s_1}} 2 \left(\frac{f - f_c}{(f_2 - f_1)^2} \right). \quad (3-92)$$

Now if we let

$$\omega = 2\pi f, \quad B\omega = \omega_2 - \omega_1 \quad \text{and} \quad \gamma = (A_{s_2} - A_{s_1})/(A_{s_2} + A_{s_1}), \quad (3-92A)$$

we obtain the following result.

$$G_s(\omega) = \begin{cases} \frac{2\pi}{B\omega} \left[1 + \gamma \frac{\omega - \omega_c}{B\omega/2} \right], & \omega_1 \leq \omega \leq \omega_2 \\ 0, & \text{otherwise.} \end{cases} \quad (3-93)$$

If Equation 3-93 is substituted into Equation 3-66, the equivalent of Equation 3-68 becomes the following.

$$\begin{aligned}
R(\tilde{t}_i) = & \frac{1}{Bw} \int_{\omega_c - Bw/2}^{\omega_c + Bw/2} \cos(\omega[\tilde{t}_i + \Delta d_0 + \frac{\Delta B}{2} T]) \text{Sinc}(\omega \frac{\Delta B}{2} T) d\omega \\
& + \gamma \frac{2}{Bw} \left[\frac{2}{Bw \Delta B T} \int_{\omega_c - Bw/2}^{\omega_c + Bw/2} \cos(\omega[\tilde{t}_i + \Delta d_0 + \frac{\Delta B T}{2}]) \sin(\omega \frac{\Delta B T}{2}) d\omega \right. \\
& \left. - \frac{\omega_c}{Bw} \int_{\omega_c - Bw/2}^{\omega_c + Bw/2} \cos(\omega[\tilde{t}_i + \Delta d_0 + \frac{\Delta B T}{2}]) \text{sinc}(\omega \frac{\Delta B T}{2}) d\omega \right] \quad (3-94)
\end{aligned}$$

The first and third integrals are effectively equivalent to Equation 3-68. All integrals are analytically or approximately analytically evaluated in Appendix B. Equation 3-94 yields a new form for the mismatch function represented by the vector $\underline{C}$. If we designate the solution of the first integral at lag time $\tilde{t}_i$ as u_i and the solution of the second and third integrals at lag time $\tilde{t}_i$ as v_i we can write the following expression for the new components c_i of the vector $\underline{C}$.

$$c_i = u_i + \gamma v_i \quad (3-95)$$

or

$$\underline{C} = \underline{U} + \gamma \underline{V} \quad (3-95A)$$

Now we substitute Equation 3-95A into Equation 3-83 to obtain a revised form of the ML estimator.

$$\text{MAX}_{\Delta B T} (\underline{U} + \gamma \underline{V})^T R^T / ((\underline{U} + \gamma \underline{V})^T (\underline{U} + \gamma \underline{V}))^{1/2} \quad (3-96)$$

Taking the derivative of Equation 3-96 with respect to γ and setting the result equal to zero yields the following equation for the estimate $\hat{\gamma}$.

$$\hat{\gamma} = \frac{\tilde{a} \tilde{b} - \tilde{c} \tilde{d}}{\tilde{a} \tilde{c} - \tilde{b} \tilde{e}} \quad (3-97)$$

where

$$\tilde{a} = \underline{U}^T \underline{V} \quad (3-97A)$$

$$\tilde{b} = \underline{U}^T \underline{R} \quad (3-97B)$$

$$\tilde{c} = \underline{V}^T \underline{R} \quad (3-97C)$$

$$\tilde{d} = \underline{U}^T \underline{U} \quad (3-97D)$$

$$\tilde{e} = \underline{V}^T \underline{V} \quad (3-97E)$$

At this point there are two directions which can now be developed. The first is to substitute the optimum expression for $\hat{\gamma}$ back into Equation 3-96 and to simplify. After extensive algebra* the following expression is obtained.

$$\text{MAX}_{\Delta BT} \left[\frac{\tilde{b}^2 \tilde{e} + \tilde{c}^2 \tilde{d} - 2 \tilde{a} \tilde{b} \tilde{c}}{\tilde{d} \tilde{e} - \tilde{a}^2} \right]^{1/2} \quad (3-98)$$

* The algebra was actually performed by a symbolic manipulation program.

Equation 3-98 can be maximized numerically for $\Delta\hat{B}T$, yielding an estimate for $\Delta\hat{B}T$ and also indirectly via Equation 3-97 an estimate for $\hat{\gamma}$. However, Equation 3-98 is significantly more complicated than Equation 3-83 and contains implicitly an estimate of the rather unstable slope parameter γ .

A second more practical technique is the following.

Assume that the slope is zero; i.e., $\gamma = 0$. This assumption via Equation 3-96 yields our original ML estimator given in Equation 3-83. Note that under the single target assumption the maximum of Equation 3-83 will occur, on average, when the expected peak of the measured cross-correlation $\underline{R}$ lines up with the symmetrical peak of the expected cross-correlation vector $\underline{C}$. The effect of the spectral slope parameter (γ) is to broaden or narrow the cross-correlation function. It does not affect the location of the peak. Therefore, Equation 3-96 can be optimized independently of the slope parameter (γ). Once this is completed, Equation 3-97 can be used to obtain an estimate of the spectral slope parameter $\hat{\gamma}$.

3.7 DERIVATION OF THE A PRIORI KNOWLEDGE PENALTY FUNCTION

We have obtained an expression for the ML portion of the MAP Equation 3-3. We shall now obtain an expression for the a priori knowledge portion of Equation 3-3.

In essence, the a priori knowledge is what ties Problem 2 and Problem 1 together. Our ML estimator assumes a piecewise linear time delay difference trajectory and yields an estimate of the time delay difference for each piecewise linear segment (Problem 2). However, we assume a priori in Problem 1 (Equations 3-4, 3-5) that the actual time delay difference trajectory is quadratic; i.e.,

$$\Delta d_i = \Delta \tilde{d}_2 t_i^2 + \Delta \tilde{d}_1 t_i + \Delta \tilde{d}_0 \quad (3-99)$$

Note that estimates of $\Delta \hat{d}_i$ from the ML estimator (Equation 3-84) are given by

$$\Delta \hat{d}_i = \Delta \hat{d}_0 + \sum_{j=1}^i \Delta \hat{\beta}_j^T \quad (3-100)$$

where $\Delta \hat{d}_0$ is the estimated or given time delay difference at the beginning of the trajectory and $\Delta \hat{\beta}^T$ are the estimates obtained from Equation 3-84.

We shall assume that the difference between the ML estimates $\Delta \hat{d}_i$ and the truth Δd_i are independent Gaussian random variables with equal variances* σ^2 . Therefore, the a priori density function $P_{\Delta \hat{D}}(\Delta \hat{D})$ is given by the following expression.

* We have assumed that all the components of $\Delta \hat{D}$ are equal variance and are uncorrelated. Neither assumption is entirely true, so theoretically there is a covariance matrix Q relating the components of $\Delta \hat{D}_i$. Equations 3-102, 3-103, and 3-104 should therefore be modified by the components of the Q matrix. However, the ML estimates $\Delta \hat{D}$ and the a priori information interact in a complex manner which obscures this relationship. Simulations have shown small gain, if any, for weighting matrixes Q other than the Identity.

$$P_{\underline{\Delta D}}(\underline{\hat{\Delta D}}) = \prod_{i=1}^M \frac{1}{\sqrt{2\pi}\sigma} e^{-[\hat{\Delta d}_i - (\Delta \tilde{d}_2 \tilde{t}_i^2 + \Delta \tilde{d}_1 \tilde{t}_i + \Delta \tilde{d}_0)]^2} \quad (3-101)$$

and

$$\ln P_{\underline{\Delta D}}(\underline{\hat{\Delta D}}) = M \ln\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \frac{1}{\sigma^2} \sum_{i=1}^M [\hat{\Delta d}_i - (\Delta \tilde{d}_2 \tilde{t}_i^2 + \Delta \tilde{d}_1 \tilde{t}_i + \Delta \tilde{d}_0)]^2 \quad (3-102)$$

Only term 2 of Equation 3-102 is dependent on $\underline{\hat{\Delta D}}$ or equivalently $\underline{\Delta \hat{B}T}$. Therefore, finally, the MAP Equation 3-3 can be written as a combination of Equations 3-102 and 3-84.

$$\underset{\underline{\Delta \hat{B}T}}{\text{MAX}} \sum_{j=1}^M \frac{R_j^T C}{\sqrt{C^T} C} - K \sum_{j=1}^M [\hat{\Delta d}_j - (\Delta \tilde{d}_2 \tilde{t}_j^2 + \Delta \tilde{d}_1 \tilde{t}_j + \Delta \tilde{d}_0)]^2 \quad (3-103)$$

We can rewrite the second summation in Equation 3-103 in the following standard matrix format.

$$\sum_{j=1}^M [\hat{\Delta d}_j - (\Delta \tilde{d}_2 \tilde{t}_j^2 + \Delta \tilde{d}_1 \tilde{t}_j + \Delta \tilde{d}_0)]^2 = [\underline{\hat{\Delta D}} - H \underline{\hat{\Delta \hat{D}}}]^T [\underline{\hat{\Delta D}} - H \underline{\hat{\Delta \hat{D}}}] \quad (3-103A)$$

where from the standard solution of least square, $\underline{\hat{\Delta \hat{D}}}$ is given by

$$\underline{\hat{\Delta \hat{D}}} = (H^T H)^{-1} H^T \underline{\hat{\Delta D}} \quad (3-103B)$$

Using Equations 3-103A and 3-103B, we can rewrite Equation 3-103 as the following.

$$\underset{\underline{\Delta \hat{B}T}}{\text{MAX}} \sum_{j=1}^M \frac{R_j^T C}{\sqrt{C^T} C} - K(\underline{\hat{\Delta D}}^T [I - H(H^T H)^{-1} H^T] \underline{\hat{\Delta D}}) \quad (3-104)$$

where the H matrix in our case is:

* The initial time delay difference Δd_0 can be added to the vector $\underline{\Delta \hat{B}T}$ as the set of unknown parameters to be estimated.

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 1 & .1 & 1 \\ 1 & 2 & 4 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 1 & M & M^2 \end{bmatrix} \quad (3-104A)$$

and I is the identity matrix.

Also note that

$$\begin{bmatrix} \hat{\Delta d}_2 \\ \hat{\Delta d}_1 \\ \hat{\Delta d}_0 \end{bmatrix} = (H^T H)^{-1} H^T \underline{\hat{\Delta d}} \quad (3-105)$$

and an estimate of the target time delay difference rate is given by^{*}:

$$\hat{\Delta d}_i = 2 \hat{\tilde{d}}_2 \tilde{t}_i + \hat{\tilde{d}}_1 \quad (3-106)$$

and the estimate of the target time delay acceleration is given by^{*}:

$$\hat{\Delta d} = 2 \hat{\tilde{d}}_2 \quad (3-107)$$

* The time delay difference rate and acceleration are referenced to the measurement averaging time T seconds.

The parameter K represents the functional relationship between the ML and the a priori information. Unfortunately, the approximate derivation of the ML term has obscured the functional form of K . Simulation will be required to determine the functional form and appropriate values for K .

A key concept to understand is that Equation 3-103 or 3-104 is not equivalent to calculating a set of ML estimates $\hat{\underline{D}}$ and then performing a quadratic LMSF to refine the estimate. The update-by-update interaction between the ML estimator and the a priori information yields a more accurate estimate and ensures stability at low SNR.

As stated earlier, the normalized measured data vector $\underline{R}_j$ will have a normalization factor of $A_s/(A_s + N_s)$ or signal power divided by signal plus noise power. Therefore, at low SNR [$N_s \gg A_s$], the ML estimate will be proportional to SNR (A_s/N_s). At high SNR ($A_s \gg N_s$), the ML estimate will approach a constant. The ML estimator can be modified to maintain an approximately linear relationship with SNR (A_s/N_s) by the following development.

Let

$$\hat{\underline{W}} = \underset{\underline{\Delta B T}}{\text{MAX}} \left[\sum_{j=1}^M \underline{R}_j^T \underline{C} / \sqrt{\underline{C}^T \underline{C}} \right] \quad (3-108)$$

which is equivalent to the following by Equation 3-37:

$$\hat{W} = \underset{\Delta \delta T}{\text{MAX}} \sum_{j=1}^M \hat{A}_j \sqrt{\underline{C}^T \underline{C}} \quad (3-109)$$

The average SNR estimate for the block of data is given by Equation 3-87 which, after some algebra, is given by:

$$M \hat{\text{SNR}}_{\text{AVG}} = \frac{M \hat{W} / \sqrt{\underline{C}^T \underline{C}}}{M - \hat{W} / \sqrt{\underline{C}^T \underline{C}}} \quad (3-110)$$

The right-hand side of Equation 3-110 is proportional to the average SNR and increases as $\hat{W}$ increases ($0 < \hat{W} < M$). Therefore, the right-hand side of Equation 3-110 can be used in place of the ML term in Equations 3-103 or 3-104 for the MAP estimator.

Wolcin⁶ derived an a priori knowledge penalty function for an assumed piecewise linear frequency trajectory which may be appropriate for our work. His assumptions are the following:

1. The densities $P_0(\Delta \hat{d}_0)$ and $P_1(\Delta \hat{d}_1 / \Delta \hat{d}_0)$ are uniform over the space of allowable choices of $\Delta \hat{d}_0$ and $\Delta \hat{d}_1$ (i.e., no start time delay difference or time delay difference rate should be preferred over any other).
2. The expected value of $\Delta \hat{d}_i$, given $\Delta \hat{d}_{i-1}$ and $\Delta \hat{d}_{i-2}$, can be obtained by linear extrapolation. (The time delay difference function is smooth and can be approximated by a piecewise linear function.)

3. The sequence $\{\hat{\Delta d}_0, \hat{\Delta d}_1 \dots \hat{\Delta d}_N\}$ is second-order Gauss-Markoff (note discussion in Section 3.5).

The resulting a priori knowledge penalty function is the following.

$$\frac{1}{\sigma^2} \sum_{j=0}^{N-2} (\hat{\Delta d}_{j+2} - 2 \hat{\Delta d}_{j+1} + \hat{\Delta d}_j)^2 \quad (3-111)$$

$$\text{where } \text{Var}(\hat{\Delta d}_i / \hat{\Delta d}_{i-1}, \hat{\Delta d}_{i-2}) = \sigma^2 \quad (3-111A)$$

Equation 3-111A can be substituted into Equation 3-103 or 3-104 for the a priori information term.

3.8 SUMMARY

In Chapter 3 we have derived an approximate MAP estimator (Equation 3-104) for a target time difference trajectory as observed between two-point sensors. We also have derived an SNR estimator (Equation 3-87) and an estimator for a spectral slope parameter (Equation 3-97) for the target signal.

Using the LMSF penalty function, we also have estimates for the target time delay difference rate and acceleration (Equations 3-106 and 3-107, respectively).

In the following chapters we shall examine techniques and problems in implementing the above estimators.

CHAPTER 4

PRACTICAL IMPLEMENTATION OF THE MAP ESTIMATORS

4.1 INTRODUCTION

In Chapter 3 we have derived an approximate MAP estimator for the unknown target time delay difference trajectory due to a moving target over an extended period of time τ . Our total measurement consists of a block of M consecutive measurements, each of duration T seconds. (Therefore, we have that $M \cdot T = \tau$.) Each of the M individual measurements consists of the cross-correlation of two-point sensors averaged for T seconds and evaluated at $2N+1$ discrete lag delay times centered about zero lag delay.

Equation 3-104 nominally represents the MAP estimator for our described input measurements. Effectively, the optimum time delay difference trajectory is that which maximizes Equation 3-104 or equivalent over the block of input lag point measurements.

Two general techniques for solving the above maximization problem become immediately apparent. The first is to discretize and limit the allowable time delay difference space ΔD (note ΔD depends on ΔST via Equation 3-100). With only a finite number of allowable time delay difference trajectories, a solution for a maximum is obtainable. This approach we shall call the dynamic programming approach.

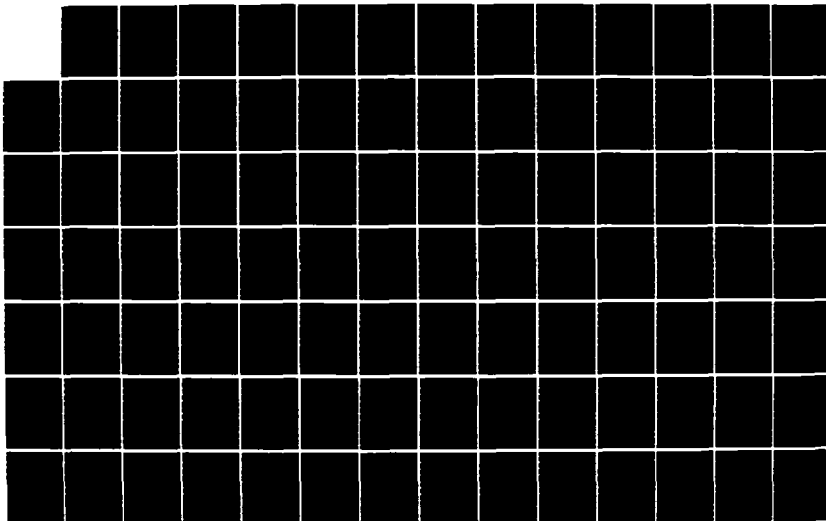
AD-A173 396

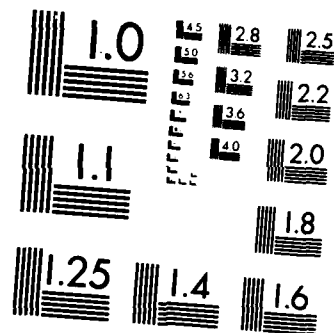
EFFICIENT RECURSIVE BATCH TIME DELAY DIFFERENCE
ESTIMATION IN THE PRESENC. (U) NAVAL UNDERWATER SYSTEMS
CENTER NEW LONDON CT NEW LONDON LAB. R A LATOURETTE
17 JUL 86 NUSC-TR-7743 F/G 17/1

2/4

UNCLASSIFIED

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

The second approach is to calculate the gradient vector of the MAP equation (Equation 3-104 or equivalent) with respect to the unknown time delay difference parameters $\Delta B T$. Using the gradient vector, a numerical search can be used to find the maximum. This approach we shall call the gradient search approach.

In the subsequent sections we shall discuss and outline the above two approaches as well as other related suboptimal approaches.

4.2 DYNAMIC PROGRAMMING APPROACH

One solution to finding the maximum of the MAP equation from a finite set of time delay difference trajectories ΔD is to simply try all of them and keep the one which maximizes the MAP equation. This approach is not practical for reasonable block sizes and reasonable number of allowable time delay difference points. Wolcin's^{7,8} solution of a similar problem for maximizing a MAP function over a discretized space of complex FFT data is the basis for our approach. Wolcin in his work used a dynamic programming technique developed by Bellman.¹⁶ Wolcin's original algorithm was modified by LaTourette, Greineder, and Wolcin⁹ into a recursive batch technique for frequency estimation which parallels our development for time delay estimation.

4.2.1 Basic Concept

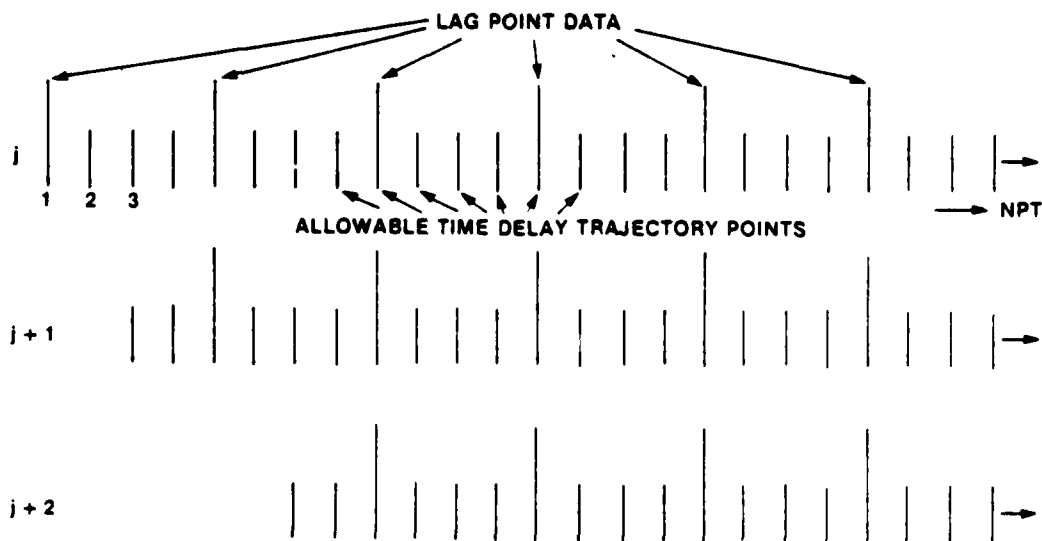
We shall now briefly* describe the dynamic programming approach and pertinent parameters used to solve our maximization problem in the time delay difference domain.

The first step is to discretize the available time delay difference space and limit our region of search. The starting point for our discretion process is the measured lag point data. We shall subdivide the allowable time delay difference trajectory to correspond to 'NP' subdivision between the discrete lag point data. The optimum time delay difference trajectory shall be limited to an initial discretized point and one additional discretized point valid at the end of each of the M consecutive measurements. To limit our search to a finite region, we shall limit our search to nominally m lag points per update centered about our latest estimate. The m lag points correspond to precisely

$$NPT = m(NP - 1) + 1 \quad (4-1)$$

* The fine development of the dynamic programming algorithm is beyond the scope of this study. Our purpose is to present enough detail to discuss the recursive modification. For more information on this algorithm, contact the author or consult References 7, 8, and 9.

discretized points. To further limit our search process we shall limit to 'NPMAX' the maximum number of discretized points that an optimal time delay difference trajectory can move in one update. As noted earlier, each update is centered about the latest optimal estimate. Therefore, although the search region for each update is precisely NPT points, the search regions for adjacent updates do not have to overlap. However, adjacent search regions can only be offset by a maximum of NPMAX discretized points. This restriction ensures at least one allowable path through every discretized point in the data block. Figure 4-1 conceptually illustrates a portion of the discretized data block. Note if $NPMAX = 4$, the magnitude of the shift between update $j + 1$ and $j + 2$ is the maximum allowed.



952 008

Figure 4-1. Partial Discretized Data Block

We shall now briefly describe the dynamic programming process. Let us assume we know and have stored in matrix $\hat{T}_j$ the optimal time delay difference trajectory from each of NPT discretized points of time frame 'j' back to our initial time delay difference estimate prior to time frame 1. The matrix $\hat{T}$ is defined below.

$$\hat{T}_j^T = [\hat{\Delta D}^T(\hat{1}_j), \hat{\Delta D}^T(\hat{2}_j), \dots, \hat{\Delta D}^T(\hat{NPT}_j)] \quad (4-2)$$

where $\hat{\Delta D}^T(i_j)$ is defined as:

$$\hat{\Delta D}^T(i_j) = [\hat{i}_0, \hat{i}_1, \hat{i}_2, \dots, \hat{i}_{j-1}, \hat{i}_j] \quad (4-3)$$

We shall also assume that we have stored in vector $\underline{FIT}(\hat{i}_j)$ the sum of the ML costs (Equation 3-84) corresponding to the optimal trajectory $\hat{\Delta D}(\hat{i}_j)$; i.e.,

$$\underline{FIT}(\hat{i}_j) = \sum_{k=1}^j ML(\hat{i}_k, \hat{i}_{k-1}) \quad (4-4)$$

where $ML(\hat{i}_k, \hat{i}_{k-1})$ is defined from Equation 3-83 as the following.

$$ML(\hat{i}_k, \hat{i}_{k-1}) = \frac{\underline{C}^T \underline{R}_k}{\sqrt{\underline{C}^T \underline{C}}} \quad (4-5)$$

$$\text{Note } \Delta BT = \hat{i}_k - \hat{i}_{k-1} \quad .$$

The next step in the dynamic programming technique is to find the optimal time delay difference trajectory for all NPT discretized points in time frame $j + 1$ back to the original estimate, assuming our prior knowledge of the optimal trajectories from the j th time frame back to the original estimate. Therefore, for each of the NPT discretized points in time frame $j + 1$, we shall perform the following calculation:

For each discretized point ' i_{j+1} ' in time frame $j + 1$, we shall calculate incremental ML functions $[ML(\hat{i}_{j+1}, i_j), i_j : \hat{i}_{j+1} - NPMAX \leq i_j \leq \hat{i}_{j+1} + NPMAX$ (Equation 4-5)] corresponding to the time segment from the end of time frame j to the end of time frame $j + 1$. From the above, there are $2 NPMAX + 1$ candidate optimum time delay difference trajectories from point $\hat{i}_{j+1}$. These candidate optimal time delay difference trajectories shall be designated as shown below.

$$\underline{\Delta D}^T(\hat{i}_{j+1}, i_j) = [i_0, i_1, i_2, \dots, i_j, \hat{i}_{j+1}] \quad , \quad (4-6)$$

$$i_j \text{ is such that } \hat{i}_{j+1} - NPMAX \leq i_j \leq \hat{i}_{j+1} + NPMAX \quad .$$

Note that $i_0, i_1, i_2, \dots, i_j$ is the optimum time delay difference trajectory from the i th point of the j th time frame back to the initial time frame.

To determine the optimal trajectory we proceed as follows. Each incremental cost $ML(\hat{i}_{j+1}, i_j)$ is added to the accumulated optimum ML

cost $\hat{FIT}(i_j)$ corresponding to discretized point i_j to form a candidate ML cost $\hat{FIT}(\hat{i}_{j+1}, i_j)$.

$$\hat{FIT}(\hat{i}_{j+1}, i_j) = ML(\hat{i}_{j+1}, i_j) + \hat{FIT}(i_j) \quad (4-7)$$

In addition, using the candidate optimal time delay difference trajectories of Equation 4-6, we can now calculate candidate sum of squared residuals $SSR(\hat{i}_{j+1}, i_j)$ about a quadratic function for each candidate trajectory. From Equation 3-104 we can write the following.

$$SSR(\hat{i}_{j+1}, i_j) = \underline{\Delta D}^T(\hat{i}_{j+1}, i_j) (I - H(H^T H)^{-1} H^T) \underline{\Delta D}(\hat{i}_{j+1}, i_j), \quad (4-8)$$

i_j is such that $\hat{i}_{j+1} - NPMAX \leq i_j \leq \hat{i}_{j+1} + NPMAX$

where

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ . & . & . \\ . & . & . \\ 1 & j & j^2 \\ 1 & j+1 & (j+1)^2 \end{bmatrix} \quad (4-9)$$

The optimal trajectory from point $\hat{i}_{j+1}$ can now be calculated using Equation 3-104 as shown below.

$$\hat{\Delta D}(\hat{i}_{j+1}) = \hat{\Delta D}(\hat{i}_{j+1}, \hat{i}_j) \text{ for } \hat{i}_j : \text{MAX}[\text{FIT}(\hat{i}_{j+1}, i_j) - K \text{SSR}(\hat{i}_{j+1}, i_j)],$$

$$\hat{i}_{j+1} - \text{NPMAX} \leq i_j \leq \hat{i}_{j+1} + \text{NPMAX} \quad (4-10)$$

where K is a constant that weighs the ML function against the a priori information.

This process is repeated for all $\hat{i}_{j+1}$ such that $1_{j+1} \leq i_{j+1} \leq \text{NPT}_{j+1}$ until we have an optimal time delay difference trajectory back from all NPT discretized points composing time frame $j + 1$.

For all the optimal trajectories $\hat{\Delta D}(i_{j+1})$, $\hat{i}_{j+1} = 1, 2, 3, \dots \text{NPT}$, we can choose the one with the largest value for Equation 4-10. This estimate, $\hat{i}_{j+1}$, we shall use to center our next input lag point data measurement.

The dynamic program process can be performed inductively until an optimal time delay difference trajectory $\hat{\Delta D}(\hat{i}_M)$ is determined for the entire block of input data measurement (M updates). At that point we have completed the transient dynamic programming process. From this point on we would like to begin the recursive sliding window update of our estimator. By this we mean that the total number of data measurements we shall process after every update shall remain fixed at M .

updates. As each new data measurement is incorporated into the block of data, we will drop the oldest data measurement. We would then like to re-estimate the optimal time delay difference trajectory $\hat{\Delta D}(\hat{i}_{M+1})$ for the new block of M data measurements.

Note that if we know the entire optimum time delay difference trajectory for a block of M measurements, we effectively have an estimate for the time delay difference at any time inclusive of the block of M data measurements. We shall nominally choose the middle time frame estimate and the most recent time frame estimate for our outputs. Since our estimates shall be recalculated every update, we shall have two time histories of estimates. The middle time frame history shall be delayed by half the block size from the most recent time frame history. We would expect the middle time frame time history to have a significant reduction in variance with respect to the most recent time frame history. In other words, we should expect that the interpolation process should be more accurate than the extrapolation process.

In the next section we shall discuss efficient numerical techniques for implementing the concept of the recursive batch process.

4.2.2 Recursive Modification

The straightforward approach to implement the recursive batch process would be to simply repeat the inductive dynamic programming algorithm over the entire sliding block of M data measurements after each update. This procedure would involve excessive processing and a

waste of previously obtained information. What we would like to do is to remove the information supplied by the oldest time frame and then add the information obtained by the newest time frame while maintaining all the information acquired in the intermediary time frames. This was the concept presented by LaTourette⁹ for a similar problem in the frequency domain.

We shall now discuss how we can apply this concept to recursively update Equation 4-10 for what we shall nominally call the $M + 1$ time frame. Equation 4-10 has two terms which we need to recursively update. We shall discuss them separately.

Using the dynamic programming approach we need to know all the optimum time delay difference trajectories $\hat{T}_M$ and associated accumulative optimum ML costs $\hat{FIT}(\hat{i}_M)$ for the Mth time frame in order to calculate the optimal time delay difference trajectory $\hat{\Delta D}(\hat{i}_{M+1})$ for the $M + 1$ time frame. After processing the Mth time frame, we have calculated both $\hat{T}_M$ and $\hat{FIT}(\hat{i}_M)$. However, these terms contain the unwanted information calculated from the first time frame. To remove the unwanted information from the first time frame, we can use the following simple procedure:

1. For $\hat{i}_M = 1, 2, 3, \dots$ NPT using $\hat{T}_M$, determine $\hat{i}_0$ and $\hat{i}_1$ corresponding to $\hat{i}_M$.
2. Use Equation 4-5 to calculate $ML(\hat{i}_1, \hat{i}_0)$.*

* In practice, $\hat{i}_1$ and $\hat{i}_0$ are almost always identical for all $\hat{i}_M$'s. Therefore, step 2 may only need to be performed once.

3. With Equation 4-7 in mind, perform the following operation to remove the effects of the first time frame from $\hat{\text{FIT}}(\hat{i}_M)$.

$$\hat{\text{FIT}}(\hat{i}_M) = \text{FIT}(\hat{i}_M) - \text{ML}(\hat{i}_1, \hat{i}_0) \quad (4-11)$$

4. Now simply delete $\hat{i}_0$ from the corresponding vector $\hat{\Delta D}(\hat{i}_M)$ in $\hat{T}_M$.

We are now in a position to add the effects of the $M + 1$ time frame using the standard dynamic programming procedure. Candidate optimum time delay trajectories $\hat{\Delta D}(\hat{i}_{M+1}, i_M)$ and their corresponding ML costs $\text{FIT}(\hat{i}_{M+1}, i_M)$ can be obtained via Equations 4-6 and 4-7, respectively. To complete our calculation, we need to calculate the corresponding candidate sum of squared residuals $\text{SSR}(\hat{i}_{M+1}, i_M)$ for the $M + 1$ time frame.

The straightforward approach to calculate $\text{SSR}(\hat{i}_{M+1}, i_M)$ would be to simply re-apply Equations 4-8 and 4-9 on the new candidate optimum time delay difference trajectory $\hat{\Delta D}(\hat{i}_{M+1}, i_M)$. However, Ng and Lambert¹⁰ have derived a recursive sliding window technique for updating estimates of polynomial Least Mean Square Fit (LMSF) coefficients which we can make use of to speed up our calculations. In their work, Ng and Lambert¹⁰ developed a recursion for a parameter vector $\underline{L}_n$ with components $l_n(i)$, $i = 1, 2, 3, \dots, p + 1$, where p is the order of the polynomial fitting function and n is the update index. In steady state the recursion assumes that a polynomial LMSF is being performed on a sliding data window of fixed data length $M + 1$. For our application, we

have $\underline{L}_{M+1}$ defined as follows.

$$\underline{L}_{M+1} = H^T \underline{\Delta D}(\hat{i}_{M+1}, i_M) \quad (4-12)$$

where the matrix H is defined in Equation 4-9 for $j = M$.

Let us assume that we have $\underline{L}_M$, then Ng and Lambert¹⁶ give the following recursion* to calculate the components of $\underline{L}_{M+1}$.

$$l_{M+1}(1) = l_M(1) + (\hat{i}_{M+1} - i_0) \quad (4-13)$$

$$l_{M+1}(2) = l_M(2) + (M + 1) \hat{i}_{M+1} - l_{M+1}(1) \quad (4-14)$$

$$l_{M+1}(3) = l_M(3) + (M + 1)^2 \hat{i}_{M+1} + 2 l_{M+1}(2) - l_{M+1}(1) \quad (4-15)$$

Note that the above recursion depends on:

1. The oldest data point i_0
2. The newest data point $\hat{i}_{M+1}$
3. The previous values of $\underline{L}_M$.

Using Equation 4-12, we can rewrite Equation 4-8 to the following form.

* Our sliding window has M data measurements but $M + 1$ estimates (note i_0 is the initial estimate). However, we begin our LMSF at $t = 0$, not $t = 1$. Equations 4-13, 4-14, and 4-15 reflect these differences.

$$\underline{SSR}(\hat{i}_{M+1}, i_M) = \underline{\Delta D}^T(\hat{i}_{M+1}, i_M) \underline{\Delta D}(\hat{i}_{M+1}, i_M) - \underline{L}_{M+1}^T (H^T H)^{-1} \underline{L}_{M+1} \quad (4-16)$$

The matrix $(H^T H)^{-1}$ in our application is a known precalculated 3×3 constant matrix. The vector $\underline{L}_{M+1}$ can be recursively calculated from $\underline{L}_M$ by Equations 4-13, 4-14, and 4-15. The only quantity remaining in Equation 4-16 to be recursively updated is $\underline{\Delta D}^T(\hat{i}_{M+1}, i_M) \underline{\Delta D}(\hat{i}_{M+1}, i_M)$. This quantity can be rewritten as follows:

$$\underline{\Delta D}^T(\hat{i}_{M+1}, i_M) \underline{\Delta D}(\hat{i}_{M+1}, i_M) = \underline{\Delta D}^T(\hat{i}_M) \underline{\Delta D}(\hat{i}_M) + (\hat{i}_{M+1})^2 - (\hat{i}_0)^2 \quad (4-17)$$

which completes our recursion for Equation 4-16.

We can now complete our process for the incorporation of information from time frame $M + 1$ by evaluating Equation 4-10 for $\hat{i}_{M+1} = 1, 2, 3, \dots, \text{NPT}$.

The recursive procedure discussed in this section represents a significant reduction in computation via an efficient use of previously calculated information and thus makes feasible the concept of a practical recursive batch process.

However, the straightforward use of the recursive batch procedure may have potential to introduce numerical instability. We shall now discuss this problem and its solution.

The first order recursive digital filter has the following simple form where $Z_i = 1, 2, 3, \dots N+1$ is the input data sequence.

$$Y_{N+1} = \alpha Y_N + \gamma Z_{N+1} \quad (4-18)$$

The current output Y_{N+1} depends on all previous inputs Z_i , $i = 0, 1, 2, \dots N+1$. For this reason these filters are called infinite impulse response filters. If $\alpha < 1$,¹⁷ the effects of old inputs and corresponding errors will decay exponentially, thus ensuring stability.

Recursive batch filters depend on a fixed finite number of previous input data points. For this reason these filters are called finite impulse response filters. These algorithms are made recursive at each update by removing the effects of the oldest data point and then adding the effects of the newest data point. Symbolically, these algorithms can be written in the following form.

$$Y_{N+1} = Y_N + F(Z_{N+1}, Z_{N+1-M}) \quad (4-19)$$

where

$M =$ block size

$Z_{N+1} =$ newest data

$Z_{N+1-M} =$ oldest data

$F(\cdot) =$ some recursion function.

Numerical problems can occur in Equation 4-19 if what is inputted at the current update (Z_N) and processed for M updates is not exactly removed at the $M + 1$ update. This error is normally the result of computer roundoff errors in the recursive computation of Equation 4-19. Note from Equation 4-19 that whatever error we may have at update N will be fully carried over to update $N + 1$. It is very important to realize that this error has nothing to do with any errors in the input data Z_N . Z_N can be as inaccurate as wanted. The only requirement we have is that whatever is inputted at update N and processed for $M - 1$ additional updates will be precisely removed at update $N + M$.

Equation 4-11 can be rewritten into the form of Equation 4-19 if we include the update cycle of Equation 4-7; i.e.,

$$\underline{FIT}(\hat{i}_{j+1}) = \underline{FIT}(\hat{i}_j) + [ML(\hat{i}_{j+1}, \hat{i}_j) - ML(\hat{i}_{j+1-M}, \hat{i}_{j-M})] \quad (4-20)$$

The input Z_{j+1} is represented by $ML(\hat{i}_{j+1}, \hat{i}_j)$.

Numerically executing Equation 4-20 will result in computer roundoff in the lower order bits due to the summation process. However, these errors should be small, uniformly distributed, zero mean and unbiased. Computer simulations have shown for reasonable computer word sizes (32 bits on the VAX 11/780) that the cumulative sum of these errors is insignificant for any reasonable practical application and thus can be ignored.

However, numerical execution of Equations 4-13, 4-14, and 4-15 for the recursive LMSF parameter $\underline{l}_{M+1}$ results in the accumulation of errors which cannot be ignored. This fact was recognized by LaTourette and is discussed in a paper by Ng and LaTourette.¹⁸

The recursive equations for $\underline{l}_{M+1}(j)$ (Equations 4-13, 4-14, and 4-15) can be rewritten in the following revealing form.

$$\begin{aligned} \underline{l}_{M+1}(j) = & \left(\sum_{i=1}^j (-1)^{i-1} B\binom{j}{i} \underline{l}_M(j+1-i) \right) + [M^{j-1} \hat{i}_{M+1}] \\ & + [(-1)^j i_0], \quad j = 1, 2, 3 \end{aligned} \quad (4-21)$$

where $B\binom{j}{i}$ is the binomial coefficient defined by

$$B\binom{j}{i} = \frac{j!}{(j-i)! i!} \quad (4-22)$$

The first term represents integer multiples of the previous states of the accumulation variables $\underline{l}_M(j)$. The second term is an integer multiple of the current data point and represents the incorporation of the current data. The third term represents the removal of the oldest data point.

Numerical problems are encountered when data incorporated in term 2 are not precisely removed $M+1$ updates later in term 3. These errors will be accumulated and expanded in term 1. Errors are primarily the result of computer roundoff due to finite word sizes. Computer simulations have indicated the errors do not significantly expand for

the constant and linear coefficient parameters $l_{M+1}(1)$ and $l_{M+1}(2)$, respectively, but expand and become significant for second order and higher coefficient parameters $l_{M+1}(j)$, $j > 2$.

Fortunately, there is a numerical technique to overcome this problem. The solution is to simply not allow any numerical roundoff. This is accomplished by representing all input data as single precision variables, all multiplicative coefficients by integers, and all accumulation variables by double precision variables. The idea assumes that a finite sum of products of integers and single precision variables will not totally use all the bits contained in a double precision word. Thus there will never be any computer roundoff and what we input at the current update will be precisely removed $M + 1$ updates later. If the local magnitudes of the input data $\hat{i}_j$ are roughly the same and the order of the polynomial LMSF small, the above assumptions should be valid. Appendix C describes an experiment to demonstrate the above ideas.

Note that if Equation 3-111 is used for the a priori penalty function it can easily be placed in a recursive form. Using the definition of Equation 4-2, Equation 3-111 can be written in the following form.

$$\underline{SSD}_{M+1}(\hat{i}_{M+1}) = \underline{SSD}_{M+1}(\hat{i}_M) + (\hat{i}_{M+1} - 2\hat{i}_M + \hat{i}_{M-1})^2 - (\hat{i}_2 - 2\hat{i}_1 + \hat{i}_0)^2 \quad (4-22)$$

where

$\text{SSD}_{M+1}(\hat{i}_{M+1}) \equiv$ sum of the squared differences of the optimum trajectory starting at point $\hat{i}_{M+1}$ of the $M + 1$ time frame.

Equation 4-22 involves only integers and therefore does not encounter the numerical problems previously discussed.

4.2.3 Initialization

The most difficult period for a recursive estimation algorithm is the period between the input of the first initial guess until a steady state condition is reached. This generalization is certainly true for recursive batch algorithms where the basic assumption is that you have available a block of M measurements. The basic problem is that many crucial decisions are being made by the algorithm prior to receiving a full complement of M data measurements.

The first critical decision imposed on the estimation algorithm is the initial estimate. The initial estimate may or may not be based on adequate information. A poor initial estimate will start the algorithm at an incorrect point and thus cause transients and possibly violate algorithmic assumptions. The above problem is especially critical, in our case, where we have assumed a priori knowledge* that the unknown

* If Wolcin's⁶ a priori information function (Equation 3-111) is used this problem should be reduced. The sum of the squared second differences is much more of a local process. The initial point only affects the first squared second difference. However, an LMSF quadratic curve fit through an outlying point will have an affect at all points.

time delay difference trajectory is a quadratic function. An incorrect initial guess will appear as a marked violation of our a priori assumption and will remain so until after the first data block is filled.

Fortunately, the dynamic programming technique used to implement our recursive block estimator has built within itself a natural solution to this problem. This concept was first recognized by LaTourette for this study and is described below.

The original dynamic programming technique employed by Wolcin^{6,7} started at an initial point and expanded in a $2N_{P\text{MAX}} + 1$ point tree structure over the block of data measurements. LaTourette⁹ introduced the concept of limiting the maximum width of the tree structure to N_{PT} points. Since in steady state the basic recursive batch dynamic programming algorithm assumes that the previous estimate contains a full complement of N_{PT} candidate optimal trajectories, why not assume N_{PT} candidate initial estimates centered about the supplied initial estimate. Thus, in essence, we are estimating the initial estimate in a natural extension of the basic algorithm.

We have made some inroads in our first decision, the initial estimate, but many more critical early decisions must be made prior to filling the data block and resulting steady state operation. For example, our a priori information assumption of a quadratic optimum time delay difference trajectory will have no affect on the selection of candidate optimal time delay difference trajectories until the third time frame. Therefore, the first three points in all the candidate

optimum time delay difference trajectories and their corresponding accumulated ML sums will contain no assistance from the a priori information. Even the third time frame will have only one degree of freedom (4 points, 3 coefficients). The basic problem is we are trying to estimate too many parameters on too few measurements, which yields poor results.

Another view of this problem is to view the LMSF process as a filter. The two-sided bandwidth of the input ML estimates is 1 divided by the sampling time (In our case, the sampling time is assumed 1 and therefore the input two-sided bandwidth is 1.) The two-sided bandwidth of the LMSF process is given by LaTourette¹⁹ and Ng²⁰ as the following.

$$BW = \frac{p + 1}{N \Delta T} \quad (4-23)$$

where p = order of the polynomial fit

N = number of measurements

ΔT = sampling time $\cong 1$

and

BW = two-sided bandwidth of the LMSF process.

Note if $BW = 1$, then the LMSF process is an all pass filter.

Examining Equation 4-23, it is apparent that the higher the order 'p' of the LMSF polynomial fit for a fixed number of data points 'N', the larger bandwidth and therefore less noise discrimination. Conversely, the lower the order of the fit the more noise discrimination.

However, if the order of the fit is not adequate, a bias error will result due to model mismatch. Therefore, a designer must balance model mismatch bias errors against noise discrimination. This trade-off leads us to the following technique for our initialization problem.

The first measurement involves the least amount of information, yet we have two points to estimate (the initial point at the beginning of the data frame and a second point at the end of the data frame). For this first case we shall assume a priori that our unknown time delay trajectory is constant (i.e., the initial point is the same as the second point). For two points, a constant ($p = 0$) should yield a minimal model mismatch and yield our maximum and only noise discrimination ($BW = 0.5$).

For the second measurement until some measurement $m < M_d$, we shall assume that our optimum time delay difference trajectory is linear ($p = 1$). For the first few data measurements this should prove, in most cases, an excellent model. We have assumed that the optimum time delay difference trajectory is quadratic over the entire block of M measurements; therefore, it would not be unreasonable to assume that the optimum trajectory can be modeled as linear for the first few time frames. As the number of time frames ' N ' increases, the noise discriminating advantage of the linear fit ($p = 1$) decreases with respect to the quadratic fit ($p = 2$), while the potential for mismatch bias errors increases for the linear model. The exact solution for the optimum value of m is highly dependent on the actual block size. A rough rule of thumb for m is given below.

$$m \leq \text{MIN}[10, M/2] \quad . \quad (4-24)$$

In order to minimize transients and smooth the initialization of the recursive sum of squared residuals SSR (Equation 4-16), we need to update the matrix $(H^T H)^{-1}$ to correspond to the number of data points 'N' associated with the current update. After M data updates, the data block will be full and the matrix $(H^T H)^{-1}$ will therefore become an invariant constant matrix for subsequent recursive updates. The $(H^T H)^{-1}$ matrix for the linear time delay difference trajectory assumption is given below as a function of N. (Note N equal the number of time frames plus 1.)

$$(H^T H)^{-1}_{\text{Linear}} = \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{1,2} & h_{2,2} \end{bmatrix} \quad (4-25)$$

where

$$h_{1,1} = \frac{2(N - 1)}{N(N + 1)} \quad (4-25A)$$

$$h_{1,2} = \frac{-6}{N(N + 1)} \quad (4-25B)$$

$$h_{2,2} = \frac{12}{N(N + 1)(N - 1)} \quad . \quad (4-25C)$$

The following equations define the $(H^T H)^{-1}$ matrix for the quadratic time delay difference trajectory assumption.

$$(H^T H)^{-1}_{\text{quadratic}} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{1,2} & h_{2,2} & h_{2,3} \\ h_{1,3} & h_{2,3} & h_{3,3} \end{bmatrix} \quad (4-26)$$

where

$$h_{1,1} = \frac{9(N^2 - N) + 6}{N(N+1)(N+2)} \quad (4-26A)$$

$$h_{1,2} = \frac{-18(2N - 1)}{N(N+1)(N+2)} \quad (4-26B)$$

$$h_{1,3} = \frac{30}{N(N+1)(N+2)} \quad (4-26C)$$

$$h_{2,2} = \frac{(2N - 1)(96N - 132)}{N(N+1)(N+2)(N-2)(N-1)} \quad (4-26D)$$

$$h_{2,3} = \frac{-180}{N(N+1)(N+2)(N-2)} \quad (4-26E)$$

$$h_{3,3} = \frac{180}{N(N+1)(N+2)(N-2)(N-1)} \quad (4-26F)$$

Note that the transient recursive updates for the vector $\underline{L}$ (Equations 4-13, 4-14, and 4-15) are not affected if ' i_0 ', which represents the removal of old data, is set equal to zero. Of course only coefficients $l(1)$ and $l(2)$ can be used in Equation 4-16 if the linear time delay difference trajectory is assumed since $(H^T H)^{-1}$ will be a 2×2 matrix.

Finally, we shall complete our initialization compensation by applying the standard correction factor for the variance to the

a priori information function (the sum of the squared error.) We will accomplish this by modifying the constant 'K' in Equation 3-97 or 3-97A to the following.

$$K' = K \frac{j+1}{j-p} \quad (4-27)$$

where

j = number of time frames (note the number of data points is j + 1)

p = order of the polynomial LMSF.

Once the data block is filled and our recursive estimator is in steady state, 'K' will be constant for all future updates.

4.3 GRADIENT SEARCH APPROACH

An exact solution for the parameter vector $\hat{\Delta D}$ which maximizes the MAP equation (Equation 3-104 or equivalent) over the block of M measurements cannot be obtained analytically. Therefore, the only technique to obtain a precise solution for $\hat{\Delta D}$ is by numerical search. However, neither Wolcin^{6,7} nor LaTourette⁹ in their foundation work in the frequency domain had derived a numerical solution.

4.3.1 Basic Concept

A basic tool in numerical optimization of a scalar function F with respect to a parameter $\underline{A}$ is the gradient vector defined as:

$$\underline{G} = \frac{\partial F}{\partial \underline{A}} \quad (4-28)$$

In our application the scalar function F will be defined as Equation 3-104*. The parameter vector $\underline{A}$ is by definition the vector $\underline{\Delta D}$ whose components Δd_j are defined by Equation 3-100. From Equation 3-100 we can define an alternate, more useful, parameter vector $\underline{A}$ shown below.

$$\underline{A}^T = [\Delta d_0, \Delta B_1 T, \Delta B_2 T, \dots, \Delta B_M T] \quad (4-29)$$

where

Δd_0 = initial time delay difference at the start of the data block

$\Delta B T_j$ = incremental time delay difference change from the end of the 'j-1' time frame to the end of the 'j' time frame.

Our objective now shall be to calculate the gradient vector $\frac{\partial F}{\partial \underline{A}}$ from Equation 3-104. To begin this process, we shall first divide the scalar function F into two components.

$$F(\underline{A}) = F_1(\underline{A}) + F_2(\underline{A}) \quad (4-30)$$

* Equation 3-104 can be modified by Equation 3-110. However, the results obtained in this section can be applied to this modification via a simple application of the chain rule for differentiation. The derivative of the right-hand expression given in Equation 3-110 is the following: $(M/(M-W))^2 \partial / \partial \underline{A}$ where W via Equation 3-108 is the ML term in Equation 3-104, 'M' is the number of updates in the data block, and $\underline{A}$ is the unknown parameter vector.

where from Equation 3-104 we can define $F_1(A)$ as

$$F_1(\underline{A}) = \sum_{j=1}^M \underline{R}_j^T \underline{C} / \sqrt{\underline{C}^T \underline{C}} \quad (4-31)$$

and $F_2(\underline{A})$ as

$$F_2(\underline{A}) = K \underline{\Delta D}^T [I - H(H^T H)^{-1} H^T] \underline{\Delta D} \quad (4-32)$$

We shall first calculate the gradient for Equation 4-31. Let us redefine Equation 4-31 as the following.

$$F_1(\underline{A}) = f_1(\underline{A}) + f_2(\underline{A}) + \dots + f_M(\underline{A}) \quad (4-33)$$

where from Equation 3-104

$$f_j(\underline{A}) = \underline{R}_j^T \underline{C}^T(\Delta d_{j-1}, \Delta \beta_j^T) / \sqrt{\underline{C}^T(\Delta d_{j-1}, \Delta \beta_j^T) \underline{C}(\Delta d_{j-1}, \Delta \beta_j^T)} \quad (4-34)$$

Let us now take the derivative of Equation 4-33 with respect to Δd_0 . Using the chain rule, we have:

$$\frac{\partial F_1}{\partial \Delta d_0} = \frac{\partial f_1}{\partial \Delta d_0} + \frac{\partial f_2}{\partial \Delta d_1} \frac{\partial \Delta d_1}{\partial \Delta d_0} + \dots + \frac{\partial f_M}{\partial \Delta d_{M-1}} \prod_{i=1}^{M-1} \frac{\partial \Delta d_i}{\partial \Delta d_{i-1}} \quad (4-35)$$

However, from our definition of a piecewise linear function (Equation 2-6), we have that

$$\Delta d_i = \Delta d_{i-1} + \Delta \beta_i T \quad (4-36)$$

and therefore

$$\frac{\partial \Delta d_i}{\partial \Delta d_{i-1}} = 1 \quad (4-37)$$

Therefore we can rewrite Equation 4-35 as the following.

$$\frac{\partial F_1}{\partial \Delta d_0} = \sum_{i=1}^M \frac{\partial f_i}{\partial \Delta d_{i-1}} \quad (4-38)$$

Now let us calculate the derivative of Equation 4-33 with respect to $\Delta \beta_1 T$. Using the chain rule, we have:

$$\frac{\partial F_1}{\partial \Delta \beta_1 T} = \frac{\partial f_1}{\partial \Delta \beta_1 T} + \frac{\partial f_2}{\partial \Delta d_1} \frac{\partial \Delta d_1}{\partial \Delta \beta_1 T} + \dots + \frac{\partial f_M}{\partial \Delta d_{M-1}} \left(\prod_{i=2}^M \frac{\partial \Delta d_i}{\partial \Delta d_{i-1}} \right) \frac{\partial \Delta d_1}{\partial \Delta \beta_1 T} \quad (4-39)$$

but from Equation 4-35 we have that

$$\frac{\partial \Delta d_i}{\partial \Delta \beta_1 T} = 1, \quad i = 1, 2, 3, \dots, M \quad (4-40)$$

Therefore using Equations 4-35 and 4-40, we can rewrite Equation 4-39 as the following.

$$\frac{\partial F_1}{\partial \Delta \beta_1 T} = \frac{\partial f_1}{\partial \Delta \beta_1 T} + \sum_{i=2}^M \frac{\partial f_i}{\partial \Delta d_{i-1}} \quad (4-41)$$

The above procedure can be repeated for the remaining parameters $\Delta\beta_j$ yielding the following result.

$$\frac{\partial F_1}{\partial \Delta\beta_j T} = \frac{\partial f_j}{\partial \Delta\beta_j T} + \sum_{i=j+1}^M \frac{\partial f_i}{\partial \Delta d_{i-1}} \quad (4-42)$$

Note that

$$\frac{\partial F_1}{\partial \Delta\beta_M T} = \frac{\partial f_m}{\partial \Delta\beta_M T} \quad (4-43)$$

Examining Equations 4-38 and 4-42, it becomes readily apparent that we need only calculate the following derivatives.

$$\frac{\partial f_j}{\partial \Delta d_{j-1}}, \quad j = 1, 2, 3, \dots M \quad (4-44)$$

and

$$\frac{\partial f_j}{\partial \Delta\beta_j T}, \quad j = 1, 2, 3, \dots M \quad (4-45)$$

However, these $2 \cdot M$ derivatives only have two symbolic forms nominally specified below for any time frame j .

$$\frac{\partial f}{\partial \Delta d_0} \text{ and } \frac{\partial f}{\partial \Delta\beta T} \quad (4-46)$$

Using a nominal symbolic form for Equation 4-34, we can begin to calculate these two derivatives. (The dependence on the j th time frame will be understood and not written.)

$$\frac{\partial f}{\partial \Delta d_0} = \frac{\left(\sqrt{\underline{C}^T \underline{C}} \right) \left(\underline{R}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right) - \frac{\left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right) \left(\underline{C}^T \underline{R} \right)}{\sqrt{\underline{C}^T \underline{C}}}}{\underline{C}^T \underline{C}} \quad (4-47)$$

$$\frac{\partial f}{\partial \Delta \beta T} = \frac{\left(\sqrt{\underline{C}^T \underline{C}} \right) \left(\underline{R}^T \frac{\partial \underline{C}}{\partial \Delta \beta T} \right) - \frac{\left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta \beta T} \right) \left(\underline{C}^T \underline{R} \right)}{\sqrt{\underline{C}^T \underline{C}}}}{\underline{C}^T \underline{C}} \quad (4-48)$$

The components of the vector $\underline{C}$ are defined by Equation 3-68 and the components of the vector $\underline{R}$ are the nominal measured lag point data. Therefore, the only unknown quantities in Equations 4-47 and 4-48 are $\partial c_i / \partial \Delta d_0$ and $\partial c_i / \partial \Delta \beta T$, $i = -N, -N + 1, \dots, 0, \dots, N$, the components of vectors $\partial \underline{C} / \partial \Delta d_0$ and $\partial \underline{C} / \partial \Delta \beta T$, respectively. These quantities we shall obtain by differentiating Equations 3-68 with respect to Δd_0 and $\Delta \beta T$, respectively. Using the Leibniz rule for differentiating an integral along with trigonometry and algebra, we obtain the following expressions.

$$\frac{\partial c_i}{\partial \Delta d_0} = \frac{\cos(\omega_c \theta_i) \text{Sinc}\left(\frac{B_w}{2} \theta_i\right) - \cos(\omega_c \Delta d_0) \text{Sinc}\left(\frac{B_w}{2} \Delta d_0\right)}{\Delta \beta T} \quad (4-49)$$

$$\frac{\partial c_i}{\partial \Delta \beta T} = \frac{\cos(\omega_c \theta_i) \text{Sinc}\left(\frac{B_w}{2} \theta_i\right) - c_i}{\Delta \beta T} \quad (4-50)$$

where

$$\theta_i = \tilde{T}_i + \Delta d_0 + \Delta \beta T \quad (4-51)$$

and ω_c , Bw , and $\tilde{T}_i$ have been previously defined for Equation 3-67 and repeated below for convenience.

$$\omega_c = \frac{\omega_2 + \omega_1}{2} \quad (4-52)$$

$$Bw = \omega_2 - \omega_1 \quad (4-53)$$

$\tilde{T}_i$ = measured correlation data lag delays.

We will completely specify Equations 4-49 and 4-50 if we give their limits when: $\Delta\beta T \rightarrow 0$; $\tilde{T}_i + \Delta d_0 + \Delta\beta T \rightarrow 0$; and $\Delta\beta T \rightarrow 0$ and $\tilde{T}_i + \Delta d_0 \rightarrow 0$.

$$\begin{aligned} \lim_{\Delta\beta T \rightarrow 0} \left(\frac{\partial c_i}{\partial \Delta d_0} \right) = & -\omega_c \sin(\omega_c \tilde{\theta}_i) \operatorname{Sinc}\left(\frac{Bw}{2} \tilde{\theta}_i\right) \\ & - \frac{\cos(\omega_c \tilde{\theta}_i) \left[\sin\left(\frac{Bw}{2} \tilde{\theta}_i\right) - \frac{Bw}{2} \tilde{\theta}_i \cos\left(\frac{Bw}{2} \tilde{\theta}_i\right) \right]}{\frac{Bw}{2} \tilde{\theta}_i^2} \end{aligned} \quad (4-54)$$

$$\lim_{\Delta\beta T \rightarrow 0} \left(\frac{\partial c_i}{\partial \Delta\beta T} \right) = \frac{1}{2} \lim_{\Delta\beta T \rightarrow 0} \left(\frac{\partial c_i}{\partial \Delta d_0} \right) \quad (4-55)$$

where

$$\tilde{\theta}_i = \tilde{T}_i + \Delta d_0 \quad (4-56)$$

$$\lim_{\substack{\tilde{T}_i + \Delta d_0 + \Delta \beta T \rightarrow 0 \\ \Delta \beta T \neq 0}} \left(\frac{\partial c_i}{\partial \Delta d_0} \right) = \frac{1 - \cos(\omega_c \tilde{\theta}_i) \operatorname{Sinc}(\frac{B_w}{2} \tilde{\theta}_i)}{\Delta \beta T} \quad (4-57)$$

$$\lim_{\substack{\tilde{T}_i + \Delta d_0 + \Delta \beta T \rightarrow 0 \\ \Delta \beta T \neq 0}} \left(\frac{\partial c_i}{\partial \Delta \beta T} \right) = \frac{1 - c_i}{\Delta \beta T} \quad (4-58)$$

$$\lim_{\substack{\tilde{T}_i + \Delta d_0 \rightarrow 0 \\ \Delta \beta T \rightarrow 0}} \left(\frac{\partial c_i}{\partial \Delta d_0} \right) = 0 \quad (4-59)$$

$$\lim_{\substack{\tilde{T}_i + \Delta d_0 \rightarrow 0 \\ \Delta \beta T \rightarrow 0}} \left(\frac{\partial c_i}{\partial \Delta \beta T} \right) = 0 \quad (4-60)$$

We have now completely defined the gradient for Equation 4-31. We shall now calculate the gradient for Equation 4-32. Fortunately, this gradient can be easily obtained by vector calculus. The derivative of Equation 4-32 with respect to the vector $\underline{A}$ can be written as:

$$\frac{\partial F_2(\underline{A})}{\partial \underline{A}} = \left(\frac{\partial F_2(\underline{A})}{\partial \underline{\Delta D}} \right)^T \frac{\partial \underline{\Delta D}}{\partial \underline{A}} \quad (4-61)$$

where

$$\left(\frac{\partial F_2(\underline{A})}{\partial \underline{\Delta D}} \right)^T = 2 K \underline{\Delta D}^T [I - H(H^T H)^{-1} H^T] \quad (4-62)$$

Note that the matrix $(I - H(H^T H)^{-1} H^T)$ is symmetric and $\partial \underline{\Delta D} / \partial \underline{A}$ from Equations 4-36, 4-37, and 4-40 can be written as the following $M+1, M+1$ lower triangular unity matrix

$$\frac{\partial \Delta D}{\partial \underline{A}} = \begin{bmatrix} 1 & 0 & 0 & 0 & . & . & . & 0 & 0 \\ 1 & 1 & 0 & 0 & . & . & . & 0 & 0 \\ 1 & 1 & 1 & 0 & . & . & . & 0 & 0 \\ . & . & . & & & & & & \\ . & . & . & & & & & & \\ . & . & . & & & & & 1 & 0 \\ 1 & 1 & 1 & 1 & . & & & 1 & 1 \end{bmatrix} \quad (4-63)$$

Equation 4-61 can now be rewritten in the following form.

$$\frac{\partial F_2(\underline{A})}{\partial \underline{A}} = \underline{\Delta D}^T \underline{M} \quad (4-64)$$

where $\underline{M}$ is a constant $M+1, M+1$ matrix defined below.

$$\underline{M} = 2 \underline{K} (\underline{I} - \underline{H}(\underline{H}^T \underline{H})^{-1} \underline{H}^T) \frac{\partial \underline{\Delta D}}{\partial \underline{A}} \quad (4-65)$$

and $\underline{\Delta D}$ is defined from the unknown parameter vector $\underline{A}$ by Equation 3-100.

We have now completely defined the gradient vector of the MAP Equation 3-104 with respect to the unknown parameter vector $\underline{A}$. With the knowledge of the gradient vector, there are numerous gradient, conjugate direction,²¹ and variable metric²² algorithms to solve for the unconstrained maximum.

In this study, computation time was considered critical for a practical algorithm. Therefore, the computational method chosen was a single line search along the gradient direction. However, in the

recursive update of the gradient algorithm, we shall see that the simple line search can evolve into a more complicated procedure.

4.3.2 Recursive Update

Let us assume that after M updates we have the optimal estimates for the unknown parameter vector $\hat{\underline{A}}$. From Equation 4-29 we can write $\hat{\underline{A}}$ as the following.

$$\hat{\underline{A}}^T = [\hat{\Delta d}_0, \hat{\Delta \beta}_1^T, \hat{\Delta \beta}_2^T, \dots, \hat{\Delta \beta}_M^T] \quad (4-66)$$

We now slide our block of M data measurements by one update measurement. Therefore, we drop our first measurement from the block and add our M + 1 measurement to the block. We now want to optimize the MAP Equation 3-104 over the updated block of data for an updated parameter vector $\tilde{\underline{A}}$ defined below.

$$\tilde{\underline{A}}^T = [\tilde{\Delta d}_1, \tilde{\Delta \beta}_2^T, \tilde{\Delta \beta}_3^T, \dots, \tilde{\Delta \beta}_M^T, \tilde{\Delta \beta}_{M+1}^T] \quad (4-67)$$

It is well accepted that the accuracy and even convergence of a single gradient line search for a maximum is critically dependent on the initial estimate. Assuming that we had convergence in our last update, we are in an excellent position to specify an accurate initial estimate for the current update maximization problem. We shall define $\tilde{\underline{A}}$ in terms of the previous optimum vector $\hat{\underline{A}}$ as shown below.

$$\tilde{\Delta d}_1 = \hat{\Delta d}_0 + \hat{\Delta \beta}_1 T \quad (4-68)$$

$$\tilde{\Delta \beta}_j T = \hat{\Delta \beta}_j T, \quad j = 2, 3, 4, \dots M \quad (4-69)$$

and for simplicity

$$\tilde{\Delta \beta}_{M+1} T = \hat{\Delta \beta}_M T^* \quad (4-70)$$

With the initial estimate vector $\tilde{\underline{A}}$ we now can obtain, via a single line search along the gradient direction, the estimate of the new optimum vector $\hat{\underline{A}}$.

Our algorithm can continue along the above lines recursively for all subsequent data measurement updates. Effectively, we use the solution of the previous update to yield excellent approximations for the current update.

Examining the above algorithm, it is apparent that a given parameter $\Delta \beta_j T$ is involved in M gradient searches from its inclusion in the unknown parameter vector until M updates later when it is used to estimate the initial time delay difference. This observation leads to the idea²³ of modifying our simple gradient line search algorithm to a modified conjugate direction or variable metric routine where we update the conjugate direction or metric once every update cycle. This represents a significant computational savings from multiple conjugate direction updates every cycle. However, due to adequate performance of

* For the first couple of updates after initialization, the previous time delay difference change estimate may not be our best choice. The obvious problem is an initialization error. A better choice for the first few updates may be $\tilde{\Delta \beta}_{M+1} T = 0$.

the simple gradient search algorithm and increased computational costs, this idea has not been pursued for this study.

Note that in performing a line search, the MAP function (Equation 3-104) must be evaluated several times for each recursive update. Since the entire optimum time delay difference trajectory is altered each update, we cannot recursively update the MAP equation as we did for the dynamic programming algorithm. We must evaluate Equation 3-104 at each new update.

In this section we have assumed that we started with an optimal estimate after M time frames. It is critical that we examine how we propagate an optimal estimate from time frame 1 through time frame M.

4.3.3 Initialization

Note that after the first data measurement the numerical gradient algorithm has two unknown parameters to estimate. They are the initial time delay difference Δd_0 and the time delay difference after the first time frame Δd_1 or, equivalently, the incremental time delay difference $\Delta \beta_1 T$ ($\Delta d_1 = \Delta d_0 + \Delta \beta_1 T$). Our initial estimate for $\tilde{\Delta d}_0$ is the initial time delay difference supplied to the algorithm and our initial estimate for $\Delta \tilde{\beta}_1 T$ is zero, the mean of all possible incremental time delays. Therefore, we shall effectively have estimated our initial estimate after the first time frame and shall continue refining the initial estimate after each subsequent data update until the data block is filled

with M measurements. The above refinement process for the initial estimate is the counterpart of the multiple hypothesis algorithm for the initial estimate used by the dynamic programming algorithm.

Using a similar argument as employed for the dynamic programming algorithm (Section 4.2.3), we can conclude that a quadratic time delay difference trajectory model makes little sense for the first few time frames. Therefore, we shall assume a linear time delay difference model for the first $m < M_d$ time frames and a quadratic model afterwards. The effect of switching time delay difference trajectory models and increasing the number of data points from 2 through M alters the H matrix. Therefore the matrix $[I - H(H^T H)^{-1} H^T]$ must be recalculated each update for the MAP Equation 3-104 and for the gradient Equation 4-65 until the data block is filled with M data measurements. This calculation is straightforward once the matrix $(H^T H)^{-1}$ is calculated via Equations 4-25 through 4-25C or Equations 4-26 through 4-26F.

Similar to the dynamic programming algorithm, we can apply the standard correction factor for the variance to the a priori information function (the second term in Equation 3-104). We accomplish this by using Equation 4-27.

Simulation studies involving initializing the gradient search algorithm, discussed in Chapter 6, indicate additional steps required to obtain good initialization characteristics for the gradient search algorithm. The discussion of the additional initialization procedure is contained in Chapter 6.

4.3.4 Sum of Squared Second Differences Penalty Function

For completeness and information we shall define the gradient vector $\partial F_2(\underline{A})/\partial \underline{A}$ if Equation 3-111 was used as the a priori information penalty function instead of Equation 4-32.

$$\frac{\partial F_2(\underline{A})}{\partial \Delta d_0} = 0 \quad (4-71)$$

$$\frac{\partial F_2(\underline{A})}{\partial \Delta \beta_1^T} = -2K [\Delta d_2 - 2 \Delta d_1 + \Delta d_0] \quad (4-72)$$

$$\frac{\partial F_2(\underline{A})}{\partial \Delta \beta_j^T} = -2k [\Delta d_{j+1} - 3 \Delta d_j + 3 \Delta d_{j-1} - \Delta d_{j-2}], \quad j = 2, 3, 4, \dots M-1 \quad (4-73)$$

$$\frac{\partial F_2(\underline{A})}{\partial \Delta \beta_M^T} = -2K [\Delta d_{M+1} - 2 \Delta d_M + \Delta d_{M-1}] \quad (4-74)$$

where

Δd_j is defined by Equation 4-36

$\underline{A}$ is defined by Equation 4-29

K is constant of proportionality.

4.4 RELATED RECURSIVE BATCH ALGORITHMS

In our derivation of the MAP estimator, several alternate estimators became apparent which are of theoretical or practical interest. In this section we shall very briefly identify these algorithms for the benefit of later comparison testing.

The first alternative algorithm uses Equation 3-111 for an a priori information function. This algorithm has been discussed in some detail in Chapter 3. We shall call this algorithm the Sum of Squared Differences (SSD) algorithms. The primary purpose of examining the SSD algorithm is to study an alternative a priori information function. The SSD algorithm was implemented using a dynamic programming approach.

4.4.1 Sequential Algorithm

In Section 3.8 we claimed at low SNR that maximizing the ML expression and the a priori information expression concurrently (Equation 3-104) is superior to maximizing the ML expression (Equation 3-84) with respect to the unknown time delay difference trajectory and then performing a quadratic LMSF to the ML estimates to obtain improved optimal estimates. The 'sequential' algorithm performs the latter operation using a gradient search scheme.

4.4.2 Spectral Estimation Algorithms

In Section 3.7 an ML estimator was derived (Equation 3-93) to simultaneously optimize for the unknown time delay difference trajectory and a spectral parameter. This ML estimator was combined with the SSR a priori information penalty function to form a MAP estimator. Algorithm 'SPECTRAL' solves for the above MAP estimator using a dynamic programming technique.

In addition, either the primary dynamic programming algorithm or the gradient search algorithm can estimate a spectral parameter off-line via Equation 3-97.

Many more recursive batch algorithms or combination of algorithms are possible, but the algorithms discussed in Chapter 4 are the ones primarily examined in this study.

CHAPTER 5

OPTIMUM PARAMETER SELECTION

The objective of this chapter is to examine the key parameters which affect our two principal MAP estimation algorithms. They are the Dynamic Programming (SSR) algorithm and the Gradient Search algorithm. Both algorithms are adequately described in Chapter 4.

The key^{*} parameters we shall address in this study are listed below:

1. K The proportionality constant between the ML estimate and the a priori information.
2. M, T The number of data measurements in a data block and the integration time per measurement.
3. N The number of measured lag points calculated for each data measurement.
4. Tau The time delay spacing between lag points.

* The dynamic programming algorithm has other parameters relating to the discretation of the allowable time delay difference space. The equivalent parameters were investigated by LaTourette, Greinader, and Wolcin.¹⁰

5.1 TEST METHOD

Our investigation into optimum parameter selection shall be by simulation. We shall vary a key parameter while holding the others constant. Our measures of optimality will in general be RMS time delay difference error plotted versus the key parameter or parameters.

In order to help ensure independence of the simulation program from the derived algorithms, we have used a stimulation/simulation program developed by H. F. Jarvis, Jr.²⁵ For our purposes the key aspects of the stimulation/simulation program are listed below.

1. The program simulates programmed target motion.
2. The program provides the measured lag point data at the desired lag times.
3. The SNR can be set to any level and held constant throughout the simulation.
4. The simulation program models the target spectrum by a piecewise linear model.
5. The program assumes oversampling of the target and noise spectrums and thus models correlated noise components for the measured lag point data. The noise correlation matrix N takes on the following form.

$$N(i,j) = \frac{\pi}{B_w KSNR T} [\text{CORR}(i-j) + KSNR \text{CORR}(i + j + N-1)] \quad (5-1)$$

where

$$\text{CORR}(t) = \cos[\omega_c \tau t] \frac{\sin[\frac{Bw}{2} \tau t]}{\frac{Bw}{2} \tau t} \quad (5-1A)$$

$$\text{KSNR} = \left(\frac{\text{SNR}}{1 + \text{SNR}} \right)^2 \quad (5-1B)$$

ω_c = the center frequency (radians)

Bw = the bandwidth (radians)

τ = the lag point spacing

i, j = the lag point indices

N = the total number of lag points

SNR = halfbeam SNR ratio (linear)

T = integration time per measurement.

The right-hand side of Equation 5-1A is a correlation function for a flat broadband power spectrum. The noise power spectrum has the same band limits as the flat target signal power spectrum. Therefore, Figure 5-1 for the auto correlation function of the target signal specifies Equation 5-1A the noise correlation function.

For all simulations, unless otherwise stated, the following shall be true.

1. The signal and noise power spectrum shall be flat.
2. The SNR shall be held constant throughout the simulation.
3. The noise component of the measured lag data shall be correlated.
4. The MAP estimation algorithm shall not prewhiten the measured lag point data.

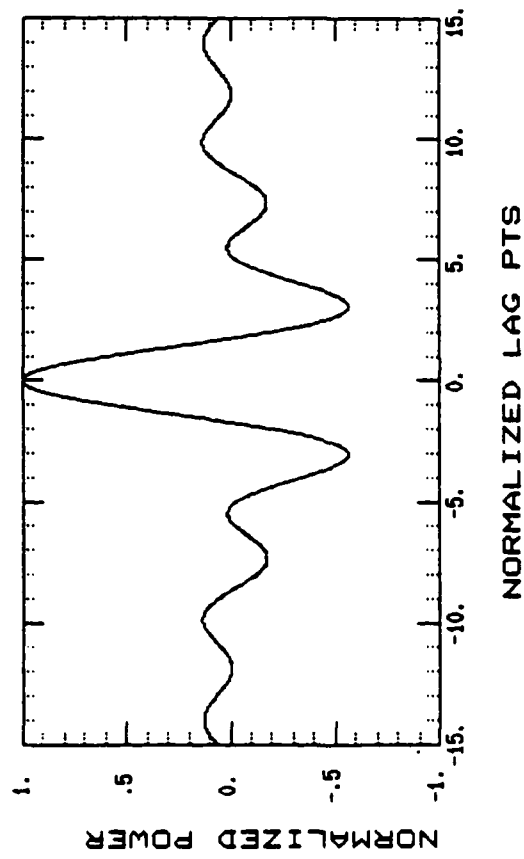


Figure 5-1 Correlation Function vs Normalized Lag Points

5. The nominal relationship between lag point spacing and the auto correlation function resulting from a stationary flat target power spectrum is shown in Figure 5-1.
6. All time delay difference errors shall be normalized by the lag point spacing and therefore shall be in terms of fractional lag point spacing. The lag point spacing is shown in Figure 5-1.
7. The dynamic programming (SSR) algorithm will operate on a time delay difference space which has been quantized to a quarter of the lag point spacing. The maximum change for the dynamic programming algorithm (SSR) is twice the lag point spacing per measurement update.
8. The nominal data block size M is 10.
9. The nominal integration time T for each data update is 5 seconds.
10. The RMS time delay difference error is for the estimate at the midpoint of the data block.
11. The nominal non-stationary dynamic target time delay difference trajectory is shown in Figure 5-2.

5.2 THE PROPORTIONALITY CONSTANT K

An important question to ask is what should be the relative proportionality (K) between the ML estimate and the a priori information function. The ML estimate is roughly proportional to SNR and the a priori information penalty function is roughly proportional to inverse SNR.

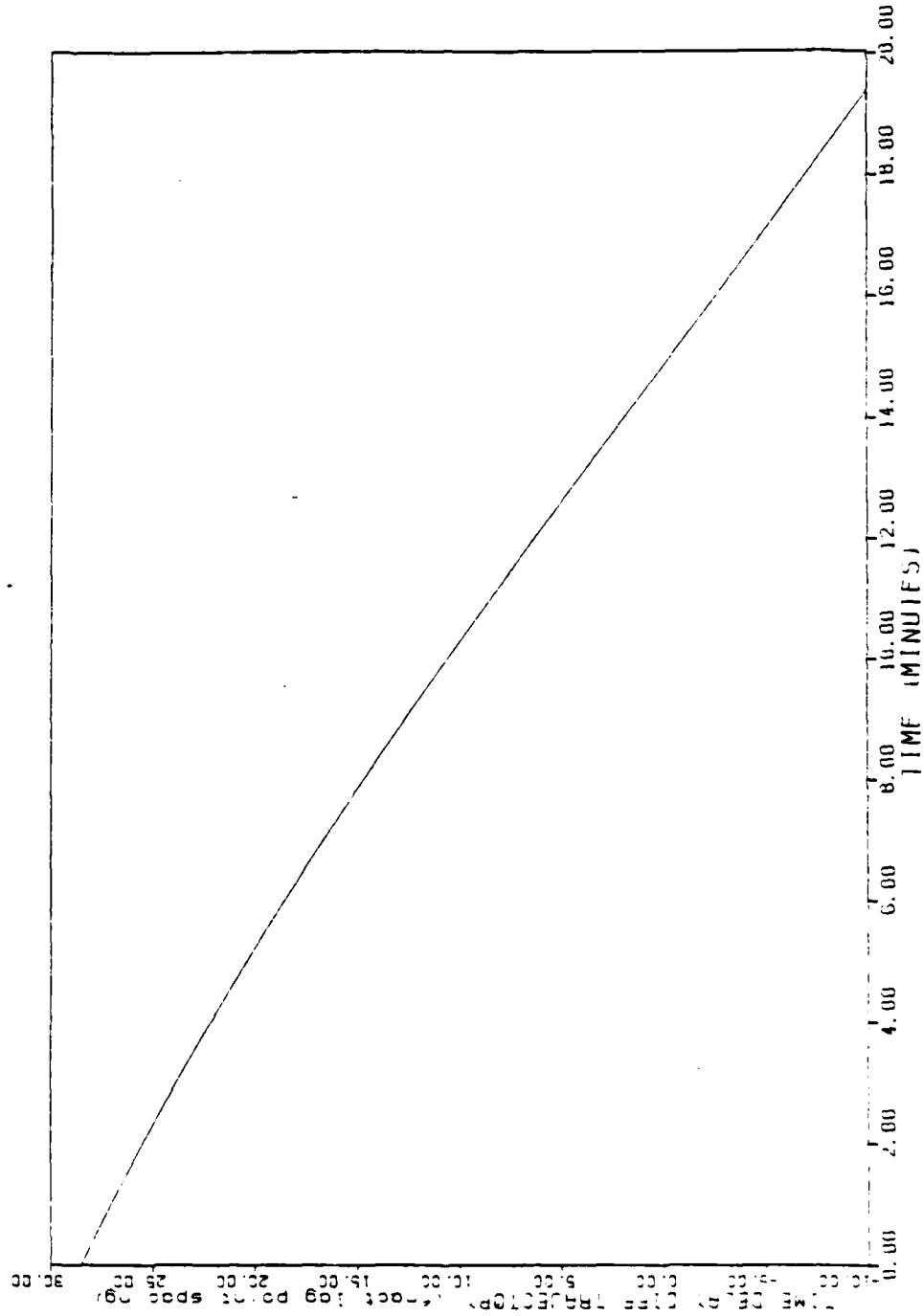


Figure 5-2 Target Time Delay Difference Trajectory vs Time

A practical view of the proportionality constant K is to view it as a measure of confidence in the individual ML estimates. If K is large, more weight is given to conforming to the time history of ML measurements. If K is small, more weight is given to the individual ML estimate. To a certain extent the ML estimate and the a priori information scale themselves. At high SNR the ML estimate will dominate, which is proper, and at low SNR the a priori information will play an ever increasing role. Therefore, a second question is how does the proportionality constant K depend on SNR.

We shall examine the above questions first with the dynamic programming (SSR) algorithm. We shall hold the SNR constant at five different levels (5 dB, -10 dB, -5 dB, -15 dB, -18 dB). For each SNR level we will vary the proportionality constant K .^{*} Figure 5-3 shows the results of the above simulations.

Viewing Figure 5-3, the first point to note is that for SNR = -18 dB the dynamic programming (SSR) algorithm diverges (does not track) for $K < 0.25$. This is an indicator that the a priori information is required for the dynamic programming (SSR) algorithm to maintain track on the target time delay difference trajectory at very low SNR.

* The absolute value for K is algorithmic dependent. Therefore, all numbers used for K should be taken in a relative sense, especially between algorithms.

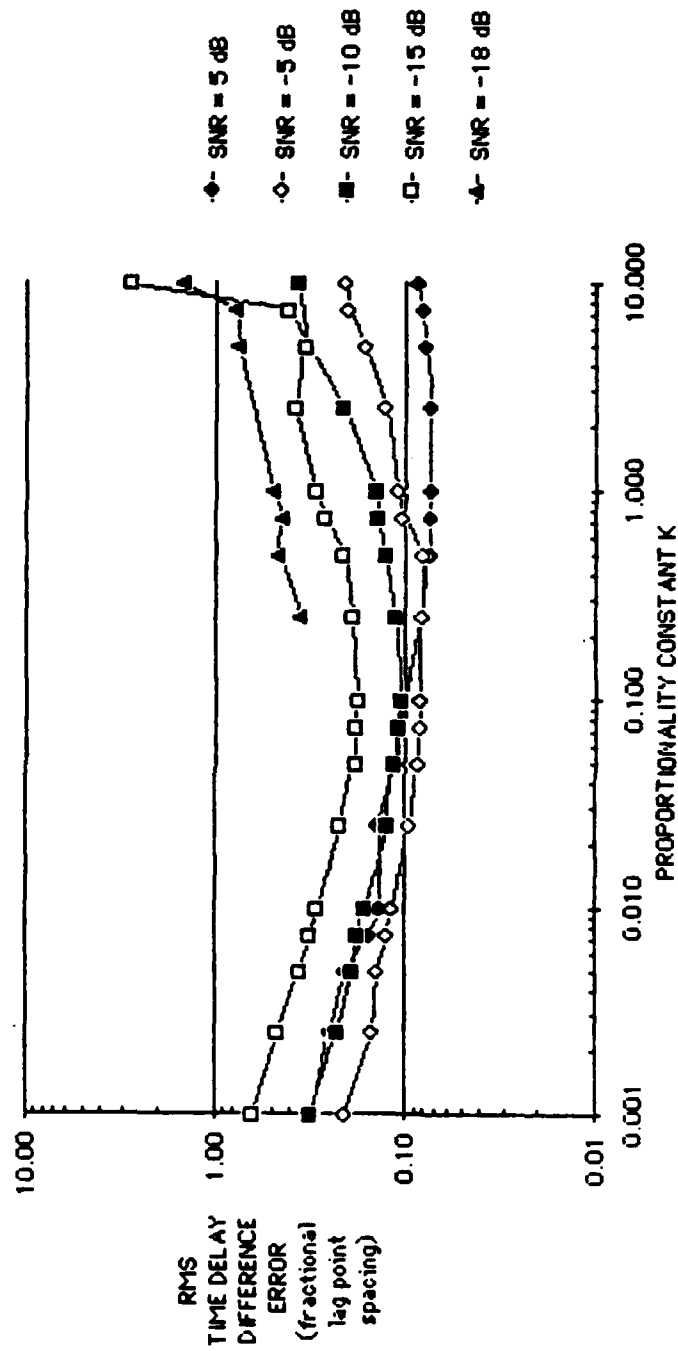


Figure 5-3. RMS Time Delay Difference Error vs Proportionality Constant K, Dynamic Program SSR ($M = 10, T = 5 \text{ sec}, N = 31$)

The most contradictory thing about Figure 5-3 is that for $K < 0.5$ the RMS time delay difference error is lower for the -5 dB case than for the +5 dB case. Not shown in Figure 5-3 is that for $K = 0$ (ML estimate only), the RMS time delay difference error for the +5 dB case is 0.332 fractional lag point spacings and, as would be expected, the RMS time delay difference error for the -5 dB case is 0.425 fractional lag point spacings. The problem is that at high SNR the RMS time delay difference error for the dynamic programming (SSR) algorithm is dominated by discretation errors. (The dynamic programming (SSR) algorithm only has resolution down to a quarter of the lag point spacing.) The a priori information tends to smooth out the discretation error for a moving target. However, since the a priori information is roughly inversely proportional to SNR, larger values of K are required to obtain the same smoothing. Therefore, at higher SNR and a given value of K , the discretation error remains constant while the effective smoothing is reduced. For the gradient search algorithm where there is no discretation error we would not expect to see this.

Note that for the +5 dB case as K becomes large, the RMS time delay difference error approaches the theoretical limit of 0.0722 fractional lag point spacings for an approximately linear target motion (Figure 5-2). This theoretical limit is calculated in Appendix D.

In general, for all SNR levels the RMS time delay difference error decreases at first as K increases but eventually increases for K large (>10). A good overall choice for K given the fixed parameters chosen

is K : $0.01 \leq K \leq 1$.^{*} The above makes sense. At first, the addition of a priori knowledge decreases the RMS time delay difference error. But in the limit as K becomes large, the a priori knowledge will totally dominate the ML estimate. The a priori information in and of itself contains no information of the actual time delay difference trajectory without the ML estimate. Therefore, in the limit as K goes to infinity there is no information.

Figure 5-4 is the corresponding graph for the gradient search algorithm. The SNR is held constant at SNR levels of 5 dB, -5 dB, -15 dB, -18 dB, and -20 dB while the proportionality constant K is varied. Figure 5-4 is more like what we would expect. At high SNR (+5 dB, -5 dB), the constant K has little effect except for large values where we see signs of a degradation. Note that the RMS time delay difference errors for SNR levels of 5 dB and -5 dB are significantly reduced from the corresponding RMS time delay difference errors shown in Figure 5-3. This is true because the gradient search algorithm yields continuous estimates and therefore does not encounter discretization errors at high SNR. Also note that, as would be expected, in Figure 5-4 the RMS time delay difference errors for the +5 dB case are always lower than the RMS time delay difference errors for the -5 dB case.

* As we shall see later, this optimum choice is only valid for the choice of parameter we have held fixed. In particular, we shall see that as the data block size is increased ($M > 10$), this optimum criteria may no longer be valid.

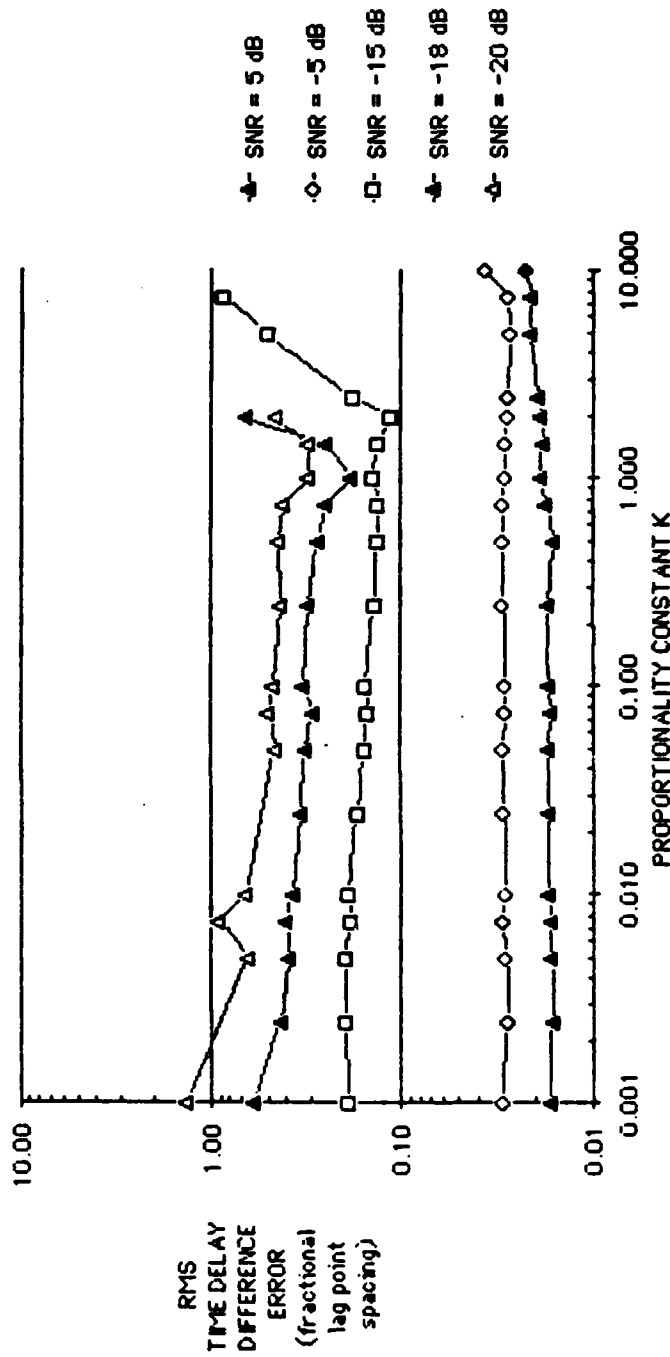


Figure 5-4. RMS Time Delay Difference Error vs Proportionality Constant K,
Gradient Search Algorithm ($M = 10$, $T = 5$ sec, $N = 31$)

From Figure 5-4, we can see in general that, as SNR is lowered, the a priori information yields a greater reduction in RMS time delay difference error. In each case the RMS time delay difference error reduction appears to be reduced monotonically by increasing K until a critical point where there is a rapid degradation in RMS time delay difference error and loss of track with increasing K. Once again a suitable range for the proportionality constant K is $K: .1 \leq K \leq 2$.

5.3 THE DATA BLOCK SIZE M AND INTEGRATION TIME T

A key question to ask is how many data measurement (M) should we include in our data block, and the related question, how long should we integrate each measurement.

We shall examine the above question first with the dynamic programming (SSR) algorithm.

The above questions are likely to be complicated by the target time delay difference trajectory. Therefore, to initially minimize complications we shall use a time delay difference geometry which is constant and, by definition of our initialization process, right on a discretized time delay difference point used by the dynamic programming (SSR) algorithm.

The first case we shall look at is for the ML estimator only ($K = 0$) with a integration time of 5 seconds and an SNR of -15 dB.

Figure 5-5 shows both the RMS time delay difference midpoint and endpoint error output for block sizes of $M = 2, 4, 8, 10, 16, 20$, and 30 . (Note for an integration time of 1 second the algorithm diverged for the above scenario.)

Examining Figure 5-5 we note that block size has almost no effect on the RMS time delay difference error for block sizes of 8 and longer. This implies that the ML estimator under static laboratory* conditions does not significantly depend on moderate block sizes but obviously depends on the integration time per update. There is, however, some dependence on adjacent states and thus minimum block size. This dependence is the result of the weak assumption of a piecewise linear model and the limit on the maximum allowable time delay difference change per measurement update. Evidence of the dependence is apparent from the reduced RMS time delay difference error of the midpoint estimate over the endpoint estimate (RMS midpoint ≈ 0.79 , RMS endpoint ≈ 1.1) and the variation in RMS time delay difference error for block sizes of 2 and 4. If each update was truly independent, we would expect the same RMS time delay difference error for any point in the block. Although in general we will only plot the midpoint RMS time delay difference errors, the endpoint RMS time delay difference errors for all simulations performed was always greater than the corresponding midpoint RMS time delay difference errors. Effectively, algorithmic decisions are being made

* In a scenario where significant signal fades are encountered, this conclusion may not be strictly true. The addition data block size will potentially allow for smoother transition over signal fades.

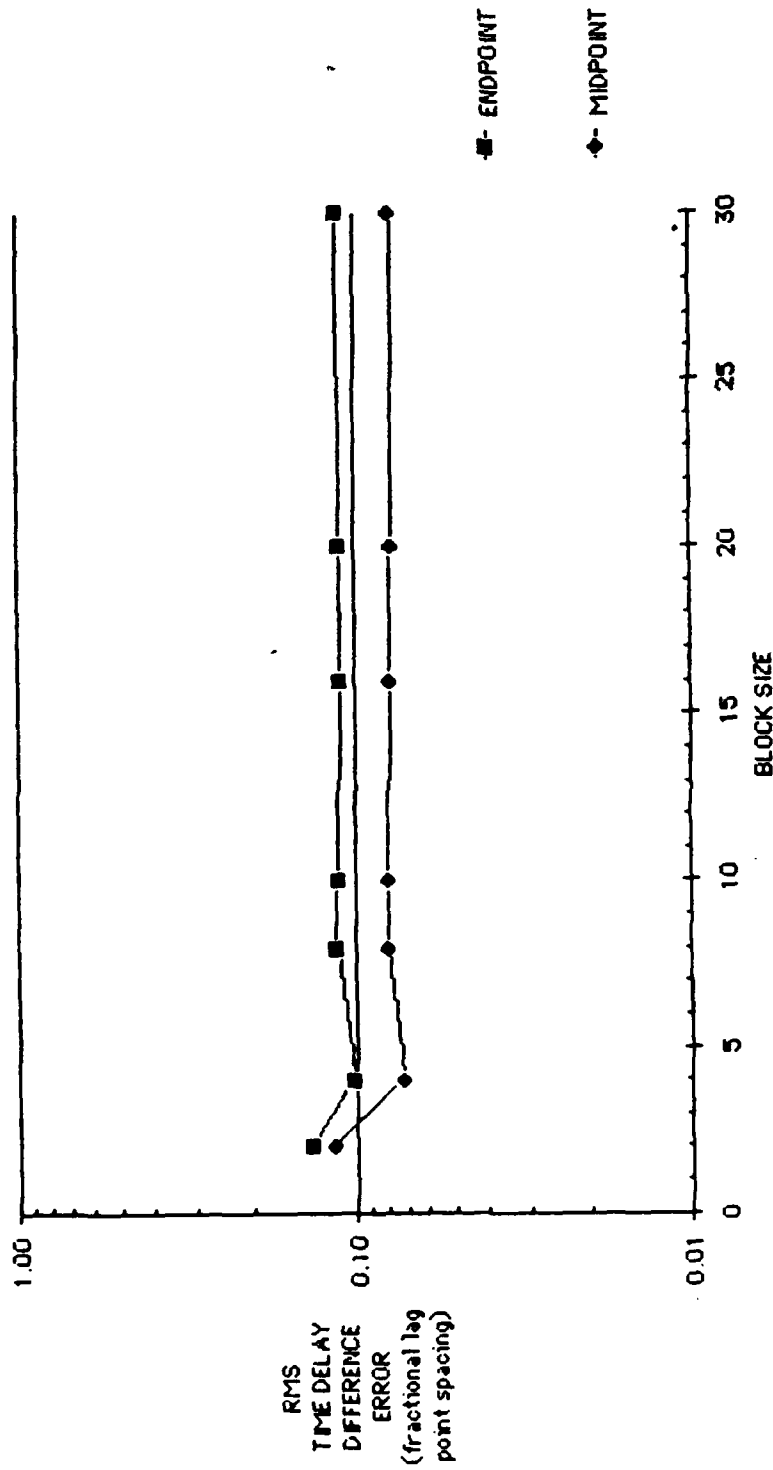


Figure 5-5. RMS Time Delay Difference Error vs Block Size,
 Static Scenario, Dynamic Programming (SSR)
 ($T = 5$ sec, $K = 0$, $N = 31$, $SNR = -15$ dB)

at each update with a minimum decaying dependence on surrounding updates. Therefore, the RMS time delay difference error of the ML estimate at the midpoint of each block is a local process with little correlation with data measurements more than one or two updates removed. As we shall see later, the slight improvement in RMS time delay difference error for a block size of 4 is rather odd. We would normally expect an increase in RMS time delay difference error.

The next logical step is to repeat the above experiment using a priori information ($k = 0.5$). For effect we shall also lower the SNR to -18 dB. Figure 5-6 shows the resulting RMS time delay difference error versus block size. Since quadratic a priori information makes little sense for block sizes of 2 and 4, we shall only process block sizes of 8 or greater.

Figure 5-6 shows an entirely different picture than Figure 5-5. The block size of 8 diverged while the block size of 30 effectively has zero^{*} error. In between we see a steady reduction in RMS time delay difference error with increased block size. As stated in Chapter 3, the a priori information is what combines the large observation interval model with the multiple measurement model. In this case our assumed model matches the actual target time delay trajectory perfectly, for any data length. Therefore we would expect and have confirmed in Figure 5-6 that increasing the observation interval decreases the variance and thus the RMS error.

* The zero error is due to the discretized allowable time delay difference trajectory and the fact that the target signal is a constant lined up with an allowable discretized point.

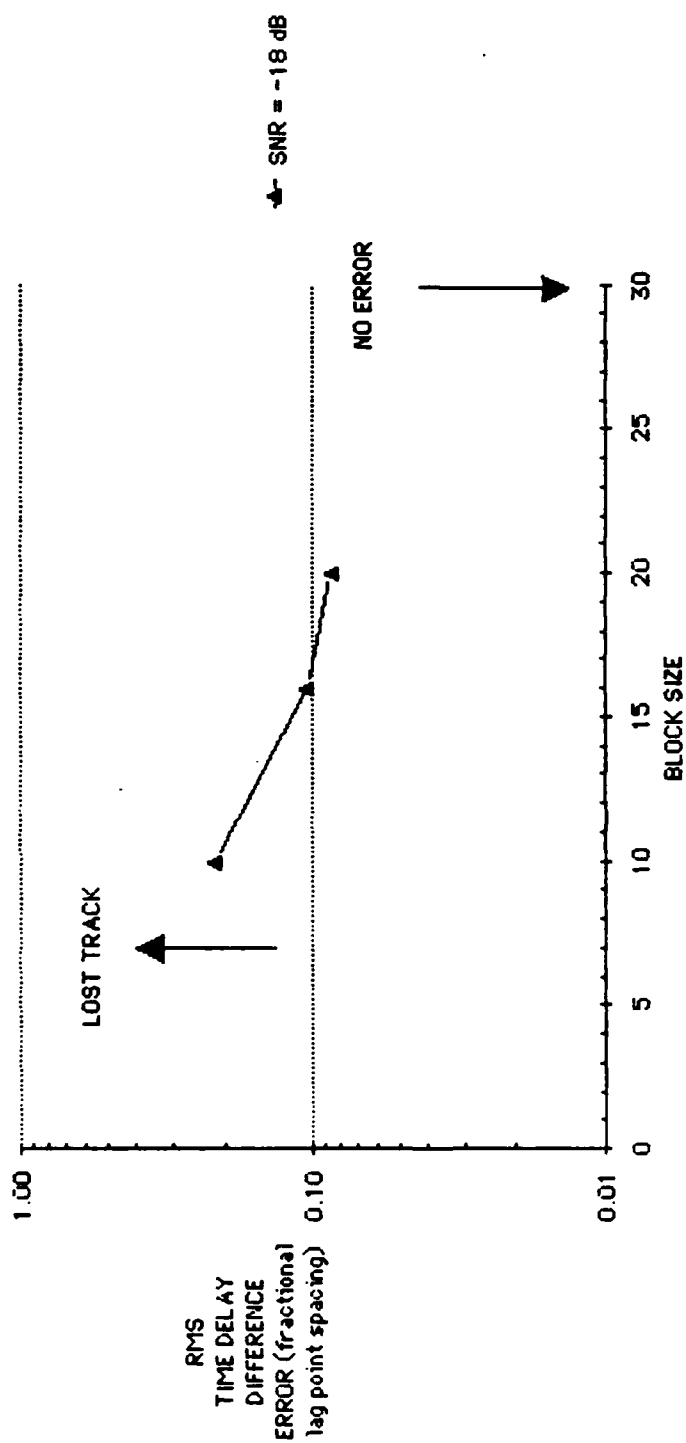


Figure 5-6. RMS Time Delay Difference Error vs Block Size,
Static Scenario, Dynamic Programming (SSR)
($T = 5$ sec, $K = 0.5$, $N = 31$)

The above experiments are very nice and confirm our basic concepts but are not very practical in the real world. Therefore, we shall perform similar experiments using moderate target motion as shown in Figure 5-2.

Figure 5-7 shows the RMS time delay difference error versus block size for the ML estimator ($K = 0$) for an integration time of 5 seconds at three different SNR levels (-5 dB, -10 dB, and -13 dB). Figure 5-8 shows corresponding plots for an integration time of 1 second at SNR levels of 0 dB, -5 dB, and -10 dB. Figures 5-7 and 5-8 for the dynamic target scenario agree with the corresponding Figure 5-5 for the static scenario. The results indicate that, for the ML estimator only ($K = 0$), moderate block sizes have little effect on the RMS time delay difference error. This result is not surprising since data measurements are still loosely coupled by our assumed piecewise linear model and therefore local variance is a local issue. For short block sizes of 4 and 2, there is a general increase in RMS time delay difference error, except at the lowest SNR level and at a block size of 4 where there is a slight decrease in RMS time delay difference error. This result, in general, confirms our theory on the loose correlation of data points due to the piecewise linear assumption. For short block sizes, increasing the block size has some affect on the RMS time delay difference error while at longer block sizes increasing the block size has little affect on RMS time delay difference error. Note, however, that for the dynamic scenario (Figure 5-7) the dynamic programming (SSR) algorithm RMS time delay difference error is significantly higher at 0 dB than the RMS time delay difference error for the dynamic programming (SSR) algorithm at

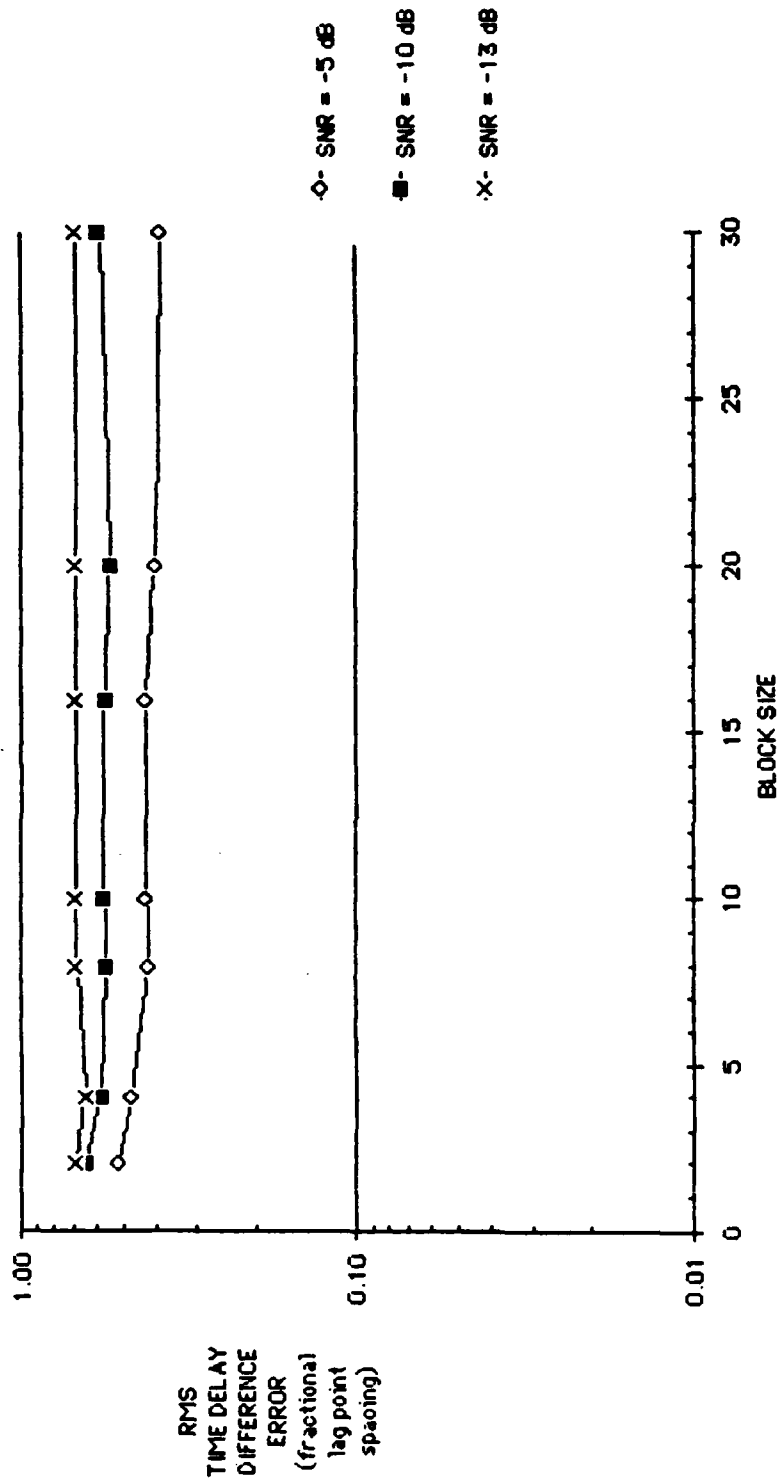


Figure 5-7. RMS Time Delay Difference Error vs Block Size,
Dynamic Scenario, Dynamic Programming
(SSR)(T = 5 sec, K = 0, N = 31)

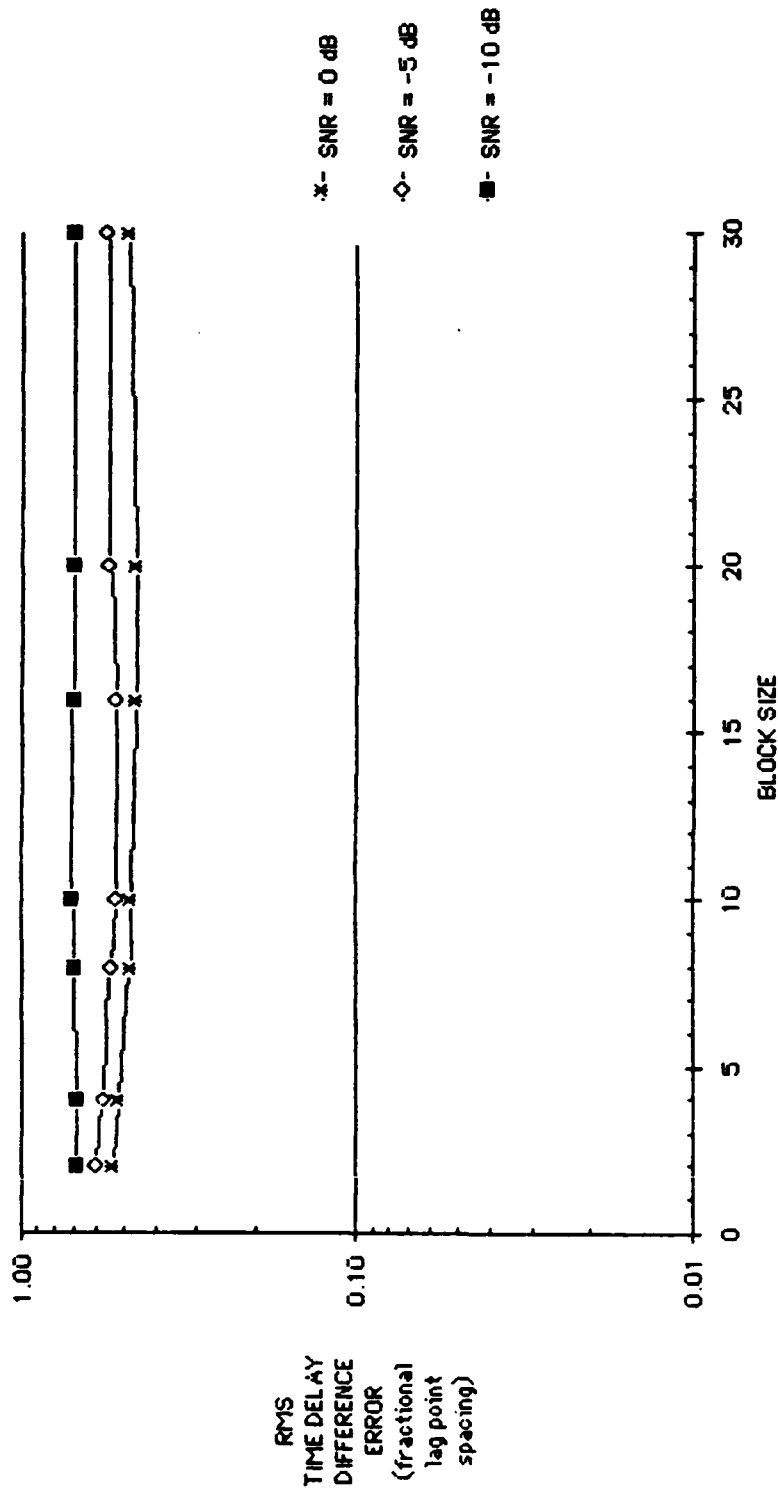


Figure 5-8. RMS Time Delay Difference Error vs Block Size,
Dynamic Scenario, Dynamic Programming
(SSR) ($T = 1$ sec, $K = 0$, $N = 31$)

-15 dB (Figure 5-5) using the static scenario. In fact, the dynamic programming (SSR) algorithm diverges at -15 dB using the dynamic scenario. This demonstrates that results obtained using the unrealistic static scenario will have little validity in practice for the dynamic programming (SSR) algorithm. Finally we should note from Figures 5-7 and 5-8 that the increased integration time from 1 second to 5 seconds reduces the corresponding RMS time delay difference errors. Since there is relatively small motion in 5 seconds there should be little coherency loss. Therefore, as would be expected, longer integration time will reduce the estimator variance.

Our next step is to repeat the above dynamic scenario simulations with the addition of the a priori information. Figure 5-9 shows the RMS time delay difference error versus block size with $K = 0.5$ and an integration time of 5 seconds for five different SNR levels (+5 dB, -5 dB, -10 dB, -15 dB, and -18 dB). Our first observation is that for high SNR (+5 dB and -5 dB) the block size has little effect. However, there is a significant RMS time delay difference error reduction from the corresponding RMS time delay difference error plots (Figure 5-7) for the ML estimate only ($K = 0$) simulations. Although the a priori information does reduce the discretation errors at high SNR, the block size has little effect on this reduction. At low SNR levels, Figure 5-9 indicates that at first longer block sizes decrease the RMS time delay difference error. However, at the longest block sizes of 20 and 30, the RMS error is increased. One possibility is that the proportionality constant K is a function of the block size. Although our algorithms are normalized for the block size, anything which would affect the validity

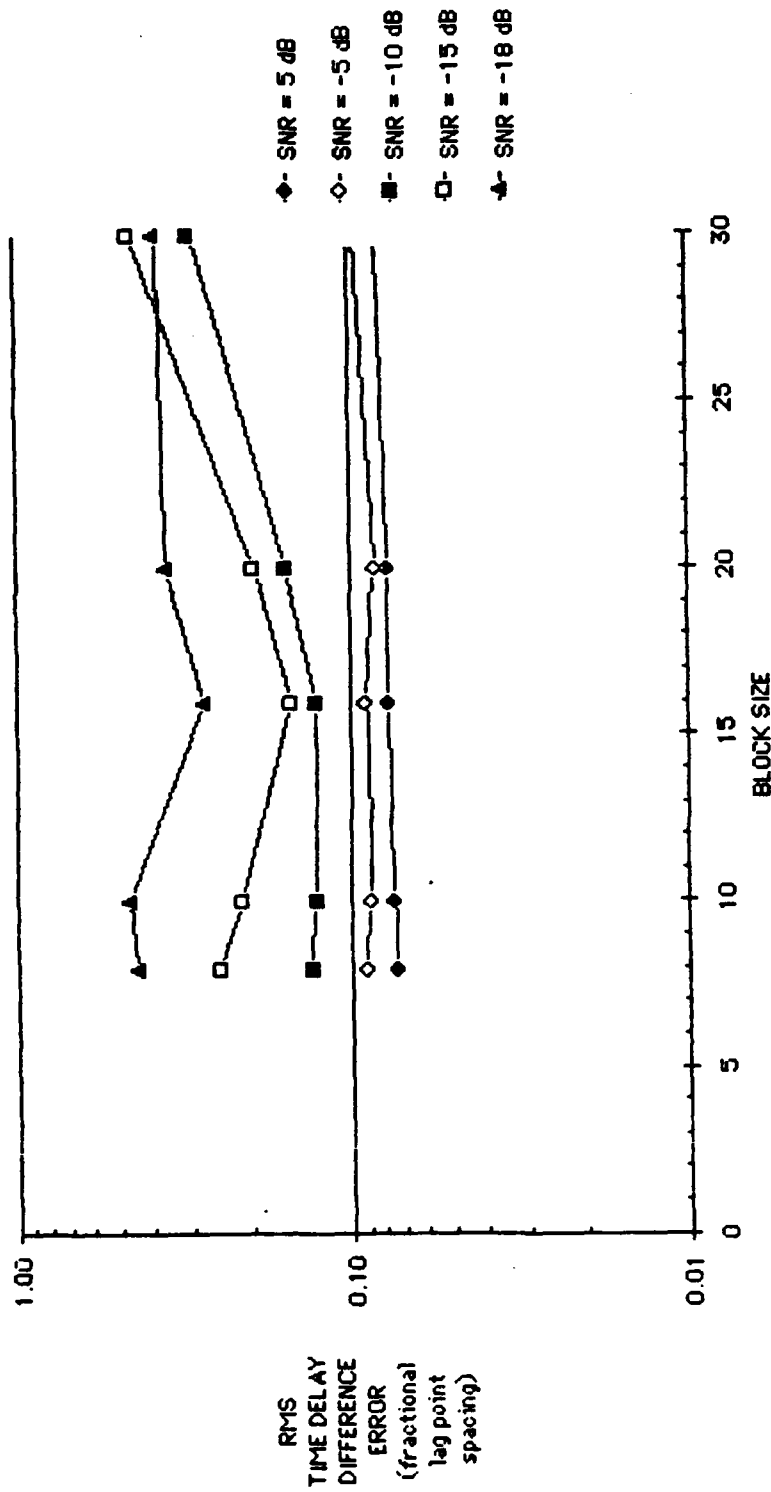


Figure 5-9. RMS Time Delay Difference Error vs Block Size,
Dynamic Scenario, Dynamic Programming
(SSR) ($T = 5$ sec, $K = 0.5$, $N = 31$)

or precision of either the ML estimates or the a priori information function will affect the proportionality constant K . Certainly for long block sizes modeling mismatches may compromise the a priori information function. To test this idea we shall plot, similar to Figure 5-3, RMS time delay difference error versus proportionality constant K for SNR levels of 5 dB, -5 dB, -10 dB, -15 dB, and -18 dB. However, in Figure 5-10, we shall fix the block size at 20 and in Figure 5-11 we shall fix the block size at 30.

Examining Figures 5-3, 5-10, and 5-11, we can see that the parameter K is a function of block size. In order to highlight this dependency, we shall plot together in Figure 5-12 the RMS time delay difference error versus parameter K at an SNR = -15 dB for block sizes of 10, 20, and 30. Figure 5-12 indicates that as block size increases the optimum value of K decreases and RMS time delay difference error degrades at lower values of K . If increased block sizes resulted in a modeling mismatch, it would be reasonable to expect that degradation due to over-dependence on the a priori information would occur at lower value of K . These conclusions are not valid at all SNR levels, indicating that K is a complex function of SNR and block size.

In order to validate the conclusions on data block size obtained from Figure 5-9, we shall now plot in Figure 5-13 the RMS time delay difference error versus block size using the optimal values of K found in Figures 5-3, 5-10, and 5-11. Although the increase in RMS time delay difference error for the block size of 30 is not as great at low SNR, Figure 5-13 basically agrees with Figure 5-9.

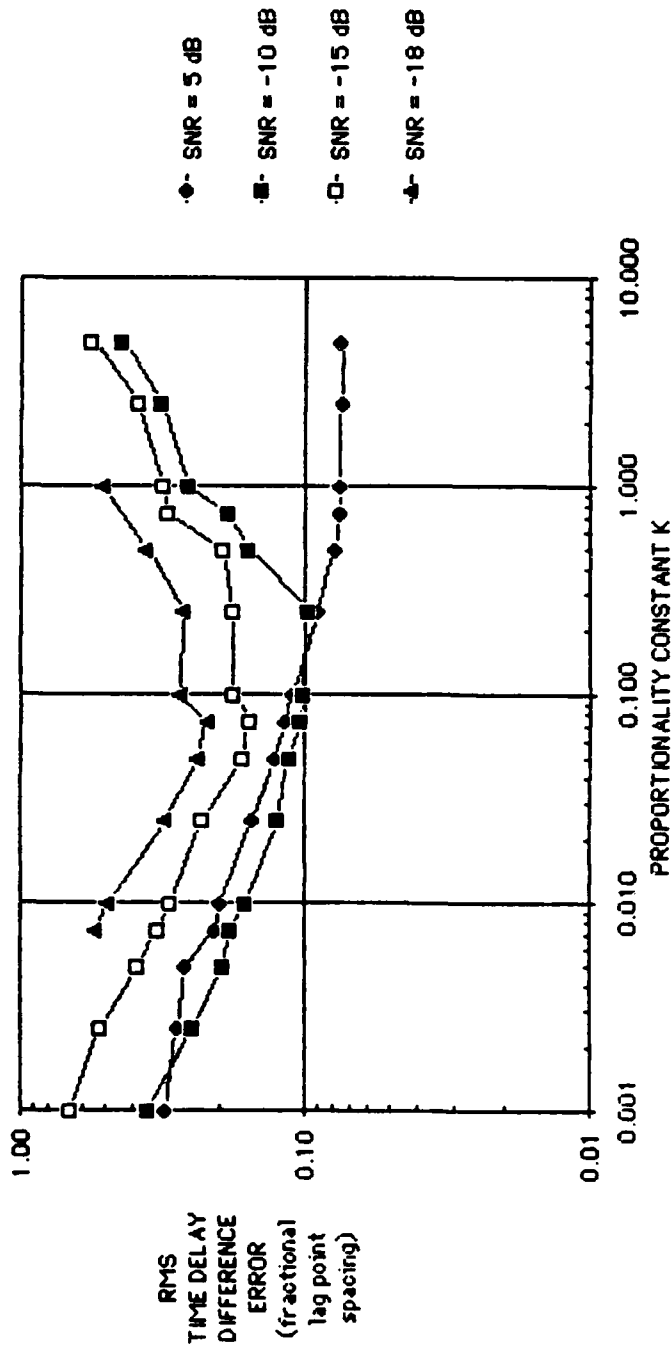


Figure 5-10. RMS Time Delay Difference Error vs Proportionality Constant K, Dynamic Program SSR ($M = 20$, $T = 5$ sec, $N = 31$)

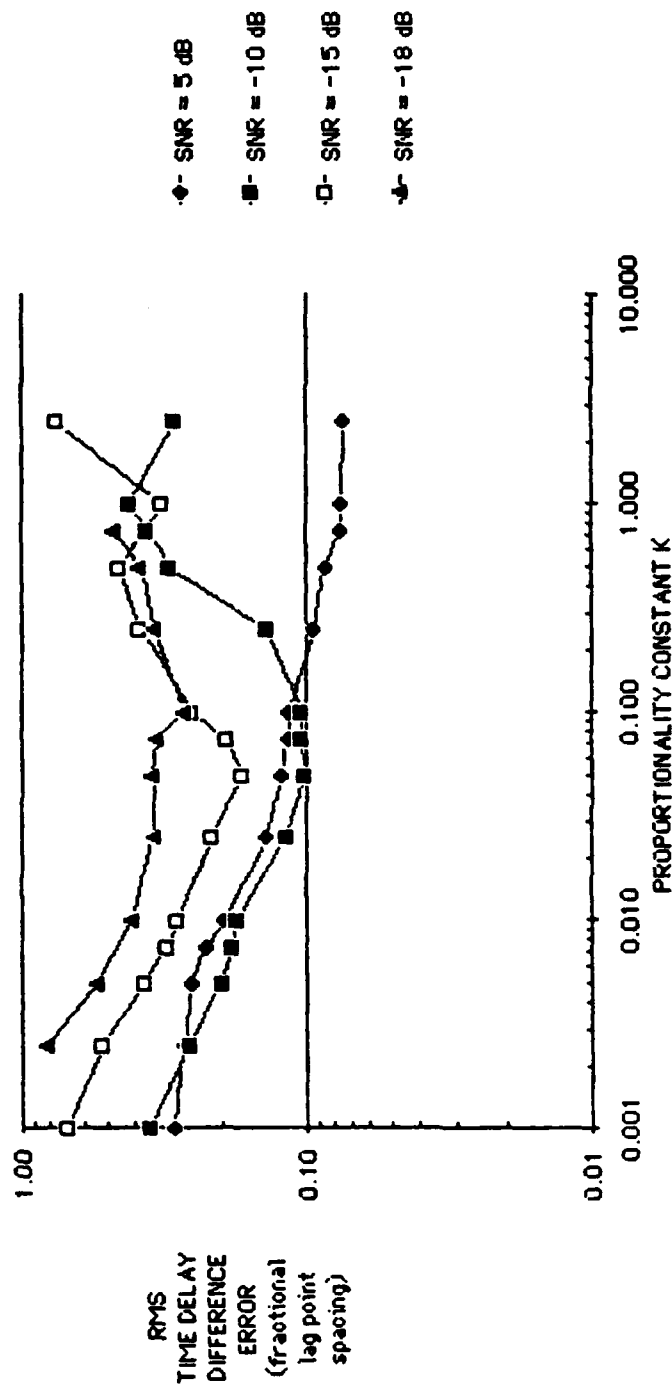


Figure 5-11. RMS Time Delay Difference Error vs Proportionality Constant K, Dynamic Program SSR ($M = 30$, $T = 5$ sec, $N = 31$)

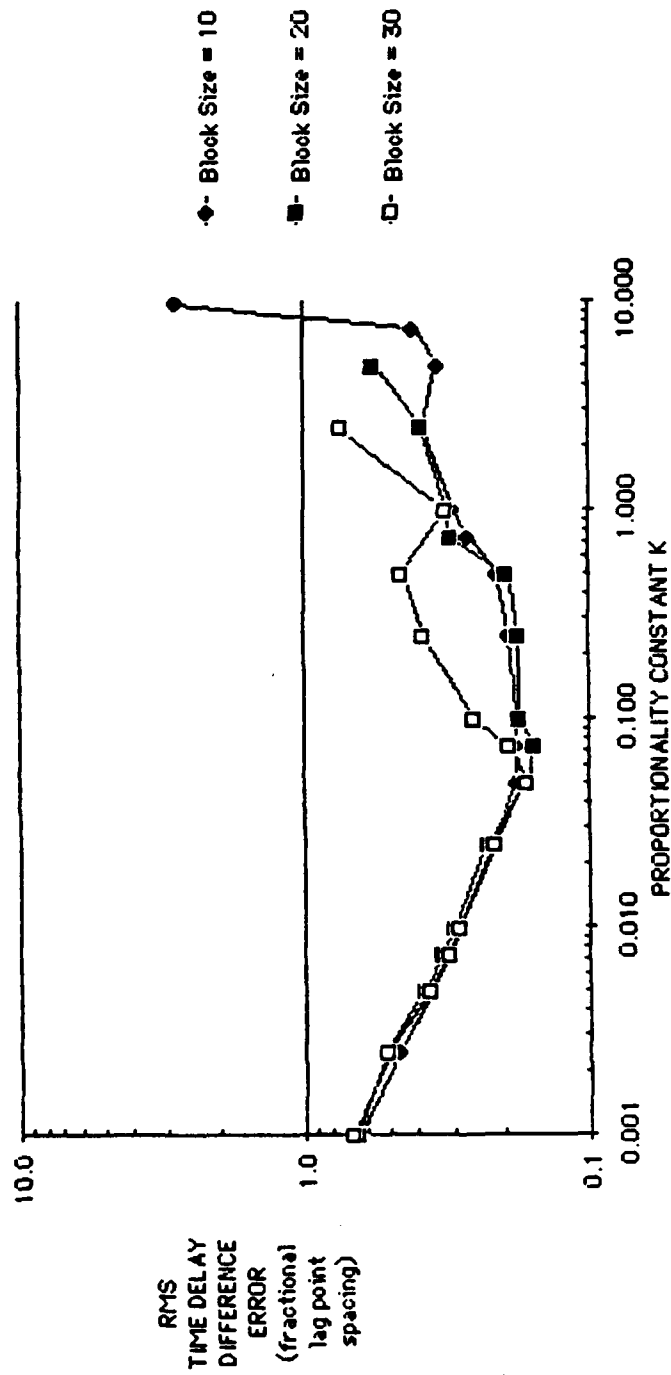


Figure 5-12. RMS Time Delay Difference Error vs Proportionality Constant K, Dynamic Program SSR
(SNR = -15 dB, T = 5 sec, N = 31)

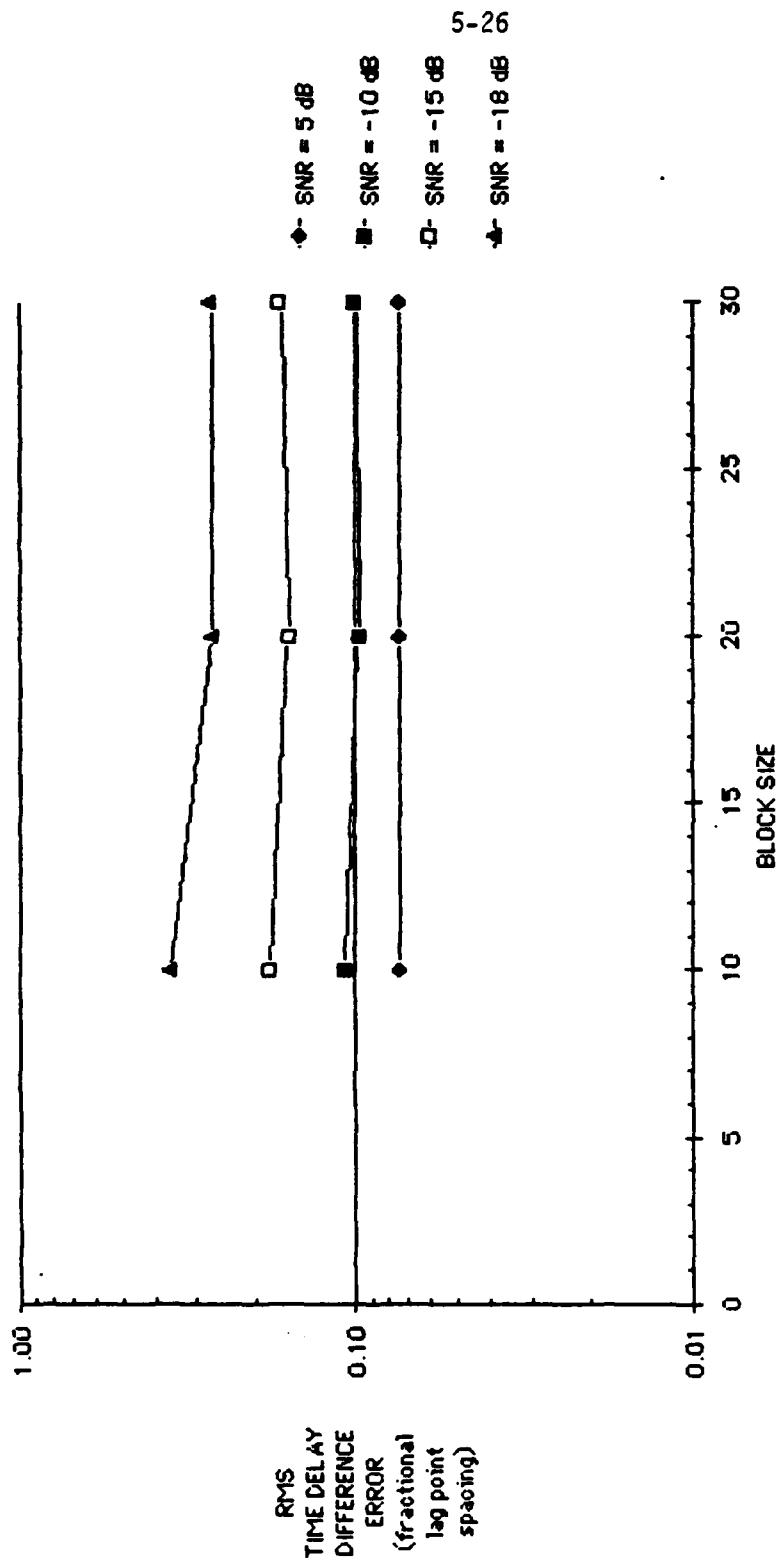


Figure 5-13. RMS Time Delay Difference Error vs Block Length,
Dynamic Scenario, Dynamic Programming
(SSR) ($T = 5$ sec, Optimal K , $N = 31$)

Therefore, we may conclude that there is an optimal block size. Since the target scenario and block size affect the validity of our assumed quadratic time delay difference model, we may conclude that longer block sizes violate our model. However, examining Figure 5-2, we can see that the time delay difference trajectory looks reasonably quadratic for time intervals as large as $5 \times 30 = 150$ seconds.

In order to resolve the above question, we shall reduce the integration time from 5 seconds to 1 seconds. Now our maximum block size is only 30 seconds and we should have minimal modeling mismatches.

Figure 5-14 shows the RMS time delay difference error versus block size with $K = 0.5$ and an integration time of 1 second. Basically, Figure 5-14 with an integration time of 1 second agrees with its counterpart Figure 5-9 with an integration time of 5 seconds. At high SNR there is little dependence on block size. At low SNR there is at first little dependence on increasing block size, but for the highest block size of 30 the RMS time delay difference error increases.

We shall confirm the above observations for a fixed value of K by replotting Figure 5-14 using the optimal values of K . Figure 5-15 contains the resulting plots for SNR levels of -10 dB and -15 dB. For the -15 dB case the search for the optimum was marginal with many values of K losing track.

Examining Figure 5-15 we can see that the -10 dB case agrees with Figure 5-14. At block sizes of 10 and 20 there is little difference in

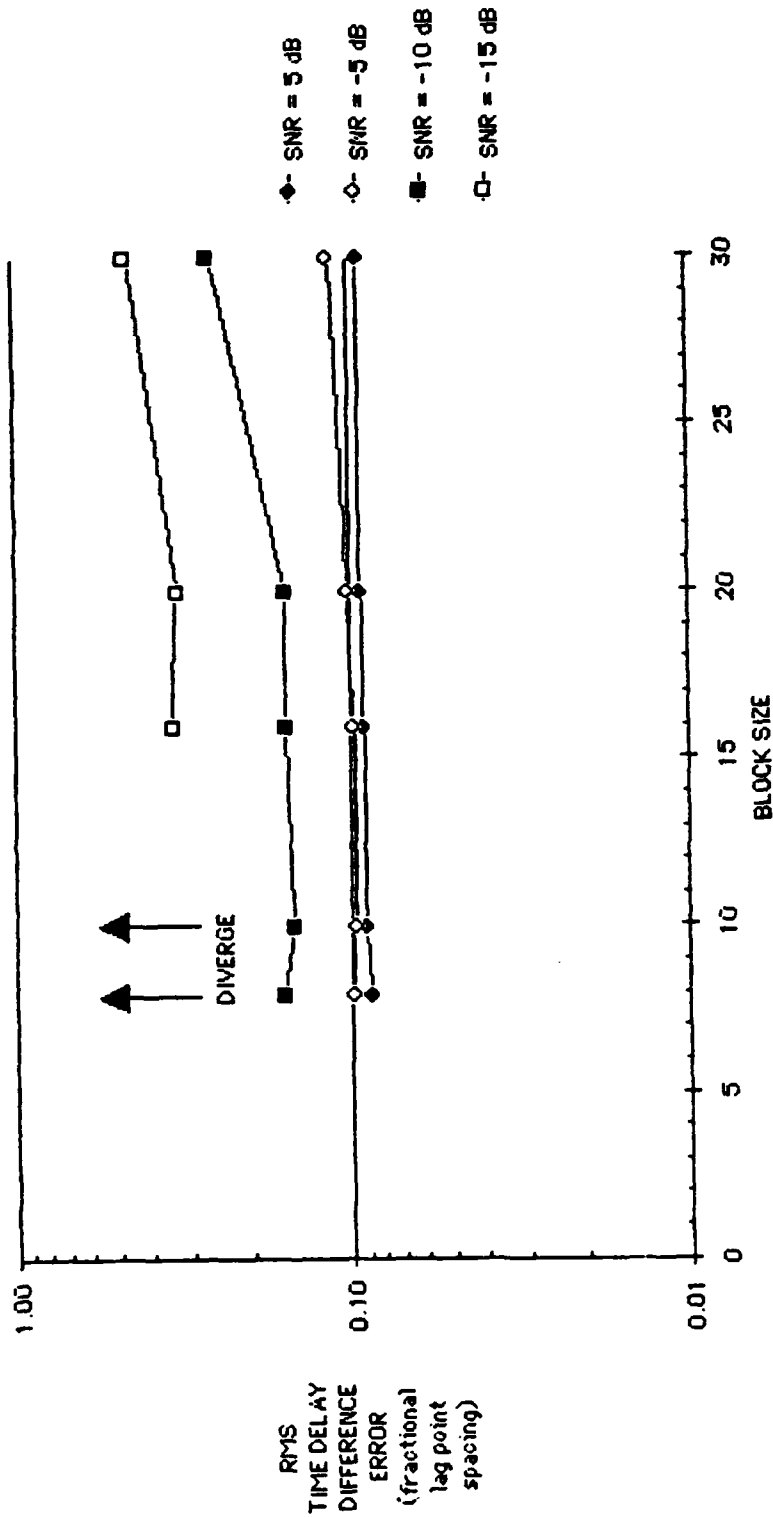


Figure 5-14. RMS Time Delay Difference Error vs Block Size,
Dynamic Scenario, Dynamic Programming
(SSR) ($T = 1$ sec, $K = 0.5$, $N = 31$)

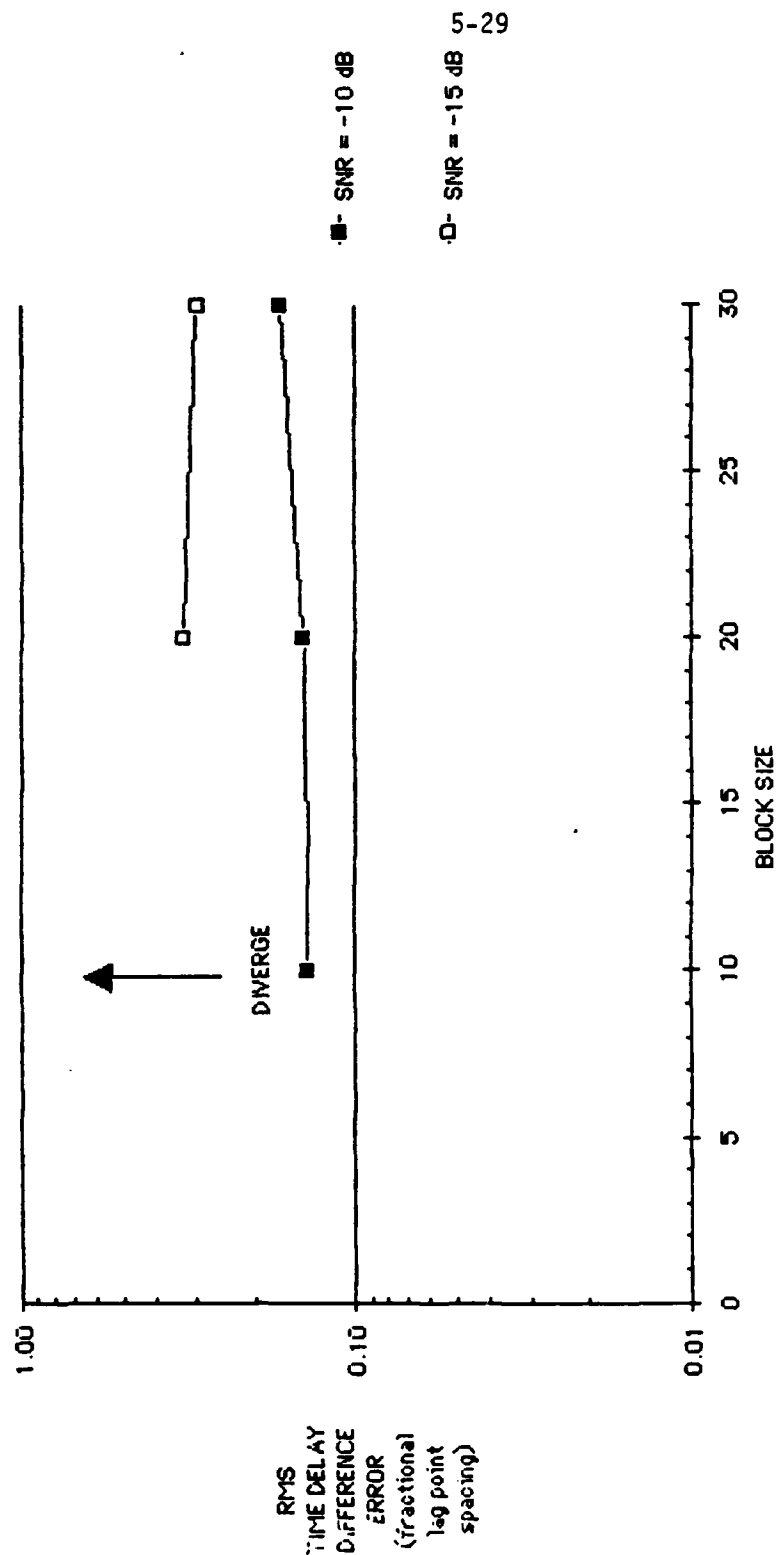


Figure 5-15. RMS Time Delay Difference Error vs Block Size,
Dynamic Scenario, Dynamic Programming
(SSR) ($T = 1$ sec, Optimal K , $N = 31$)

the RMS time delay difference error but a small but noticeable larger RMS time delay difference error occurs at the block size of 30. For the -15 dB case we note that the block size of 10 did not track for any value of K . In addition, the block size of 30 now shows an RMS time delay difference error decrease from the block size of 20.

The results of the simulations of RMS time delay difference error versus block size for an integration time of 1 second are somewhat inconclusive. For the -15 dB case increased block size certainly produces convergence in the estimator. However, at other higher SNR levels, increased block size results initially in no change in the RMS time delay difference error and finally, at the longest block size, an increase in RMS time delay difference error.

In conclusion, for the dynamic programming algorithm there are restrictions on the choice of block sizes. It appears that the optimal choice of block size is dependent on more than target dynamics and potential model mismatches. Since the block size affects the RMS time delay difference error via the LMSF a priori information penalty function, it is reasonable to conclude that numerical problems may be entering computation of the a priori information penalty function for large block sizes. Regardless, it would be unreasonable to assume that increasing the number of unknowns and calculation in numerical algorithm will not eventually lead to computational problems.

We shall now examine the effects of block size on the gradient search algorithm. Since the gradient search algorithm yields continuous estimates, we will at least be relieved of the complication of the discretation error.

Once again to initially simplify the experiment, we shall use the static geometry at an SNR equal to -15 dB and we shall look at the ML only ($K = 0$) estimator. Figure 5-16 gives the RMS time delay difference error versus block size for the above experiments. Examining Figure 5-16, we notice that something is wrong. For a block size of 30 there is a significant increase in RMS time delay difference error. Since the piecewise linear assumption is common to both the gradient search and dynamic programming algorithms, we would expect that long block size should not effect the RMS time delay difference error. However, as will become more apparent as we proceed, the gradient search algorithm has increasing numerical problems with block size. The obvious problem is that as the block size increases, so does the dimension of the numerical search. As expected, we notice a significant increase in RMS time delay difference error at very short block sizes of 2 and 4, and possibly 8. This we may attribute to the weak coupling of the piecewise linear assumption.

Our next step is add a priori knowledge. Therefore, similar to the dynamic programming experiment, we shall set $K = 0.5$ and repeat the experiment at an SNR = -18 dB. Figure 5-17 gives the RMS time delay difference error versus block size for the above experiment. We notice that RMS time delay difference error generally decreases with block

-A- SNR = -15 dB

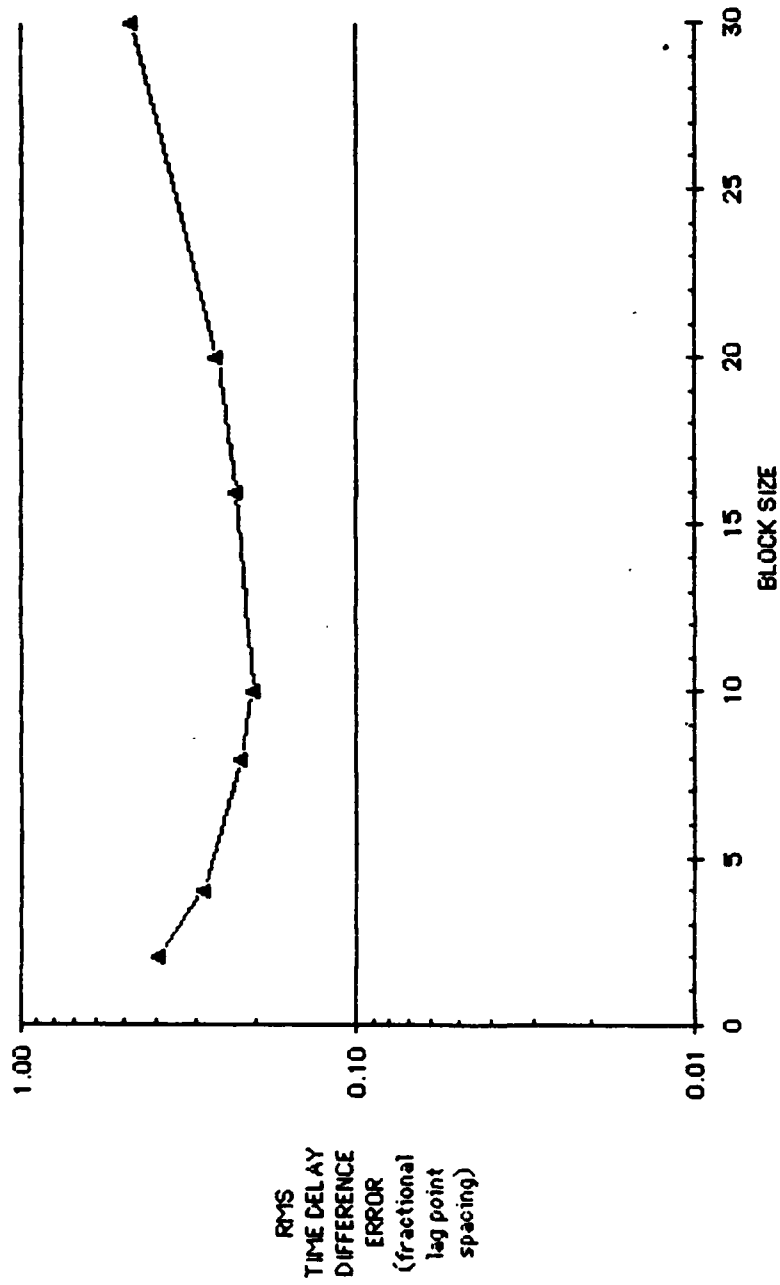


Figure 5-16. RMS Time Delay Difference Error vs Block Size, Gradient Search, Static Geometry
($T = 5$ sec, $K = 0$, $N = 9$)

▲ SNR = -18 dB

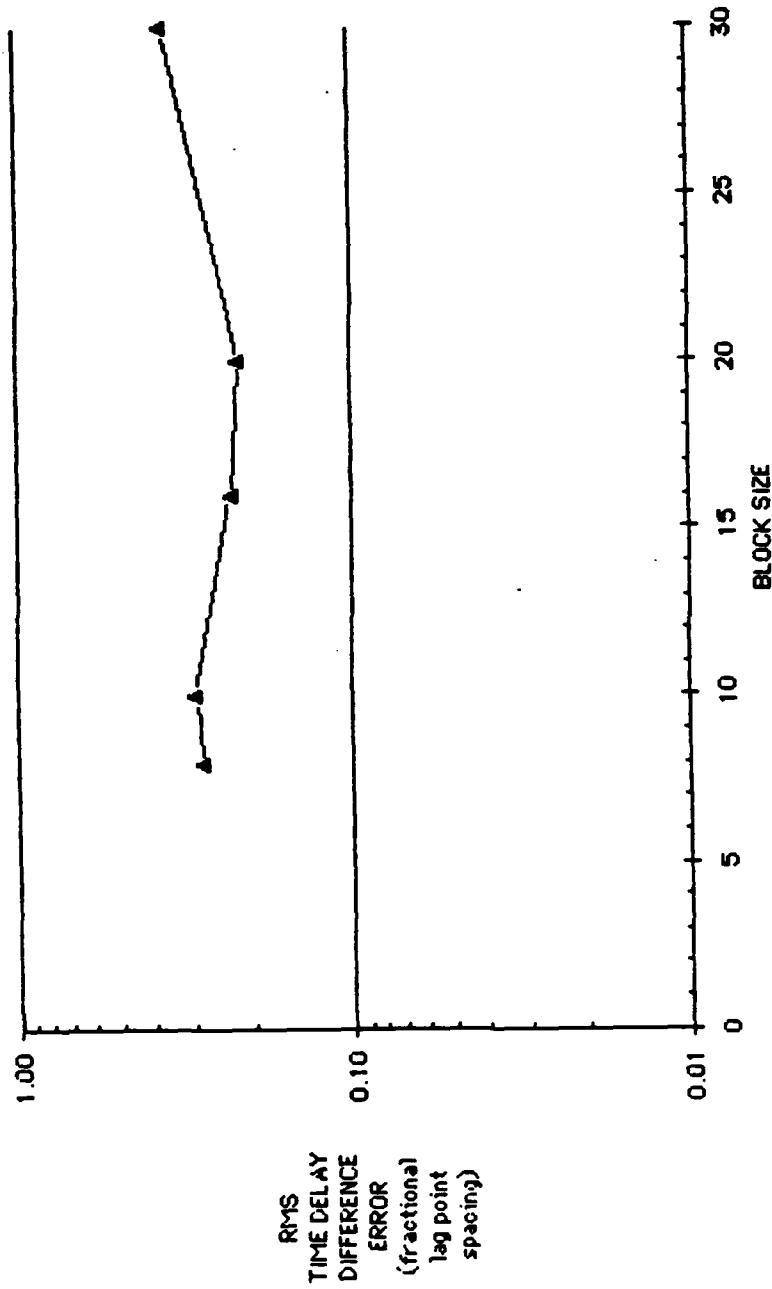


Figure 5-17. RMS Time Delay Difference Error vs Block Size, Gradient Search, Static Geometry ($T = 5$ sec, $K = 0.5$, $N = 9$)

size with a noticeable exception at a block size of 30. We should not be deceived by the apparently improved performance of the dynamic programming algorithm (Figure 5-6) at certain block sizes. As stated earlier, the dynamic programming algorithm has the artificial advantage of having the target always right on a discretized point. This experiment confirms that something is happening to the gradient search algorithm at long block sizes even for a very nice geometry.

We shall now perform experiments using the more realistic dynamic target geometry shown in Figure 5-2. Figure 5-18 shows the RMS time delay difference error versus block size for the gradient search ML estimator ($K = 0$) for an integration time of 5 seconds at three different SNR levels (-5 dB, -10 dB, -15 dB). Examining Figure 5-18 we can see the sharp increase in RMS time delay difference error with block size. This sharp increase occurs at lower block sizes at higher SNR levels. A reasonable cause for the SNR dependency can be seen from taking the derivative of the right-hand expression of Equation 3-10^{*} (ML term) with respect to the parameter vector $\underline{A}$.

$$\frac{\partial}{\partial \underline{A}} [\text{ML term}] = (M/(M - W))^2 \frac{\partial W}{\partial \underline{A}} \quad (5-2)$$

* For both the gradient search and dynamic programming algorithms, the right-hand expression of Equation 3-10 is used as the ML term in Equation 3-104.

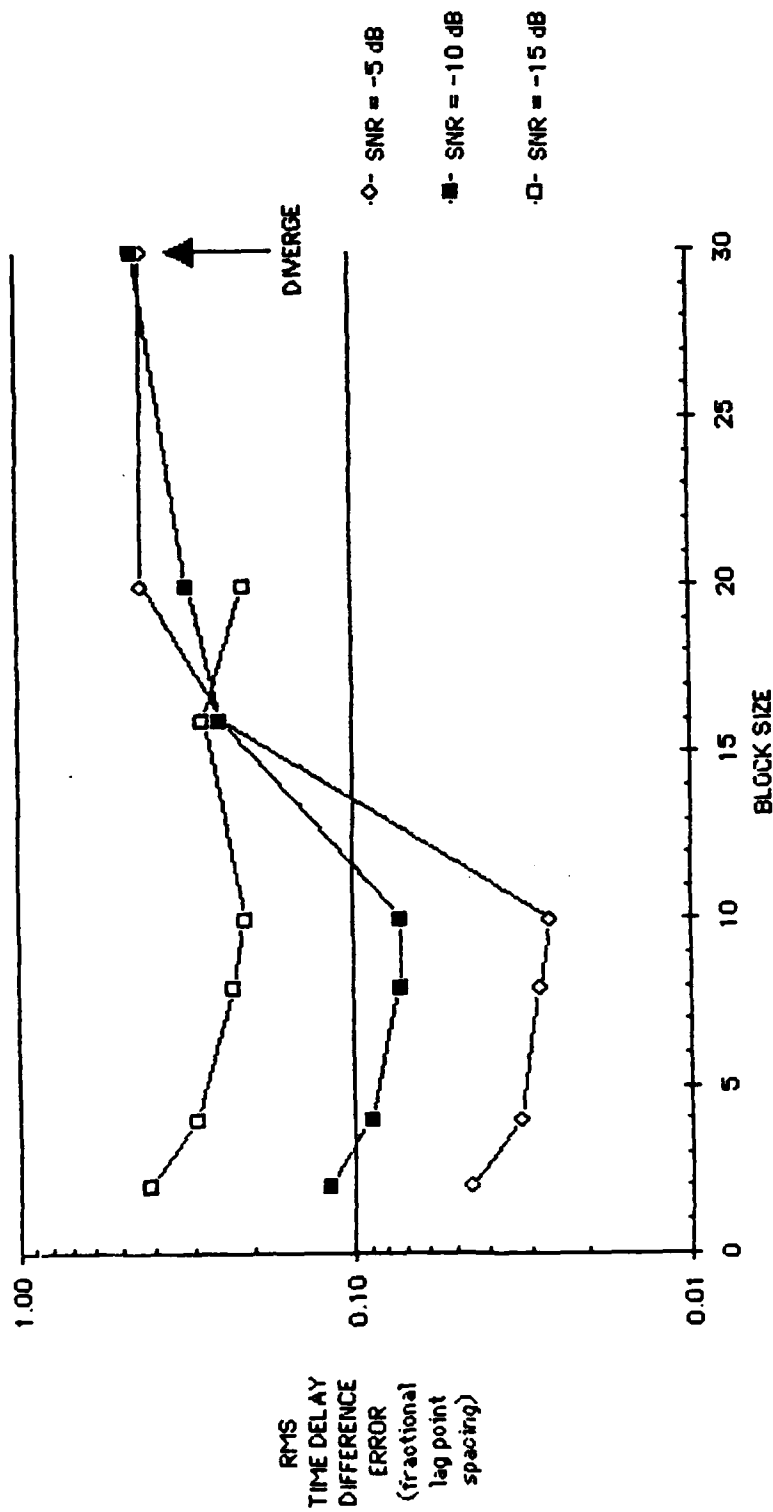


Figure 5-18. RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Geometry ($T = 5$ seconds, $K = 0$, $N = 9$)

where

W = the ML expression given in Equation 3-108

M = the block size

$\underline{A}$ = a parameter vector representing the target time delay difference trajectory.

At high SNR W approaches M and therefore Equation 5-1, representing a portion of the gradient vector, becomes very sensitive to numerical problems.

Note in Figure 5-18, that for block size equal to 20, the RMS time delay difference error actually decreases as SNR is decreased! A reasonable cause for this would be if the numerical problems dominated the nominal RMS time delay difference error and if the numerical problems decreases with SNR level.

At all SNR levels for short block sizes, we notice an increasing RMS time delay difference error with decreasing block size. This we attribute to the weak piecewise linear assumption and the resulting loose correlation of data points.

We shall introduce a priori information ($K = 0.5$) into the dynamic scenario simulation. Figure 5-19 shows the RMS time delay difference error versus block size for the gradient search algorithm at SNR levels of 5 dB, -5 dB, -10 dB, -15 dB, -18 dB, and -20 dB. The integration time is 5 seconds.

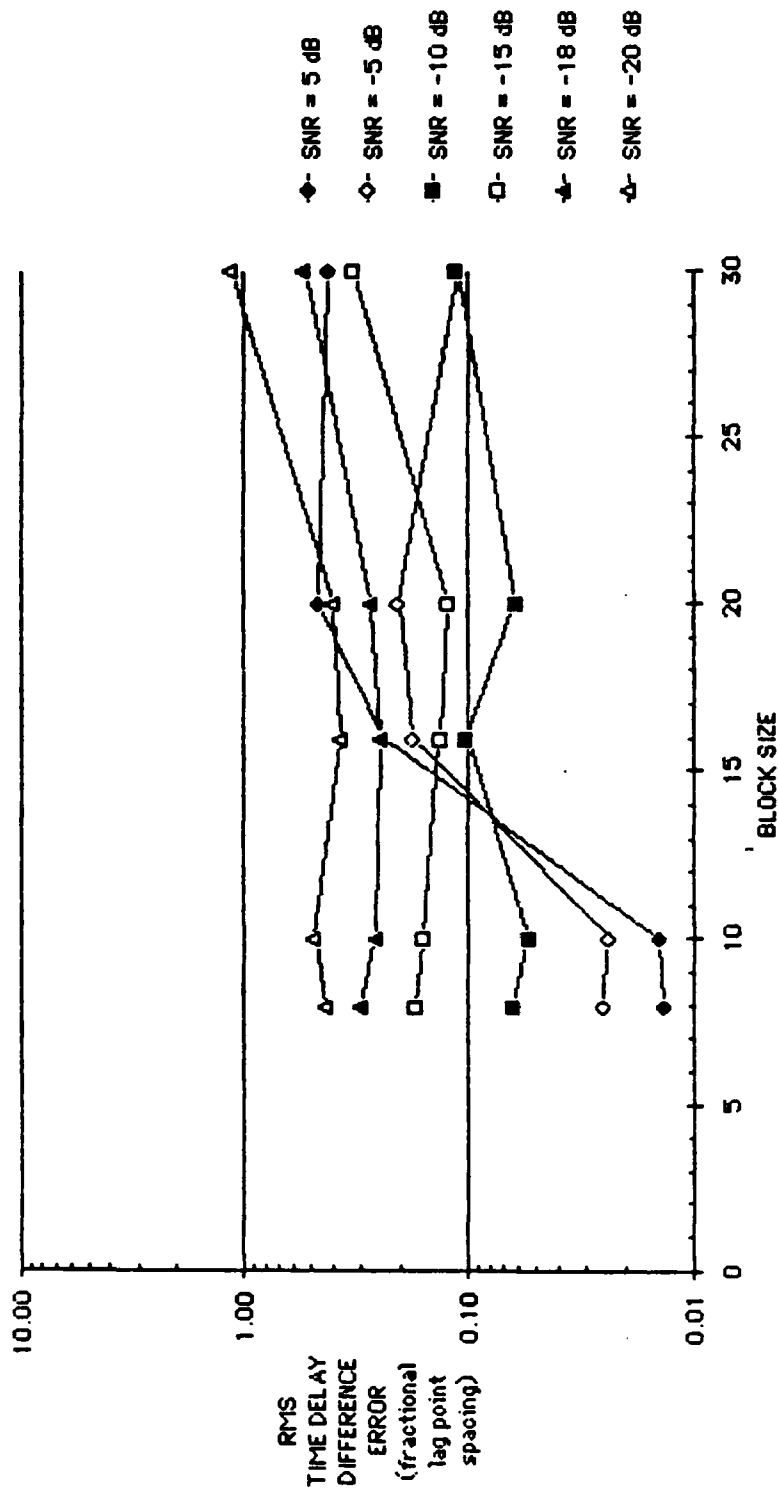


Figure 5-19. RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Scenario ($T = 5$ sec, $K = 0.5$, $N = 9$)

Once again, examining Figure 5-19, we can see, in general, an initial decrease in RMS time delay difference error with increased block size, but at the longest block sizes the RMS time delay difference error increases. At the higher SNRs (+5 dB, -5 dB), the increase in RMS time delay difference error occurs at shorter block sizes. These results are effectively identical with those obtained for $K = 0$ (Figure 5-18). However, as would be expected, the addition of a priori knowledge reduces the RMS time delay difference error.

We shall now see what improvement can be obtained by using the optimum value of K^* for each block size at SNR levels of +5 dB, -10 dB, -15 dB, and -18 dB. Figure 5-20 plots the RMS time delay difference error versus block size for the optimum choices of K . The integration time is 5 seconds. Figure 5-20 also gives the optimal values of K used.

From Figure 5-20 we can see that using the optimal value of K significantly reduces the RMS time delay difference error at the longer block sizes. However, at the longest block size of 30 the RMS time delay difference error is always greater than the shorter block sizes for all SNR levels. At the higher SNR levels (+5 dB, -5 dB), there is an increase in RMS time delay difference error from a block size of 10 to 20 to 30. At the lower SNR levels (-15 dB, -18 dB), the RMS time delay difference error decreases from a block size of 10 to a block size of 20, but increases at the block size of 30. These results indicate

* The values of K were the best found.

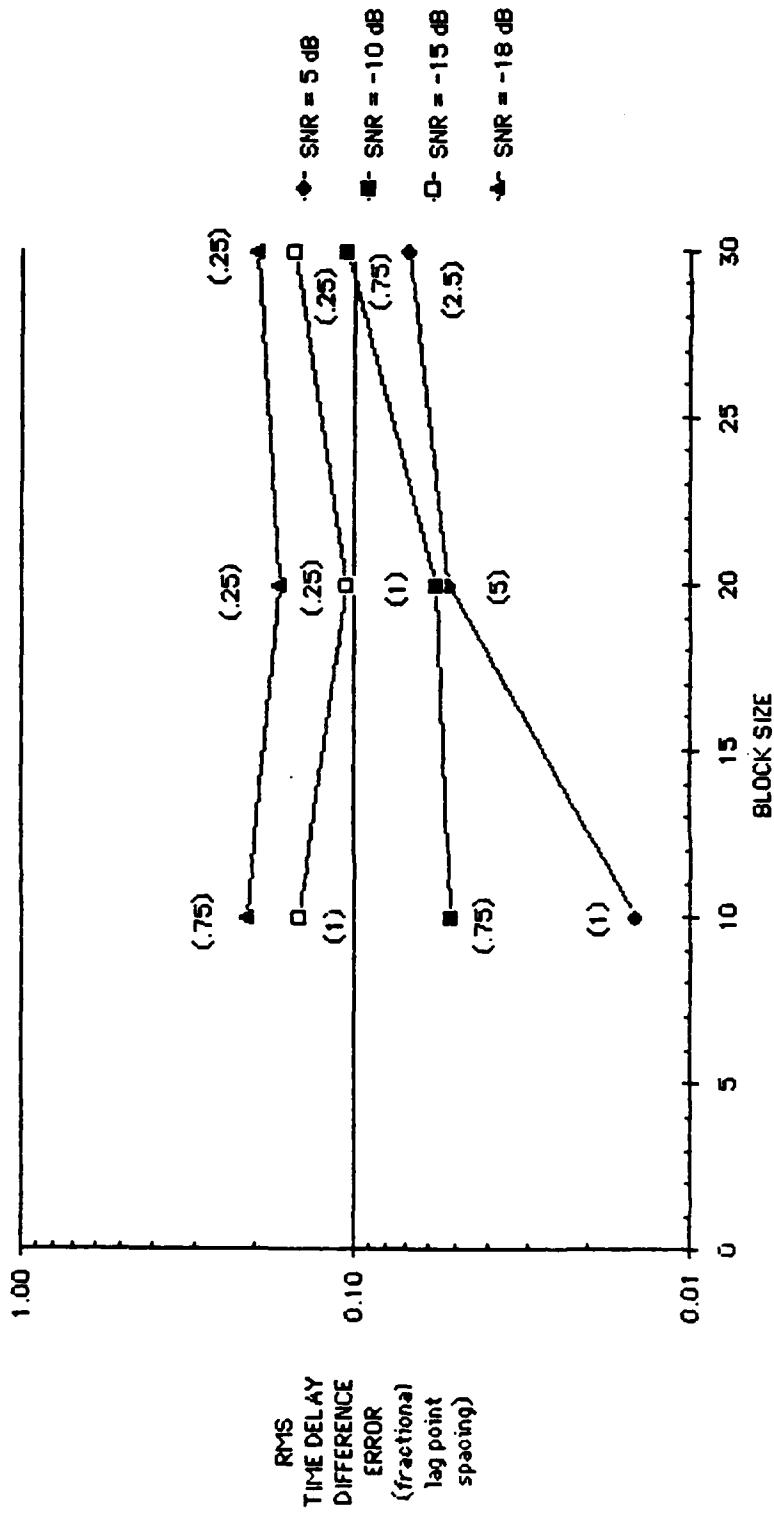


Figure 5-20. RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Scenario ($T = 5$ sec, Optimal K , $N = 9$)

that increasing the weighting of the a priori information can significantly reduce the effects of numerical problems at longer block sizes and high SNR (+5 dB, -10 dB). At low SNR levels (-15 dB, -18 dB), decreasing the weight of the a priori information yields better results. The increase in RMS time delay difference error at the longest block size of 30 indicates potential modeling mismatch or numerical problems.

We shall now reexamine our gradient search algorithm performance as a function of block size with an integration time of 1 second. At this short integration time our modeling errors should be minimum. We shall use the realistic scenario described in Figure 5-2 for all simulations.

Figure 5-21 shows the RMS time delay difference error versus block size for the ML estimator only ($K = 0$). The results of Figure 5-21 show large increases in RMS time delay difference with increased block size, especially at high SNR (-5 dB). This result is the same as Figure 5-18 for an integration time of 5 seconds. Therefore, as would be expected, we still have a numerical instability at long block sizes and high SNR. Note, however, the RMS time delay difference errors in Figure 5-21 are more than the corresponding RMS time delay difference errors in Figure 5-18 (at least for block sizes not totally dominated by numerical problems). At short block sizes we notice the same increase in RMS time delay difference error with decreasing block size as we observed in Figure 5-18 for an integration time of 5 seconds.

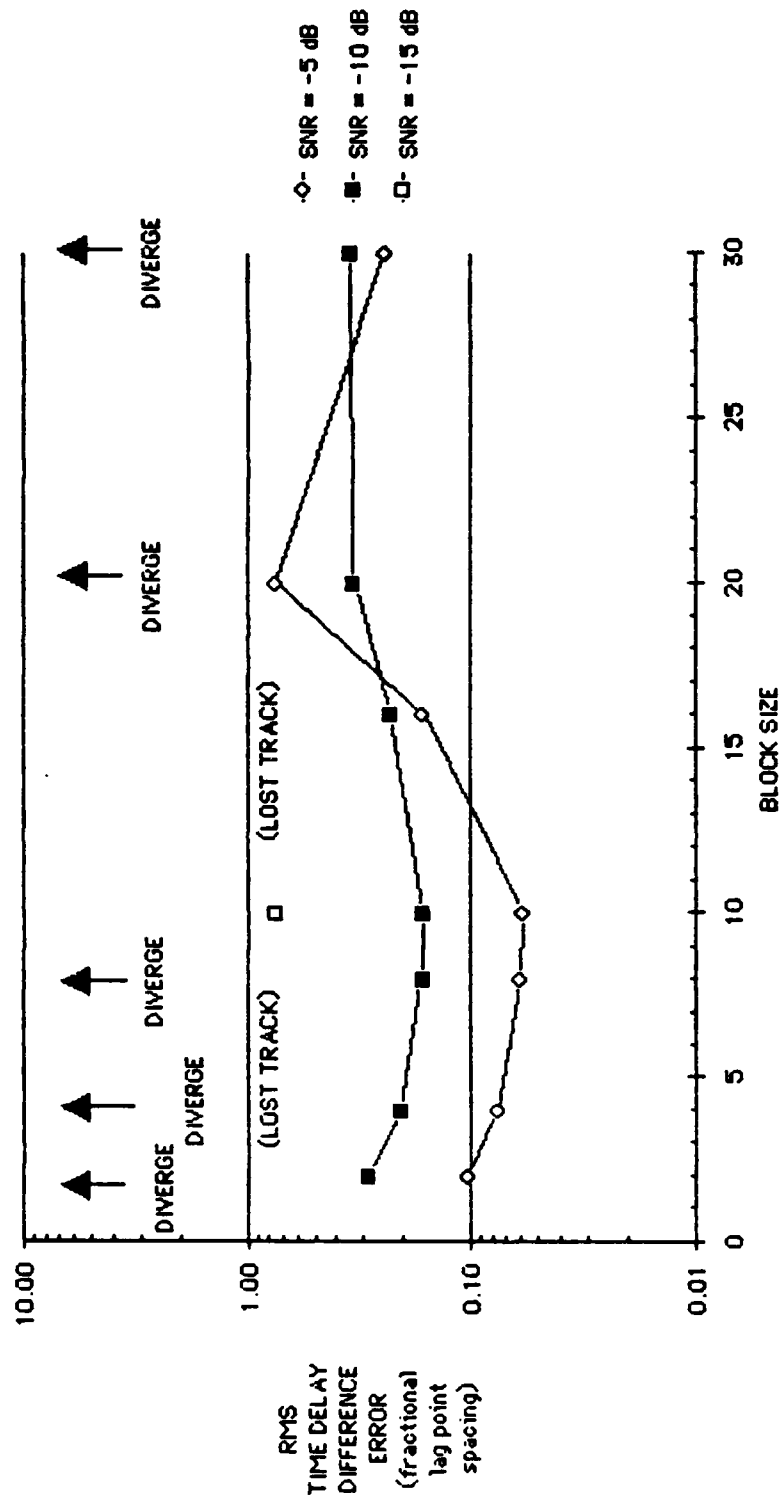


Figure 5-21. RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Scenario ($T = 1$ sec, $K = 0$, $N = 9$)

We shall now add a priori information to our simulation ($K = 0.5$). Figure 5-22 shows the RMS time delay difference error versus block size for our full MAP estimator ($K = 0.5$).

Examining Figure 5-22, we can still see a rather dramatic but reduced increase in RMS time delay difference error with increasing block size at high SNR levels (+5 dB, -5 dB). At -10 dB the RMS time delay difference error decreases for increased block size except at the longest block size 30 where the RMS time delay difference error increases. At -15 dB the gradient search algorithm diverges at block sizes of 8 and 10 and shows an RMS time delay difference error reduction going from a block size of 20 to a block size of 30. The above indicates that the numerical problems still exist at high SNR for the 1 second integration. However, at lower SNR the RMS time delay difference error tends to decrease with block size. Since the shorter integration time would reduce modeling mismatches we might expect this.

We shall now replot Figure 5-22, but now using RMS time delay difference error corresponding to the optimal value of K found. Figure 5-23 shows the results of the above simulations for block sizes of 10, 20, and 30. The optimal values of K are shown in brackets below the plotted points.

Examining Figure 5-23, we see that increasing block size now decreases RMS time delay difference error for all but the highest SNR level (+5 dB). The optimal choices of K generally indicate that larger values of K are required at high SNR and also at shorter block sizes.

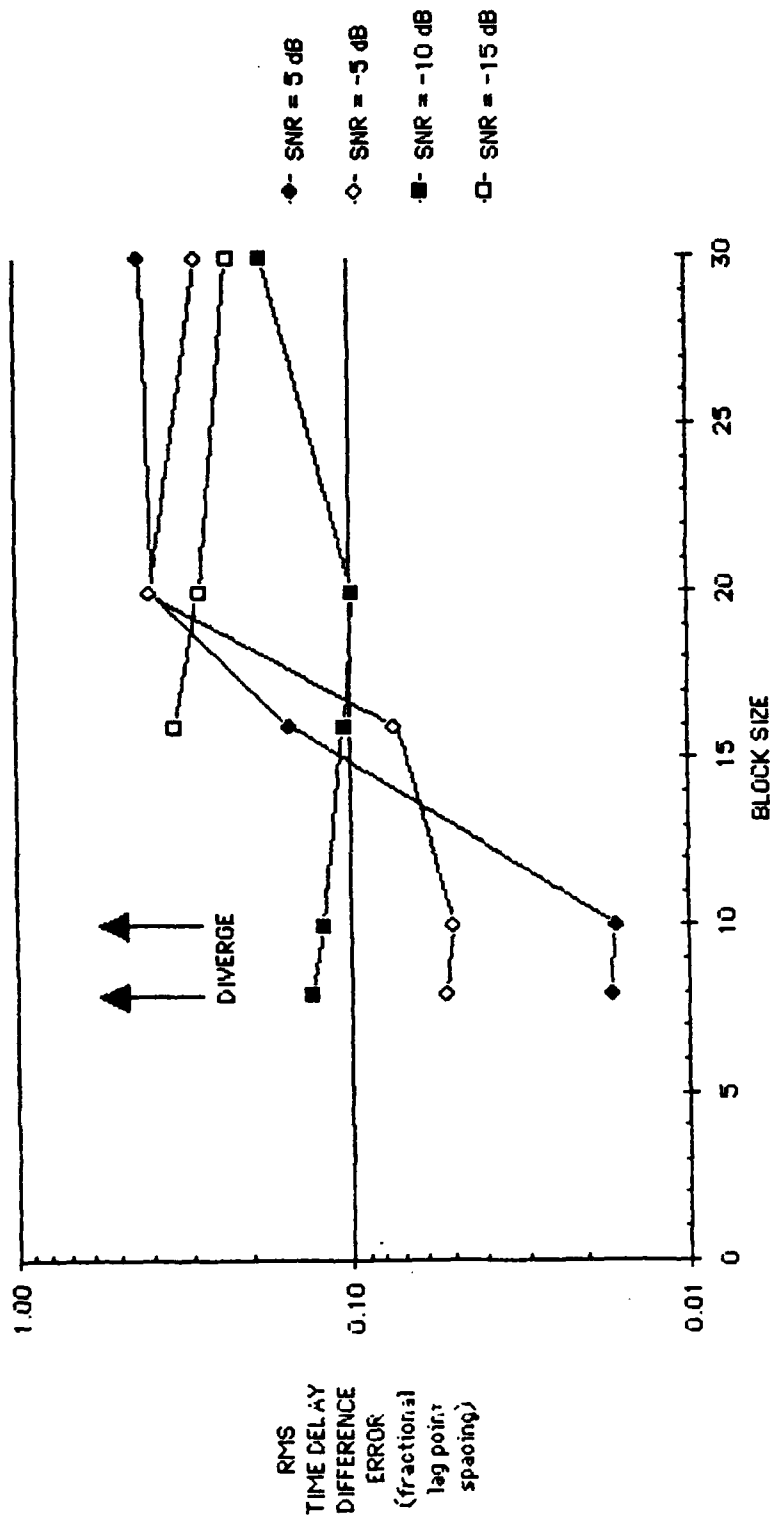


Figure 5-22. RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Scenario ($T = 1$ sec, $K = 0.5$, $N = 9$)

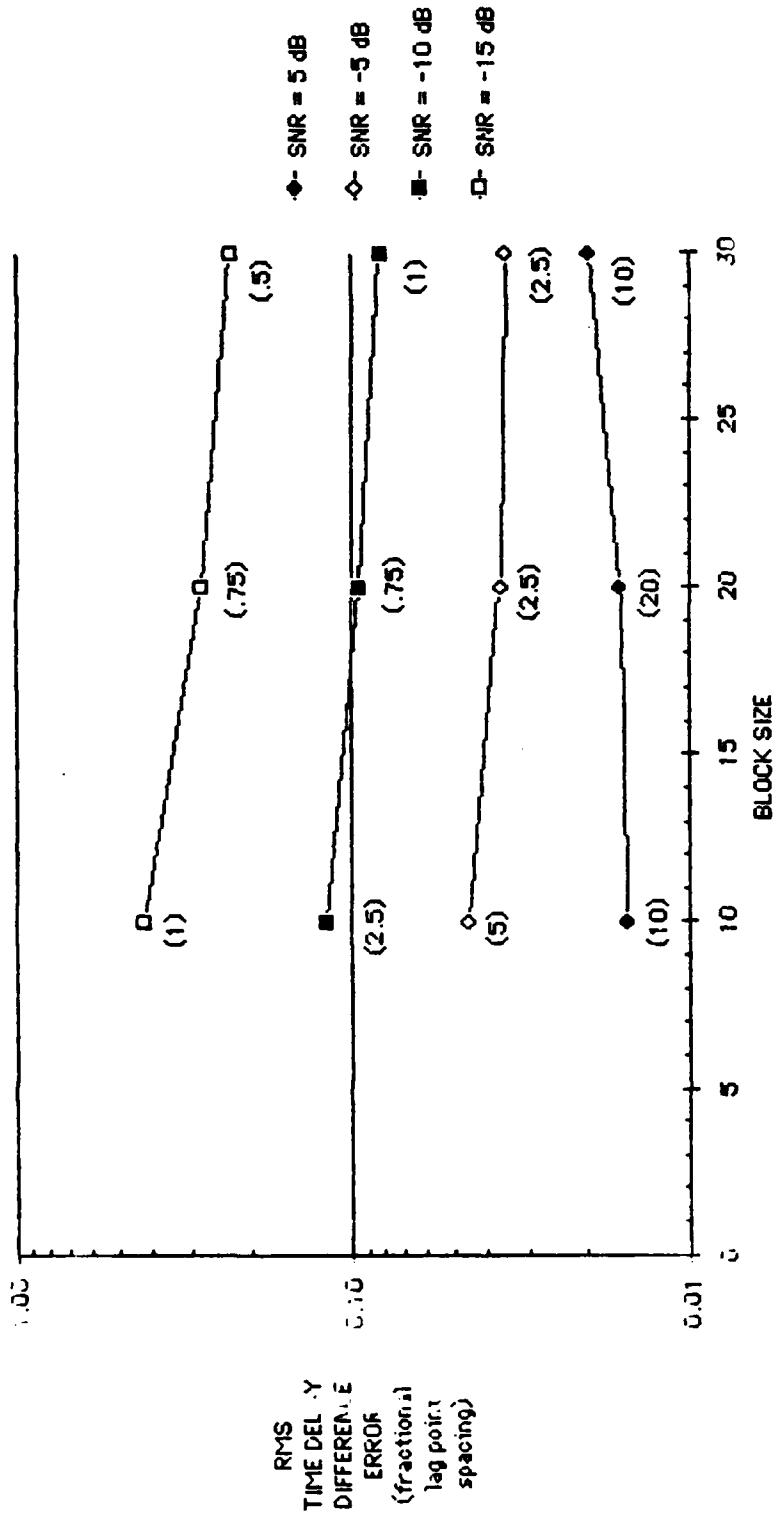


Figure 5-23. RMS Time Delay Difference Error vs Block Size, Gradient Search, Dynamic Scenario ($T = 1$ sec, Optimal K , $N = 9$)

The numerical problems at high SNR evidently require more than the expected smoothing from the a priori information.

Figure 5-20, using optimal values of K for an integration time of 5 seconds, always shows an increase in RMS time delay difference error at the longest block size. The corresponding Figure 5-23 for the integration time of 1 second effectively shows a decrease in RMS time delay difference error with longer block size. From this we may conclude that some of the increased RMS time delay difference errors at a longer block size for the 5 second integration time are the result of modeling mismatches. Note that at several SNR levels at the longer block sizes of 20 and 30 the 1 second integration time has a lower RMS time delay difference error than the corresponding 5 second integration time.

In summary, longer block sizes introduce numerical and modeling problems for the gradient search algorithm. At high SNR the numerical problems are more severe, but can be reduced by increasing the weighting (K) on a priori information. Therefore, the optimal choice of block size, integration time is model-dependent with an upper limit determined by numerical stability. To determine the optimum selection will require experimentation.

We will now examine the affect on our estimators due to the number of lag points N utilized in our ML estimator. Since this parameter does not directly affect our a priori information function, we will perform this study on only the ML portion ($K = 0$) of our estimators.

For this parameter we can calculate some special case analytical work to predict our simulation results. We will restrict ourselves to a high SNR assumption and to an ML estimate on a single measurement. For this case we can only obtain theoretical estimates for a single unknown parameter (initial offset Δd_0 or slope change $\Delta \beta T$). Simultaneous theoretical estimates for both parameters cannot be obtained since for a single measurement their resolution is ambiguous.

We will therefore restrict our attention to obtaining theoretical estimates for two simple problems. The first is to estimate the mean and variance of the estimated initial offset $\hat{\Delta d}_0$. For this problem we shall assume that the target motion is zero (i.e., $\Delta \beta T = 0$) and for convenience that the actual target offset is as shown below.

$$\Delta d_0 = 0 \quad (5-3)$$

In the second problem we will estimate the mean and variance of the slope parameter estimate $\hat{\Delta \beta T}$. We will again assume that the actual initial offset $\Delta d_0 = 0$. However, we shall look at several constant choices of the rate parameter $\Delta \beta$ and the observation interval T .

Although the above special cases are very simplified, it is reasonable to expect that results obtained should apply to our more complicated estimators at least at high SNR. By this we mean if varying the number of lag points used significantly affects our simplified estimators, it would be reasonable to assume a similar effect on our related more complicated estimators.

We shall examine first the simpler problem of calculating the statistics of the initial offset $\hat{\Delta d}_0$ (Problem 1). Note, however, that the basic mathematics is effectively identical for both problems.

For convenience we shall rewrite our ML estimator below. From Equation 3-83 we have:

$$\underset{\Delta d_0}{\text{MAX}} \left[\frac{\underline{C}^T \underline{R}}{\sqrt{\underline{C}^T \underline{C}}} \right] = F(\hat{\Delta d}_0) \quad (5-4)$$

where

$\underline{C}$ = the normalized vector matching function

$\underline{R}$ = the vector of normalized measured lag points.

Note that the parameter of interest is the number of components in vectors $\underline{C}$ and $\underline{R}$. Also note that we are maximizing with respect to Δd_0 and that $\Delta \beta T = 0$.

About the optimum ($\Delta d = 0$) we can approximate our estimator $\hat{\Delta d}$ by a standard Newton approximation given below

$$\hat{\Delta d}_0 = - \left(\frac{\partial F}{\partial \Delta d_0} / \frac{\partial^2 F}{\partial \Delta d_0^2} \right)_{\substack{\hat{\Delta d}_0=0 \\ \Delta \beta T=0}} \quad (5-5)$$

We can obtain $\left(\frac{\partial F}{\partial \Delta d_0}\right)_{\substack{\Delta d_0=0 \\ \Delta B T=0}}$ from Equation 4-47.

The result is shown below.

$$\left(\frac{\partial F}{\partial \Delta d_0}\right)_{\substack{\Delta d_0=0 \\ \Delta B T=0}} = \left[\frac{(\underline{C}^T \underline{C}) \left(\underline{R}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right) - \left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right) (\underline{C}^T \underline{R})}{(\underline{C}^T \underline{C})^{3/2}} \right]_{\substack{\Delta d_0=0 \\ \Delta B T=0}} \quad (5-6)$$

We can now take the second partial of Equation 5-6 with respect to Δd_0 to obtain the following

$$\left(\frac{\partial^2 F}{\partial \Delta d_0^2}\right)_{\substack{\Delta d_0=0 \\ \Delta B T=0}} = \left[\begin{aligned} & \left[\left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right) \left(\underline{R}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right) - \left(\frac{\partial \underline{C}^T}{\partial \Delta d_0} \frac{\partial \underline{C}}{\partial \Delta d_0} \right) (\underline{R}^T \underline{C}) \right. \\ & + (\underline{C}^T \underline{C}) \left(\underline{R}^T \frac{\partial^2 \underline{C}}{\partial \Delta d_0^2} \right) - \left(\underline{C}^T \frac{\partial^2 \underline{C}}{\partial \Delta d_0^2} \right) (\underline{R}^T \underline{C}) - (\underline{C}^T \underline{C})^{3/2} \\ & - 3(\underline{C}^T \underline{C})^{1/2} \left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right) \left[(\underline{C}^T \underline{C}) \left(\underline{R}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right) \right. \\ & \left. \left. - \left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right) (\underline{C}^T \underline{R}) \right] \right] / (\underline{C}^T \underline{C})^3 \end{aligned} \right] \quad (5-7)$$

We shall represent the vector of normalized measured lag points $\underline{R}$ by a scaled version of our mismatch vector $\underline{C}$ plus a noise vector $\underline{N}$ (for convenience the scale factor will be 1).

$$\underline{R} = \underline{C} + \underline{N} \quad (5-8)$$

The noise vector $\underline{N}$ is zero mean Gaussian with a covariance matrix $\underline{N}$.

Substituting Equation 5-8 into Equation 5-6 and taking the expected value, the following result is easily obtained.

$$E \left[\left(\frac{\partial F}{\partial \Delta d_0} \right) \hat{\Delta d_0} \right]_{\Delta \hat{d}_0=0, \Delta \hat{B}T=0} = 0 \quad (5-9)$$

We will make the standard assumption at high SNR that $\left(\frac{\partial^2 F}{\partial \Delta d_0^2} \right) \hat{\Delta d_0}$ (Equation 5-7) is insensitive to the noise vector $\underline{N}$ and $\Delta \hat{B}T=0$

has a nominal value equal to its noise free or expected value. Therefore, Equation 5-7 can be rewritten as the following after substituting $\underline{R} = \underline{C}$.

$$\left(\frac{\partial^2 F}{\partial \Delta d_0^2} \right) \hat{\Delta d_0} \Big|_{\Delta \hat{B}T=0} = \left[\left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right)^2 - \left(\frac{\partial \underline{C}^T}{\partial \Delta d_0} \frac{\partial \underline{C}}{\partial \Delta d_0} \right) (\underline{C}^T \underline{C}) \right] / (\underline{C}^T \underline{C})^{3/2} \quad (5-10)$$

(Note that Equation 5-10 oddly does not depend on $\frac{\partial^2 \underline{C}}{\partial \Delta d_0^2}$.)

Therefore noting Equations 5-9 and 5-10 we can obtain the expected value of our estimator $\hat{\Delta d}_0$ from Equation 5-5.

$$E[\hat{\Delta d}_0] = 0 \quad . \quad (5-11)$$

This result states that our estimator is unbiased in the limit at high SNR.

The variance of $\hat{\Delta d}_0$ can be obtained from Equations 5-5, 5-6, and 5-10 and is given below.

$$\text{Var}[\hat{\Delta d}_0] = \frac{(\underline{C}^T \underline{C})^2 \left(\frac{\partial \underline{C}^T}{\partial \Delta d_0} N \frac{\partial \underline{C}}{\partial \Delta d_0} + \left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right)^2 (\underline{C}^T N \underline{C}) - 2(\underline{C}^T \underline{C}) \left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right) \left(\underline{C}^T N \frac{\partial \underline{C}}{\partial \Delta d_0} \right) \right)}{\left[\left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta d_0} \right)^2 - \left(\frac{\partial \underline{C}^T}{\partial \Delta d_0} \frac{\partial \underline{C}}{\partial \Delta d_0} \right) (\underline{C}^T \underline{C}) \right]^2} \quad (5-12)$$

To complete our expression for the variance of the parameter $\hat{\Delta d}_0$ we need expressions for the components of the vectors $\underline{C}$ and $\partial \underline{C} / \partial \Delta d_0$ evaluated at $\hat{\Delta d}_0 = 0$, $\Delta B T = 0$ in addition to expressions for the components of the noise correlation matrix N .

An expression for the components of the noise correlation matrix N was obtained from Jarvis²⁵ and is given by Equation 5-1.

The components of the mismatch vector $\underline{C}$ can be obtained from Equation 3-68. The expression is given below.

$$(C_i)_{\substack{\Delta d_0=0 \\ \Delta \beta T=0}} = \cos(\omega_c \tau_i) \text{Sinc}\left(\frac{Bw}{2} \tau_i\right) \quad (5-13)$$

Where τ_i is the time delay associated with each lag point.

The components of the vector $\partial \underline{C} / \partial \Delta d_0$ can be obtained from Equation 3-68. The expression is given below.

$$\left(\frac{\partial C_i}{\partial \Delta d_0} \right)_{\substack{\Delta d_0=0 \\ \Delta \beta T=0}} = - \omega_c \sin(\omega_c \tau_i) \text{Sinc}\left(\frac{Bw}{2} \tau_i\right) - \frac{\cos(\omega_c \tau_i) \left[\sin\left(\frac{Bw}{2} \tau_i\right) - \frac{Bw}{2} \tau_i \cos\left(\frac{Bw}{2} \tau_i\right) \right]}{\frac{Bw}{2} \tau_i^2} \quad (5-14)$$

(Note that the expression for $\left(\frac{\partial C_i}{\partial \Delta \beta T} \right)_{\substack{\Delta d_0=0 \\ \Delta \beta T=0}} = \frac{1}{2} \left(\frac{\partial C_i}{\partial \Delta d_0} \right)_{\substack{\Delta d_0=0 \\ \Delta \beta T=0}}$.

For completeness we shall give the limits of Equations 5-13 and 5-14 when $\tau_i = 0$

$$(C_0)_{\Delta d_0=0}^{\Delta T=0} = 1 \quad (5-15)$$

$$\frac{\partial C_0}{\partial \Delta d_0} \bigg|_{\Delta d_0=0}^{\Delta T=0} = 0 \quad (5-16)$$

We can now use Equation 5-12 to perform three theoretical predictions concerning the number of lag points used by our ML estimator.

1. What is the effect of increasing the number of lag points in our ML estimator if we assume that the lag point noise components are independent? (i.e., the off-diagonal components of the matrix N are zero). This case is more for theoretical interest than practical interest.
2. What is the effect of increasing the number of lag points in our ML estimator if we assume that the lag point noise components are correlated? (Primary case)
3. Will prewhitening as described in Section 3.5 improve our estimates?

In order to answer the third question, Equation 5-12 must have the following modification to reflect the prewhitening.

First since the data has been prewhitened, the off-diagonal terms of the noise correlation matrix must be set to zero.

Also the vectors $\underline{C}$ and $\partial \underline{C} / \partial \Delta d_0$ undergo the following transformation.

$$\underline{C} = \underline{L}^{-1} \underline{C} \quad (5-17)$$

and

$$\frac{\partial \underline{C}}{\partial \Delta d_0} = \underline{L}^{-1} \frac{\partial \underline{C}}{\partial \Delta d_0} \quad (5-18)$$

where $\underline{L}^{-1}$ is determined by the following equation.*

$$\underline{N}^{-1} / \text{NORM} = \underline{L}^{-1T} \underline{L}^{-1} \quad (5-19)$$

and

$$\text{NORM} = \frac{\pi(1 + \text{SNR})^2}{B_w T \text{SNR}^2} \quad (5-20)$$

* In the actual prewhitening algorithm the matrix $\underline{L}$ is calculated for a low SNR assumption. This is done in order to have a prewhitening matrix independent of SNR which is valid at low SNR where it may prove useful.

Bw = bandwidth (in radians)

T = integration time per measurement

SNR = halfbeam signal-to-noise ratio (linear).

The following figures (5-24, 5-25, 5-26) show the results of our theoretical predictions.

Figure 5-24 shows the predicted standard deviation in the time delay difference offset parameter $\hat{\Delta d}_0$ as a function of the number of lag points centered about zero lag delay. The lag noise components are assumed to be independent. Figure 5-24 contains results for both high SNR (+5 dB) and low SNR (-15 dB).

As would be expected for both SNR levels, Figure 5-24 shows a decrease in standard deviation as the number of lag points is increased. The reduction is initially rapid but asymptotically approaches a constant value for large number of lag points. Since the noise is independent and the increasing number of lag points yield additional but decaying information on the signal, this result is as expected.

Figure 5-25 is identical to Figure 5-24 except that now the noise components of the lag points are correlated.

Now, however, the standard deviation of Δd_0 effectively remains constant with minor variations for all numbers of lag points. There is

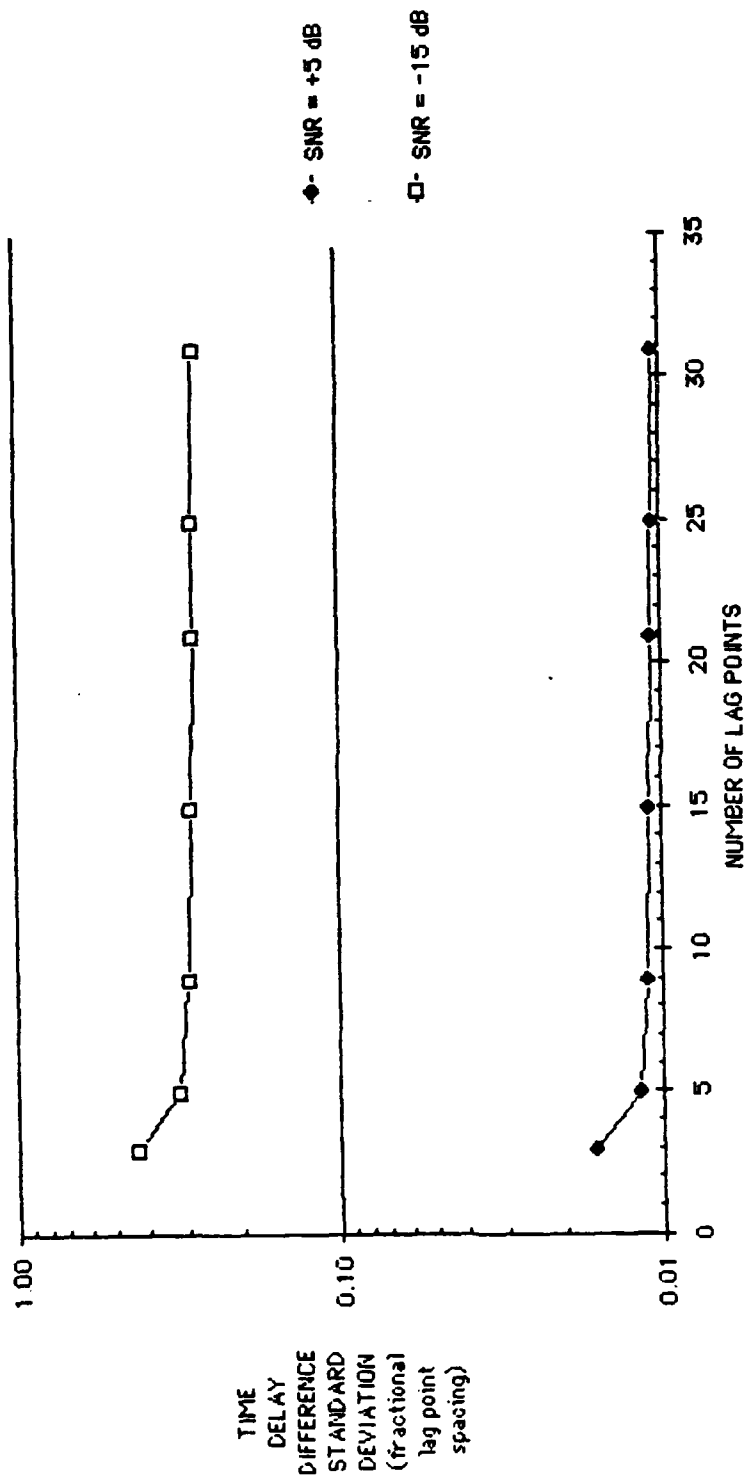


Figure 5-24. Theoretical Time Delay Difference Standard Deviation
vs Number of Lag Points (Independent Noise)

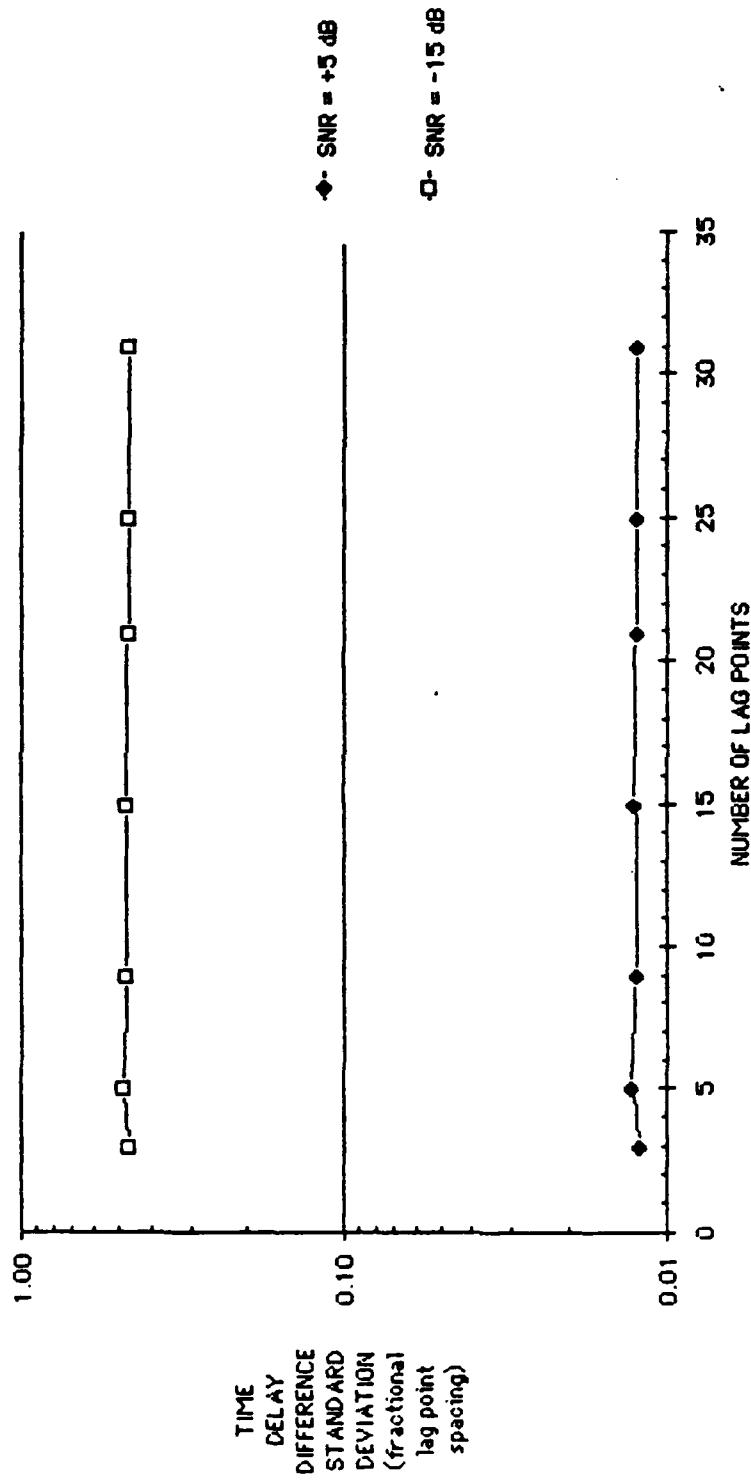


Figure 5-25. Theoretical Time Delay Difference Standard Deviation vs Number of Lag Points (Correlated Noise)

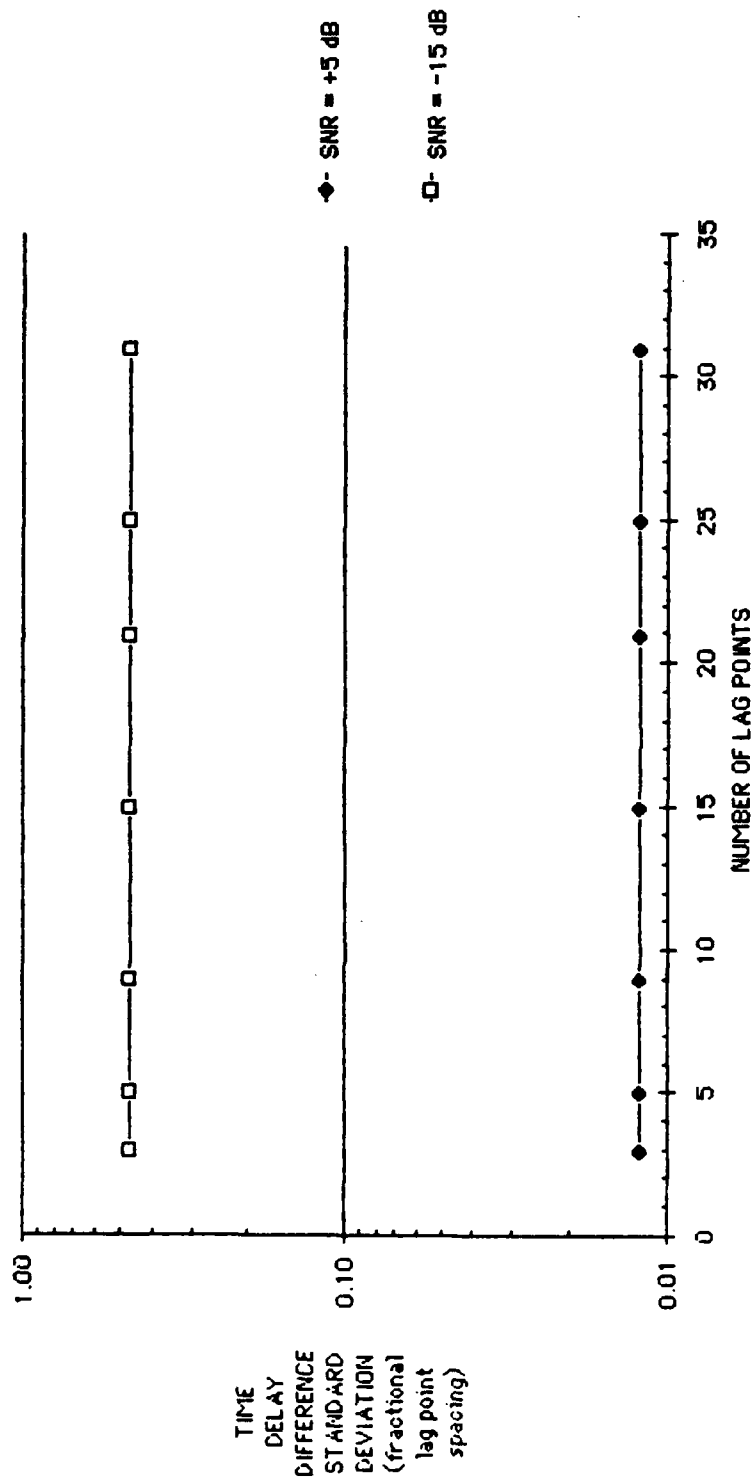


Figure 5-26. Theoretical Time Delay Difference Standard Deviation vs
Number of Lag Points (Correlated Noise with Prewhitening)

no reduction in variance with increased number of lag points. Effectively the correlated noise has offset the advantage of increasing the number of lag points.

Figure 5-26 contains results under the same conditions as Figure 5-25 but using the prewhitening assumption.

The prewhitening modifications effectively eliminates the minor variations in the standard deviation of $\hat{\Delta d}_0$ as a function of the number of lag points. However, increasing the number of lag points does not improve our estimate and the prewhitening modification shows an insignificant improvement over corresponding results in Figure 5-25. Therefore, from the above, prewhitening does not yield any gain.

We shall now present some simulation results on our actual estimators to compare to our theoretical results. Although our theoretical results assumed zero target motion and estimated means and variance, we shall use RMS error as a measurement and the modest target motion shown in Figure 5-2. Both modifications are more meaningful for our actual estimators and will not alter our basic conclusions.

All simulations were conducted for the ML estimator only ($K = 0$). Each plotted point represents a Monte Carlo average of five simulations of 20 minute duration each. As per standard procedure, all simulations were conducted with correlated Gaussian noise added to the measured lag point data.

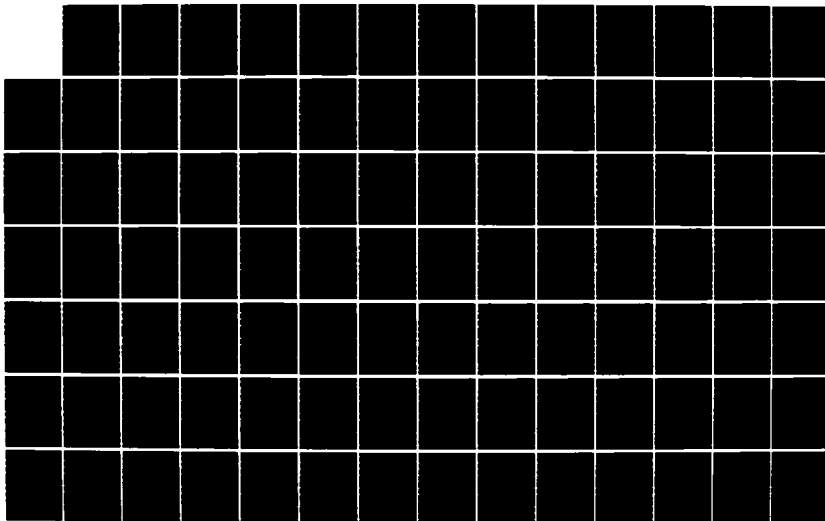
AD-A173 396

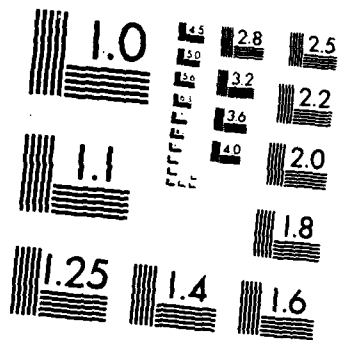
EFFICIENT RECURSIVE BATCH TIME DELAY DIFFERENCE
ESTIMATION IN THE PRESENC. (U) NAVAL UNDERWATER SYSTEMS
CENTER NEW LONDON CT NEW LONDON LAB. R A LATOURETTE
17 JUL 86 NUSC-TR-7743 F/G 17/1

3/4

UNCLASSIFIED

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Figure 5-27 shows the RMS time delay difference error versus the number of lag points using the gradient search algorithm for an SNR of 5 dB and an SNR of -15 dB. The results agree with the theoretical and show no significant variation with the number of lag points at either SNR level.

Figure 5-28 shows the RMS time delay difference error versus the number of lag points using the gradient search algorithm with pre-whitening at SNR levels of 5 dB and -15 dB. Once again the results agree with the theory, showing no significant variation in RMS time delay difference error with the number of lag points. In addition, the RMS time delay difference errors of the prewhitening algorithm are effectively the same as the standard algorithm. This result indicates that pre-whitening does not improve our estimator.

For completeness, Figure 5-29 shows the RMS time delay difference error versus the number of lag points for the dynamic programming algorithm. At low SNR (-15 dB), Figure 5-29 agrees with Figure 5-27 and 5-28 and shows no significant variation in RMS time delay difference error with the number of lag points used. At high SNR (5 dB) the results appear to be modal. The RMS time delay difference errors measured fall into two levels. This variation in theory is attributed to the quantization effect of the dynamic program algorithm at high SNR.

We will now examine the statistics of the slope parameter $\hat{\Delta}T$ in our second problem. Since we will be examining cases where the actual

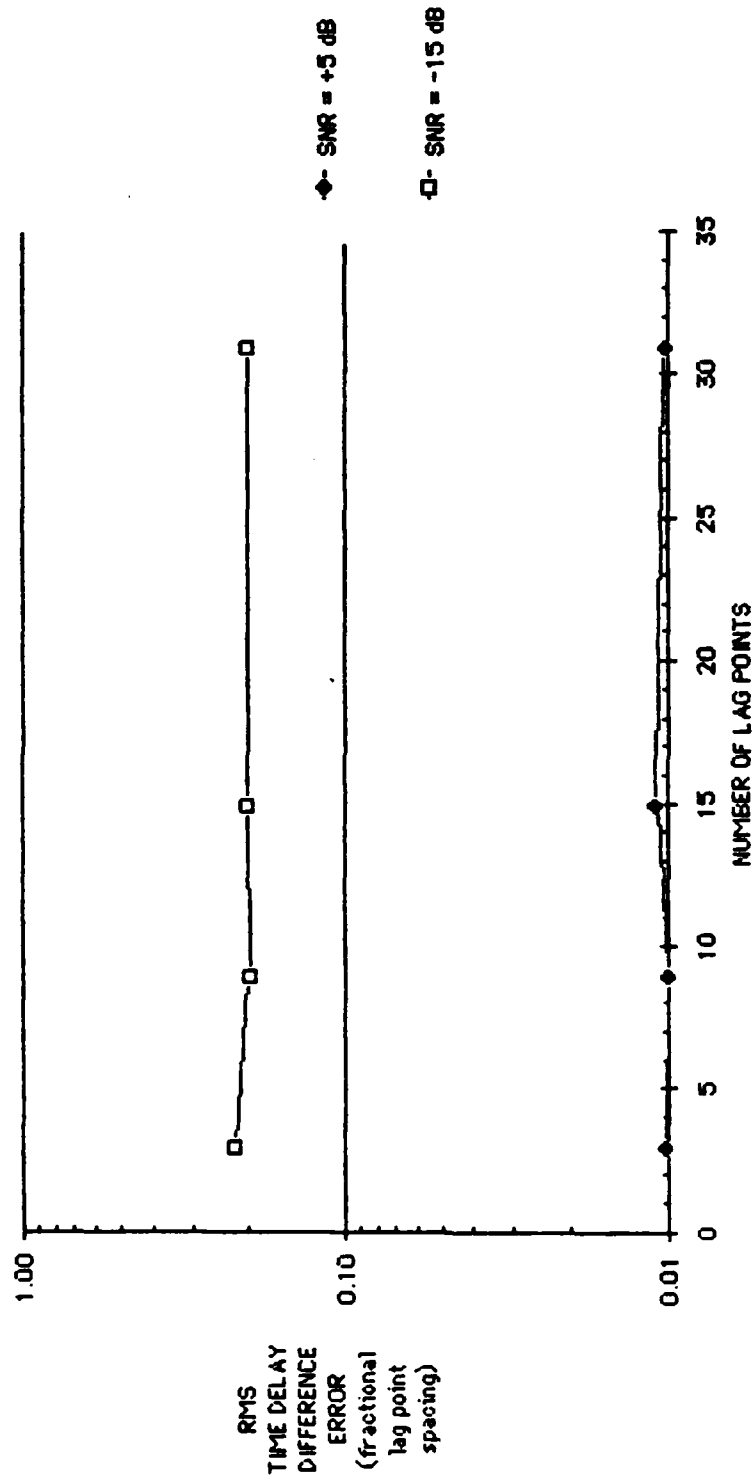


Figure 5-27. RMS Time Delay Difference Error vs Number of Lag Points,
Gradient Search, Dynamic Scenario ($T = 5$, $K = 0$)

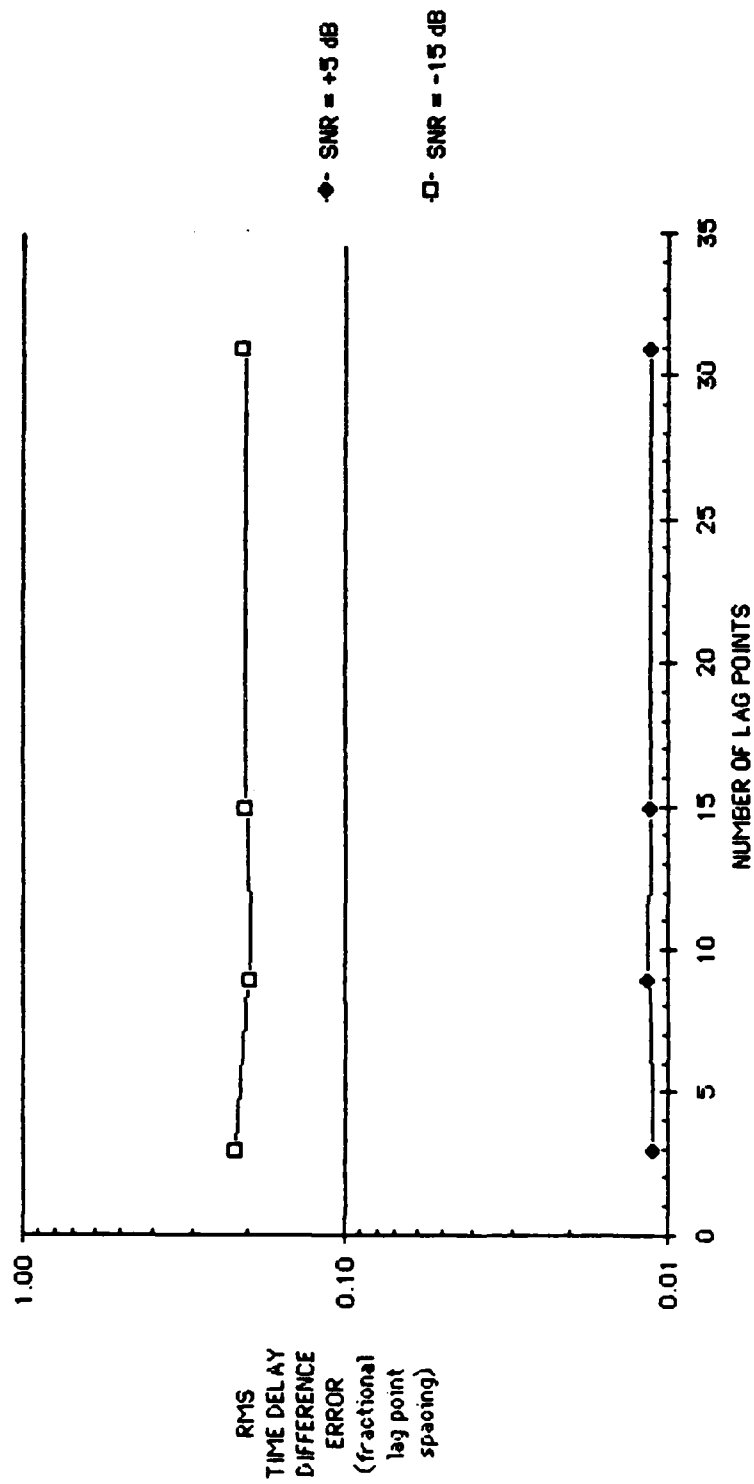


Figure 5-28. RMS Time Delay Difference Error vs Number of Lag Points, Gradient Search with Prewhitening, Dynamic Scenario ($T = 5$, $K = 0$)

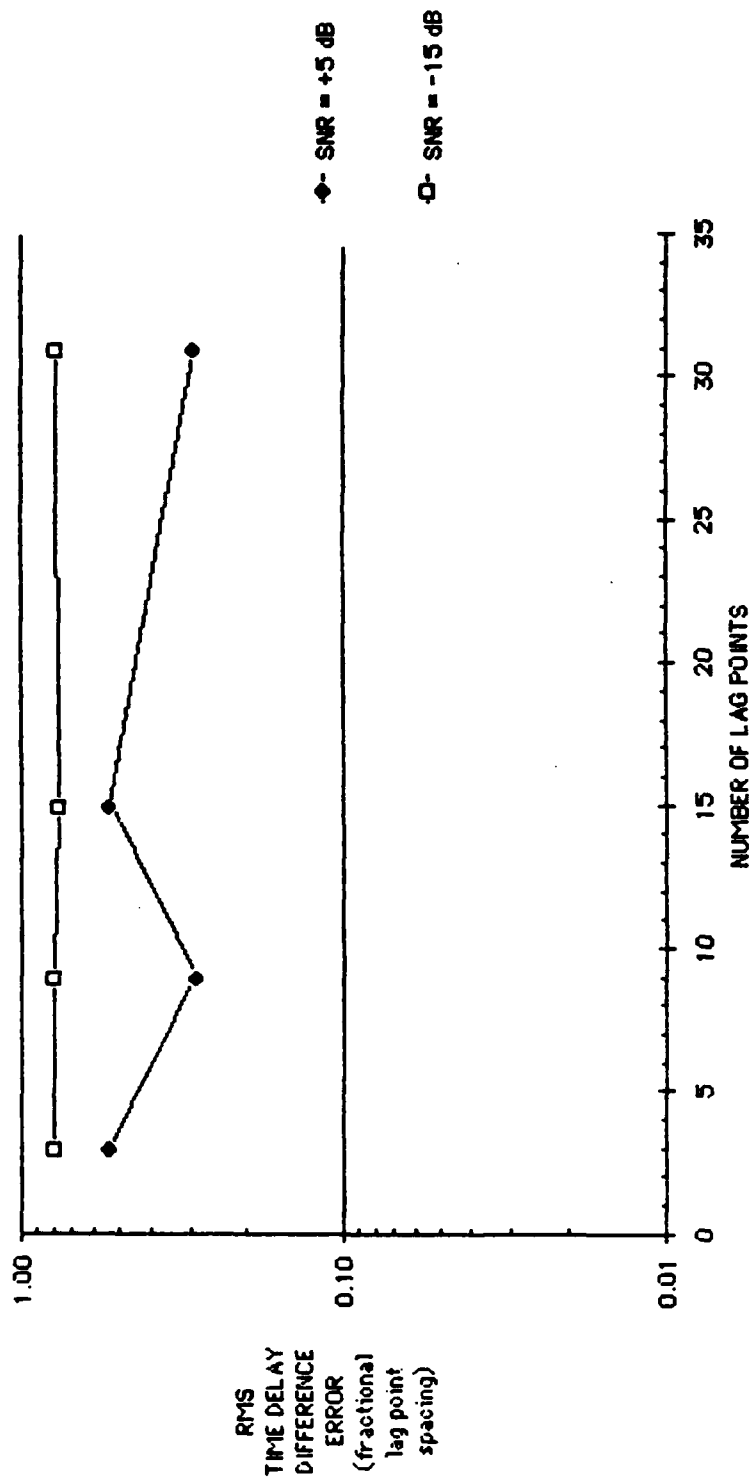


Figure 5-29. RMS Time Delay Difference Error vs Number of Lag Points, Dynamic Program, Dynamic Scenario ($T = 5$, $K = 0$)

slope parameter $\Delta\beta T$ is not zero, Equation 5-5 needs to be slightly modified to the following form.

$$\hat{\Delta\beta T} = \Delta\beta T - \frac{\partial F}{\partial \Delta\beta T} \left(\frac{\partial^2 F}{\partial \Delta\beta T^2} \right)^{-1} \Delta d_0 = 0 \quad (5-21)$$

$\hat{\Delta\beta T} = \Delta\beta T$

The analytical work of this problem follows that of Problem 1 with the obvious modification. The results are summarized below.

The estimator is unbiased at high SNR.

$$E[\hat{\Delta\beta T} - \Delta\beta T] = 0 \quad (5-22)$$

The variance of $\hat{\Delta\beta T}$ is given below.

$$\text{Var}(\hat{\Delta\beta T}) = \frac{(\underline{C}^T \underline{C})^2 \left(\frac{\partial \underline{C}^T}{\partial \Delta\beta T} \frac{\partial \underline{C}}{\partial \Delta\beta T} \right) + \left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta\beta T} \right)^2 \left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta\beta T} \right)^2 - 2(\underline{C}^T \underline{C}) \left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta\beta T} \right) \left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta\beta T} \right)}{\left[\left(\underline{C}^T \frac{\partial \underline{C}}{\partial \Delta\beta T} \right)^2 - \left(\frac{\partial \underline{C}^T}{\partial \Delta\beta T} \frac{\partial \underline{C}}{\partial \Delta\beta T} \right) (\underline{C}^T \underline{C}) \right]^2} \quad (5-23)$$

To complete the above expression, we need an equation for the components of the vector $\frac{\partial \underline{C}}{\partial \Delta\beta T}$ evaluated at $\Delta d_0 = 0$ and $\hat{\Delta\beta T} = \Delta\beta T$. This we can obtain from Equation 4-50 which is shown below.

$$\left(\frac{\partial C_i}{\partial \Delta\beta T} \right)_{\Delta d_0=0} = \frac{\cos[\omega_c(\tau_i + \Delta\beta T)] \text{Sinc}\left[\frac{B_w}{2}(\tau_i + \Delta\beta T)\right] - C_i}{\Delta\beta T} \quad (5-24)$$

$\hat{\Delta\beta T} = \Delta\beta T$

If $\Delta\delta T = 0$, the following is true.

$$\left(\frac{\partial C_i}{\partial \Delta\delta T}\right)_{\substack{\Delta d_0=0 \\ \Delta\delta T=0}} = \frac{1}{2} \left(\frac{\partial C_i}{\partial \Delta d_0}\right)_{\substack{\Delta d_0=0 \\ \Delta\delta T=0}} \quad (5-25)$$

where $\left(\frac{\partial C_i}{\partial \Delta d_0}\right)_{\substack{\Delta d_0=0 \\ \Delta\delta T=0}}$ is given by Equation 5-14.

Using Equation 5-23 we can now make some theoretical predictions concerning our ML estimator, the number of lag points, the averaging time, and the target time delay rate. Our unit of measurement will be the standard deviation of our slope parameter time delay difference ($\Delta\delta T$). (The standard deviation is the square root of the variance.) The slope parameter $\Delta\delta T$ is a measure of the change in the time delay difference trajectory from the beginning of the measurement interval to the end of the measurement interval.

Figure 5-30 shows several plots of standard deviation of time delay difference error versus time delay difference rate. Each plot represents a different selection for the number of lag points. The SNR for all plots in Figure 5-30 was set at 5 dB; the noise components of the measured lag point data is correlated with covariance matrix N ; the integration time T is set to 5 seconds.

Examining Figure 5-30, we notice that for any selection of lag points, the standard deviations in time delay difference error are

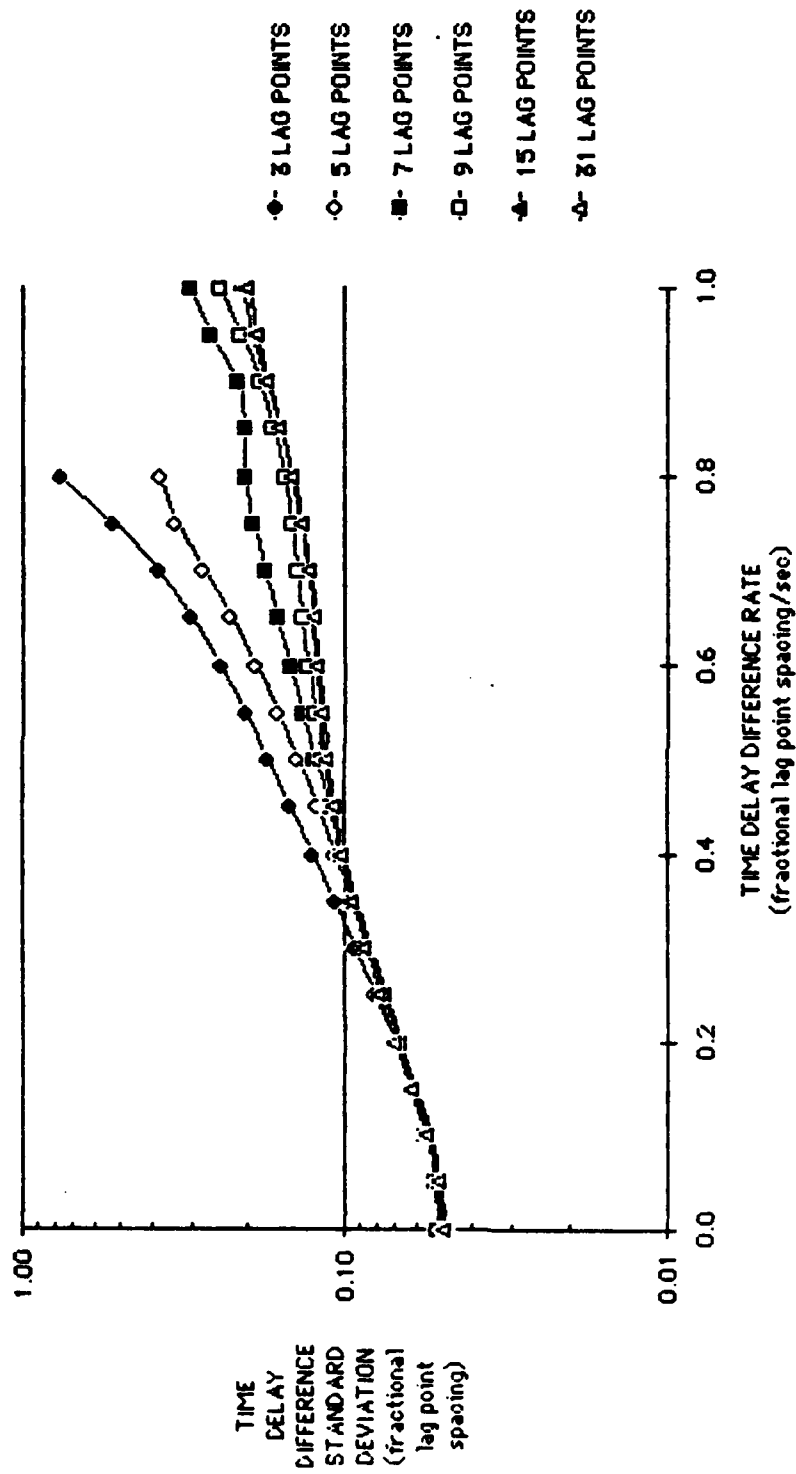


Figure 5-30. Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = 5 dB, Correlated Noise, $T = 5$ sec)

initially equivalent for low time delay difference rates and increase monotonically as time delay difference rate increases. However, for smaller number of lag points, a rapid increase in standard deviation in time delay difference error occurs at a low time delay difference rate. Note that even for 3 lag points a time delay difference rate of 0.2 fractional lag point spacing per second must be exceeded before there is a significant increase in standard deviation in time delay difference error.

The data plotted in Figure 5-31 is for conditions identical to Figure 5-30 except that the noise components of the measured lag point data are assumed uncorrelated. This case is not realistic and is presented for academic interest.

In essence, Figure 5-31 agrees with Figure 5-30. Increasing the time delay difference rate monotonically increases the standard deviation in time delay difference error ($\Delta \hat{B}T$). For a lower number of lag points, a rapid increase in standard deviation in time delay difference error occurs at a lower value of time delay difference rate. The difference in Figure 5-31 is that increasing the number of lag points reduces the standard deviation in time delay difference error even at low time delay difference rates. In addition, the increase in standard deviation of time delay difference error is reduced with respect to Figure 5-30 for all selections of the number of lag points. The above effects can be attributed to the additional information obtained due to the uncorrelated noise components.

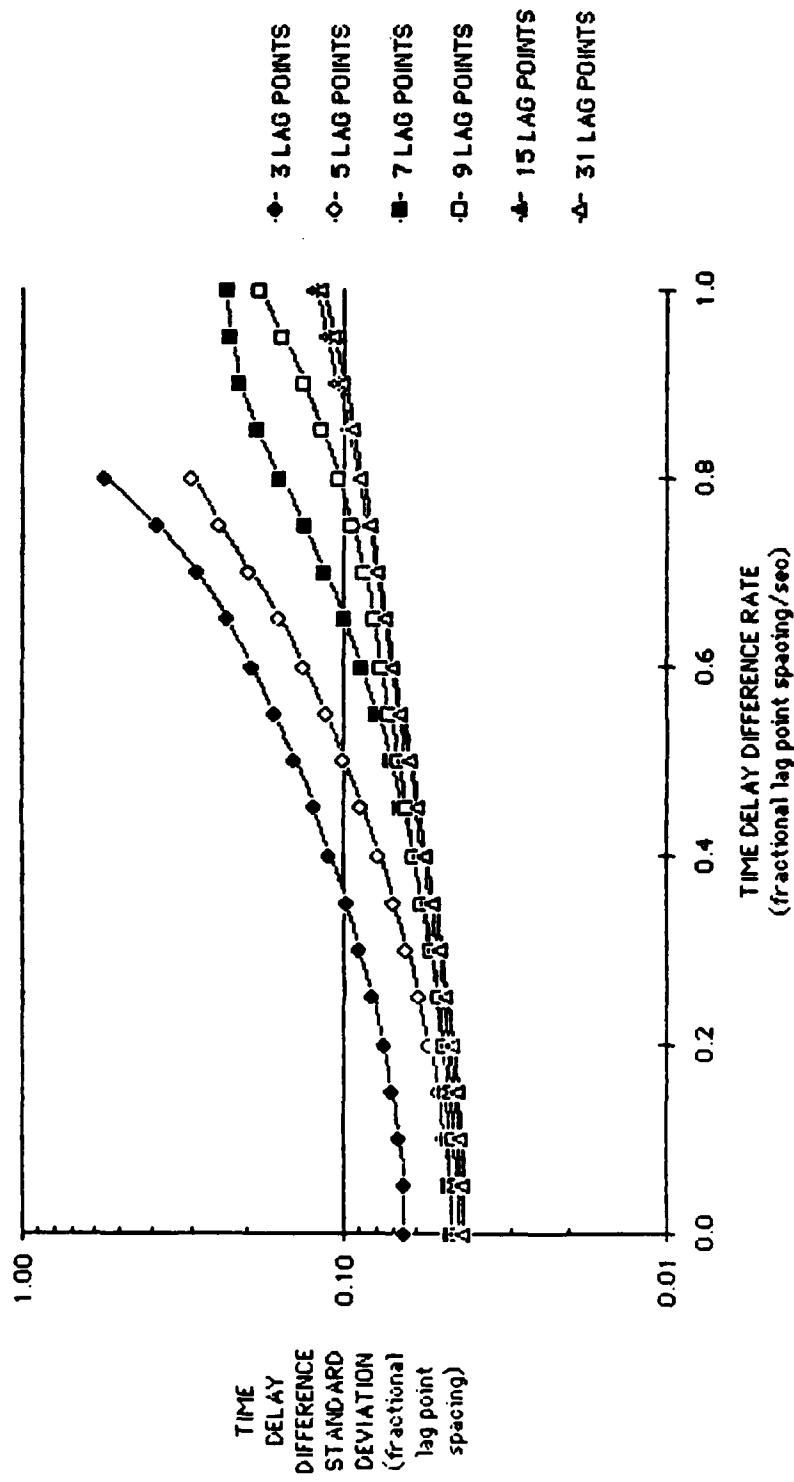


Figure 5-31. Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = 5 dB, Uncorrelated Noise, $T = 5$ sec)

The data presented in Figure 5-32 is also for conditions similar to Figure 5-30. However, the SNR was set to -15 dB instead of the 5 dB for Figure 5-30. The lag point noise components were assumed correlated and the averaging time T equals 5 seconds.

The resulting data shown in Figure 5-32 is similar to the data shown in Figure 5-30. The obvious differences are the larger standard deviation in time delay difference error and the higher time delay difference rates required before there is a significant increase in standard deviation in time delay difference error. Both observations can be largely attributed to the dominance of noise at lower SNR.

The final plotted theoretical data (Figure 5-33) was calculated with conditions similar to Figure 5-30. The key difference is that for Figure 5-33 the averaging time T is set to 1 second where in Figure 5-30 the averaging time is 5 seconds. Once again the SNR is set at 5 dB and the lag point noise components are assumed correlated.

Figure 5-33 demonstrates the obvious affects of the shorter integration time. For all selected numbers of lag points the standard deviations in time delay difference error ($\Delta\hat{\delta T}$), at the lower time delay difference rates, are higher than their counterparts in Figure 5-30. However, for all time delay difference rates and number of lag points selected, there is a significant reduction in the increase of standard deviation in time delay difference error with increasing time delay difference rate. In fact, the standard deviation in time delay difference error is effectly equivalent for all number of lag point chosen.

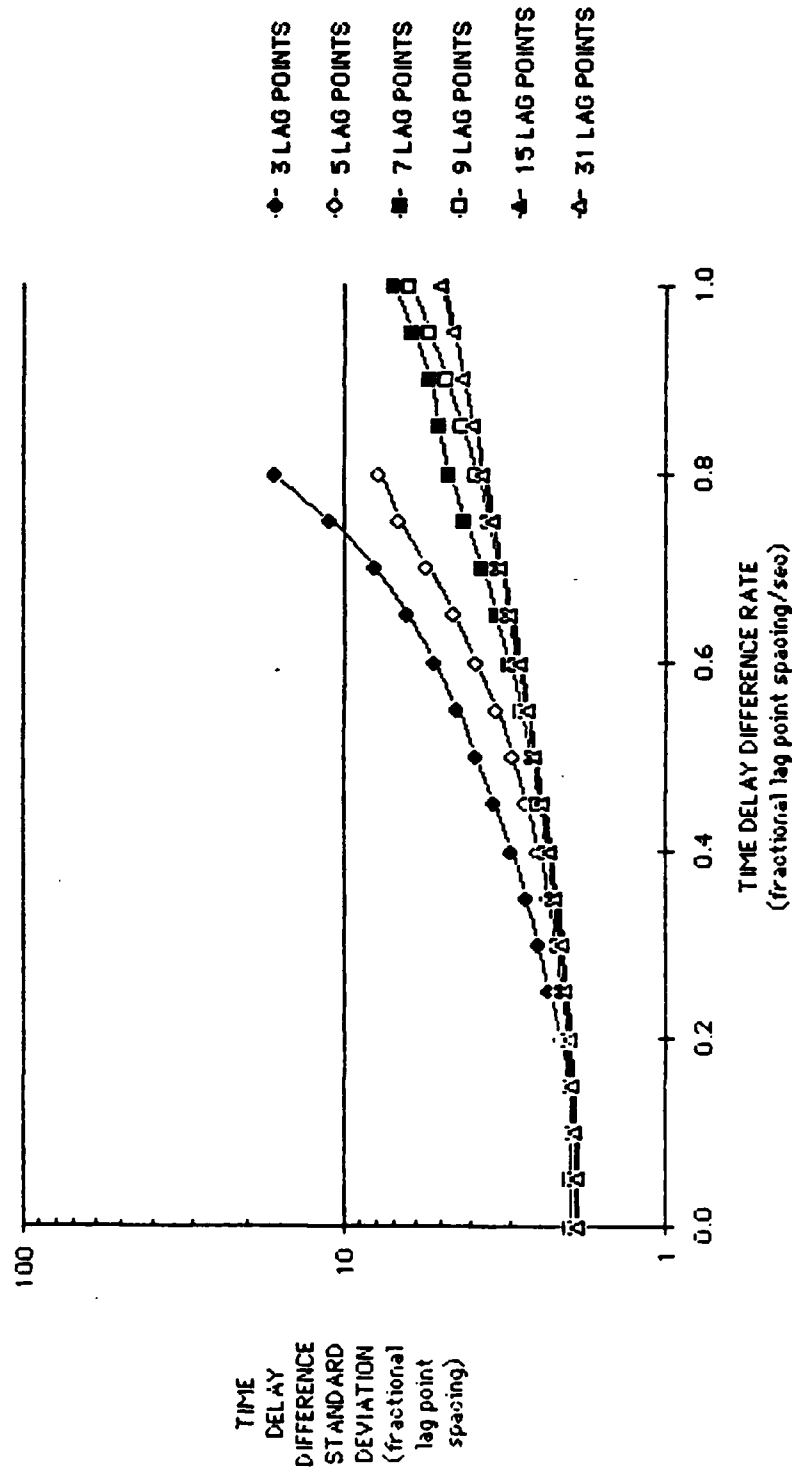


Figure 5-32. Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = -15 dB, Correlated Noise, $T = 5$ sec)

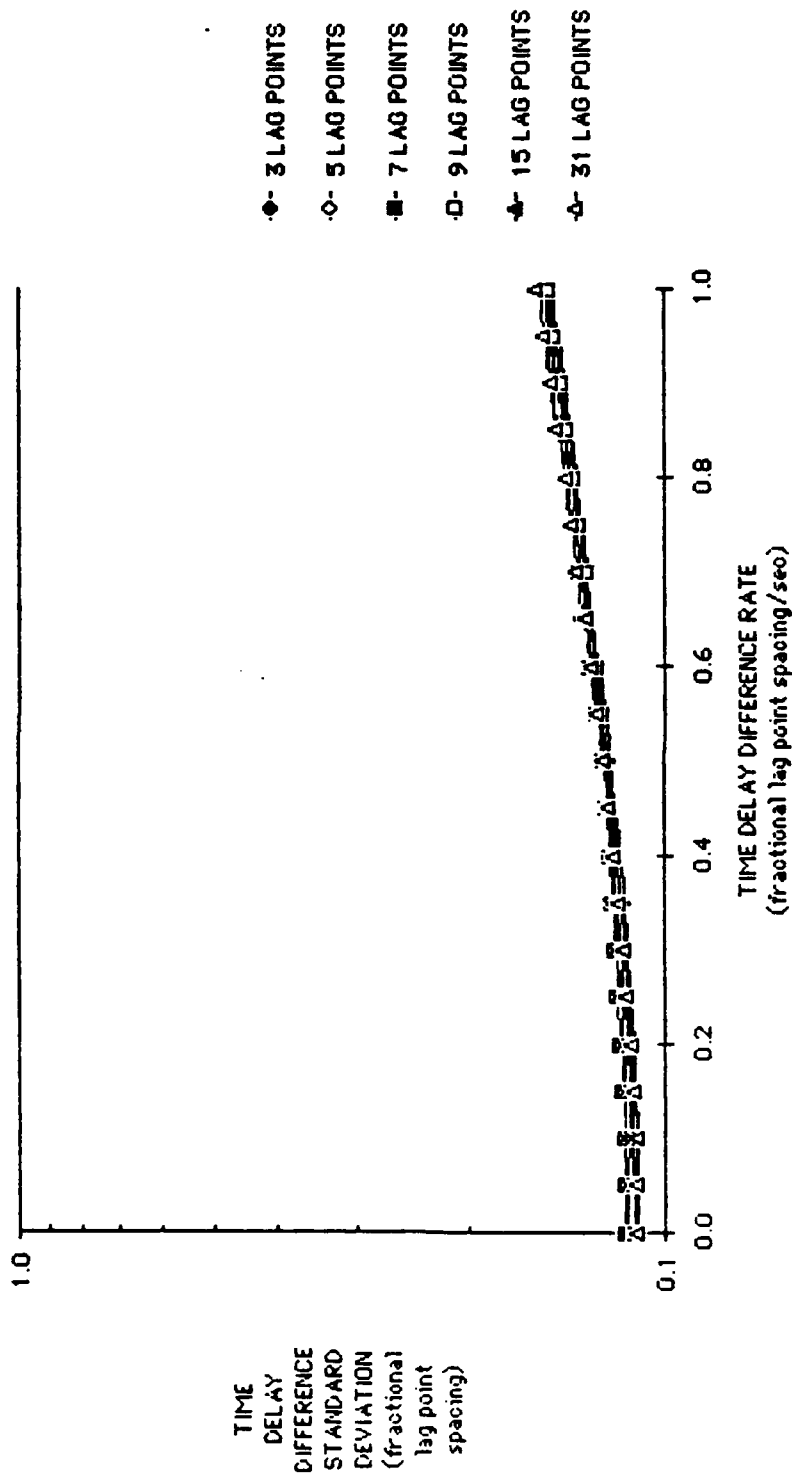


Figure 5-33. Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = 5 dB, Correlated Noise, $T = 1$ sec)

The above figures indicate that, for a given SNR time delay difference rate and number of lag points, there may be an optimum averaging time. This concept is developed in the data presented in Figures 5-34 and 5-35. Figure 5-34 plots standard deviation of time delay difference error versus time delay difference rate for various selections of averaging time T . The SNR is set at 5 dB; the number of lag points is set at 9; and the noise between lag points is correlated. The conditions for Figure 5-35 are the same as Figure 5-34 except that the SNR is set at -15 dB.

Two key points should be obtained from Figure 5-34 and 5-35. The first is that for a given time delay difference rate and number of lag points there is an optimal averaging time which is a function of SNR. The second point is that excessive average time ($T = 10$ seconds) is risky for any target scenarios featuring significant dynamics.

The following figures represent simulation data concerning the number of lag points, averaging time, and the time delay difference rate.

Performing realistic statistical simulations on significant dynamic target scenarios is almost as difficult as actual physical dynamic testing. Significant dynamic target motion limits meaningful test time and the limited test time tends to be dominated by initialization and transient conditions. Therefore, for the most part, we make no claim that our simulations confirm our theoretical predictions. (The maximum target dynamics that we shall use is only approximately

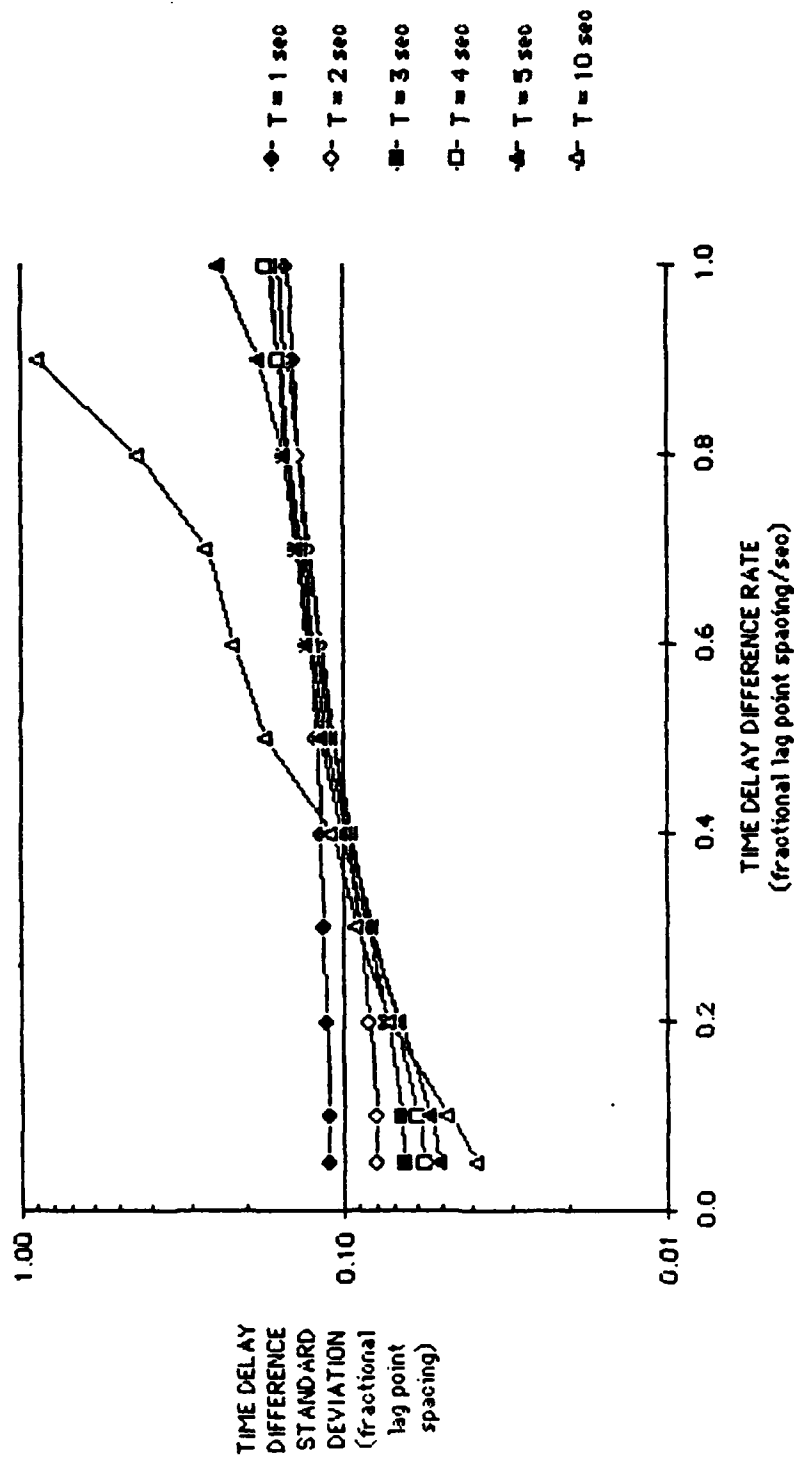


Figure 5-34. Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = 5 dB, Correlated Noise, $N = 9$)

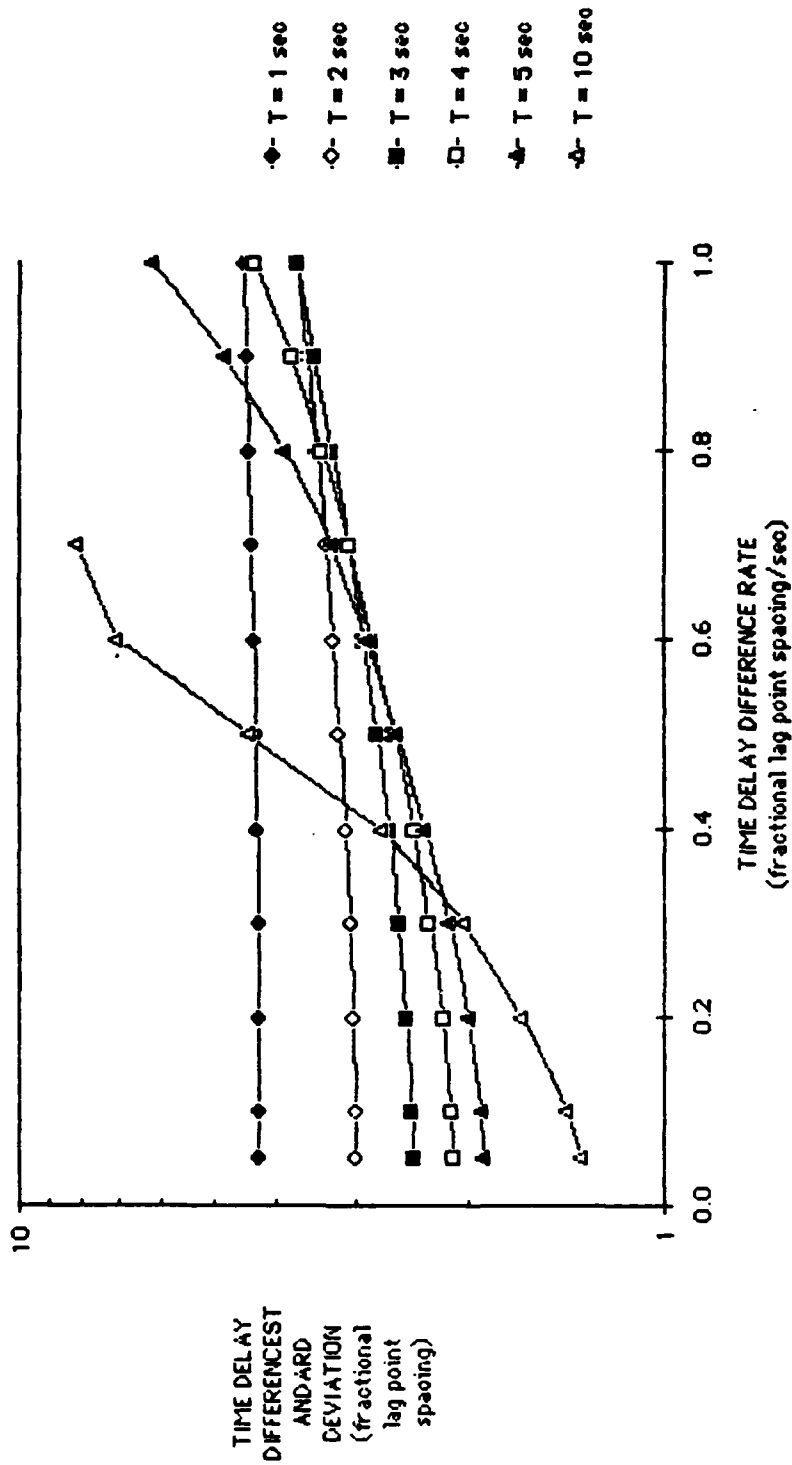


Figure 5-35. Theoretical Time Delay Difference Standard Deviation vs Time Delay Difference Rate (SNR = -15 dB, Correlated Noise, $N = 9$)

.13 fractional lag points per second.) However, we do claim that our simulations do demonstrate a significant role that averaging time and the number of lag points play in a dynamic scenario.

The following simulation data all assumes correlated noise between lag points. Once again we shall only use our ML estimator ($K = 0$) and a nominal data block size of 10. In order to match our theoretical predictions, our simulation will be limited to the continuous gradient search algorithm. We shall use the basic dynamic scenario shown in Figure 5-2. However, we shall multiply the speed of the target by integer factors, effectively multiplying the resulting time delay difference rate. Obviously, the higher the time delay difference rate the shorter the duration of the simulation. All our simulated plotted data represents the Monte Carlo average of 5 simulation runs. For simulation run at 4 times the nominal time delay difference rate, the simulations are 5 minutes in duration. For simulation runs at 2 times the nominal time delay difference rate, the simulations are 10 minutes in duration.

Figure 5-36 shows the RMS error of our time delay difference estimator versus averaging time for 2 and 4 times the nominal time delay difference rate. The number of lag points was set to 3 and the SNR is 5 dB. The data in Figure 5-36 is almost certainly dominated by transient conditions. However, it does demonstrate the adverse effects of high time delay difference rates and long averaging times. In addition, Figure 5-36 indicates the potential of an optimal match between averaging time and target dynamics.

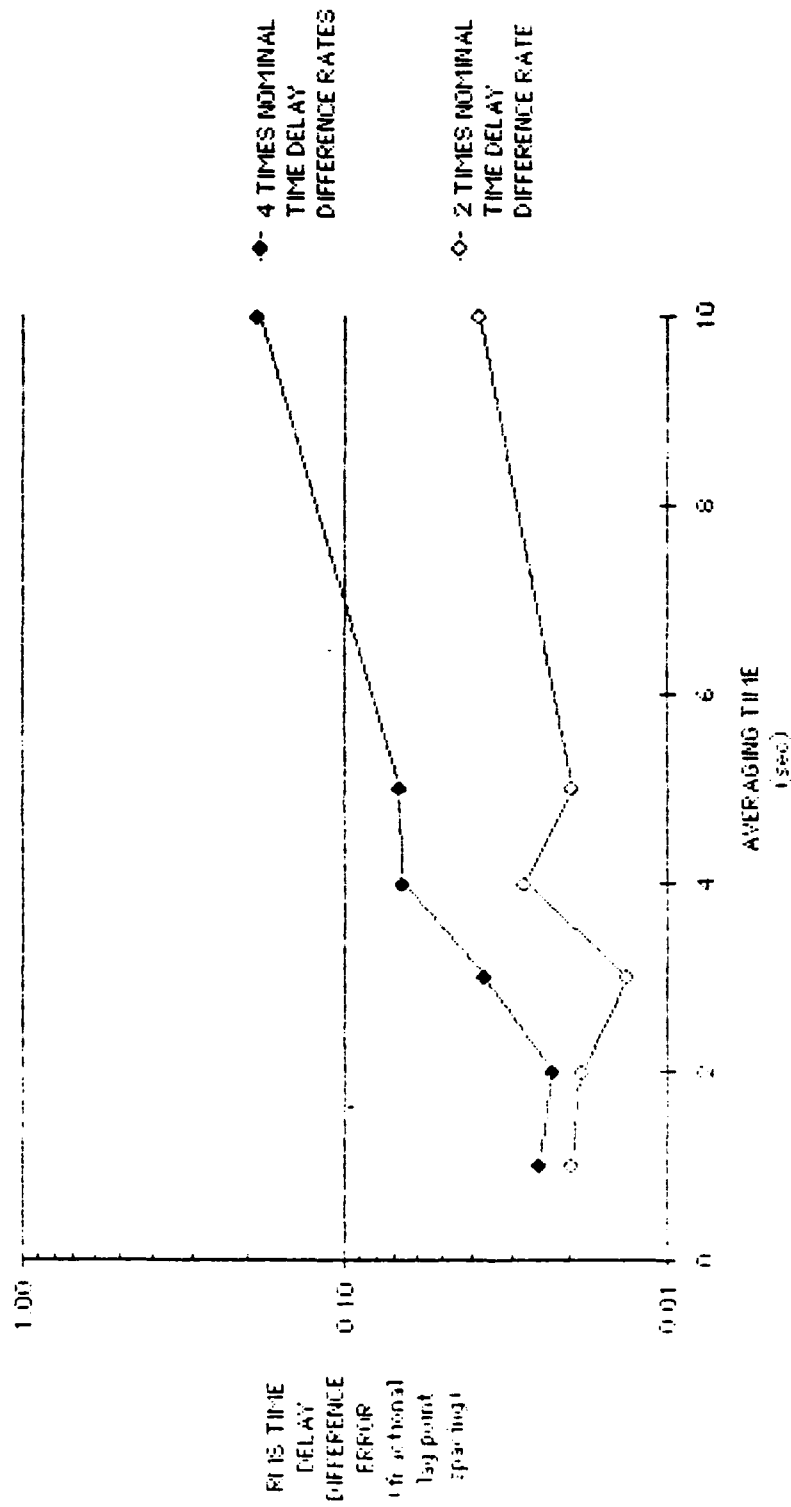


Figure 5-36. RMS Time Delay Difference Error vs Averaging Time
(SNR = 5 dB, $N = 3$, Dynamic Scenario, Gradient Search)

Figure 5-37 plots the RMS error of our time delay difference estimator versus time delay difference rate for the number of lag points set at 3 and 9. The SNR is 5 dB and the averaging time is 5 seconds.

Once again the data in Figure 5-37 at the higher time delay difference rates will be dominated by transient condition. However, Figure 5-37 does demonstrate the adverse effect of increasing time delay difference rate. In addition, the data in Figure 5-37 hints that a small number of lag points may adversely affect our estimator at high time delay difference rates.

The final set of simulation we performed at an SNR of -15 dB and for an averaging time of 10 seconds. The purpose of the simulations was to determine the affect of the number of lag points on the RMS time delay difference error at low SNR. The time delay difference rate was set at 4 times the nominal rate.

Figure 5-38 presents the results of the above simulations. Figure 5-38 indicates a potential degradation in RMS time delay difference error if a small number of lag points is used and high target time delay difference dynamics are encountered. This result is in agreement with Figure 5-37.

Despite quantization error which complicates the dynamic programming algorithm, limited simulation results indicate similar behavior as recorded in Figure 5-36 through 5-38 for the gradient search algorithm. The dynamic programming algorithm is hard-limited to the maximum time

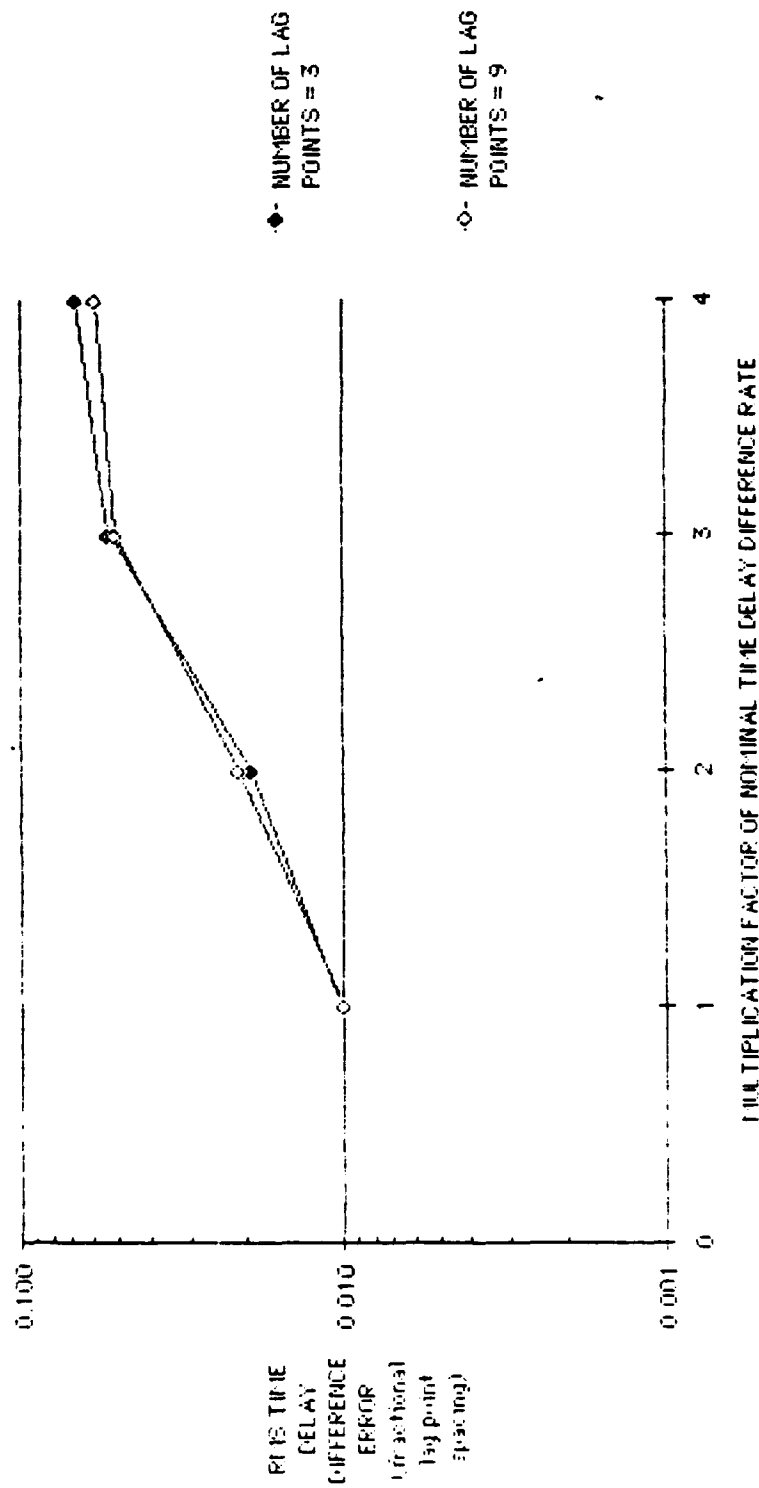


Figure 5-37. PTS Time Delay Difference Error vs Time Delay Difference Rate (SNR = 5 dB, T = 5, Gradient Search)

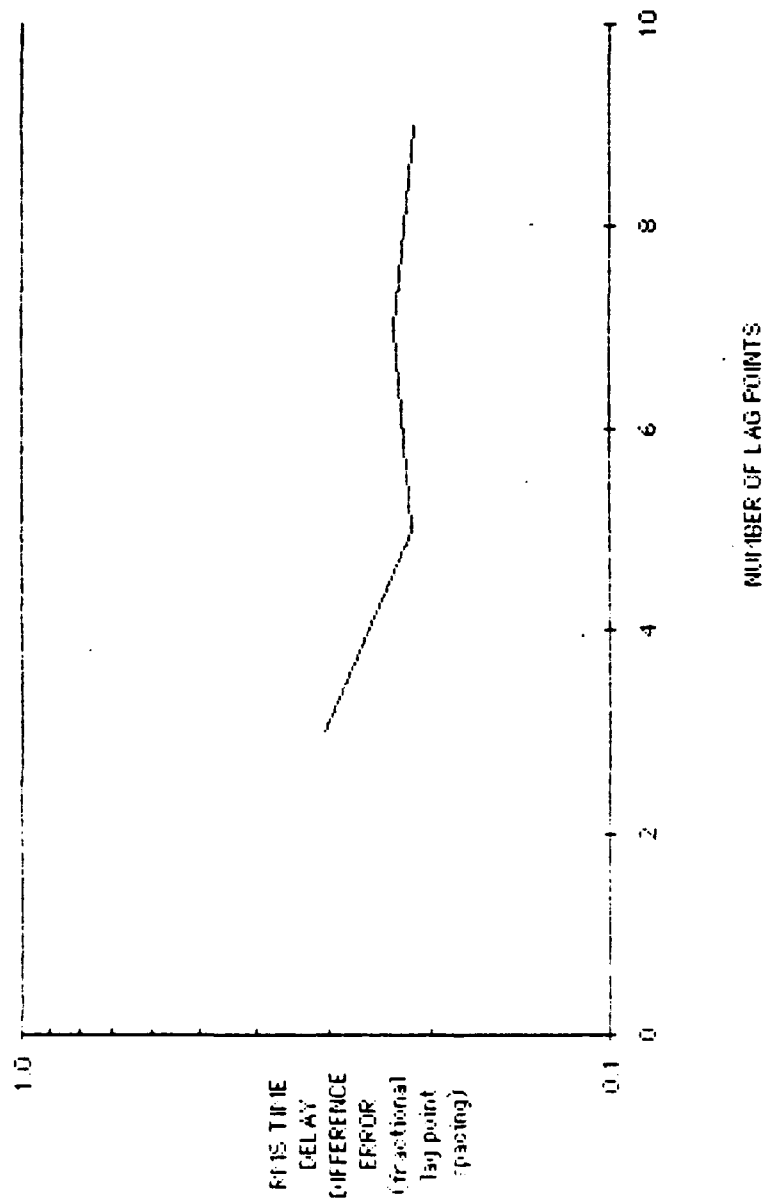


Figure 5-38 RMS Time Delay Difference Error vs Number of Lag Points (SNR = -15 dB, $T = 10$ sec, 4 X Nominal Time Delay Difference Rate)

delay difference rate that it can follow by the parameter NPMAX. This parameter limits the maximum delay difference change allowed during a single measurement update. If NPMAX is fixed, the longer the averaging time the smaller the maximum allowable time delay difference rate.

In summation, we have determined the following conclusions concerning the number of lag points N and the averaging time T. Our conclusions will be limited to the realistic assumption of correlated noise between measured lag points.

1. For low target time delay difference dynamics and reasonable averaging times, the variance of our ML estimator is effectively independent of the number of lag points. Since processing additional lag points is computationally costly we would want to minimize the numbers of lag points (3) processed.
2. For high target time delay difference dynamics and reasonable averaging times, there is potential degradation in our ML for a very small number of lag points (3).
3. Combining conclusions 1 and 2, a recommended number of lag points is either 5, 7, or 9.
4. Depending on the severity of the target time delay difference dynamics, there is an optimum averaging time. Reasonable averaging times appear to be in the range of 1 to 5 seconds.

The final parameter of interest is the lag point spacing τ . We can use Equation 5-23 to make some theoretical predictions about the effects of lag point spacing on our theoretical ML estimator. In the theoretical predictions and simulations to follow, we shall use multiples of our basic lag point spacing illustrated in Figure 5-1.

In our first theoretical analysis, we shall calculate standard deviation of $\Delta \hat{B}T$ (time delay difference change over the measurement interval) versus lag point spacing (based on multiples of the basic time delay between lag points). The above calculation was performed for several time delay difference rates varying from 0 to 1 fractional lag point spacing per second. In all cases, the fractional lag point spacing is referenced to our nominal lag point spacing. The SNR is set to 5 dB, averaging time T is 5 seconds and the number of lag points N is set at 9. The noise between lag points is correlated.

Figure 5-39 illustrates the above calculations. Examining Figure 5-39 we notice that for all* time delay difference rates the standard deviation of our time delay difference estimate degrades for lag point spacing multiples greater than twice our nominal lag point spacing. This result is not surprising after looking at Figure 5-1 which shows the relationship between the nominal lag point spacing and the nominal target signal auto correlation function. Sampling at 4 or more times the nominal lag point spacing will yield at most 1 lag point between the first two zero crossings. Therefore, we are obviously undersampling which results in loss of information.

* Unplotted points represent invalid results for our theoretical estimate of standard deviation.

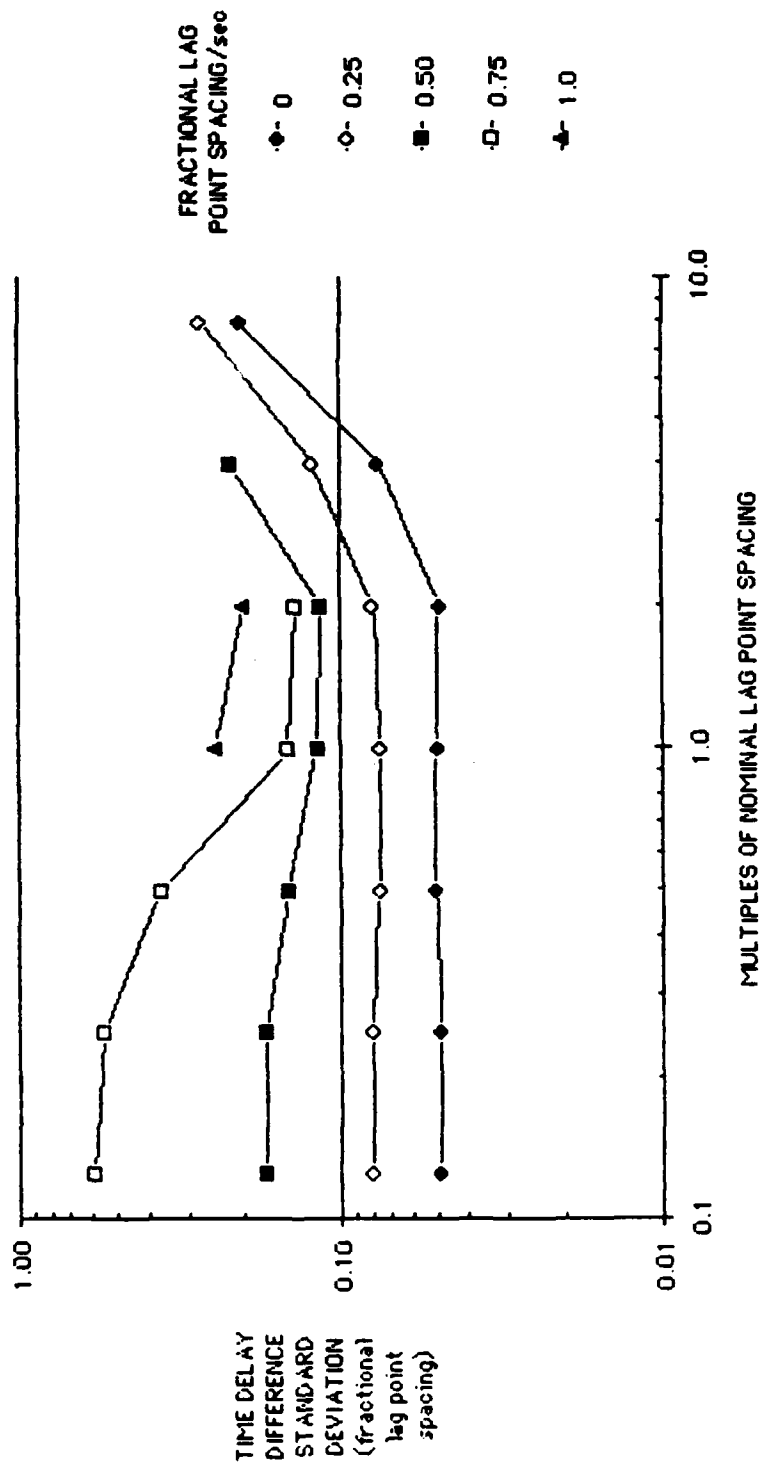


Figure 5-39. Theoretical Time Delay Difference Standard Deviation vs Lag Point Spacing (SNR = 5 dB, Correlated Noise, $N = 9$, $T = 5$)

At moderate time delay difference rates we notice from Figure 5-39 that decreasing the lag point spacing from twice the nominal value has very little affect on our estimator. As time delay difference rates are increased, lowering the lag point spacing results in an increasing degradation of our time delay difference estimator standard deviation.

Overall from the theoretical results presented in Figure 5-39 we can conclude that either 1 or 2 times our nominal lag point spacing will be an optimum choice.

Figure 5-40 is a repeat of Figure 5-39 except that the number of lag points is set to 3. The general results for Figure 5-40 are the same as Figure 5-39. However, the noticeable improvement in Figure 5-39 at high time delay difference rates for 1 or 2 times the nominal lag point spacing is noticeably reduced in Figure 5-40. This result agrees with our earlier results indicating that at high time delay difference rates too few lag points will cause a degradation in our ML estimator.

The final figure is for academic purposes. The theoretical predictions are for conditions identical to Figure 5-39 except that the noise between lag points is uncorrelated.

In Figure 5-41 the standard deviation of our time delay difference estimator is degraded at small lag point spacings for all time delay difference rates. There is a clear optimal lag point spacing ranging

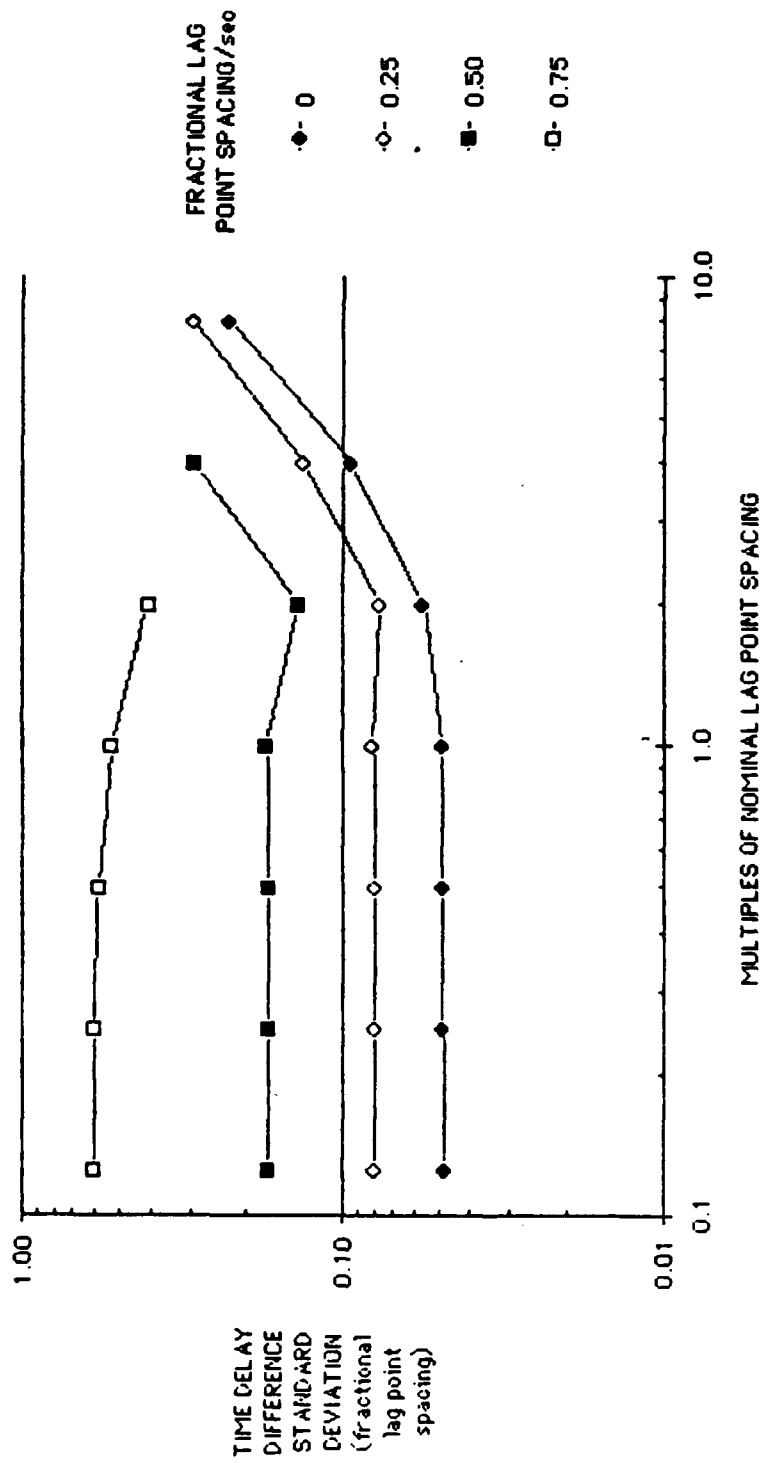


Figure 5-40. Theoretical Time Delay Difference Standard Deviation vs Lag Point Spacing (SNR = 5 dB, Correlated Noise, $N = 3$, $T = 5$)

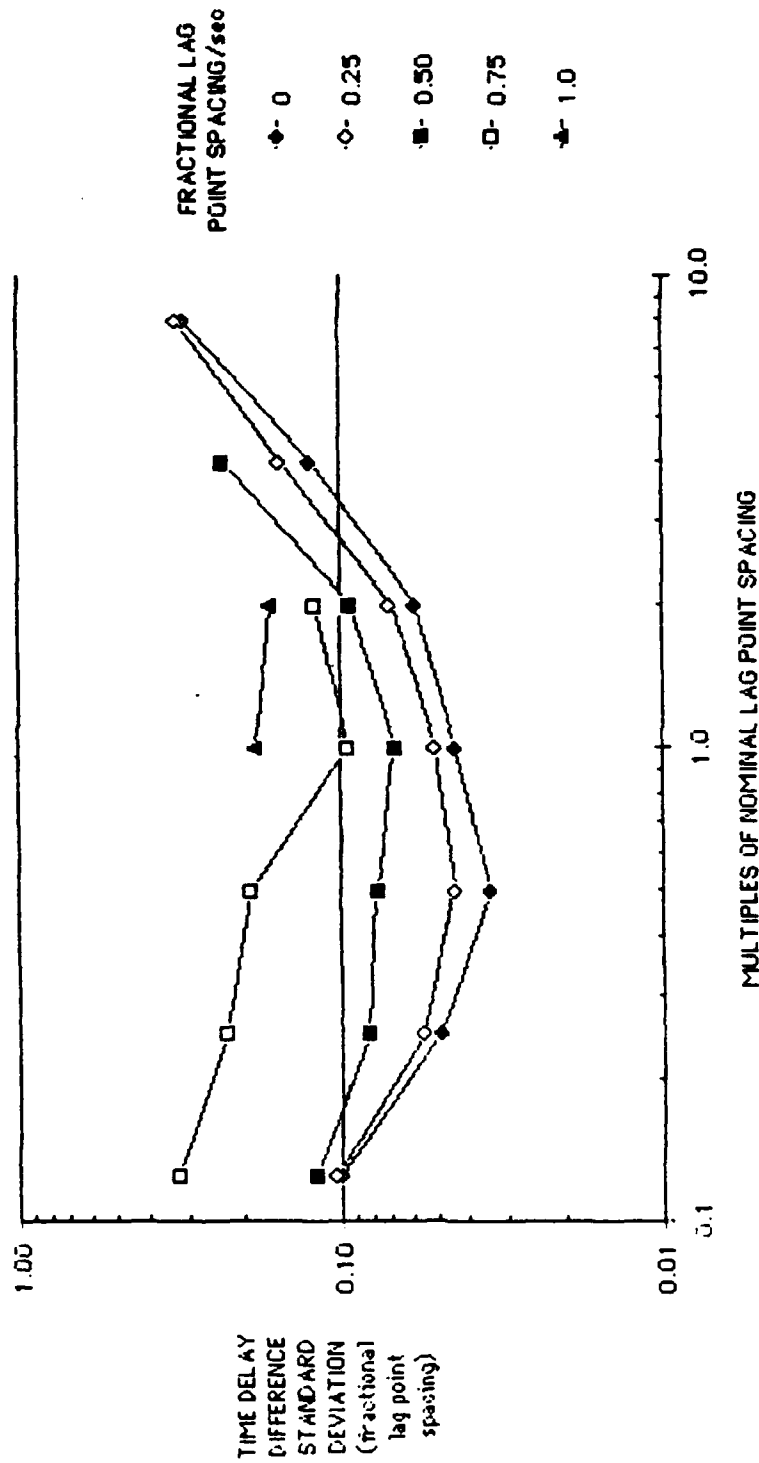


Figure 5-41. Theoretical Time Delay Difference Standard Deviation vs Lag Point Spacing (SNR = 5 dB, Uncorrelated Noise, $N = 9$, $T = 5$)

from 0.5 times the nominal lag point spacing at low time delay difference rates to 2 times the nominal lag point spacing at high time delay difference rates.

Theoretical predictions calculated at low SNRs essentially yield identical results as Figures 5-39 through 5-41 except with proportionately higher time delay difference standard deviations. These results will not be presented.

We will now attempt to confirm our theoretical prediction with simulations on our major algorithms. Since high time delay difference rates result in questionable simulation results, we will only perform simulations with our nominal low dynamic target geometry (Figure 5-2).

All simulation results represent the average of 5 Monte Carlo simulation, each of 20 minutes duration. The averaging time is set to 5 seconds, the number of lag points is set to 9, and the noise between measured lag points is correlated. Once again we shall limit our simulations to our ML estimator only ($K = 0$). As always, all RMS time delay difference errors are referenced to our nominal lag point spacing.

Figure 5-42 presents the results of the above simulations for the gradient search algorithm at SNR levels of 5 dB and -15 dB.

At +5 dB the RMS error of our time delay difference estimator varies wildly with lag point spacing. There is a distinct optimum at the nominal lag point spacing. Fortunately, the -15 dB simulation

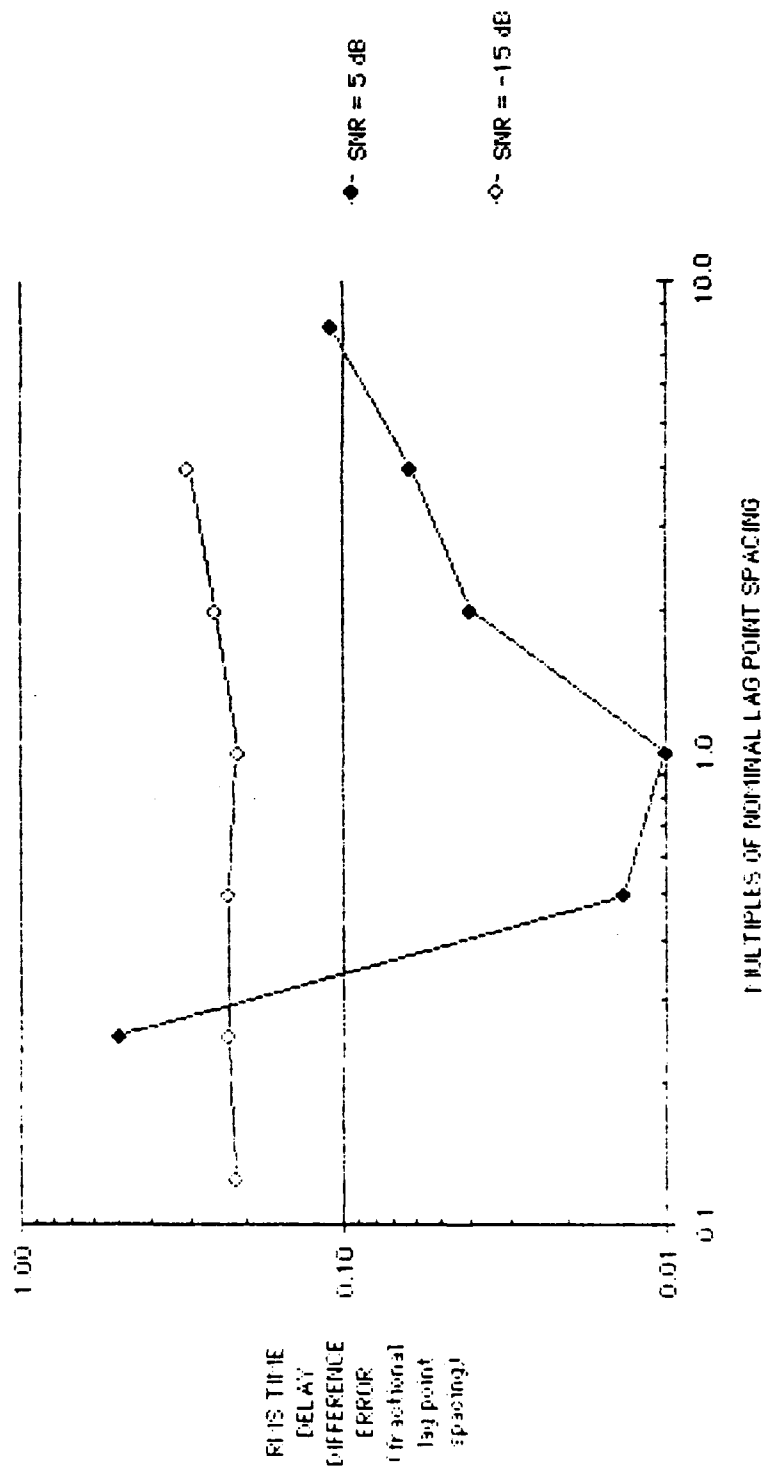


Figure 5-40 RMS Time Delay Difference Error vs Lag Point Spacing, Gradient Search (Correlated Noise, $N = 9$, $T = 5$, $K = 0$)

results essentially agree with our theoretical predictions. At short lag point spacings there is small variation in the time delay difference RMS error. At lag point spacings greater than our nominal lag point spacing, there is a gradual increase in time delay difference RMS error until at lag point spacing of 8 times our nominal lag point spacing the gradient search algorithm diverges. The erratic results obtained at 5 dB we will attribute to numerical problems that have been examined earlier.

The dynamic programming algorithm complicates our simulation due to its quantization. Reducing the lag point spacing automatically reduces lag point resolution and the maximum time delay difference change allowed in a given measurement.

Figure 5-43 presents simulation results for the dynamic programming algorithm at SNR levels of 5 dB and -13 dB. The conditions are identical to the gradient search algorithm shown in Figure 5-42.

The results shown in Figure 5-43 are interesting. At the lowest lag point spacing, the maximum time delay difference update (NPMAX) is about the order of the actual time delay difference rate. This should result in a degradation in RMS time delay difference error as seen in Figure 5-43. At the high undersampled lag point spacing the RMS time delay difference error increases and the algorithm diverges as would be expected. For moderate lag point spacings, Figure 5-43 shows the RMS time delay difference errors for the 5 dB case remaining constant whereas an increase in RMS time delay difference error can be seen from

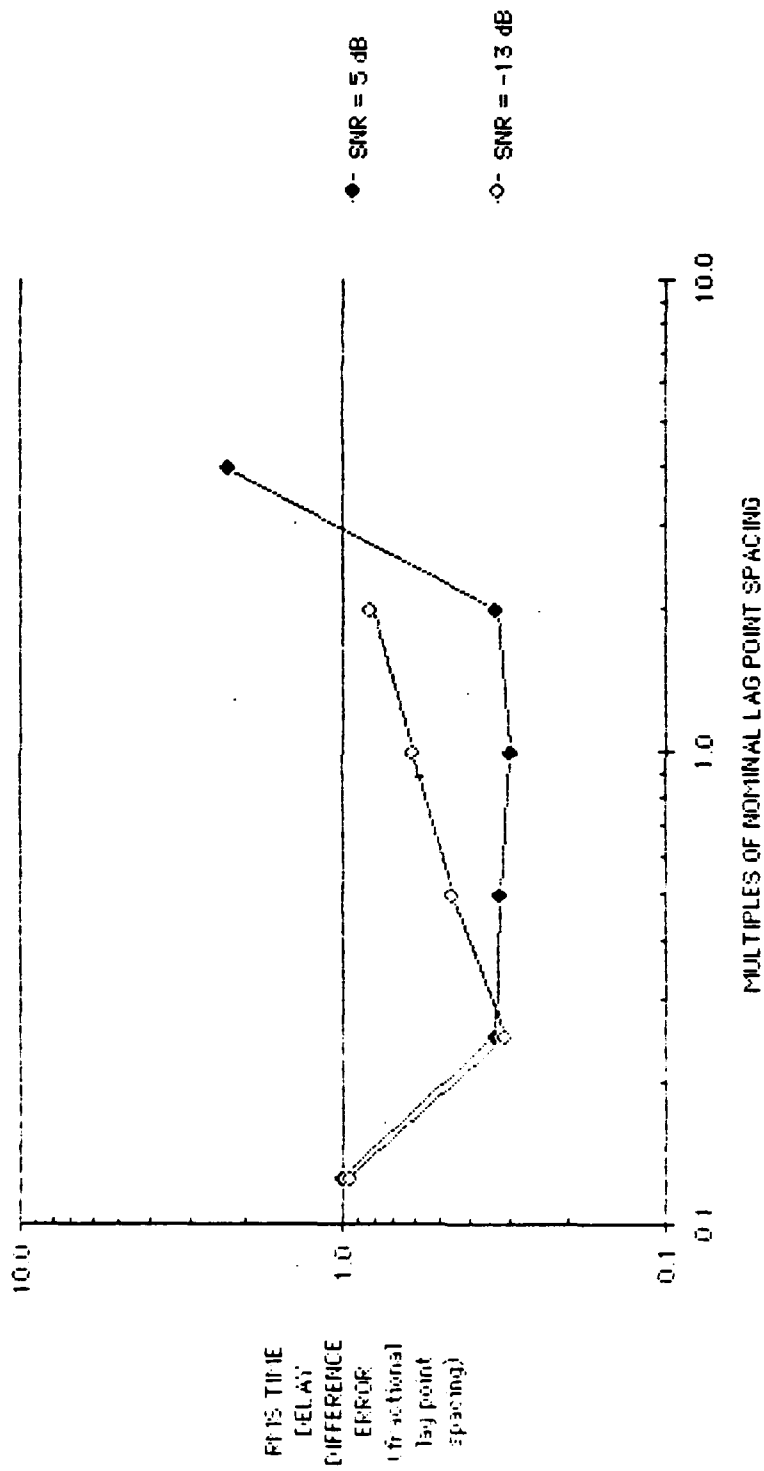


Figure 5-43. RMS Time Delay Difference Error vs Lag Point Spacing, Dynamic Programming (Correlated Noise, $N = 9$, $T = 5$, $K = 0$)

1/4 the nominal lag point spacings to 2 times the nominal lag point spacing. The increase in RMS time delay difference error with increasing lag point spacing would be reasonable since the resolution of the time delay differences is decreasing linearly with increasing lag point spacing. At high SNR, numerical resolution problems may account for the relative constant RMS time delay difference error with lag point spacing.

One should not be deceived by apparent improvement in the dynamic programming algorithm performance at 1/4 the lag point spacing. The dynamic following ability for the same complexity algorithm has been severely limited by the resulting smaller maximum allowable time delay difference change (NPMAX). Increasing NPMAX will result in significantly more computational requirements. In the final analysis, the a priori information will almost certainly reduce any apparent gain to zero. Therefore, even for the dynamic programming algorithm we will recommend that the nominal lag point spacing be adopted.

CHAPTER 6

COMPARATIVE ALGORITHM PERFORMANCE

6.1 INTRODUCTION

In this chapter we have two basic objectives. The first is to present the performance of our best MAP estimation algorithm referenced to a basic standard. The best MAP estimation algorithm is the gradient search algorithm and the basic standard was chosen as the Cramer Rao Lower Bound (CRLB) for coherent integration on a stationary signal (no a priori information). The second objective is to present relative performance for other alternatives/approaches discussed in Chapters 3 and 4.

The above objectives were accomplished by conducting Monte Carlo simulations on the gradient search and other related algorithms. The results of these simulations are plotted for visual comparison. Unless otherwise stated, all plotted data points consist of the average of five 20 minute simulation runs.

The simulation data is presented using five different criteria as outlined below.

1. Semi static^{*} performance of the gradient search algorithm.

* The geometry used is shown in Figure 5-2. This geometry is not static, but has low time delay difference rates. The reasons for not using a strictly static geometry are discussed in Chapter 5.

2. Semi-static performance comparison between the gradient search algorithm and other related MAP algorithms.
3. Performance of the gradient search algorithm with target dynamics.
4. Performance of the gradient search algorithm with spectral mismatch.
5. Initialization performance of the gradient search and dynamic programming algorithms.

6.2 SEMI STATIC PERFORMANCE OF THE GRADIENT SEARCH ALGORITHM

The time delay difference trajectory shown in Figure 5-2 was used for all simulations in this section.

In order to minimize the effects of initialization on steady state algorithm performance, all simulations were artificially updated with the correct trajectory for the first 10 updates (block size). From the 11th update onward the algorithm proceeded normally. This procedure emphasizes a key point. Tracking and/or estimation algorithms are capable of maintaining track of valid estimates in steady state at a lower SNR than they are capable of initializing on. This points out the important aspect of track initiation in cases of target dynamics.

For our reference function we shall use the CRLB modified to reflect standard deviation in time delay difference error in units of fractional lag point spacing.

The CRLB bound for the case of a single stationary target as seen from two sensors without a priori information can be obtained from Ng⁵ or Carter.²⁶ Their result is given below.

$$\text{VAR}(\hat{t}) \geq 2\pi \left(2T \int_0^\infty \frac{c(\omega)}{1 - c(\omega)} \omega^2 d\omega \right)^{-1} \quad (6-1)$$

where

$$c(\omega) = \left(\frac{G_s(\omega)}{G_s(\omega) + G_n(\omega)} \right)^2 \quad (6-2)$$

$G_s(\omega)$ = the signal power spectrum as seen in identical receive channels

$G_n(\omega)$ = the noise power spectrum as seen in identical receive channels

T = the observation interval

$\hat{t}$ = the time delay difference estimate.

We shall assume that the signal and noise are constant spectral densities as specified in Equations 2-3 and 2-4. Therefore, Equation 6-2 can be rewritten to the following convenient form.

$$c(\omega) = \begin{cases} \left(\frac{\text{SNR}}{1 + \text{SNR}} \right)^2, & \omega_c - \frac{Bw}{2} \leq \omega \leq \omega_c + \frac{Bw}{2} \\ 0, & \text{otherwise} \end{cases} \quad (6-3)$$

where

SNR = the signal-to-noise ratio referenced to a receive channel
(halfbeam)

$$\omega_c = \pi(f_2 + f_1)$$

$$Bw/2 = \pi(f_2 - f_1)$$

f_2 = upper bandpass cutoff frequency

f_1 = lower bandpass cutoff frequency.

Since $c(\omega)$ is flat over the ranges $\omega_c - \frac{Bw}{2}$ through $\omega_c + \frac{Bw}{2}$ the integral in Equation 6-1 can be written as follows.

$$\int_{\omega_c - Bw/2}^{\omega_c + Bw/2} \omega^2 d\omega \quad (6-4)$$

The above integral evaluates to the following.

$$\frac{Bw}{3} [3\omega_c^2 + (Bw/2)] \quad (6-5)$$

Using Expression 6-1 through 6-5 we obtain

$$\text{VAR}(\hat{t}) = \frac{3\pi}{B_w T [3\omega^2 + (B_w/2)^2]} \frac{1 + 2\text{SNR}}{\text{SNR}^2} \quad (6-6)$$

We will now modify Equation 6-6 to reflect time delay difference standard deviation in units of fractional lag point spacing.

$$\text{STD}(\hat{t}) = \sqrt{\text{VAR}(\hat{t})/\text{Tau}} \quad (6-7)$$

where

STD = our modified CRLB

Tau = the nominal lag point spacing.

We shall use Equation 6-7 to calculate the modified CRLB for observation time $T = 5$ and $T = 50$. $T = 5$ is the observation time for our nominal measurement update and $T = 50$ is the observation time for our nominal block size of 10.

Figure 6-1 consists of three pairs of plots. The first pair is the midpoint and endpoint RMS time delay difference errors versus SNR (dB) for a 5 second ($T = 5$) measurement update integration time. The second pair is the midpoint and endpoint RMS time delay difference errors versus SNR (dB) for a 2 second ($T = 2$) measurement update integration time. The third set of plots is our modified CRLB for an integration time of $T = 5$ and $T = 50$ seconds.

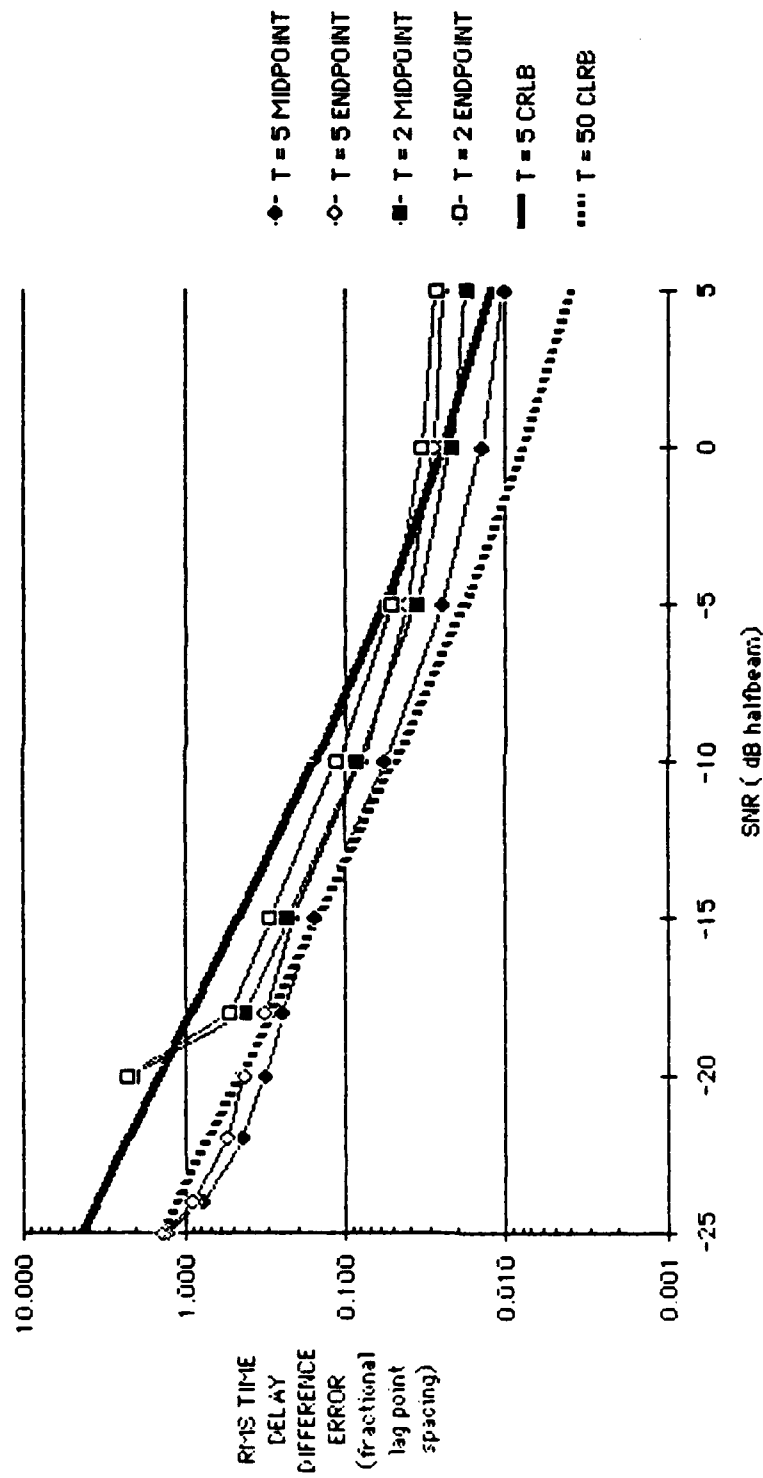


Figure 6-1. RMS Time Delay Difference Error vs SNR, Gradient Search ($K = 1.5$, $N = 9$, $M = 10$)

The gradient search algorithm was used for all simulations shown in Figure 6-1. The block size was fixed at 10 ($M = 10$). The number of lag points was 9 ($N = 9$) and the a priori information proportionality constant was set to 1.5 ($K = 1.5$).

Examining Figure 6-1 we can make the following observations.

1. The RMS time delay difference error for the endpoint measurement is always greater than the corresponding midpoint measurement. At high SNR this difference is greater and at low SNR the difference is smaller. In between, the RMS time delay difference error difference is relatively constant at about 1.5 dB. At high SNR our midpoint and endpoint estimators should be dominated by the weak piecewise linear assumption. As the SNR is lowered, our midpoint and endpoint estimators will become more dominated by our quadratic time delay difference trajectory assumption.
2. The RMS error for the 2 second measurement update integration time is always greater than the corresponding RMS error for the 5 second measurement update integration time. Since the time delay difference rate is small, this result is as expected.
3. The midpoint RMS time delay difference error for the 5 second integration time is always less than the corresponding 5 second CRLB and at low SNR is less than the 50 second CRLB.

This result is not a contradiction of the CRLB. The proper calculation of the CRLB¹² requires an additional term due to a priori information. We have neglected this term. Even at high SNR the weak piecewise linear assumption is a priori information, thus we would expect our estimator to outperform the $T = 5$ CRLB as a result of this information. As the SNR is lowered, the stronger quadratic time delay difference assumption will add more a priori information to our MAP estimator via the LMSF penalty function (see Equation 3-104). This is true since the relative magnitude between our ML term and our a priori information term will favor the a priori information term as SNR is lowered. This a priori information is not accounted for in our CRLB calculation.

4. Our final observation is that our RMS time delay difference error curves for our gradient search algorithm apparently deviate from the expected lower trend at high SNR. This may be a result of numerical problems at high SNR which is discussed earlier.

We will now examine the overall effect of the proportionality constant K on our gradient search algorithm.

Figure 6-2 presents the RMS time delay difference errors versus SNR for three different choices of the proportionality constant K . Once again included in Figure 6-2 are the $T = 5$ and $T = 50$ second modified CRLB curves. The choices for the proportionality constant K are

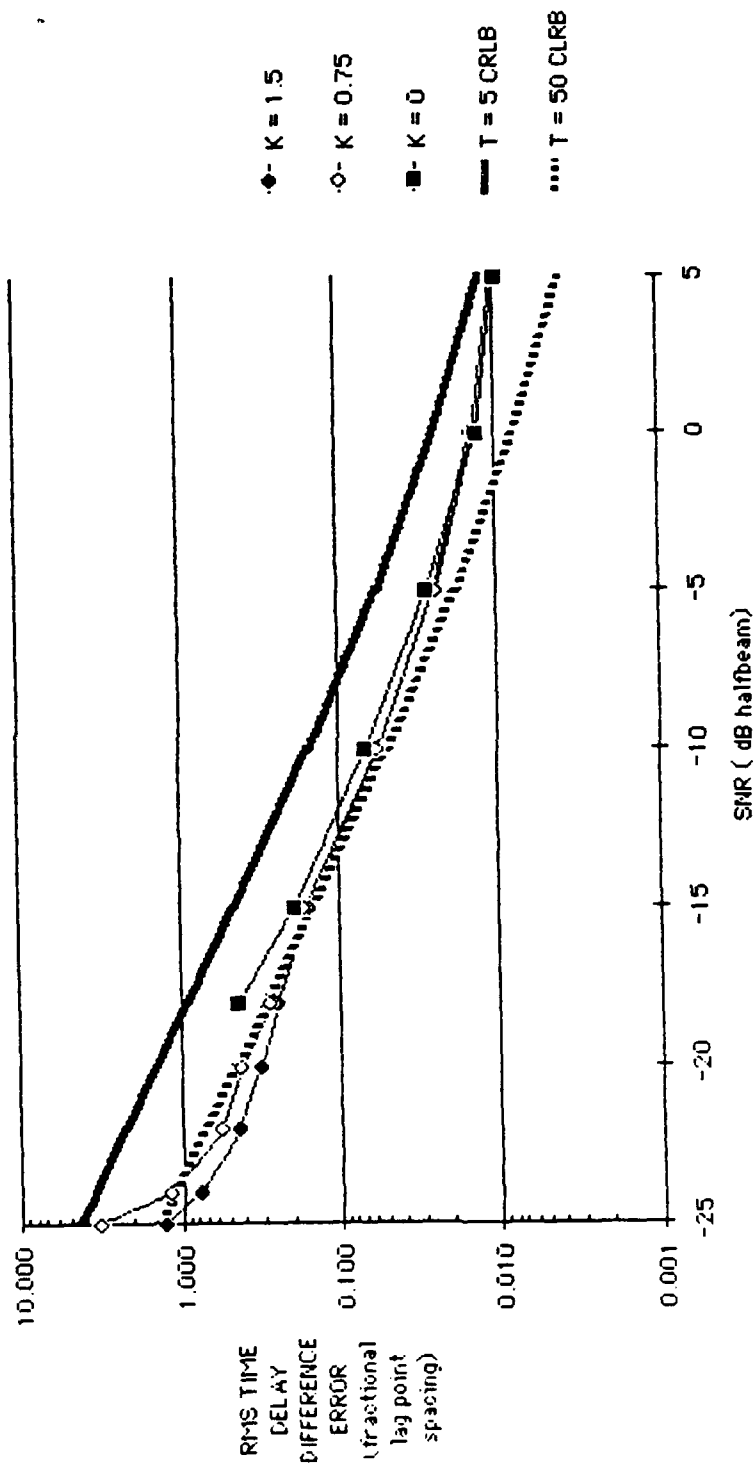


Figure 6-2. RMS Time Delay Difference Error vs SNR, Gradient Search ($T = 5$, $N = 9$, $M = 10$)

$K = 1.5$, $K = 0.75$ and $K = 0$. Note that $K = 0$ is our ML estimator only. The block size M is set to 10 and the number of lag points N is 9.

Examining Figure 6-2 we can make the following observations.

1. At low SNR (< -15 dB), the $K = 1.5$ option has the smallest RMS time delay difference error. This fact is in agreement with data analyzed in Chapter 5. Since good low SNR performance is desirable, we have chosen $K = 1.5$ as our nominal design.
2. Below -18 dB the ML estimator ($K = 0$) diverges. This results states that strong a priori information is required for maintaining track or stability for the given test conditions.
3. Neglecting algorithm stability at low SNR, Figure 6-2 indicates a relatively small variation in RMS time delay difference error between $K = 0$ and $K = 1.5$. This result states that the a priori information function is primarily required for stability at low SNR.
4. The modified CRLB would be roughly equivalent to the ML only ($K = 0$) RMS time delay difference error curve if we set the CRLB averaging time to 25 seconds. Therefore, if we equate averaging time to a priori information, we may conclude that the piecewise linear assumption is equivalent to averaging for five 5 second measurement updates. This result would be in agreement with test results in Chapter 5. In Chapter 5 the ML

only estimator ($K = 0$) indicated variation in RMS time delay difference error for block size of 2 and 4; however, there was no significant variation in RMS time delay difference error for block sizes greater than 8.

The final plot in this section will present the performance of the SNR estimate for the gradient search algorithm. Since there is a known bias due to the data generation program at high SNR, we shall present only the standard deviation in the SNR estimate. The standard deviation was calculated from the SNR estimate in units of dB.

Figure 6-3 presents the standard deviation of the SNR estimate versus SNR. Examining Figure 6-3 we notice a deviation from the trend at an SNR level of 5 dB where the standard deviation in SNR increases with respect to the 0 dB measurement. This deviation may be due to numerical problems in the actual algorithm or in the stimulations software. However, overall the SNR estimates appear quite reasonable with a standard deviation error of less than 1 dB for SNR levels greater than or equal to -15 dB.

6.3 SEMI-STATIC PERFORMANCE COMPARISON BETWEEN THE GRADIENT SEARCH ALGORITHMS AND OTHER RELATED MAP ALGORITHMS

In this section we will compare our chosen gradient search algorithm against related MAP estimators. The comparison will be further broken down into two categories. The first will be a comparison against

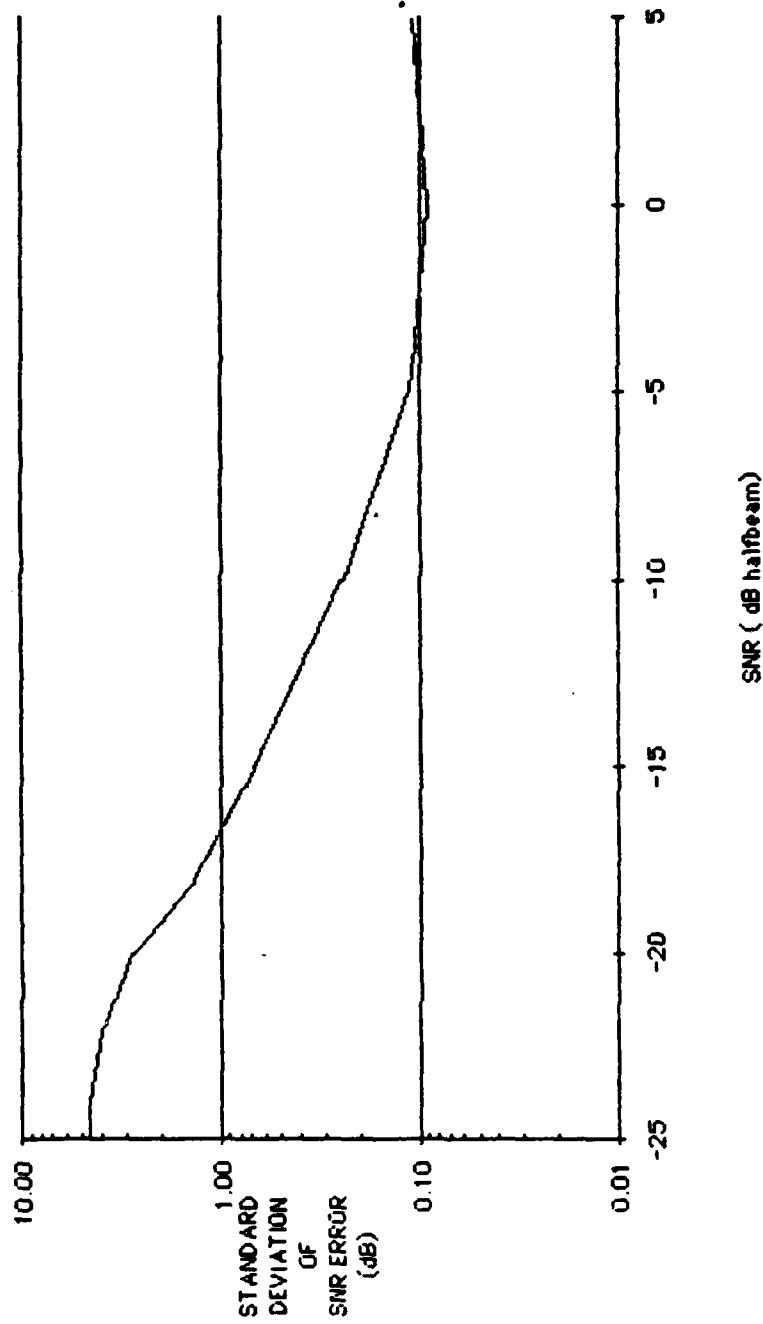


Figure 6-3. Standard Deviation in SNR Error vs SNR, Gradient Search ($K = 1.5$, $T = 5$, $N = 9$, $M = 10$)

related gradient search algorithms. The second will be a comparison against dynamic programming algorithms.

As will be true for the remaining simulations presented in this chapter, all estimates will be from the midpoint of the data block and the integration time T per measurement is set to 5 seconds.

Figure 6-4 shows the RMS time delay difference errors versus SNR for the following gradient search algorithms.

1. The standard gradient search algorithm highlighted in Section 6.2. The proportionality constant $K = 1.5$, the block size $M = 10$, the number of lag points $N = 9$, and the integration time $T = 5$ seconds.
2. The standard gradient search ML estimator followed by a separate quadratic LMSF smoother. For this algorithm there is no a priori information penalty function. The block size $M = 10$, the number of lag points $N = 9$, and the integration time $T = 5$ seconds.
3. The standard gradient search algorithm with prewhitening. (Identical parameters as the standard gradient search algorithm.)

The above algorithms are discussed in Chapters 3 and Chapter 4.

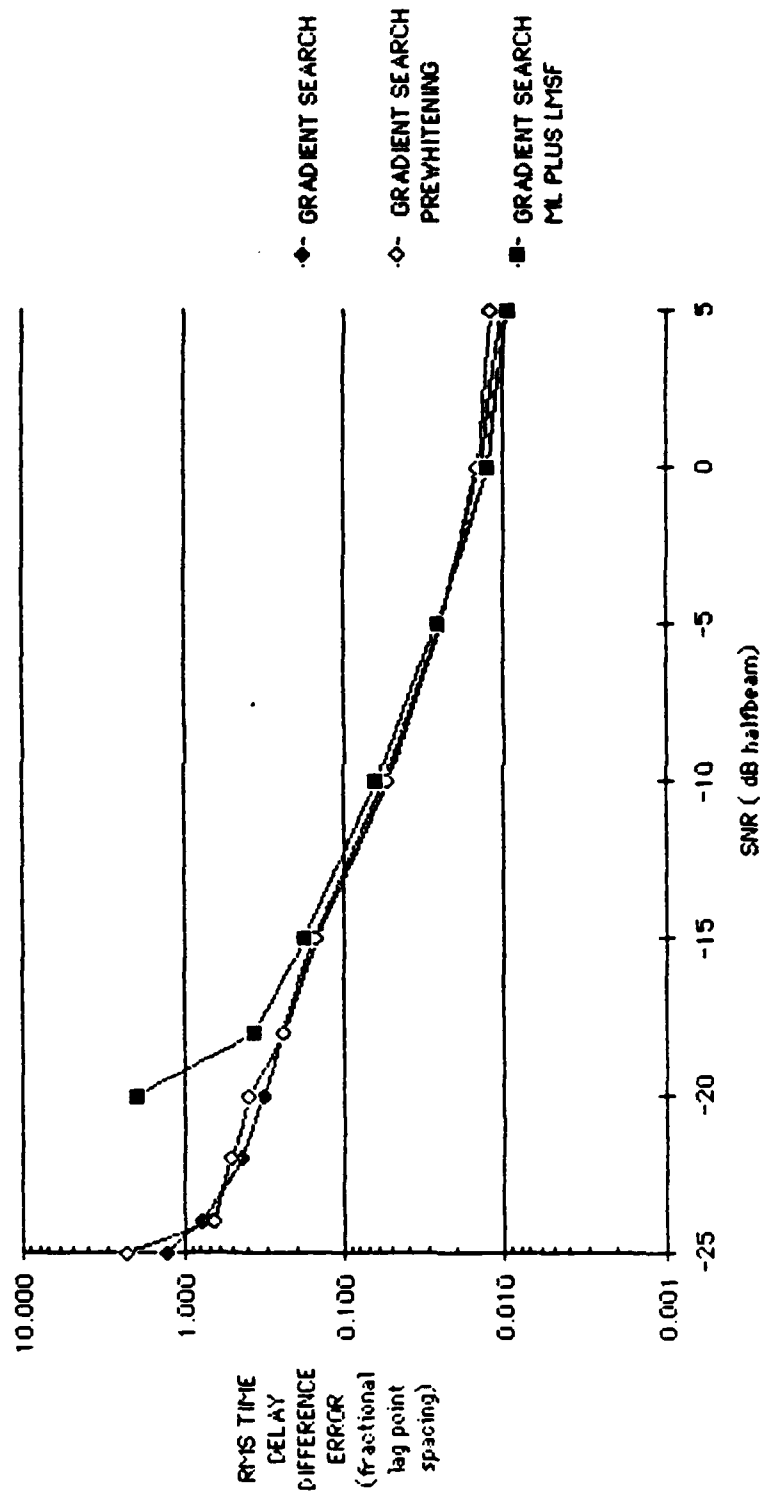


Figure 6-4. RMS Time Delay Difference Error vs SNR

Examining Figure 6-4 we can make the following observations.

1. The gradient search algorithm with prewhitening is at best equivalent to the nominal gradient search algorithm. Since the prewhitening involves a lot more processing for no gain, we can conclude that prewhitening is not a viable option.
2. The gradient ML estimator followed by the LMSF smoother demonstrates instability at SNR levels below -18 dB. Note that the -20 dB point represents one successful estimation in five attempts (i.e., four attempts diverged). This experiment verifies our conclusion in Chapter 3. An ML estimation over a block of data followed by an LMSF smoother is not equivalent to a simultaneous ML estimation and LMSF a priori information penalty function operating on a block of data.

Figure 6-5 shows the RMS time delay difference error versus SNR for the gradient search algorithm and the following dynamic programming algorithms.

1. A dynamic programming algorithm with a Sum of Square Residual (SSR) a priori information penalty function. (This is the same a priori information penalty function used by the gradient search algorithm.)
2. A dynamic programming algorithm with a Sum of Squared Differences (SSD) a priori information penalty function. (This is

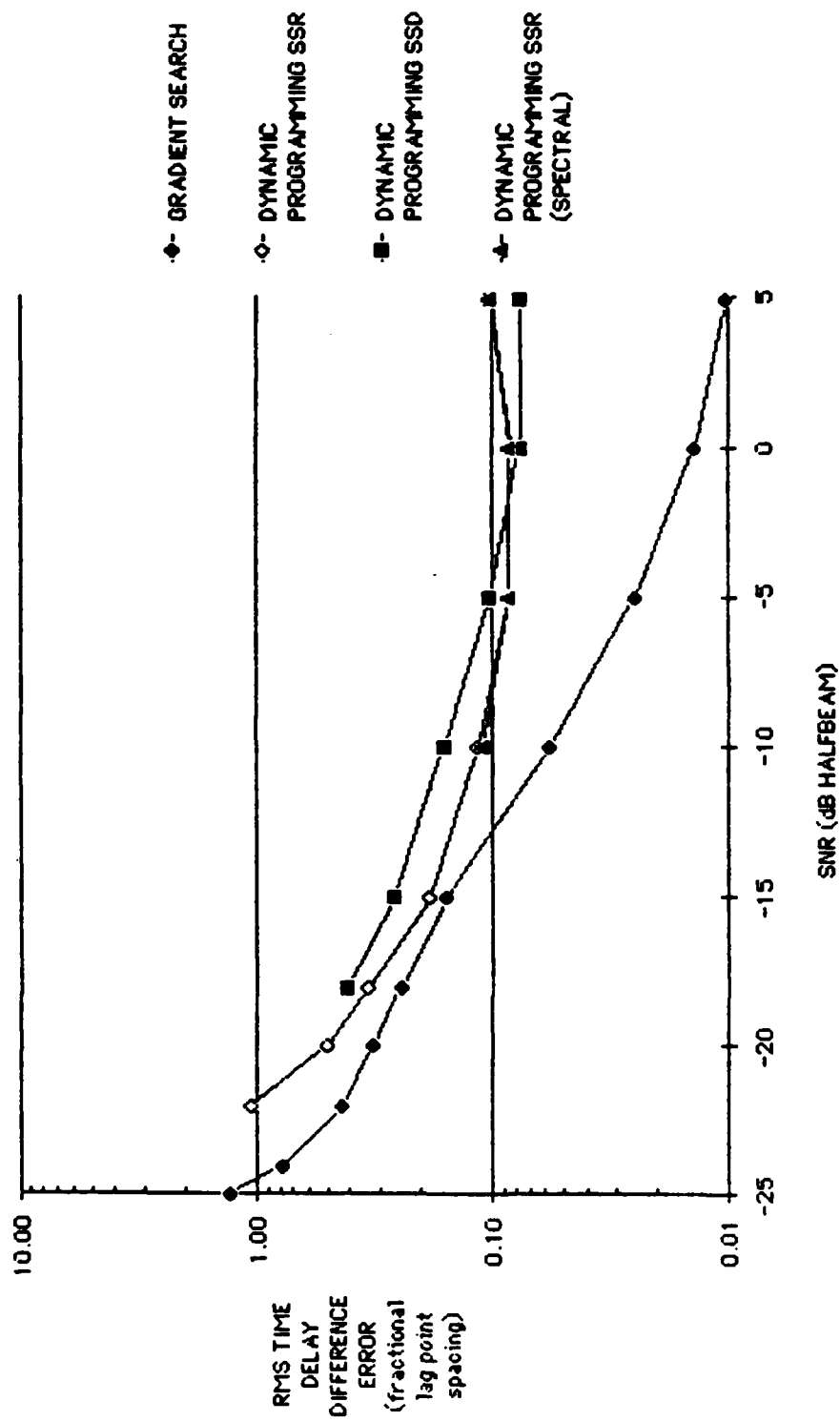


Figure 6-5. RMS Time Delay Difference Error vs SNR

the a priori information penalty function derived by Wolcin.^{6,7,8)}

3. A dynamic programming algorithm with an SSR a priori information penalty function which contains a spectral estimation algorithm within the ML estimator.

Note all algorithms used the optimum parameters found. Block size $M = 16$, proportionality constant $K = .075$, number of lag points $N = 9$, and integration time $T = 5$ seconds.

Examining Figure 6-5 we can make the following observations.

1. Our first observation is that the dynamic programming algorithm containing the spectral estimation parallels the dynamic programming SSR algorithm until -10 dB. At SNR levels below -10 dB the spectral dynamic programming algorithm diverges. The source of the instability is apparently the spectral estimation which we shall examine in more detail in a following section.
2. The second observation is that although the dynamic programming SSD has better performance at high SNR it degrades in performance with respect to the dynamic programming SSR algorithm at low SNR. The SSR algorithm appears to have a lower threshold than the SSD algorithm; however, this may not actually be the case. The -22 dB SNR point for the SSD algorithm is a

single successful measurement in five attempts. However, the superior performance at low SNR for the SSR algorithm over the SSD algorithm is our reason for selecting the LMSF a priori information penalty function.

3. The third observation is that the gradient search algorithm is clearly superior in RMS time delay difference error and threshold than any of the dynamic programming algorithms. However, the current gradient search algorithm requires at least four times the computation time as the current dynamic programming algorithm.
4. The final observation is the obvious quantization problem at high SNR for the dynamic programming algorithms. This point is highlighted by the RMS time delay difference error being smaller at 0 dB than at 5 dB as discussed in Chapter 5.

6.4 GRADIENT SEARCH ALGORITHM DYNAMIC PERFORMANCE

In this section we shall examine the degradation in the gradient search algorithm with increased target dynamics. We have discussed previously the difficulty in performing precise realistic dynamic testing. Therefore we claim only a relative measure of the effects of dynamics on the gradient search algorithm. We shall also present the performance of the gradient search algorithms rate estimate. The acceleration estimate for the gradient search algorithm did not yield meaningful estimates and therefore shall not be presented.

The increased target dynamics represents a doubling and quadrupling of the nominal target speed. Although the target speed has been doubled twice, the average time delay difference rate only increased approximately by factors of 1.95 and 3.8. The observation platform has a nominal speed. Therefore the change in target speed has altered the nominal target and observer relative position throughout the scenario. We shall, however, continue to refer to these dynamic scenarios as doubling and quadrupling the nominal time delay difference rate. For reference, the average nominal time delay difference rate is 1.8 fractional lag point spacing per minute.

Figure 6-6 illustrates the RMS time delay difference errors versus SNR of the gradient search algorithm for three different target time delay difference trajectories. The time delay difference trajectories consist of the nominal dynamic time delay difference trajectory (shown in Figure 5-2) and similar trajectories, discussed above, where the target speed has been doubled and quadrupled.

The simulation for the doubled nominal target dynamics were 10 minutes in duration and the simulations for the quadrupled target dynamics were 5 minutes in duration. The reduction in test time was required to maintain a similar time delay difference domain for all three dynamic scenarios. The integration time $T = 5$ seconds, the number of lag points $N = 9$, the block size $M = 10$, and the proportionality constant $K = 1.5$. These are the standard parameters for our chosen gradient search algorithm highlighted in Section 6.2.

Examining Figure 6-6 we can make the following observations.

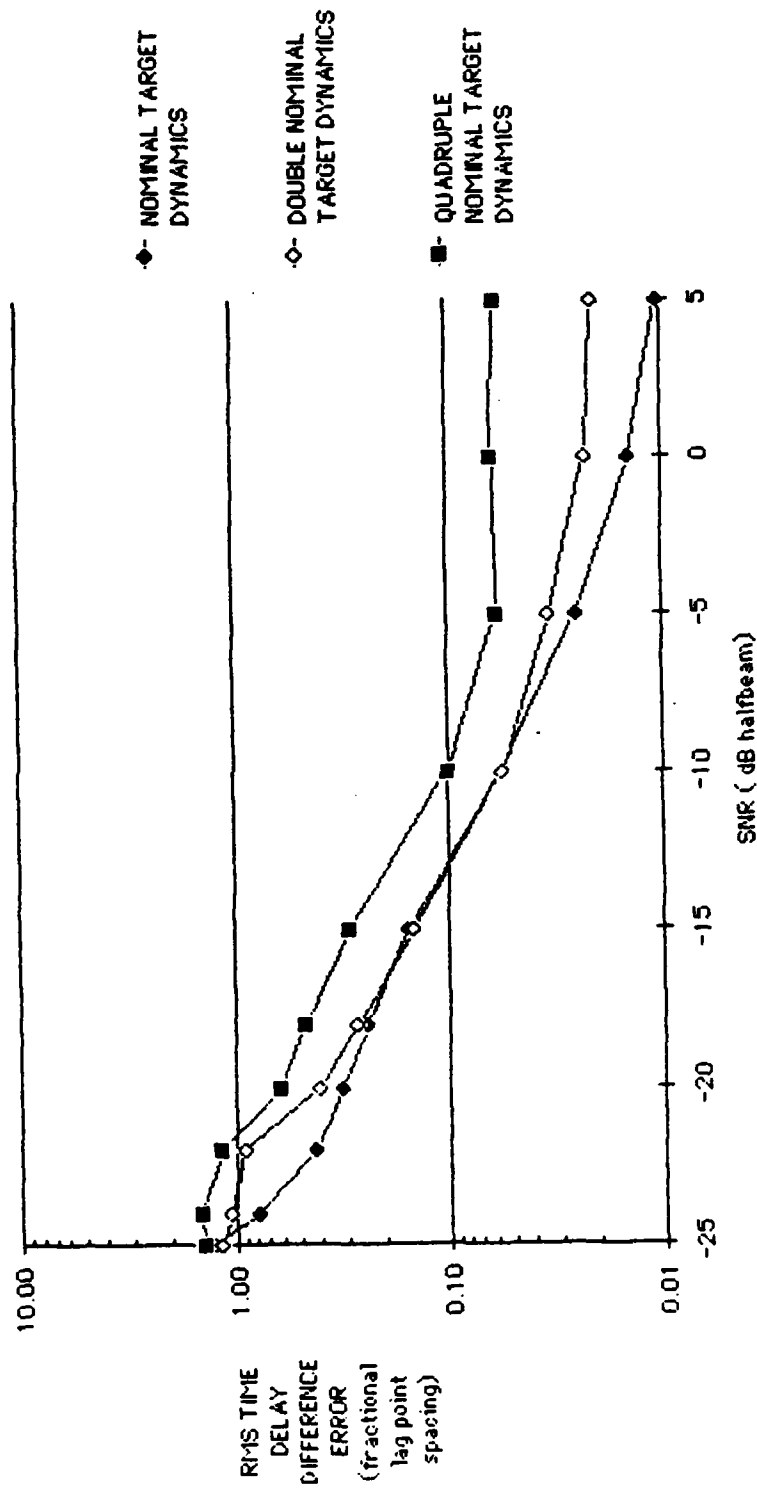


Figure 6-6. RMS Time Difference Error vs SNR, Gradient Search
($N = 9$, $M = 10$, $K = 1.5$, $T = 5$ sec)

1. At high SNR (> -5 dB) the increased target dynamics results in a quite noticeable degradation in RMS time delay difference error. At lower SNR this difference in RMS time delay difference error is reduced. This result may be due to initialization transients which are masked at lower SNR.
2. At low SNR the RMS time delay difference errors for the double nominal target dynamic scenario are effectively equivalent to the corresponding RMS time delay difference errors for the nominal target dynamic scenario. The apparent discrepancy at -22 dB may be attributed to marginal measurement accuracy at very low SNR.
3. The RMS time delay difference error for the quadruple nominal target dynamic scenario has a noticeable increase in error over the RMS time delay difference error for the lower target dynamic scenarios. Apparently, the quadruple nominal target dynamics is sufficient to degrade the gradient search algorithm at all SNR levels.

Figure 6-7 shows the RMS time delay difference rate error versus SNR for the same target dynamic scenarios as in Figure 6-6. The RMS time delay difference rate errors will be in units of fractional lag point spacing per minute. The averaging time delay difference rate for the nominal target dynamic scenario, double nominal target dynamic scenario are 1.8, 3.5, and 6.8 fractional lag point spacing per minute, respectively.

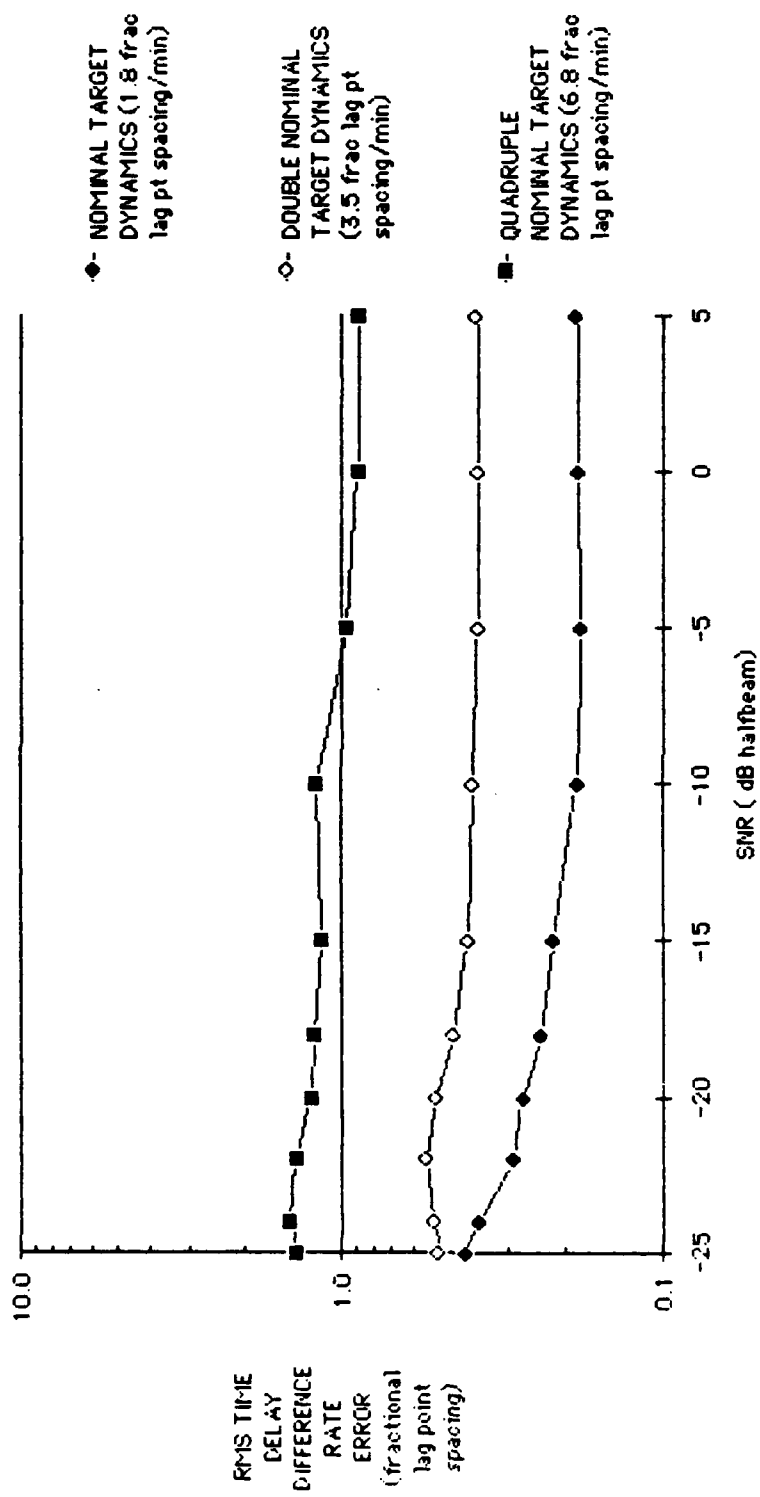


Figure 6-7. RMS Time Delay Difference Rate Error vs SNR

Examining Figure 6-7 we can make the following observations about the rate estimate of the gradient search algorithm. Note, however, that the time delay difference rate estimate results from the coefficients of the quadratic LMSF a priori information penalty function (Equation 3-106).

1. Our first observation is that the maximum RMS time delay difference rate errors are significantly below the average time delay difference rate for all three target dynamic scenarios at all SNR levels measured. This result states that the time delay difference rate estimate is a valuable and meaningful estimate at all SNR levels.
2. At high SNR we notice that doubling and quadrupling the target dynamics (actual 1.95 and 3.8) has increased our RMS time delay difference rate error by factors of 2.04 and 4.74. This result indicates that the RMS time delay difference rate error is proportional to the actual time delay difference rate with an increased degradation as the actual time delay difference rate becomes larger.
3. Our most striking observation is the flat RMS time delay difference rate error versus SNR at high SNR for all three target dynamic scenarios.

6.5 GRADIENT SEARCH ALGORITHM PERFORMANCE WITH SPECTRAL MISMATCH

The gradient search algorithm was derived under the assumption of a flat target power spectrum. In this section we shall examine the degradation in the gradient search algorithm RMS time delay difference error with respect to target power spectrums not conforming to our model. Our target power spectrums will be limited to linear models (i.e., the spectrum power changes linearly with frequency from the lower cut off frequency f_1 to the upper cut off frequency f_2). The following is a description of the power spectral models used in the simulation results. Note that although the power spectral models differ they have been normalized to contain equivalent total power over the band pass (f_1 through f_2).

Model 1	$10 \log (f_2/f_1) = -6 \text{ dB}$
Model 2	$10 \log (f_2/f_1) = -3 \text{ dB}$
Model 3	$10 \log (f_2/f_1) = 0 \text{ dB}$
Model 4	$10 \log (f_2/f_1) = 3 \text{ dB}$
Model 5	$10 \log (f_2/f_1) = 6 \text{ dB}$

Although the power spectral models are specified in dB, they are linear in power and not in dB.

Figure 6-8 show the RMS time delay difference errors versus SNR for the five target power spectral models outlined above. The algorithm under test is the gradient search algorithm highlighted in Section 6.2.

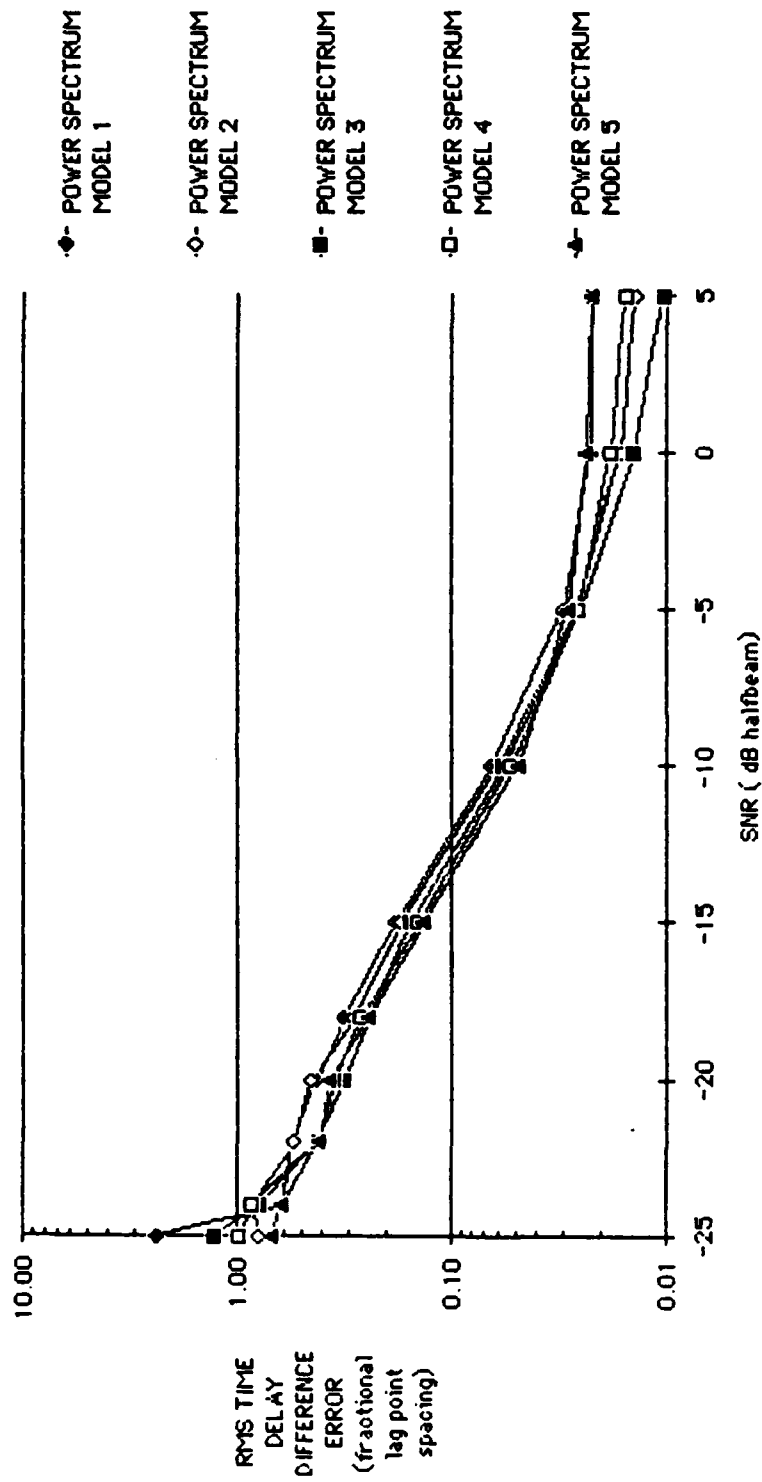


Figure 6-8. RMS Time Delay Difference Error vs SNR, Gradient Search ($K = 1.5$, $N = 9$, $M = 10$, $T = 5$ sec)

The parameters are: the number of lag points $N = 9$; the block size $M = 10$; the proportionality constant $K = 1.5$; and the integration time $T = 5$ seconds.

Examining Figure 6-8 we can make the following observations.

1. From an SNR level of -5 dB to -22 dB the RMS time delay difference error among the various target power spectral models do not differ by much more than 2 dB. This result states that our gradient search estimation algorithm is not very sensitive to moderate target power spectral modeling mismatches. Since in practice target power spectral mismatches are quite likely, it is fortunate that our estimator is robust with respect to these modeling errors.
2. From about -10 dB to -18 dB the RMS time delay difference error for the positive power spectral slope models (models 4 and 5) are less than the RMS time delay difference error for the no mismatch zero power spectral slope (model 3). This result is not surprising since the positive power spectral slope emphasizes the higher frequencies. Substituting Equation 3-92 into Equation 6-1 and performing numerical integration, we can predict the expected gain or loss in RMS time delay difference error due to our target power spectrum models. The results of this prediction indicate about a 1 dB gain for the positive sloped power spectrum (model 5) and a 1 dB loss for the negative slope power spectrum (model 1) for

all SNR levels. This is in agreement with our first observation. Therefore below -5 dB the differences in RMS time delay differences error curves can be related to theory relating to the power spectral shapes rather than modeling mismatches.

3. For SNR levels above -5 dB there is a distinct increase in RMS time delay difference errors directly related to the proportion of the power spectral model mismatch (i.e., the bigger the power spectral mismatch the greater the RMS time delay difference error). This fact indicates that power spectral modeling mismatches (at least for moderate modeling mismatches) are only a factor at high SNR. At low SNR the noise field dominates most of the distinguishable target power spectral features, thus masking modeling errors.

Figure 6-9 presents the RMS spectral slope parameter γ (Equation 3-92A) error versus SNR. The spectral slope parameters were estimated during the same simulation runs shown in Figure 6-8. The following is a list of the actual values of the spectral slope parameter γ for each of the five target power spectral models.

Model 1	$\gamma = -0.598$
Model 2	$\gamma = -0.332$
Model 3	$\gamma = 0$
Model 4	$\gamma = 0.332$
Model 5	$\gamma = 0.598$

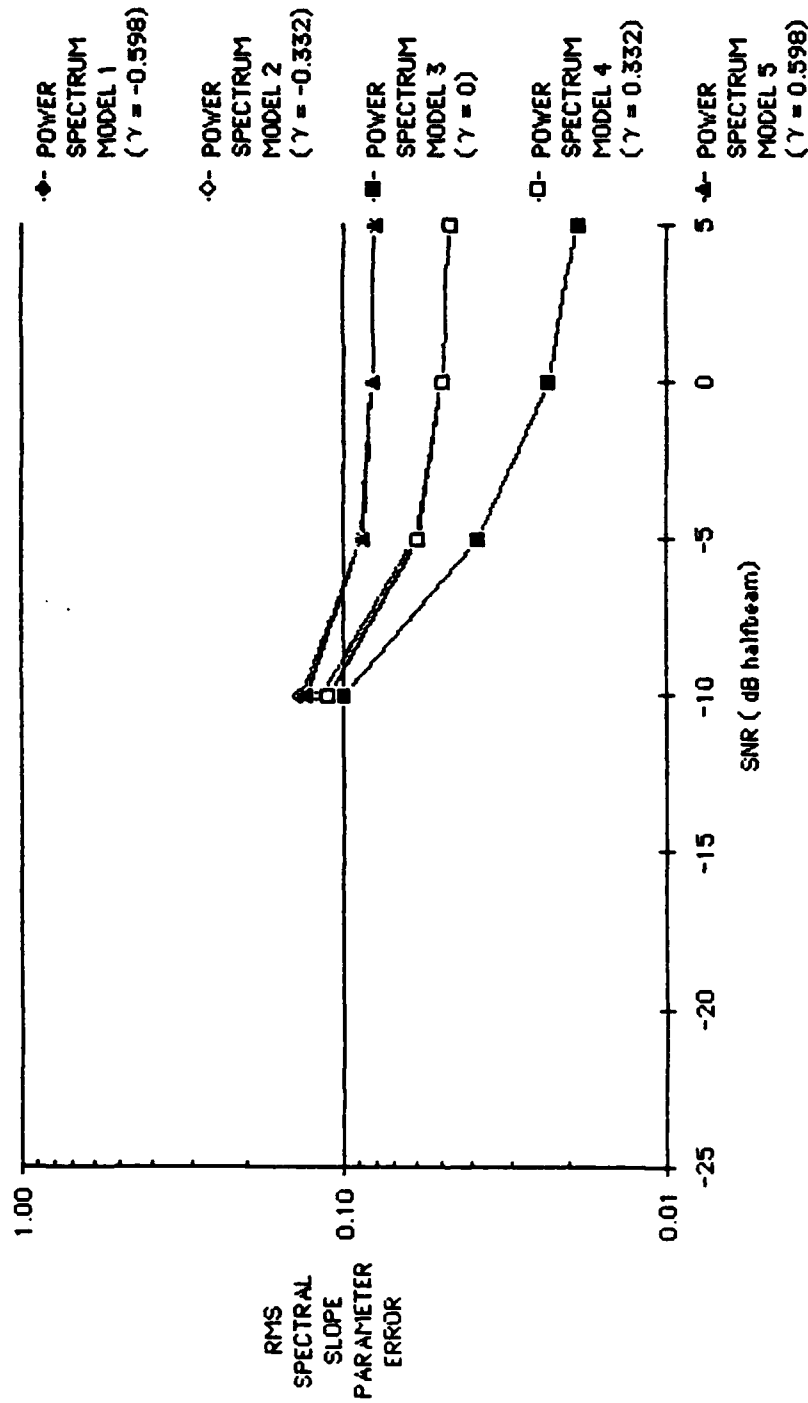


Figure 6-9. RMS Spectral Parameter γ vs SNR, Gradient Search ($K = 1.5$, $N = 9$, $M = 10$, $T = 5$ sec)

Examining Figure 6-9 we can make the following observations.

1. The most important observation is that below -10 dB (-15 dB for certain) the spectral parameter estimate is unstable and meaningless. This is convincing evidence why in Chapter 3 we did not want to include a spectral parameter estimate as an integral part of our ML estimator. Note that the dynamic programming ML estimator which contained a spectral parameter estimation as an integral part diverged at SNR levels below -10 dB (see Figure 6-5). From Figure 6-8 we conclude that modeling mismatches tend to be masked by noise below a -5 dB SNR level. In support of that conclusion we now conclude that our spectral shape estimation is meaningless below -10 dB.
2. Above -10 dB the RMS spectral shape estimates errors are reasonable proportioned to their actual values, thus we have acceptable estimators.
3. The larger power spectrum model mismatch, the greater is the RMS error in the spectral shape parameter. At higher SNR this degradation in spectral shape parameter estimation becomes larger. At higher SNR, target power spectrum modeling mismatches are a more significant factor on the algorithm.

6.6 INITIALIZATION PERFORMANCE

Most theoretical work and corresponding empirical observations concern themselves with the steady state performance of the estimator or algorithm in question. In Section 6.2 through 6.5 we concerned ourselves with the steady state performance of the gradient search and related algorithms. We have advertised that our recursive batch algorithms will have good initialization properties. In this section we shall empirically examine the initialization properties of the dynamic programming (SSR) and gradient search algorithms.

Many algorithms have good initialization characteristics for unrealistic zero target dynamics. We shall examine initialization characteristics of our algorithms with both target dynamics and initial estimate errors. Since initialization is a stochastic process, our unit of measure shall be the number of successful initializations for given initialization conditions in 10 random attempts. We shall perform initialization tests on two target dynamic scenarios. The first will be the nominal target dynamic scenario illustrated in Figure 5-2 and the second will be the double nominal target dynamic scenario. This scenario results from doubling the target speed in the nominal target dynamic scenario. Although the average time delay difference rate does not exactly double (1.95) due to nominal observer motion, certainly at the critical initialization stage the time delay difference rate should be almost exactly doubled.

Using both dynamic target scenarios, we shall initialize our algorithms under test -2, -1, 0, 1, and 2 lag points away from the actual initial time delay difference. We shall perform this experiment for three SNR levels (-15 dB, -18 dB and -20 dB).

The first algorithm that we shall examine is the dynamic programming (SSR) algorithm described in detail in Chapter 4. Since this algorithm simultaneously examines a block of potential time delay difference trajectories, we have claimed that it should have very good initialization properties.

The following is a list of the chosen parameters for the dynamic programming (SSR) algorithm.

Block size $M = 16$

Number of lag points $N = 9$

Proportionality constant $K = .075$

Integration time $T = 5$ seconds

Number of subdivisions per lag spacing $NP = 4$

Maximum subdivision change per measurement update $NPMAX = 8$.

The results for the initialization simulations for the dynamic programming SSR algorithm are given in Table 6-1 for the nominal target dynamic scenario and in Table 6-2 for the double nominal target dynamic scenario.

Table 6-1. Dynamic Programming (SSR) Nominal Target Dynamics

SNR (dB)	Initial Lag Point Offset				
	-2	-1	0	+1	+2
-15	10/10	10/10	10/10	10/10	10/10
-18	9/10	9/10	9/10	10/10	10/10
-20	2/10	4/10	2/10	4/10	6/10

Ratio of successful initializations to attempts

Table 6-2. Dynamic Programming (SSR) Double Nominal Target Dynamics

SNR (dB)	Initial Lag Point Offset				
	-2	-1	0	+1	+2
-15	10/10	10/10	10/10	10/10	10/10
-18	10/10	9/10	9/10	9/10	10/10
-20	0/10	3/10	4/10	5/10	3/10

Ratio of successful initializations to attempts

Examining Tables 6-1 and 6-2 we can make the following observations on the dynamic programming SSR algorithm initialization performance.

1. Our first observation is that the dynamic programming (SSR) algorithm has excellent initialization properties down to -18 dB. The increased target dynamics appears to have little affect at SNR levels of -15 dB and -18 dB.
2. At -20 dB the apparent initialization performance falls considerably off. However, -20 dB is near the steady state threshold for the dynamic programming (SSR) algorithm. Many of the failed initialization attempts occurred at times many minutes after the initialization and potentially could represent unfortunate noise field variations.
3. As would be expected, at -20 dB the double nominal target dynamics scenarios have poorer initialization ratios than the nominal target dynamic scenario.
4. The initialization success ratios for the initial positive lag point offset are in general superior to the 0 or negative initial lag point offsets. Since both dynamic target scenarios are progressing in a positive lag point sense, this fact indicates that better performance can be obtained by initializing at a time delay difference point leading the dynamic target.

Overall, we claim that the dynamic programming (SSR) algorithm has excellent initialization properties.

We shall now examine the gradient search algorithm described in Chapter 4 and examined in Sections 6.1 through 6.5. We have tried to give this algorithm as good initialization properties as the dynamic programming SSR algorithm. However, the gradient search algorithm is more dependant on the initial estimate and does not perform multiple simultaneous searches.

The parameter choices for the gradient search algorithm are repeated below for convenience.

Block size $M = 10$

Number of lag points $N = 9$

Proportionality constant* $K = 1.5$

Averaging time $T = 5$ seconds.

Simulations have indicated that although larger values of the proportionality constant K yield better steady state estimator performance, the initialization performance degrades with higher proportionality constant values. This point is illustrated in Figures 6-10 through 6-12. They each plot the time delay difference error versus time for the gradient search algorithm initialized on target (0 lag

* The final value of the proportionality constant K will be 1.5.

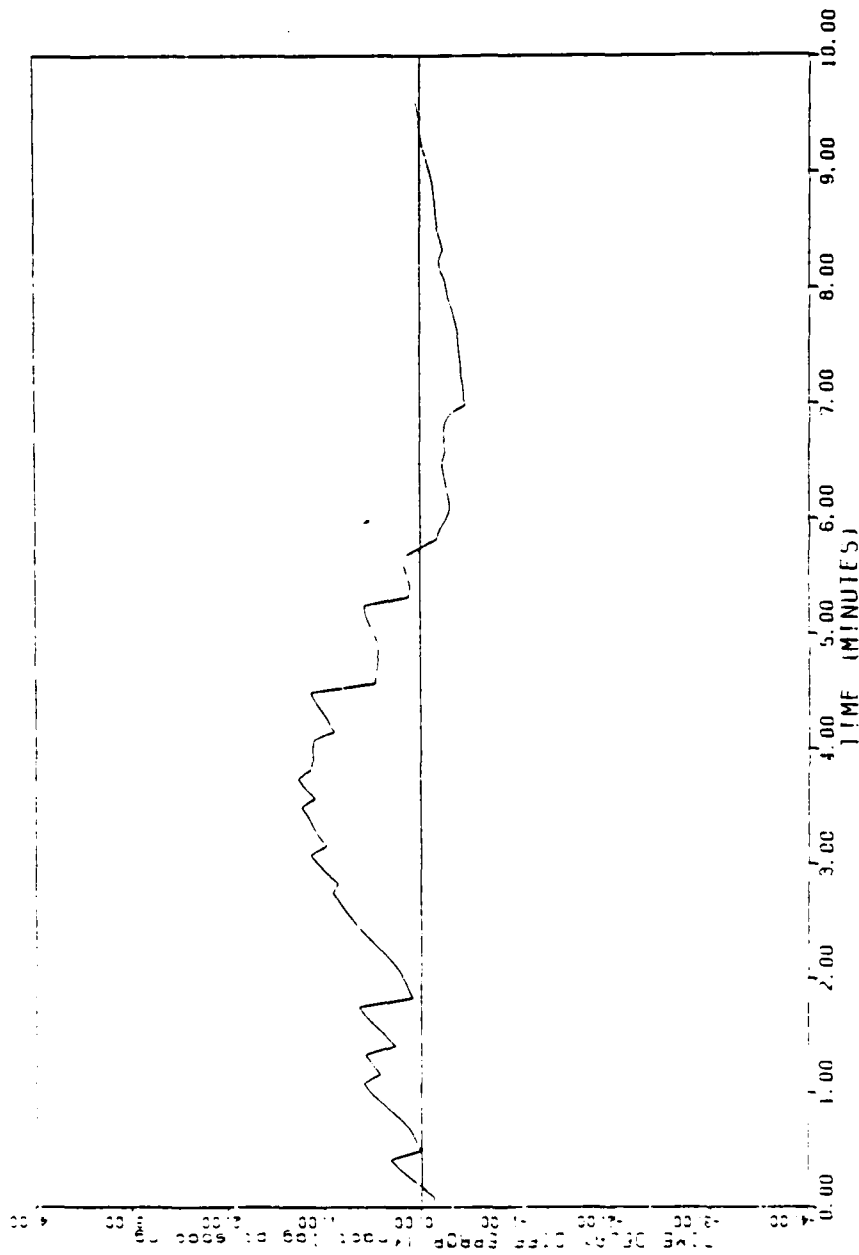


Figure 6-10. Time Delay Difference Error vs Time,
Gradient Search ($K = 1.5$, $\text{SNR} = -18$)

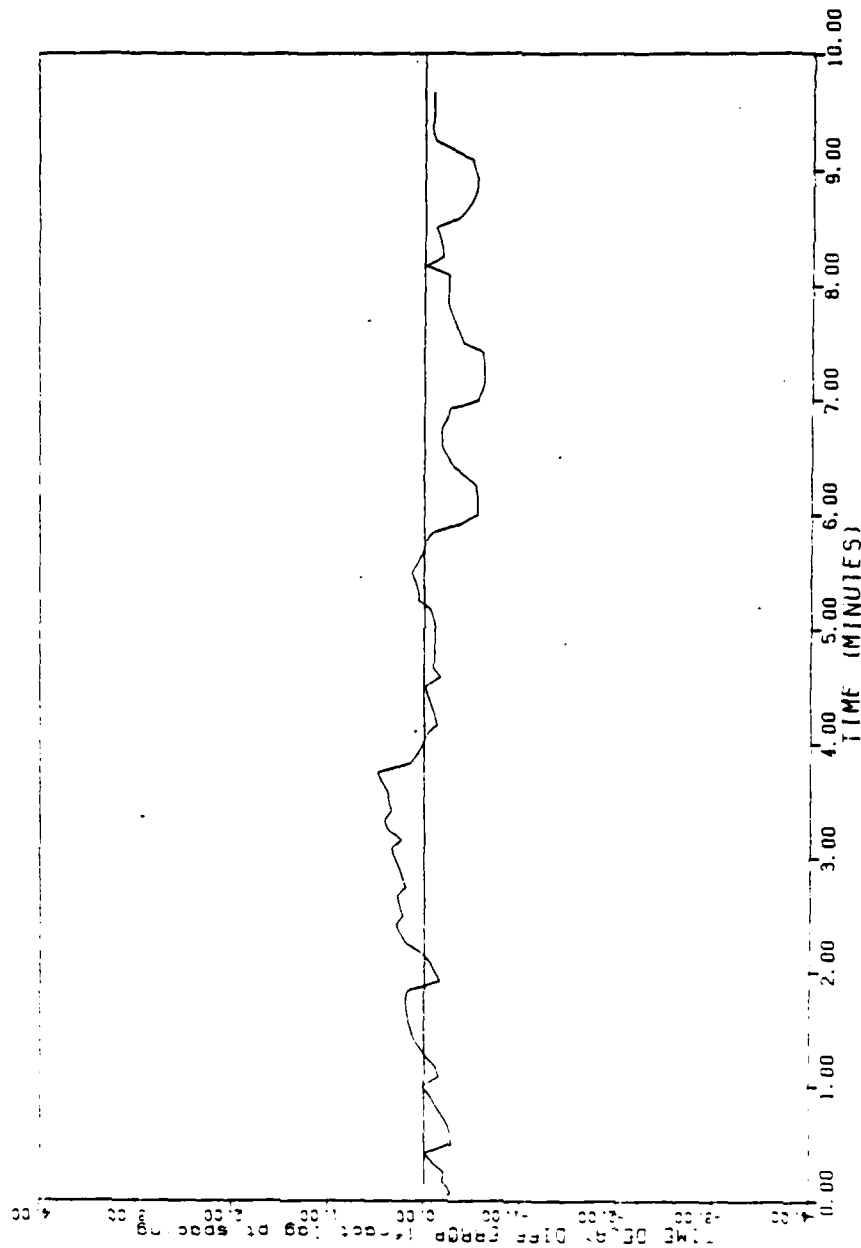


Figure 6-11. Time Delay Difference Error vs Time,
Gradient Search ($K = 0.5$, $SNR = -18$)

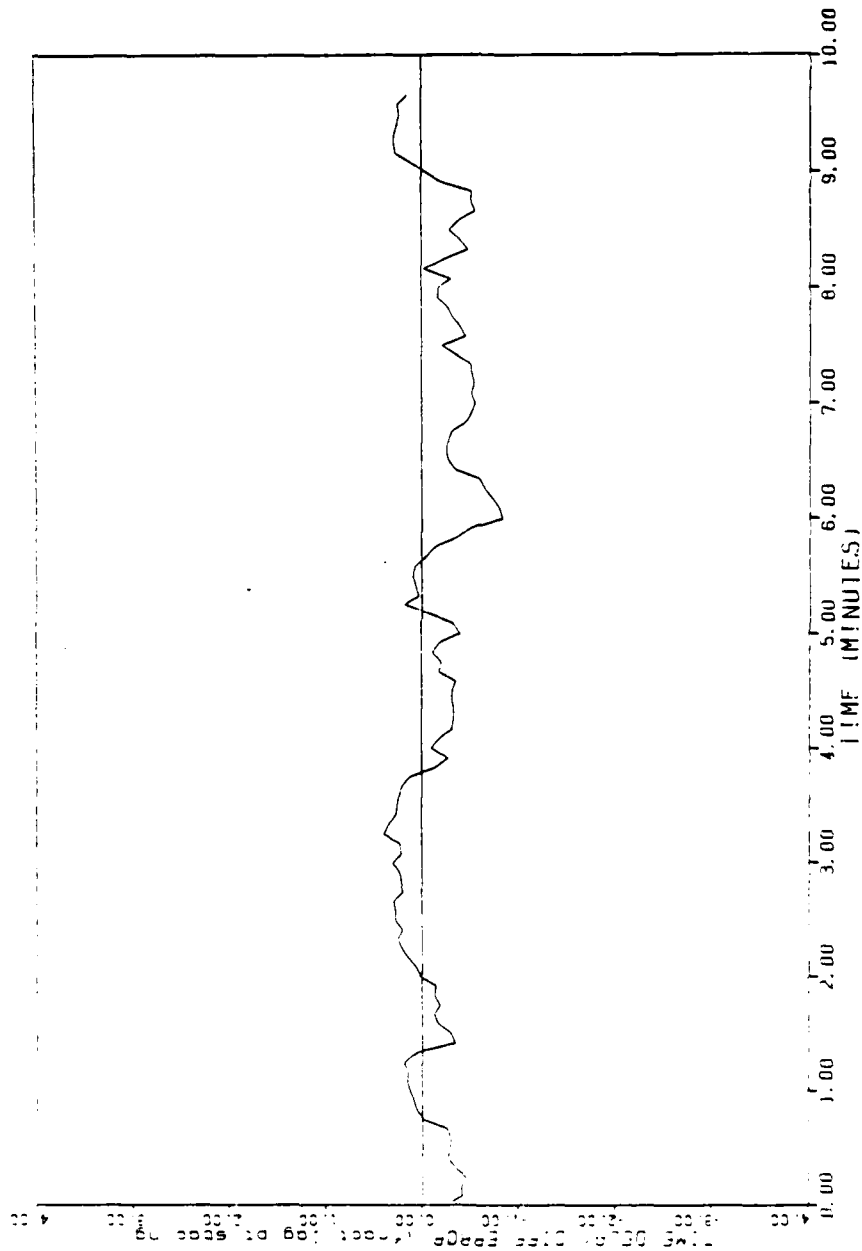


Figure 6-12. Time Delay Difference Error vs Time,
Gradient Search ($K = 0.1$, $\text{SNR} = -18$)

offset) for a SNR level of -18 dB. However, in Figure 6-10 the proportionality constant is set to 1.5; in Figure 6-11 the proportionality constant K is set to 0.5; and in Figure 6-12 the proportionality constant is set to 0.1.

Examining Figure 6-10 ($K = 1.5$), we can observe a large initialization transient. In several other simulations under similar conditions, this initialization transient has led to a loss of track or instability.

Examining Figures 6-11 and 6-12 for the lower value of the proportionality constant K , we can see that the initialization transient is removed but the random variations are greater. Similar simulations at -20 dB have shown poor transient behavior for the $K = 0.5$ proportionality constant with only $K = 0.1$ proportionality constant having satisfactory characteristics.

The basic problem is that increasing the proportionality constant increases the weight of the LMSF a priori information function. Our a priori information is that the target time delay difference trajectory is a quadratic function in time over the block size. The a priori information depends largely on the actual ML estimates. In the case of initialization, the initial estimate and the following transient ML estimates are unreliable for use by the LMSF a priori information function. Therefore, relying too much initially on the LMSF a priori information function compounds the error which is difficult to correct for as time goes on.

We do, however, desire to have the optimum smoothing qualities of the larger proportionality constant K once in steady state. To obtain the best of both initialization and steady state properties, the following initialization algorithm was adopted.

1. From time = 0 seconds until time = 110 seconds (12 updates after the block is filled), the proportionality constant is set to 0.1.
2. From time = 115 seconds until time = 250 seconds (28 updates), the proportionality constant is incrementally update by 0.05.
3. From time 255 seconds onwards, the proportionality constant K is fixed at 1.5.

The slow transition in step 2 from $K = 0.1$ to $K = 1.5$ is required to minimize introducing transients into the algorithm.

Figure 6-13, performed on the identical scenario as Figures 6-10, 6-11, and 6-12, illustrates the performance of our new initialization algorithm.

Examining Figure 6-13, we can see at the beginning the same output as Figure 6-12 for the $K = 0.1$ proportionality constant. At the end, Figure 6-13 resembles Figure 6-10 for the $K = 1.5$ proportionality constant. In between, there are no apparent initialization transients.

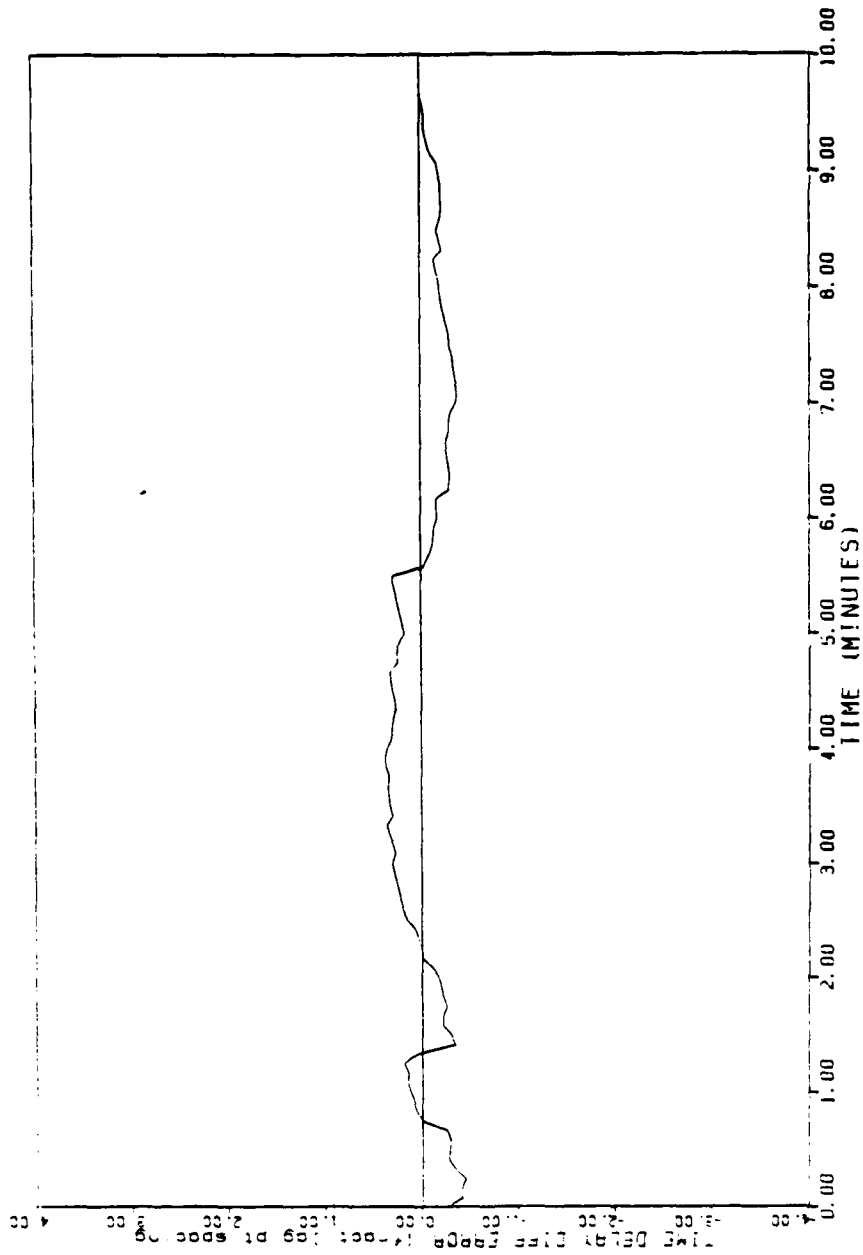


Figure 6-13. Time Delay Difference Error vs Time, Gradient Search (K = 0.1 through 1.5, SNR = -18)

We do not claim that our initialization algorithm is optimum. It is certainly a subject for future work. However, it is the algorithm for which we present the equivalent initialization test results for the gradient search algorithm as we did earlier for the dynamic programming (SSR) algorithm.

Tables 6-3 presents the nominal target dynamic initialization test results for the gradient search algorithm and Table 6-4 presents the double nominal target dynamics initialization test results for the gradient search algorithm.

Examining Tables 6-3 and 6-4 we can make the following observations about the initialization properties of the gradient search algorithm.

1. At -15 dB for either target dynamic scenario, the gradient search algorithm is perfect in all initialization attempts.
2. At -18 dB we notice a slight degradation in successful initializations. However, at double target dynamics this degradation becomes significant at an initial offset of +2 lag points.
3. At -20 dB for the nominal target dynamics, the initialization performance is excellent for negative and zero initial lag point offsets but poor at positive initial lag point offsets.

Table 6-3. Gradient Search Nominal Target Dynamics

SNR (dB)	Initial Lag Point Offset				
	-2	-1	0	+1	+2
-15	10/10	10/10	10/10	10/10	10/10
-18	9/10	9/10	10/10	9/10	8/10
-20	8/10	8/10	9/10	5/10	3/10

Ratio of successful initializations to attempts

Table 6-4. Gradient Search Double Nominal Target Dynamics

SNR (dB)	Initial Lag Point Offset				
	-2	-1	0	+1	+2
-15	10/10	10/10	10/10	10/10	10/10
-18	8/10	9/10	10/10	9/10	4/10
-20	2/10	4/10	3/10	0/10	1/10

Ratio of successful initializations to attempts

4. At -20 dB for the double normal target dynamics, the initialization falls way off. Once again, the negative and zero initial lag point offsets have better initialization performance than the positive lag point offsets.
5. It is apparent that the gradient search algorithm initializes significantly better for a lagging or negative initial lag point offsets than for leading or positive initial lag point offsets. This result is the opposite of the dynamic programming (SSR) algorithm. The reason for the above is almost certainly the result of the gradient search algorithm initially estimating the target time delay difference trajectory slope in the wrong direction. (A lead offset will result in a negative initial time delay difference slope when the actual time delay difference trajectory slope is positive.) Although steps have been taken to minimize this problem, they obviously have not been totally successful. Therefore, there is obvious improvements which can be incorporated in the gradient search algorithm initialization procedure.

Overall, the gradient search initialization characteristics are slightly worse than the dynamic programming (SSR) algorithm initialization characteristics. However, the gradient search algorithm is capable of operating at SNR level at least 3 dB lower than the dynamic programming (SSR) algorithm. Therefore, we can conclude that the gradient search initialization procedure is not as good as the dynamic programming initialization procedure and that there is a definite gap between

initialization thresholds and steady state thresholds, especially in the presence of target dynamics for the gradient search algorithm.

CHAPTER 7

SUMMARY CONCLUSIONS, AND RECOMMENDATIONS

We have studied in detail the problem of estimating time delay difference parameters of a moving broad spectrum target from two observation sensors. The objective of this study was to derive recursive batch estimation algorithms with good initialization, threshold and dynamic properties. The practical problem which motivated our study is primarily in, but not limited to, the field of underwater acoustics. The acoustical problem is to estimate the time delay difference and/or bearing trajectory of a moving broad spectrum acoustical target.

Our theoretical approach to the above problem was to derive a MAP estimator similar to the estimator derived by Wolcin⁶ for narrowband signal parameters estimation. The MAP approach was chosen to take advantage of known a priori information on expected acoustical target trajectories. We assumed that over a significant period of time, time delay difference trajectories will be smooth, continuous and can be modeled by a low order (quadratic) polynomial in time.

The derivation of the maximum likelihood (ML) portion of the MAP estimator was along similar lines adopted by Knapp and Carter² and Ng.⁵ Our principal contribution in this area was the precised derivation of the expected cross correlation (MISMATCH) function in the presence of a linear time delay difference trajectory. This derivation contained an additional factor predicting a coherency loss with increase observation

time and time delay difference rate. In addition, we have expanded the derivation of the mismatch function for the more general case of a linear target power spectrum. This work includes an estimator for a parameter related to the slope of the linear target power spectrum.

Our derived ML estimator assumes as input the discrete noisy measurements of the cross correlation between our two observation sensors averaged for T seconds and evaluated at N equally spaced lag delays centered at zero lag delay. The noise components between measured lag points are assumed independent, although the correlated noise components case is examined.

A block of input data consists of a fixed number of M consecutive measured lag point data updates. A recursive batch process is a batch process where at every update the oldest measured data is deleted from the data block and the newest measured data is added to the data block. After every update, the recursive batch process calculates new estimates based solely on the new block of data.

Similar to Ng,⁵ we adopted two different time delay difference trajectory models to the same problem to facilitate our derivation of a batch estimator. To maximize response time, minimize coherency loss and provide a tractable model for our ML estimator, we have assumed that the unknown time delay difference trajectory is a piecewise linear function over the block of measurements of interest. The piecewise linear segments correspond to individual measurement update or observation times. In order to gain the advantage of long observation time, we

have assumed that our unknown time delay difference trajectory can be modeled by a quadratic function in time over the block of time of interest. To tie these two models together, we have derived an a priori information penalty function based on minimizing the squared error of the individual ML time delay difference estimate over our block of data measurements with respect to a quadratic time delay difference model. The concept of this LMSF a priori information penalty function operating simultaneously with our individual ML estimates is unique from either the work of Ng⁵ or Wolcin.⁶

From our theoretical work we have derived two major algorithmic implementations. The first is based on a dynamic programming technique adopted by Wolcin.⁷ We claim credit for obtaining an efficient recursive implementation of this algorithm. In our recursive implementation of the LMSF a priori information function, we have used a technique developed by Ng.¹⁰ However we have uncovered potential numerical instability in Ng's algorithm and have derived a numerical technique to eliminate this instability.

The second algorithmic implementation is based on numerical gradient search techniques. Although the numerical techniques are straightforward, neither Wolcin, Ng, nor any known researcher has applied these recursive search techniques to this type of problem.

Other theoretically related algorithms were also implemented for comparison studies. The most notable of these is a algorithm which prewhitens the correlated input correlogram lag point data to effectively obtain independent noise on all the input correlogram lag points.

Using empirical result based on simulations and where possible theoretical predictions, we have sought to find the optimum algorithmic parameter selections for our two major time delay estimation algorithms. The algorithmic parameters we have researched are listed below.

1. The proportionality constant K between the ML estimator and the a priori information penalty function.
2. The number M of data measurements in our data block and the integration time T per measurement.
3. The number of lag points N calculated for each measurement update.
4. The lag spacing τ between measured lag points.

The following major conclusions were drawn from the above simulation experiments. The simulations were for the most part performed using a moderate dynamic time delay difference trajectory. The stimulator used for the simulation experiments was independently developed.

The stimulator produced simulated correlogram lag point data with realistic correlated noise added to the deterministic lag points.

1. There are optimal values for the proportionality constant K for both the gradient search and dynamic programming algorithms. The optimum choice of K is a function of SNR, block size M , and target dynamics. A corollary of this observation is that for reasonable target motion the proper introduction of a priori information ($K > 0$) will always improve the variance of our estimators.
2. For the dynamic programming algorithm, the ML estimator ($K = 0$) has little, if any, dependence on block size after a minimum block size of 4. The minimum dependence of block sizes of 4 or less is due to the weak piecewise linear assumption of the ML estimator. For the moderate test dynamics, the ML estimator is always improved by moderate increased averaging time.

The above observation should also be true for the gradient search algorithm as well. However, numerical and algorithmic problems, especially at high SNR, indicate an optimum choice of block size.

3. With the introduction of optimum a priori information, there is an optimum choice of block size dependent on integration time and target dynamics. Barring numerical problems due to the

increased number of unknowns, the following is the explanation for the dependence of block size on averaging time and target dynamics. The more numbers of measurements and the more severe the target dynamics the greater the probability of violating algorithmic assumptions (quadratic time delay difference trajectory model). The longer the integration time per measurement the more likely a modeling mismatch will occur. (For large block sizes, shorter integration times actually can improve algorithm performance.)

4. For relatively low dynamic targets the number of lag points N does not affect algorithmic performance. This conclusion is based upon the realistic assumption of correlated noise between lag point measurements. If the noise was independent this conclusion would be valid after 9 lag points.
5. The prewhitening option does not significantly improve our ML estimator in the case of correlated noise.
6. For high target dynamics a small number of lag points (3) may degrade performance. Since increasing the number of lag points increases computations for our algorithm, there exists an optimum choice for the number of lag points. This optimum choice lies in the range of 3 to 9 lag point measurements.
7. Given target dynamics, there is an optimum choice of averaging time T for the measured lag point data supplied to our ML

estimator. The more severe the target time delay rate, the shorter the optimum averaging time T . Theoretically, this optimal is a balance between averaging time and coherency loss due to target motion.

8. Theoretically, the choice of lag point spacing will not affect our ML estimators performance unless it is larger than the Nyquist rate for the highest processed frequency.

In practice, too small choices of lag point spacing severely degrade algorithmic performance. Therefore, the optimum choice of lag point spacing is in the order of $1/2$ the Nyquist rate to the actual Nyquist rate for the highest processed frequency.

Our final effort was to illustrate the performance of the chosen gradient search algorithm with respect to a basic standard, other related algorithm alternatives, and varying spectral and dynamic test conditions. The basic reference was selected as the Cramer-Rao Lower Bound (CRLB) for coherent integration of a stationary target (no a priori information). We shall name this basic reference the Maximum Likelihood Estimate (MLE) CRLB. The performance tests consisted of Monte Carlo simulations conducted on the gradient search and other related algorithms. Once again, our stimulator assumes correlated noise between lag point measurements.

The following major conclusions were drawn from the above Monte Carlo simulations.

1. The gradient search algorithm has the best steady state RMS time delay difference error performance characteristics of all the algorithms tested. However, the gradient search algorithm currently requires 4 times the processing time of the dynamic programming algorithm and also has slightly inferior initialization characteristics compared to the dynamic programming algorithm.

The RMS time delay difference errors for the gradient search algorithm are always less than the corresponding MLE CRLB for an integration time of 5 seconds and less than, at low SNR, the corresponding CRLB for an integration time of 50 seconds. Five seconds of integration is equal to the individual lag point measurement update to the gradient search algorithm. Fifty seconds of integration is equal to the entire observation time for the sliding block of data measurements used by the gradient search algorithm. The apparent performance improvement of the gradient search algorithm over the MLE CRLB is a result of good a priori information.

2. Prewhitening the input lag point measurements in the presence of correlated noise does not improve algorithm performance. Since prewhitening is computationally expensive, it is not recommended.

3. Performing an ML estimation process first and then smoothing the resulting time delay difference estimates with an LMSF is not equivalent to the simultaneous ML and LMSF a priori information penalty function estimation. The former process becomes unstable at low SNR.
4. The simultaneous estimation of time delay difference parameters and a spectral parameter becomes unstable at relatively high SNR (< -10 dB). The instability results from the spectral estimation process.
5. High target time delay difference dynamics will degrade the time delay difference estimates.

High target time delay difference dynamics will also degrade the time delay difference rate estimates. However, the rate estimate have acceptable RMS error for all SNR levels and target dynamics tested.

The time delay difference acceleration estimate proved not to be a useful estimator.

6. Moderate target power spectrum modeling mismatches proved to have a minimal effect on our gradient search estimator. The actual modeling mismatch only appeared to have an affect at high SNR (> -5 dB). At low SNR the gain or loss in time delay

estimation performance could be attributed to the actual target power spectral shape (i.e., not to our incorrect modeling of the target power spectral shape).

The separate estimation of a spectral parameter was only stable for SNR levels of -10 dB or greater. At these SNR levels the spectral parameter estimates were reasonable.

7. The initialization properties of the dynamic programming algorithm are excellent for moderate target dynamics. The dynamic programming algorithm had good initialization properties to within a few dB of its steady state performance limit.
8. The initialization levels of the gradient search algorithm are almost as good as the dynamic programming algorithm. However, its steady state threshold is 3 dB lower than the dynamic programming algorithm. Therefore, there is a significant gap between initialization threshold and steady state threshold. This difference is increased in the presence of increased target dynamics.

7.1 FUTURE WORK

Our work has been a complete research on the development of efficient recursive batch time delay difference estimation algorithm. Two theoretical areas requiring addition work are: 1) A theoretical

development on the functional dependencies of the proportionality constant K ; and 2) The related theoretical development of a lower performance bound for the ML estimator and a priori information.

The following areas are recommended for future work in the algorithmic area.

1. Development of an improved initialization scheme and related theory.
2. Development of adaptive parameter selection. Optimum parameter selection is related to the target dynamic state and SNR. Therefore, an adaptive parameter algorithm could be developed to conform to changing target dynamic conditions and SNR levels.
3. Extension of algorithms to include the multi-parameter case such as multiple frequency bands, multiple beams, and multiple target's.

REFERENCES

1. C. H. Knapp, G. C. Carter, 'The Generalized Correlation Method for Estimation of Time Delay', IEEE ASSP Vol. 24, August 1976.
2. C. H. Knapp, G. C. Carter, 'Estimation of Time Delay in the Presence of Source or Receiver Motion', JASA Vol. 61 No. 6, June 1977.
3. Y. T. Chan, J. M. Riley, J. B. Plant, 'Modeling of Time Delay and its Application to Estimation of Non Stationary Delays, IEEE ASSP Vol. 29, June 1981.
4. B. Friedlander, 'Multitarget Tracking Study, Phase I Final Report', System Control, Inc. Report No. 5334-01.
5. L. C. Ng, 'Optimum Multisensor, Multitarget Localization and Tracking'. PhD. Dissertation, University of Connecticut, 1983.
6. J. J. Wolcin, 'Maximum A-Posteriori Estimation of Narrowband Signal Parameters', NUSC TM No. 791115, 21 June 1979
(UNCLASSIFIED).
7. J. J. Wolcin, 'Maximum A-Posteriori Line Extraction, A Computer Program Discription', NUSC TM No. 801042, 20 March 1980
(UNCLASSIFIED)

REFERENCES (Continued)

8. J. J. Wolcin, 'Maximum A-Posteriori Estimation of Narrowband Signal Parameters', JASA Vol. 68, July 1980.
9. R. A. LaTourette, S. G. Greineder, J. J. Wolcin, 'Development of the Batch-Oriented MAPLE Frequency Line Detection/Tracking Algorithm into a Fast Recursive MAPLE Frequency Line Tracking Algorithm', NUSC TM No. 851048, 12 July 1985 (UNCLASSIFIED)
10. L. C. Ng, P. R. Lambert, 'Fast Moving Average Recursive Least Mean Square Fit', NUSC TM No. 841143, 30 September 1984 (UNCLASSIFIED)
11. B. D. Steinberg, 'Principles of Aperture and Array System Design', John Wiley and Sons, 1976.
12. H. L. Van Trees, 'Detection, Estimation and Modulation Theory PART 1', John Wiley and Sons, 1968.
13. L. C. Ng, Informed Correspondence, 1985.
14. A. D. Whalen, 'Detection of Signals in Noise', Academic Press, Inc., 1971.
15. G. Dahlquist, A. Bjorck, 'Numerical Methods', Prentice-Hall, 1974.

REFERENCES (Continued)

16. R. E. Bellman, 'Dynamic Programming', Princeton University, Princeton, N.J., 1957.
17. B. Gold, C. M. Rader, 'Digital Processing of Signals', McGraw-Hill, 1969.
18. L. C. Ng, R. A. LaTourette, 'Fast Moving Average Recursive Least Mean Square FIT', IEEE 24th CDC, Dec. 11-13, 1985.
19. R. A. LaTourette, L. C. Ng, 'Equivalent Bandwidth of a General Class of Polynomial Smoothers: With Application to Bearing Tracker Random Error Evaluation', NUSC TECH Report 6601, 19 July 1982 (UNCLASSIFIED)
20. L. C. Ng, R. A. LaTourette, 'Equivalent Bandwidth of a General Class of Polynomial Smoothers', JASA 74(3), September 1983.
21. M. Hestenes, 'Conjugate Direction Methods in Optimization', Springer-Verlag, 1980.
22. D. Goldfarb, 'A Family of Variable Metric Methods Derived by Variational Means', Mathematics of Computation, 1970, Vol. 24, 23-26.

REFERENCES (Continued)

23. C. Wenk informal conversation 8/85.
24. G. E. Forsythe, M. A. Malcolm, C. B. Moler, 'Computer Methods for Mathematical Computations', Prentice-Hall, 1977.
25. D. B. Cretella and H. F. Jarvis, Jr., 'Wide Aperture Array ADM Subsystem Computer Simulation Program Description and Users Guide', Analysis & Technology, Inc. Report No. P-1143-03-82.
26. G. C. Carter, 'IEEE Special Issue on 'Time Delay Estimation'', IEEE Transactions on ASSP, Volume 29(3), June 1981.

APPENDIX A
ALTERNATIVE DERIVATION OF THE MISMATCH FUNCTION

The following is a slight modification of derivation suggested by Ng¹³ for the evaluation of Equation 3-54. Equation 3-54 is repeated below for convenience.

$$\frac{1}{4\pi} \int_{-B}^B G_s(\omega) \lim_{T \rightarrow \infty} \left(\text{Sinc}[(\beta_1 \omega - \omega_k) \frac{T}{2}] \cdot \text{Sinc}[(\beta_2 \omega - \omega_k) \frac{T}{2}] \cdot e^{j\omega(\Delta d_0 + \Delta B(T/2))} \right) d\omega \quad (A-1)$$

Let $\beta = \text{MAX}(\beta_1, \beta_2)$ and assume that $G_s(\omega) e^{j\omega(\Delta d_0 + \Delta B(T/2))}$ changes very little over the interval $[(\omega_k - 2\pi/T)/\beta, (\omega_k + 2\pi/T)/\beta]$, then Equation A-1 can be rewritten as the following:

$$G_s(\omega_k) e^{j\omega_k(\Delta d_0 + \Delta B(T/2))} \int_0^B \lim_{T \rightarrow \infty} \left(\text{Sinc}[(\beta_1 \omega - \omega_k) \frac{T}{2}] \cdot \text{Sinc}[(\beta_2 \omega - \omega_k) \frac{T}{2}] \right) \frac{d\omega}{2\pi} \quad (A-2)$$

As τ becomes large (i.e., T becomes large), the width of the $\text{Sinc}(\cdot)$ functions are much smaller than B , the signal bandwidth. Therefore, we can approximate the following expression from Equation A-2.

$$G_s(\omega) e^{j\omega(\Delta d_0 + \Delta\beta(T/2))} \int_{-\infty}^{\infty} \text{Sinc}[(\beta_1\omega - \omega_k)\frac{T}{2}] \text{Sinc}[(\beta_2\omega - \omega_k)\frac{T}{2}] \frac{d\omega}{2\pi} . \quad (\text{A-3})$$

From this point onwards we shall be concerned with evaluating the integral portion of expression A-3.

The key idea we shall exploit is Parseval's theorem stated below.

$$\int_{-\infty}^{\infty} f_1(t) f_2(t) dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(-\omega) F_2(\omega) d\omega \quad (\text{A-4})$$

where

$$f_1(t) \leftrightarrow F_1(\omega)$$

$$f_2(t) \leftrightarrow F_2(\omega)$$

are Fourier transform pairs.

Note that if we define $f(t)$ as the following:

$$f(t) = \frac{1}{T}, \quad |t| \leq \frac{T}{2}, \quad (\text{A-5})$$

$$0, \quad |t| > \frac{T}{2}$$

then the corresponding Fourier transform $F(\omega)$ is defined as the following:

$$F(\omega) = \text{Sinc}(\omega T/2) . \quad (\text{A-6})$$

Furthermore, we have the following relationships for Fourier transforms.

$$f(at) \leftrightarrow \frac{1}{|a|} F\left(\frac{\omega}{a}\right) = \frac{1}{|a|} \text{Sinc}\left(\frac{\omega T}{2a}\right) \quad (\text{A-7})$$

$$f(at)e^{j\omega_0 t} \leftrightarrow \frac{1}{|a|} F\left(\frac{\omega - \omega_0}{a}\right) = \frac{1}{|a|} \text{Sinc} \frac{(\omega - \omega_0)T/2}{a} \quad (\text{A-8})$$

In order to evaluate expression A-4, let

$$F_1(-\omega) = \text{Sinc}\left[\beta_1\left(\omega - \frac{\omega_k}{\beta_1}\right)\frac{T}{2}\right] \quad (\text{A-9})$$

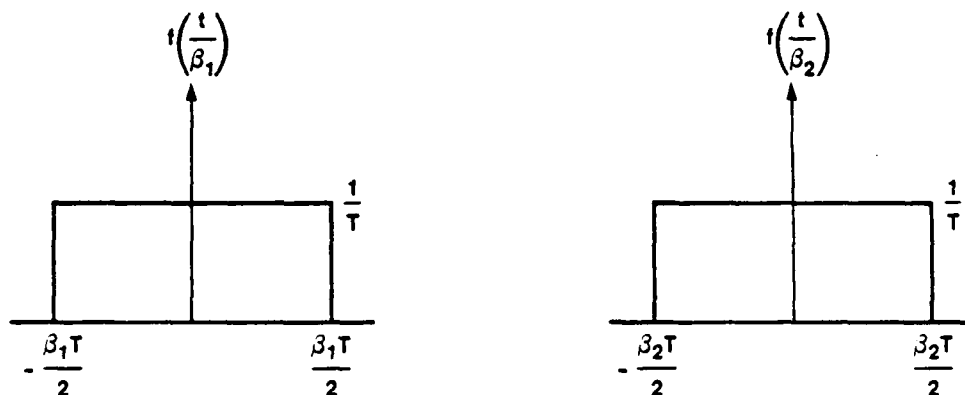
$$F_2(\omega) = \text{Sinc}\left[\beta_2\left(\omega - \frac{\omega_k}{\beta_2}\right)\frac{T}{2}\right] \quad (\text{A-10})$$

With Equations A-7 and A-8 in mind, the inverse Fourier transform of Equations A-9 and A-10 is given below.

$$F^{-1}\{F_1(\omega)\} = \frac{f\left(\frac{t}{\beta_1}\right) e^{-j\omega_k t/\beta_1}}{\beta_1} \quad (\text{A-11})$$

$$F^{-1}\{F_2(\omega)\} = \frac{f\left(\frac{t}{\beta_2}\right) e^{-j\omega_k t/\beta_2}}{\beta_2} \quad (\text{A-12})$$

where $f(\cdot)$ is defined by Equation A-5 and diagrammed below



580.097

Therefore, using Equations A-11, A-12, and A-4 we can write the following.

$$\int_{-\infty}^{\infty} \text{Sinc}\left[(\beta_1 \omega - \omega_k) \frac{T}{2}\right] \text{Sinc}\left[(\beta_2 \omega - \omega_k) \frac{T}{2}\right] \frac{d\omega}{2\pi} =$$

$$\int_{-\tilde{\beta}T/2}^{\tilde{\beta}T/2} \frac{1}{\beta_1 \beta_2 T^2} e^{j\omega_k (1/\beta_2 - 1/\beta_1)t} dt \quad (\text{A-13})$$

where

$$\tilde{\beta} = \min(\beta_1, \beta_2).$$

The left-hand side of Equation A-13 evaluates to the following after integration and algebra.

$$\frac{1}{\beta T} \text{Sinc}[\omega_k \Delta B T / 2\beta] \quad (\text{A-14})$$

where

$$\beta = \text{MAX}[\beta_1, \beta_2]$$

$$\Delta B = \beta_2 - \beta_1$$

Combining the result for expression A-14 with expression A-2 and noting that $G_s(\omega)$ is equal to zero for $|\omega| > B$, we have the following expression for the integral of expression A-1.

$$G_s(\omega_k/\beta) \frac{e^{j\omega_k/\beta(\Delta d_0 + \Delta B T/2)}}{\beta T} \text{Sinc}[\omega_k \Delta B T / 2\beta] \quad , \quad -\beta B \leq \omega_k \leq \beta B \quad (\text{A-15})$$

This is the result identical to Equation 3-64.

APPENDIX B EVALUATION OF MISMATCH FUNCTION INTEGRALS

This appendix shows the analytical approximations used to evaluate integrals related to the mismatch function.

From Chapter 3 we have the integral of Equation 3-68 to evaluate which is repeated below for convenience.

$$\frac{1}{Bw} \int_{\omega_c - Bw/2}^{\omega_c + Bw/2} \text{Sinc}\left(\omega \frac{\Delta B T}{2}\right) \cos\left(\omega \left[\tilde{t} + \left(\Delta d_0 + \frac{\Delta B T}{2}\right)\right]\right) d\omega \quad . \quad (B-1)$$

The above integral can be symbolically written as the following.

$$\frac{1}{Bw} \int_{\omega_c - Bw/2}^{\omega_c + Bw/2} \text{Sinc}[A \omega] \cos[B \omega] d\omega \quad (B-2)$$

where

$$A = \frac{\Delta B T}{2} \quad (B-3)$$

and

$$B = \left(\tilde{t}_i + \Delta d_0\right) + \frac{\Delta B T}{2} \quad . \quad (B-4)$$

we shall assume that

$$A \omega \ll 1, \quad \omega_c - Bw/2 \leq \omega \leq \omega_c + Bw/2 \quad (B-5)$$

AD-A173 396

EFFICIENT RECURSIVE BATCH TIME DELAY DIFFERENCE
ESTIMATION IN THE PRESENC. (U) NAVAL UNDERWATER SYSTEMS
CENTER NEW LONDON CT NEW LONDON LAB. R A LATOURETTE

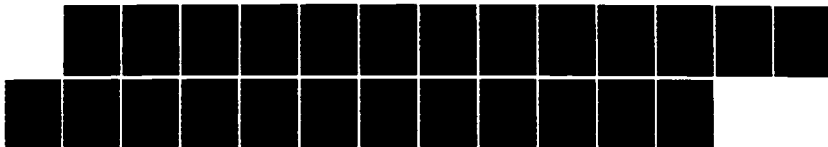
4/4

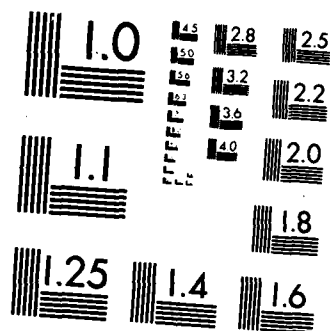
UNCLASSIFIED

17 JUL 86 NUSC-TR-7743

F/G 17/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

and therefore we can write the following expansion

$$\text{Sinc}[A \omega] \approx 1 - \frac{A^2 \omega^2}{6} . \quad (\text{B-6})$$

Substituting Equation B-6 into Equation B-2 yields the sum of two integrals. The first is given below.

$$\frac{1}{B\omega} \int_{\omega_c - B\omega/2}^{\omega_c + B\omega/2} \cos[B \omega] d\omega = \cos[B \omega_c] \text{Sinc}[B \frac{B\omega}{2}] . \quad (\text{B-7})$$

Equation B-7 is the standard correlation function for a stationary flat power spectrum.

Therefore, the second integral yields the correction factor due to target motion. This integral is shown below.

$$- \frac{A^2}{6B\omega} \int_{\omega_c - B\omega/2}^{\omega_c + B\omega/2} \omega^2 \cos[B \omega] d\omega . \quad (\text{B-8})$$

The integral of expression B-8 can be integrated by parts or look-up in a table of integrals to yield the following expression.

$$\left[\frac{2 - B^2 \omega^2}{B^3} \sin(B \omega) - \frac{2\omega}{B^2} \cos(B \omega) \right] \bigg|_{\omega_c - B\omega/2}^{\omega_c + B\omega/2} . \quad (\text{B-9})$$

Applying the limits to expression B-9 and using algebra and trigonometry yields the following result.

$$\begin{aligned}
 & - \frac{A^2}{3B\omega_c B^3} [B B\omega_c \cos(B \omega_c) \cos(B B\omega/2) \\
 & \quad - 2B \omega_c \sin(B \omega_c) \sin(B B\omega/2) \\
 & \quad + (B^2(\omega_c^2 + (B\omega/2)^2) - 2) \cos(B \omega_c) \sin(B B\omega/2) \\
 & \quad + \omega_c B\omega B^2 \sin(B \omega_c) \cos(B B\omega/2)] \quad . \quad (B-10)
 \end{aligned}$$

Note that if the target radial motion is zero, then $\Delta B = 0$ which implies that $A = 0$ which implies that expression B-10 is zero.

To summarize our result, the integral of expression B-1 evaluates to the sum of expression B-7 and B-10.

Also from Chapter 3 we have to evaluate the integrals from Equation 3-94. The first and third integrals from Equation 3-94 are identical to the integral of Equation B-1 which we have already evaluated. The second integral of Equation 3-94 is the result of a linear sloped power spectrum. This integral is written below for convenience.

$$\frac{2}{B\omega \Delta\beta T} \int_{\omega_c - B\omega/2}^{\omega_c + B\omega/2} \cos(\omega[\tilde{t}_i + \Delta d_0 + \frac{\Delta\beta T}{2}]) \sin(\omega \frac{\Delta\beta T}{2}) d\omega \quad (B-11)$$

The above integral can be symbolically written as the following

$$\frac{2}{B\omega \Delta\beta T} \int_{\omega_c - B\omega/2}^{\omega_c + B\omega/2} \cos[A \omega] \sin[B \omega] d\omega \quad (B-12)$$

where A and B are defined by Equations B-3 and B-4, respectively.

The integral shown in expression B-12 can be evaluated exactly.

From a table of integrals we obtain the following expression

$$\frac{2}{B\omega \Delta\beta T} \left[-\frac{\cos[B - A]\omega}{2[B - A]} - \frac{\cos[B + A]\omega}{2[B + A]} \right] \Bigg|_{\omega_c - B\omega/2}^{\omega_c + B\omega/2} \quad (B-13)$$

Applying the limits and using algebra and trigonometry, expression B-13 evaluates to the following.

$$\frac{1}{\Delta\beta T} [\sin(\omega_c[A + B]) \text{Sinc}(\frac{B\omega}{2}[A + B]) - \sin(\omega_c[B - A]) \text{Sinc}(\frac{B\omega}{2}[B - A])] \quad (B-14)$$

Expression B-14 is the analytical result of the integration of expression B-12.

To completely specify expression B-14, we should give the limits when $\Delta\beta T \rightarrow 0$, $\tilde{t}_i + \Delta d_0 \rightarrow 0$ (or equivalently $B - A \rightarrow 0$) and $\Delta\beta T + \tilde{t}_i + \Delta d_0 \rightarrow 0$ (or equivalently $B + A \rightarrow 0$).

$$\lim_{\substack{\Delta \beta T \rightarrow 0 \\ \Delta d_0 + \tilde{t}_i \neq 0}} [\text{Expression B-14}] = \omega_c [\cos(\omega_c \theta) \text{Sinc}(\frac{Bw}{2} \theta) + [\cos(\frac{Bw}{2} \theta) - \text{Sinc}(\frac{Bw}{2} \theta)] \text{Sinc}(\omega_c \theta)] \quad (\text{B-15})$$

$$\text{where } \theta = \Delta d_0 + \tilde{t}_i = B - A$$

$$\lim_{\substack{\Delta \beta T \rightarrow 0 \\ \Delta d_0 + \tilde{t}_i \rightarrow 0}} [\text{Expression B-14}] = \omega_c \quad (\text{B-16})$$

$$\lim_{\substack{\Delta \beta T \neq 0 \\ \Delta d_0 + \tilde{t}_i \neq 0 \\ \Delta \beta T + \Delta d_0 + \tilde{t}_i \rightarrow 0}} [\text{Expression B-14}] = -\sin[\omega_c \theta] \text{sinc}[\frac{Bw}{2} \theta] / \Delta \beta T \quad (\text{B-17})$$

APPENDIX C
RECURSIVE LMSF ERROR ACCUMULATION EXPERIMENT

This appendix describes the simulation experiment to verify the accumulation of errors in the recursive LMSF procedure as presented by Ng¹⁰ and outlined in Section 4.2. In addition, the simulation experiment will demonstrate that the modifications recommended by LaTourette¹⁸ effectively eliminate the accumulation of error by the recursive procedure.

The simulation experiment was designed as follows:

1. The input signal time series is described by the following equation.

$$Y_i = A \sin\left(\frac{2\pi}{T} t_i\right) + r_i(0, \text{var}), \quad t_i = 0, 1, 2, 3 \dots, N \quad (\text{C-1})$$

where

Y_i = the input time series

A = a constant amplitude

T = the period

r_i 's = independent Gaussian random variables with
mean = 0 and variance = var

N = the number of points in the time series.

The theoretical matching function is a quadratic equation listed below.

$$c_2 t_i^2 + c_1 t_i + c_0 \quad (C-2)$$

where c_2 , c_1 , c_0 are unknown coefficients.

2. The time series Y_i (Equation C-1) is recursively applied to Equations 4-13, 4-14, 4-15, and 4-17 which are rewritten below in a convenient format.

$$l_i(1) = l_{i-1}(1) + Y_i - Y_{i-M-1} \quad (C-3)$$

$$l_i(2) = l_{i-1}(2) + (M+1) Y_i - l_i(1) \quad (C-4)$$

$$l_i(3) = l_{i-1}(3) + (M+1)^2 Y_i + 2l_i(2) - l_i(1) \quad (C-5)$$

$$SY_i^2 = SY_{i-1}^2 + Y_i^2 - Y_{i-M-1}^2 \quad (C-6)$$

where

$l_i(k)$ = the recursive coefficient parameters $k = 1, 2, 3$

$M+1$ = the fixed number of data point in the block

SY_i^2 = the recursive sum of squared input Y_i within the sliding window.

The sum of the squared residual SSR for the i th time frame is obtained via Equation 4-16 and rewritten below in a convenient format.

$$SSR_i = SY_i^2 - \underline{L}_i^T (H^T H)^{-1} \underline{L}_i \quad (C-7)$$

where the vector $\underline{L}_i$ is defined below

$$\underline{L}_i^T = [l_i(1), l_i(2), l_i(3)] \quad (C-8)$$

and the matrix H is defined in Equation 4-9.

Therefore, the recursive parameters of interest are $l_i(1)$, $l_i(2)$ and $l_i(3)$ defined by Equations C-3, C-4, C-5 and the recursive estimate of the sum of squared errors SSR_i defined by Equation C-8.

3. To determine the accumulated error in our recursive parameters of interest, we shall calculate their values using the following standard non-recursive equations.

From Equation 4-12, $l_i(1)$, $l_i(2)$ and $l_i(3)$ can be calculated from the following equation.

$$\underline{L} = H^T \underline{\Delta D} \quad (C-9)$$

where the vector $\underline{\Delta D}$ is defined as

$$\underline{\Delta D}^T = [Y_i, Y_{i-1}, \dots, Y_{i-M}] \quad (C-10)$$

Therefore,

$$\bar{T}_i(1) = \sum_{k=0}^M Y_{i-M+k} \quad (C-11)$$

$$\bar{T}_i(2) = \sum_{k=0}^M (k - 1) Y_{i-M+k} \quad (C-12)$$

$$\bar{T}_i(3) = \sum_{k=0}^M (k - 1)^2 Y_{i-M+k} \quad (C-13)$$

Also from the definition

$$\overline{SY_i^2} = \sum_{k=0}^M Y_{i-M+k}^2 \quad (C-14)$$

We can now use Equation C-7 to calculate the non-recursive (truth) sum of squared residual $\overline{SSR}_i$ using Equations C-11 through C-14.

4. In all our simulations we shall calculate the following statistics as a function of the t_i (time or iteration index).

$$\text{ABS Error } l_i(1) = |l_i(1) - \bar{T}_i(1)| \quad (\text{C-15})$$

$$\text{ABS Error } l_i(2) = |l_i(2) - \bar{T}_i(2)| \quad (\text{C-16})$$

$$\text{ABS Error } l_i(3) = |l_i(3) - \bar{T}_i(3)| \quad (\text{C-17})$$

$$\text{Percentage error SSR} = \frac{\text{ABS} [SSR_i - \overline{SSR}_i] 100}{\overline{SSR}_i} \quad (\text{C-18})$$

Figures C-1 through C-8 give the above statistics for the recursive modification using single precision arithmetic on a VAX 11/780 computer. The amplitude A is set at 1. The period T is set at 100. The other parameters are varied as stated.

Note for all simulations and for all parameter settings, the modification to eliminate the accumulation of errors in the recursive LMSF procedure described in Section 4.2 yields identically zero errors for all statistics.

Examining Figures C-1 through C-8, we notice a moderate growth in absolute error of the linear parameter $l(2)$ and a substantially larger growth in absolute error in the quadratic parameter $l(3)$. Overall, there is an erratic but steady growth in error of the sum of squared residuals.

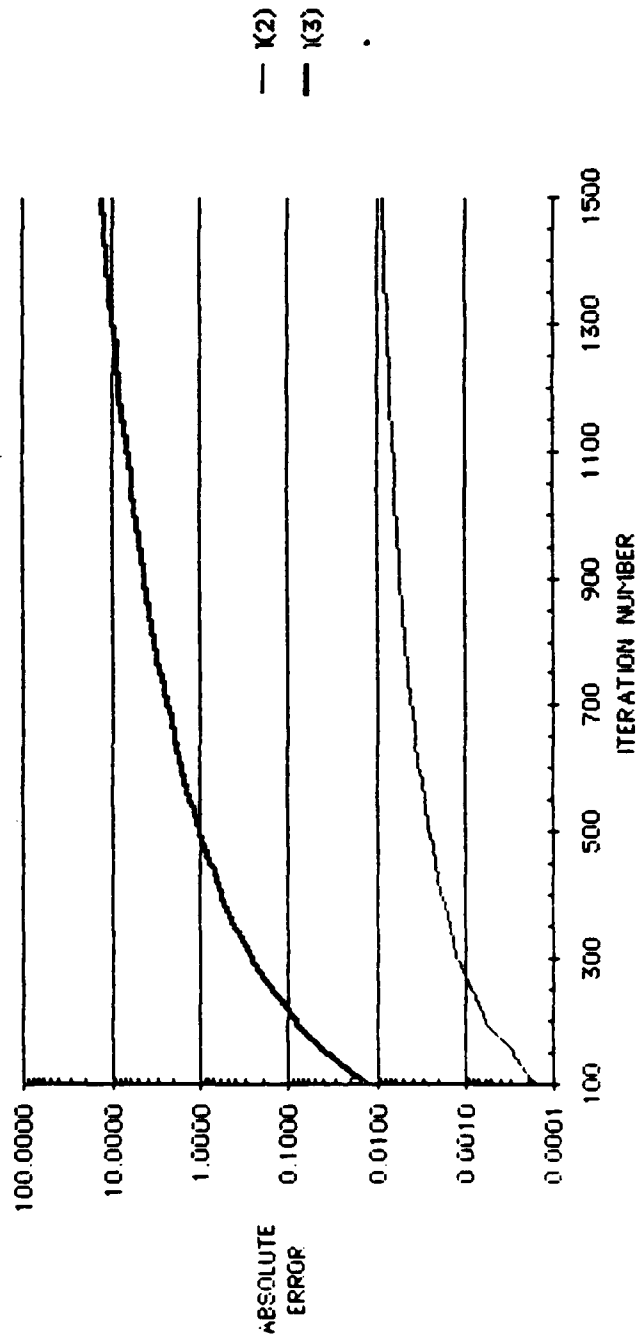


Figure C-1. Absolute Error vs Iteration Number (Block Size = 10, Noise Variance = 0.01)

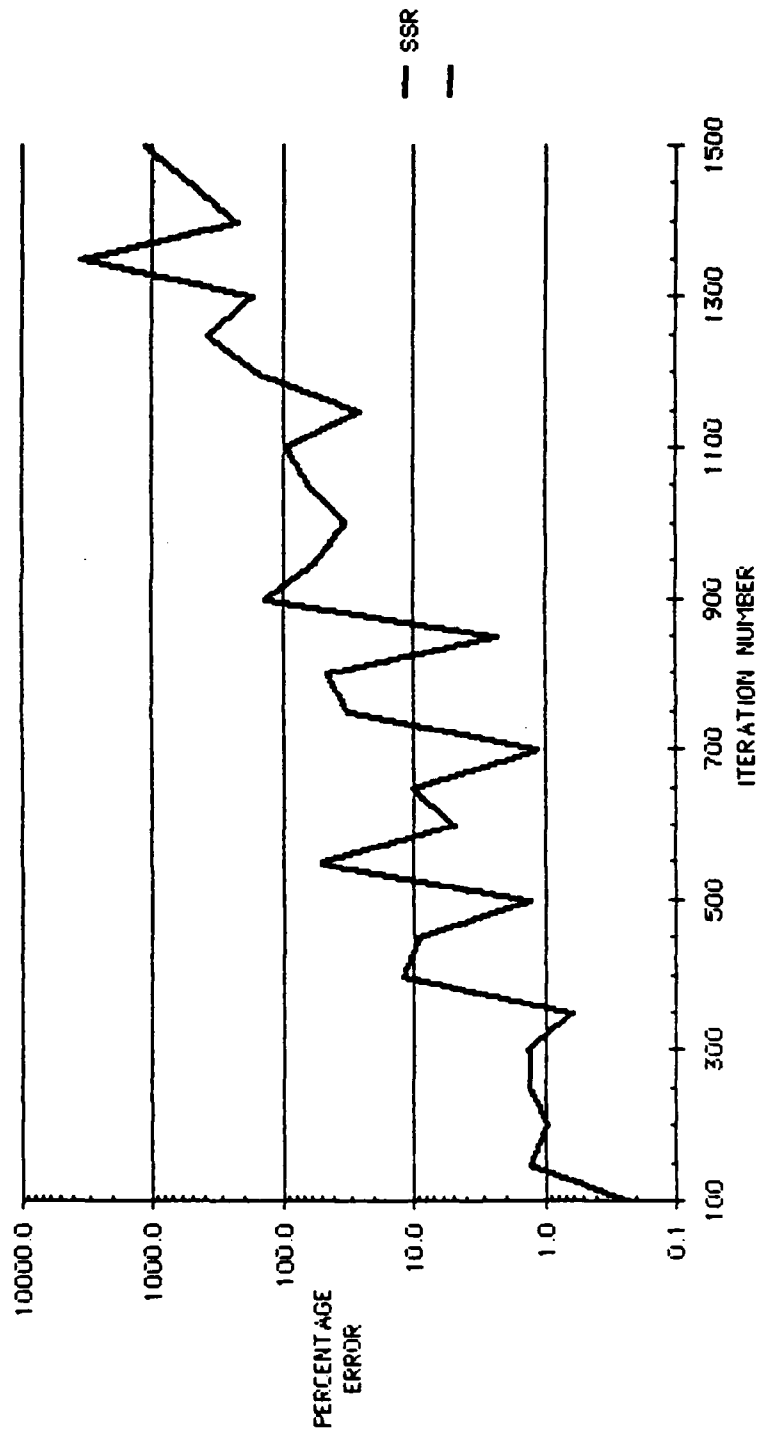


Figure C-2. Percentage Error vs Iteration Number (Block
Size = 10, Noise Variance = 0.01)

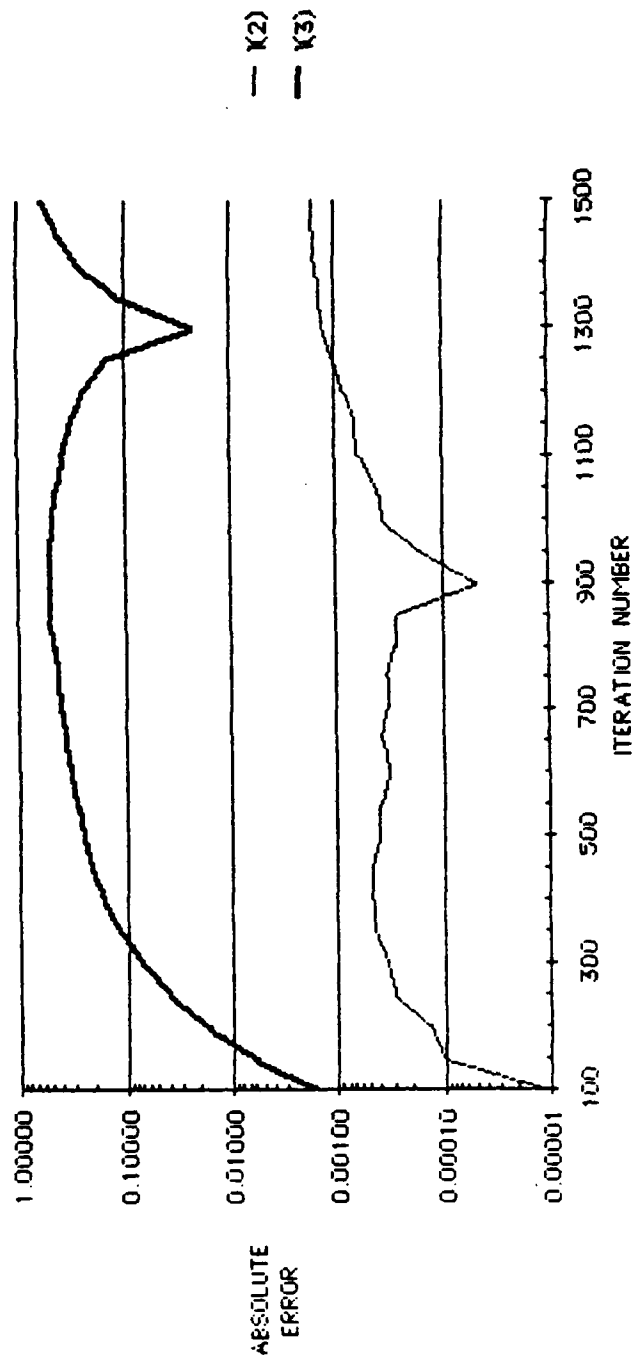


Figure C-3. Absolute Error vs Iteration Number (Block Size = 10, Noise Variance = 0.1)

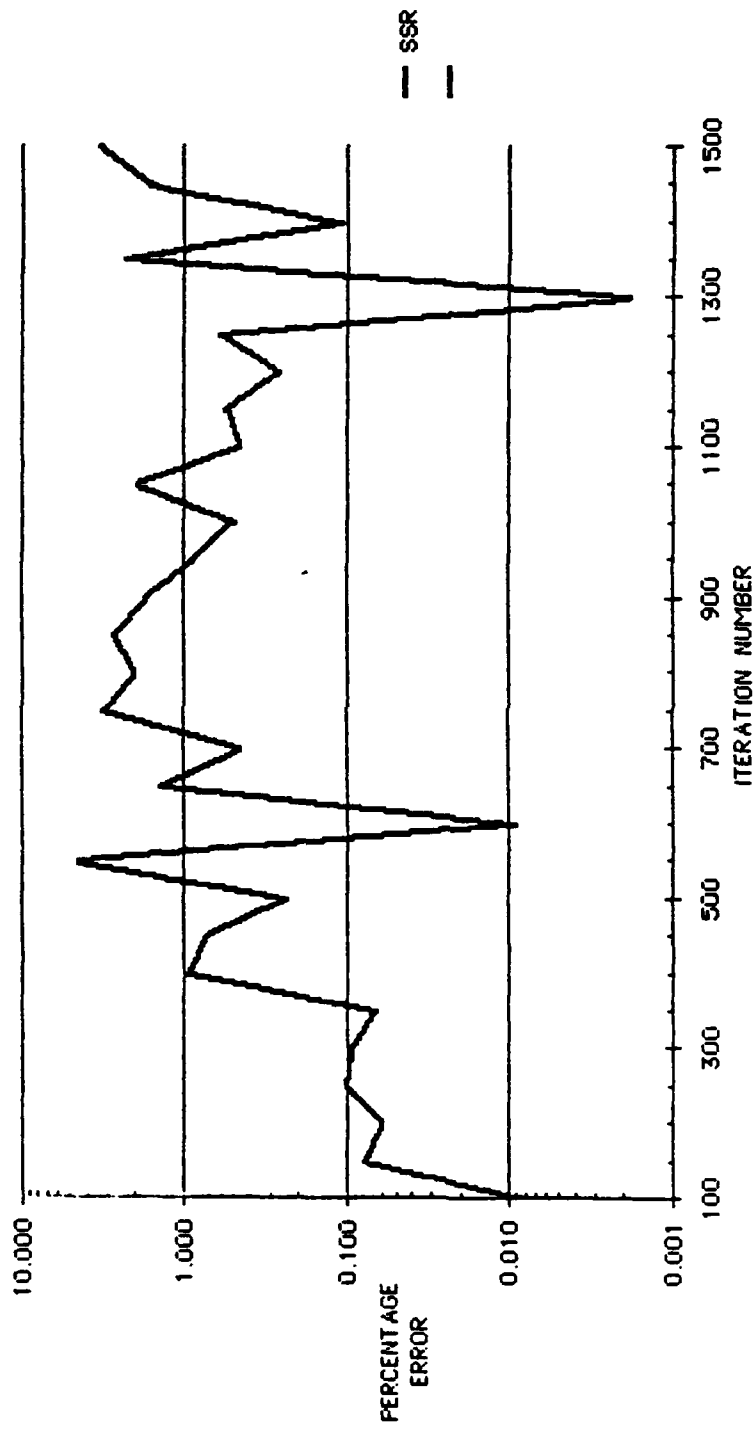


Figure C-4. Percentage Error vs Iteration Number (Block
Size = 10, Noise Variance = 0.1)

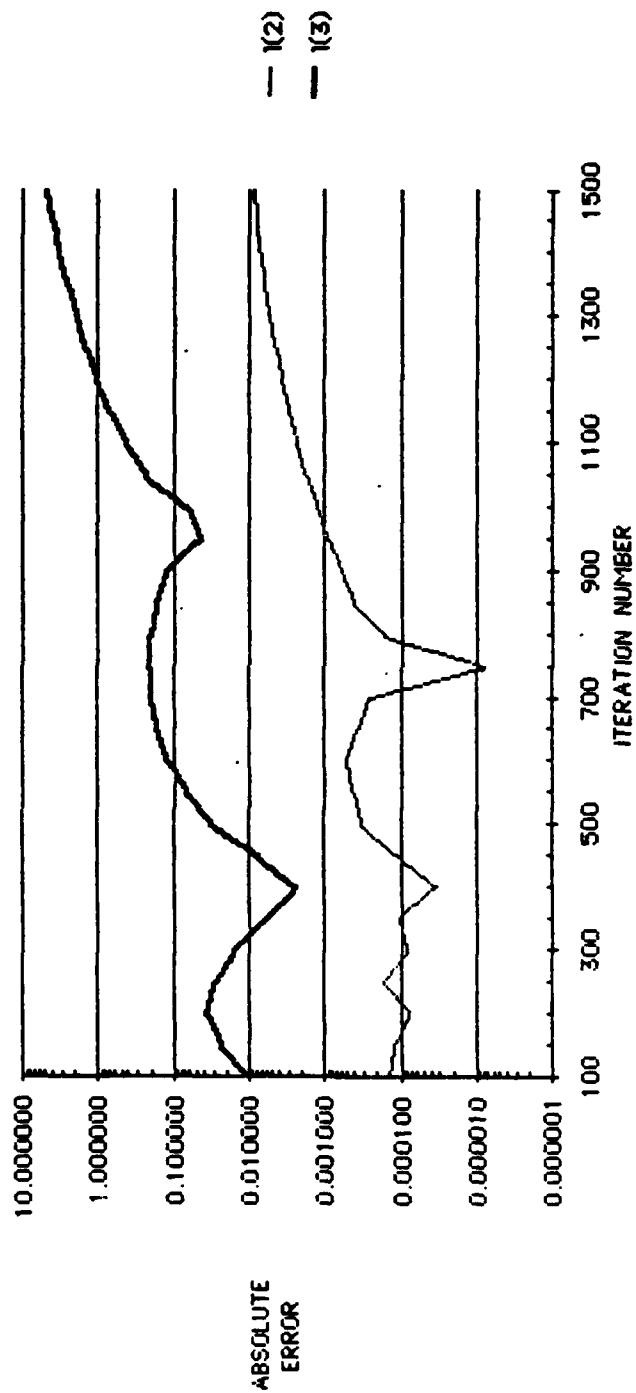


Figure C-5. Absolute Error vs Iteration Number (Block Size = 10, Noise Variance = 1)

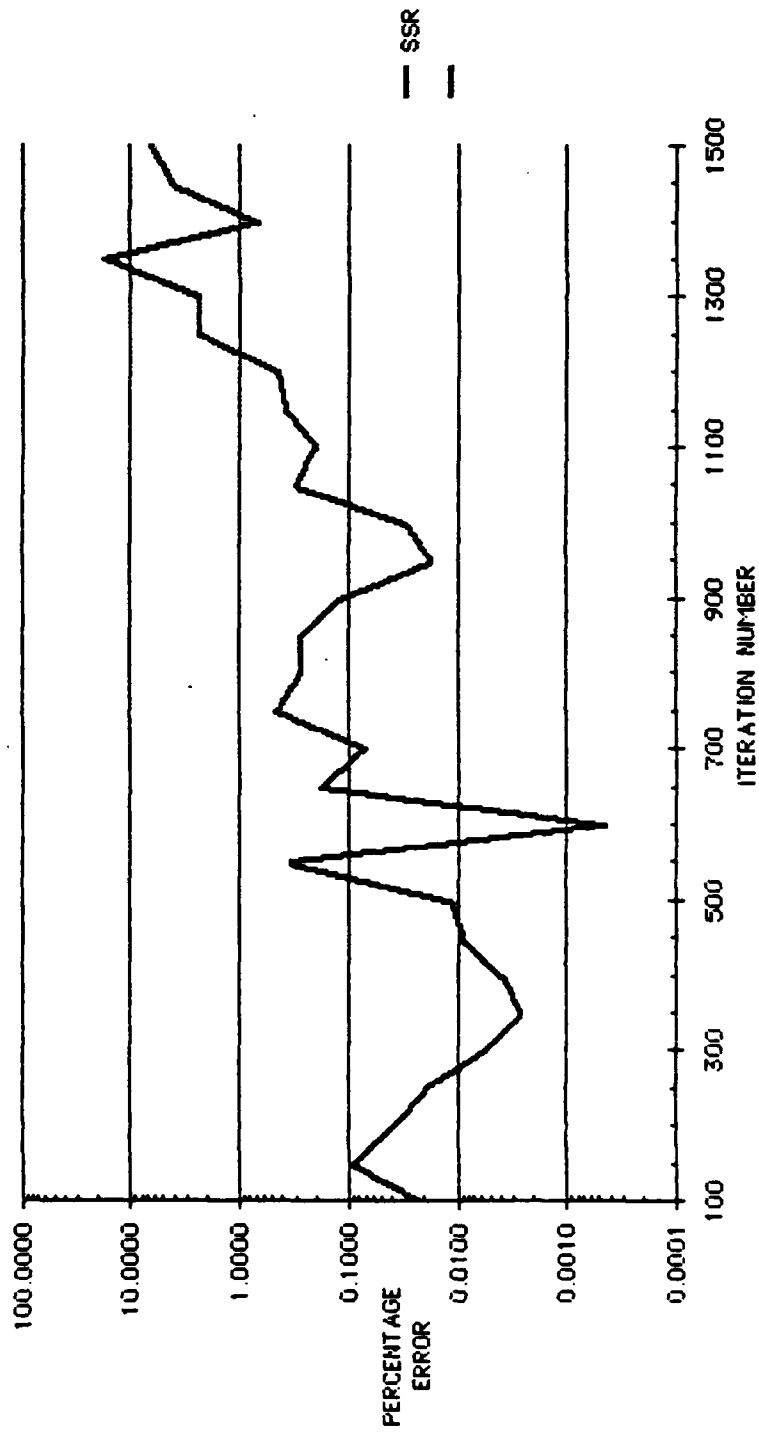


Figure C-6. Percentage Error vs Iteration Number (Block Size = 10, Noise Variance = 1)

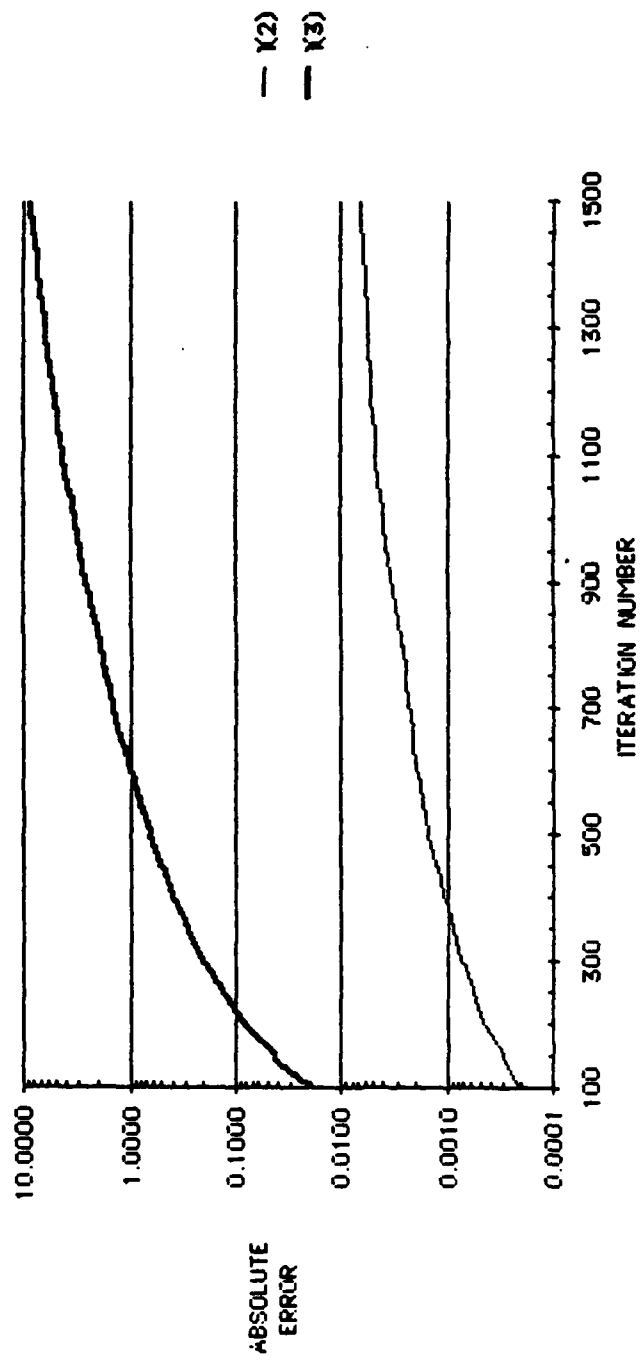


Figure C-7. Absolute Error vs Iteration Number (Block Size = 5, Noise Variance = 0.1)

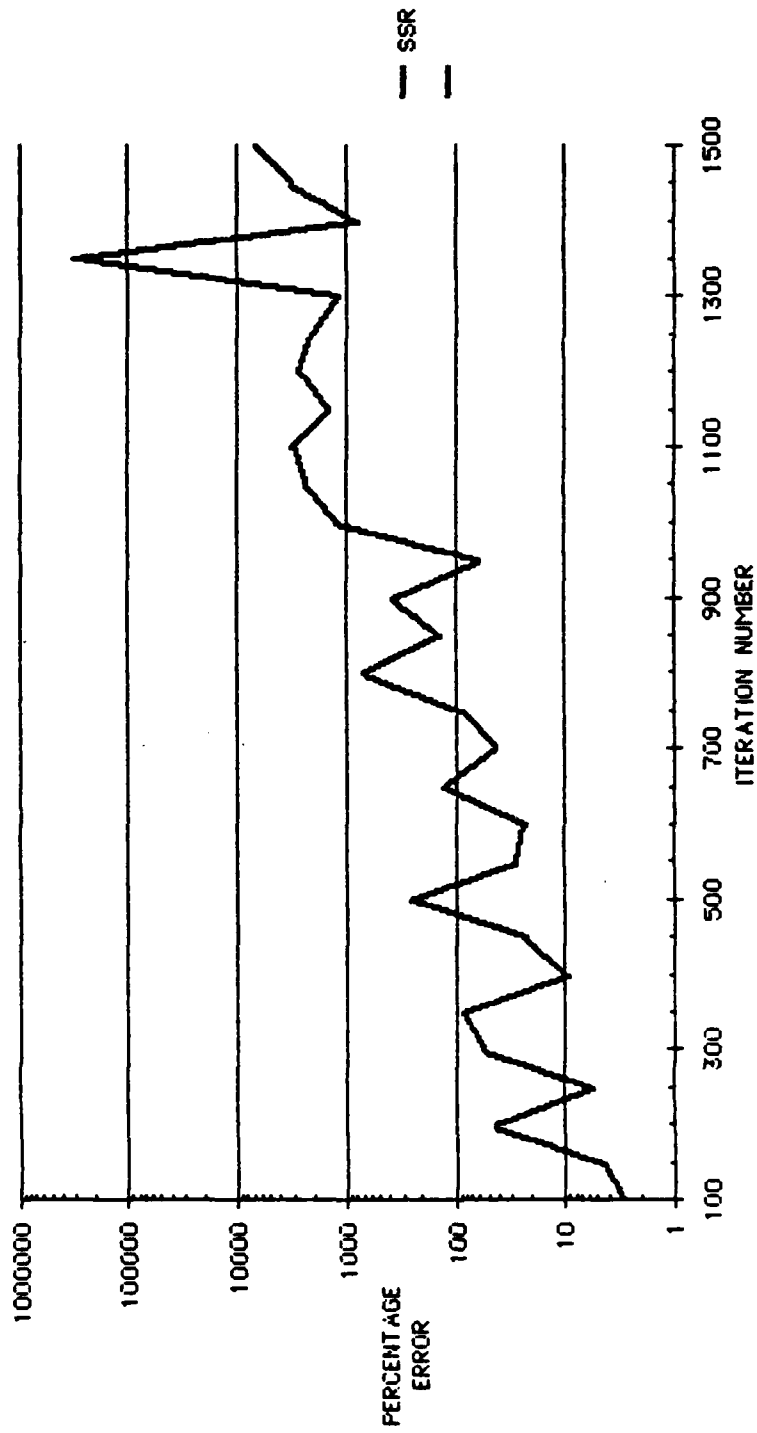
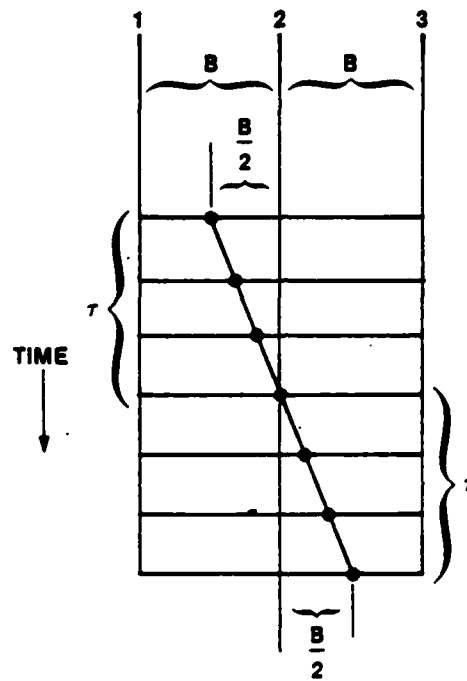


Figure C-8. Percentage Error vs Iteration Number (Block Size = 5, Noise Variance = 0.1)

APPENDIX D

THEORETICAL LOWER RMS ERROR LIMIT

FOR THE DYNAMIC PROGRAMMING ALGORITHM DESCRETIZED POINTS



From the above diagram, we can model discretation error E as the following linear function

$$E = \frac{B}{2\tau} t, \quad -\tau \leq t \leq \tau. \quad (D-1)$$

This model is repeated many times over many discretized points during a simulation.

With our model for the error E we can calculate the RMS error from the definition shown below.

$$\text{RMS} = \left(\frac{1}{2\tau} \int_{-\tau}^{\tau} \left[\frac{B}{2\tau} t \right]^2 dt \right)^{1/2} . \quad (\text{D-2})$$

The above evaluates to the following.

$$\text{RMS} = \sqrt{B^2/12} . \quad (\text{D-3})$$

Now for our simulations $B = 0.25$ (a quarter of the lag spacing).

Therefore, the limit on the RMS time delay difference error is given below.

$$\text{RMS} = 0.0722 . \quad (\text{D-4})$$

BIBLIOGRAPHY

Bellman, R. E. Dynamic Programming. Princeton: Princeton University, 1957.

Carter, G. C. IEE Special Issue on 'Time Delay Estimation'. IEEE Transactions on ASSP, Volume 29(3), June 1981.

Chan, Y. T., Riley, J. M., and Plant, J. B. Modeling of Time Delay and its Application to Estimation of Non Stationary Delays. IEEE ASSP Vol. 29, June 1981.

Cretella, D. B., and Jarvis, H. F. Jr. Wide Aperture Array ADM Sub-system Computer Simulation Program Description and Users Guide. Analysis & Technology, Inc. Report No. P-1143-03-82.

Dahlquist, G., and Bjorck, A. Numerical Methods. Prentice-Hall, 1974.

Forsythe, G. E., Malcolm, M. A., and Moler, C. B. Computer Methods for Mathematical Computations. Prentice-Hall, 1977.

Friedlander, B. Multitarget Tracking Study, Phase I Final Report. System Control, Inc. Report No. 5334-01.

Gold, B., and Rader, C. M. Digital Processing of Signals. McGraw-Hill, 1969.

BIBLIOGRAPHY (Continued)

Goldfarb, D. A Family of Variable Metric Methods Derived by Variational Means. Mathematics of Computation, 1970, Vol. 24, 23-26.

Hestenes, M. Conjugate Direction Methods in Optimization. Springer-Verlag, 1980.

Knapp, C. H., and Carter, G. C. Estimation of Time Delay in the Presence of Source or Receiver Motion. JASA Vol. 61 No. 6, June 1977.

Knapp, C. H., and Carter, G. C. The Generalized Correlation Method for Estimation of Time Delay. IEEE ASSP Vol. 24, August 1976.

LaTourette, R. A., Greineder, S. G., and Wolcin, J. J. Development of the Batch-Oriented MAPLE Frequency Line Detection/Tracking Algorithm into a Fast Recursive MAPLE Frequency Line Tracking Algorithm. NUSC TM No. 851048, 12 July 1985 (UNCLASSIFIED)

LaTourette, R. A., and Ng, L. C. Equivalent Bandwidth of a General Class of Polynomial Smothers: With Application to Bearing Tracker Random Error Evaluation. NUSC TECH Report 6601, 19 July 1982 (UNCLASSIFIED)

Ng, L. C. Informed Correspondence, 1985.

BIBLIOGRAPHY (Continued)

Ng, L. C. Optimum Multisensor, Multitarget Localization and Tracking.
PhD Dissertation, University of Connecticut, 1983.

Ng, L. C., and Lambert, P. R. Fast Moving Average Recursive Least Mean
Square Fit. NUSC TM No. 841143, 30 September 1984 (UNCLASSIFIED)

Ng, L. C., and LaTourette, R. A. Equivalent Bandwidth of a General
Class of Polynomial Smoothers. JASA 74(3), September 1983.

Ng, L. C., and LaTourette, R. A. Fast Moving Average Recursive Least
Mean Square FIT. IEEE 24th CDC, Dec. 11-13, 1985.

Steinberg, B. D. Principles of Aperture and Array System Design. John
Wiley and Sons, 1976.

Van Trees, H. L. Detection, Estimation and Modulation Theory PART 1.
John Wiley and Sons, 1968.

Wenk, C. informal conversation 8/85.

Whalen, A. D. Detection of Signals in Noise. Academic Press, Inc.,
1971.

Wolcin, J. J. Maximum A-Posteriori Estimation of Narrowband Signal
Parameters. JASA Vol. 68, July 1980.

BIBLIOGRAPHY (Continued)

Wolcin, J. J. Maximum A-Posteriori Estimation of Narrowband Signal Parameters. NUSC TM No. 791115, 21 June 1979 (UNCLASSIFIED).

Wolcin, J. J. Maximum A-Posteriori Line Extraction, A Computer Program Discription. NUSC TM No. 801042, 20 March 1980 (UNCLASSIFIED)

END

12-86

DTIC