NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A246 439

DTIC
ELECTE
FEB 26 1992
S B D

THESIS

KALMAN SMOOTHER APPLIED TO LACE

by

Frank R. Thorngren

December 1991

Thesis Advisor: Jeffrey B. Burl

Approved for public release; distribution is unlimited

92-04598

92 2 24 014

8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

KALMAN SMOOTHER APPLIED TO LACE

12. PERSONAL AUTHOR(S)

THORNGREN, Frank R.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM _____ TO _____	1991 December	59

16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	ground based Doppler resolved measurements; LACE; Kalman filter; fixed interval smoother; spacecraft structural dynamics; heterodyne signal

18. cont.
envelope.

19. cont.
based method employed to obtain a Doppler separation history. This thesis addresses the implementation of the Kalman filter algorithm in conjunction with the Rauch-Tung-Striebel fixed-interval optimal smoother algorithm to perform this filtering task. The Kalman smoother filtering based method of processing the data produced superior results when compared with the histogram filtering based method.

Approved for public release; distribution is unlimited

Kalman Smoother Applied to LACE

by

Frank R. Thorngren
Lieutenant, United States Navy
B.A., Rutgers University, 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1991

Author: _____
Frank R. Thorngren

Approved by: _____
Thesis Advisor, Jeffrey B. Burl

_____
Second Reader, Ralph Hippenstiel

_____
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

COPY INSPECTED 4

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

ABSTRACT

The ability to determine the structural dynamics of space-based platforms from ground-based radar resolved Doppler measurements will aid in the study of control/structure interaction. The Naval Research Laboratory and Lincoln Laboratory conducted an experiment to determine the feasibility of this method. To accomplish this experiment the LACE satellite was equipped with retroreflectors and the ground-based Firepond laser radar facility was employed. Vibrational information is found from the difference between the reflected Doppler frequencies of the retroreflectors. The method of extracting the Doppler separation was to obtain the power spectrum of the heterodyne signal envelope. A pulse-by-pulse processing of the data yields the Doppler separation history over time. Due to a relatively large amount of clutter in the processed data, a filtering mechanism was employed. The histogram technique is the current filtering-based method employed to obtain a Doppler separation history. This thesis addresses the implementation of the Kalman filter algorithm in conjunction with the Rauch-Tung-Striebel fixed-interval optimal smoother algorithm to perform this filtering task. The Kalman smoother filtering based method of processing the data produced superior results when compared with the histogram filtering based method.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. THE LACE DYNAMICS EXPERIMENT 5

II. LACE SIGNAL PROCESSING 10

III. KALMAN FILTER 14

IV. FIXED INTERVAL OPTIMAL SMOOTHER 21

V. KALMAN SMOOTHER APPLIED TO LACE 25

VI. CONCLUSION 40

Appendix SUBROUTINES DEVELOPED TO PROCESS THE DATA 42

List of References 49

Initial Distribution List 50

List of Figures

Figure 1. LACE Spacecraft [Fig 1a from Ref 1].....2

Figure 2. Primary Experiment [Fig 1b from Ref 1]..... 3

Figure 3. LACE IQ Envelope, GMT DAY 200, 7603.875 Seconds After Midnight 6

Figure 4. Power Spectrum of IQ Envelope, GMT DAY 200, 7603.875 Seconds After Midnight.....7

Figure 5. Doppler Separation, GMT DAY 195, 2 Second Iteration [Fig 7a from Ref 2].....8

Figure 6. Doppler Separation, GMT DAY 200, 2 Second Iteration [Fig 7b from Ref 2].....9

Figure 7. LACE Signal Processing Block Diagram 10

Figure 8. Kalman Filter Block Diagram 14

Figure 9. Fixed Interval Optimal Smoother Block Diagram 21

Figure 10. Fixed Window, Unfiltered Data, GMT DAY 195 29

Figure 11. Accurate IQ Envelope Information 30

Figure 12. Accurate Power Spectrum Information 30

Figure 13. IQ Envelope Corrupted by Noise 32

Figure 14. Power Spectrum Corrupted by Noise 32

Figure 15. Amplified Noise in Time Domain 33

Figure 16. Amplified Noise in Frequency Domain 33

Figure 17. Doppler Separation, GMT DAY 200, Q = 292.969, R = 292.969 37

Figure 18. Doppler Separation, GMT DAY 200, Q = 146.485, R = 292.969 37

Figure 19. Doppler Separation, GMT DAY 200, Q = 146.485, R = 2929.690 38

Figure 20. Doppler Separation, GMT DAY 200, Q = 7.324, R = 2929.690 39

Figure 21. Doppler Separation, GMT DAY 195, Q = 7.324, R = 2929.690 39

Figure 22. Doppler Separation Composite, GMT DAY 195 [Fig 7a after Ref 2].....40

Figure 23. Doppler Separation Composite, GMT DAY 200 [Fig 7b after Ref 2].....41

I. INTRODUCTION

Control/structure interaction (CSI) is the interaction between control systems and the platform or the structural appendages. The problem of control/structure interaction is of great interest in the development of new space-based platforms. There are unique problems associated with the ground-based testing of structures designed for weightless environments. In strong gravitational fields spaced-based platforms exhibit different structural characteristics than those found in a weightless environment. This presents problems in developing models to simulate the structural dynamics based on data obtained from ground level experimentation. An ideal way to develop accurate models using experimental data is to obtain the data while the platform is in orbit. There are various methods that could be utilized to accomplish this task. A method that does not involve telemetric links or sophisticated electronic hardware installed on the platform is remote ground-based Doppler resolved measurements. These ground-based Doppler resolved measurements will then be used to analyze the structural dynamics of the platform. The Naval Research Laboratory [Ref. 1] and Lincoln Laboratory [Ref. 2] have been sponsoring a series of experiments to determine the feasibility of using the ground-based Doppler resolved measurement approach.

To accomplish this study, the Laser Atmospheric Compensated Experiment (LACE) Satellite (object number 20496) was equipped with three germanium IR retroreflectors prior to launch. These IR retroreflectors were located on the forward boom, trailing boom, and body of the satellite as indicated in Figure 1.

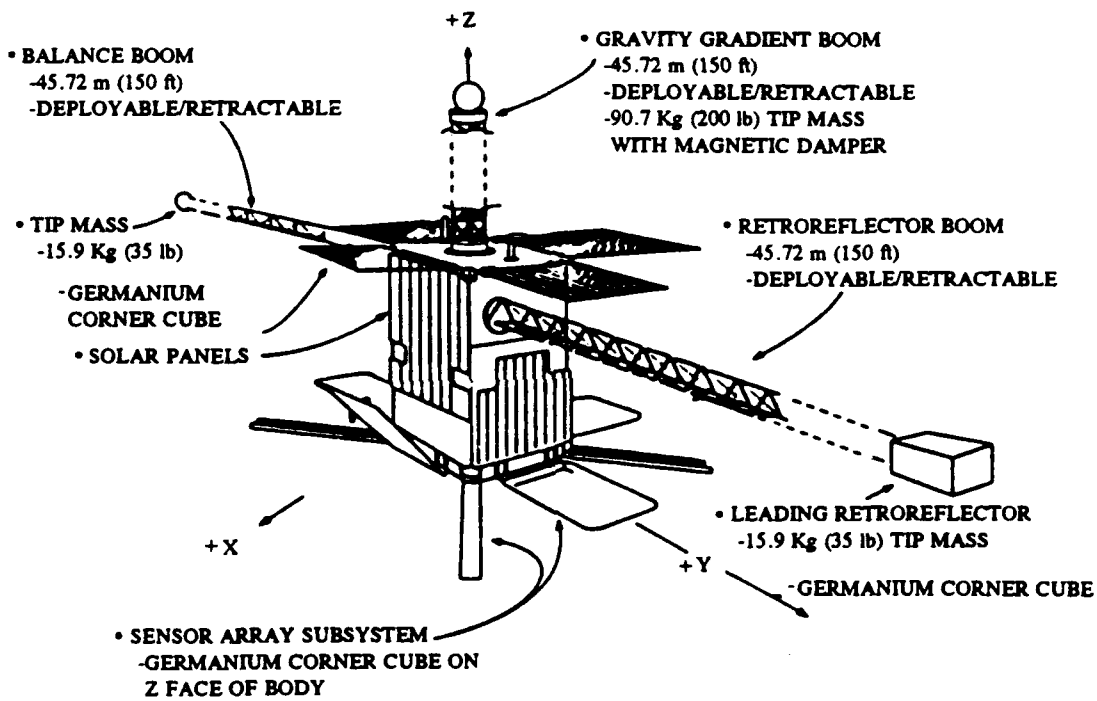

Figure 1. LACE Spacecraft [Fig 1a from Ref. 1]

The retroreflectors are then illuminated with the ground-based Firepond laser radar as depicted in Figure 2. The Firepond is a coherent narrowband 10.6 micrometer laser radar

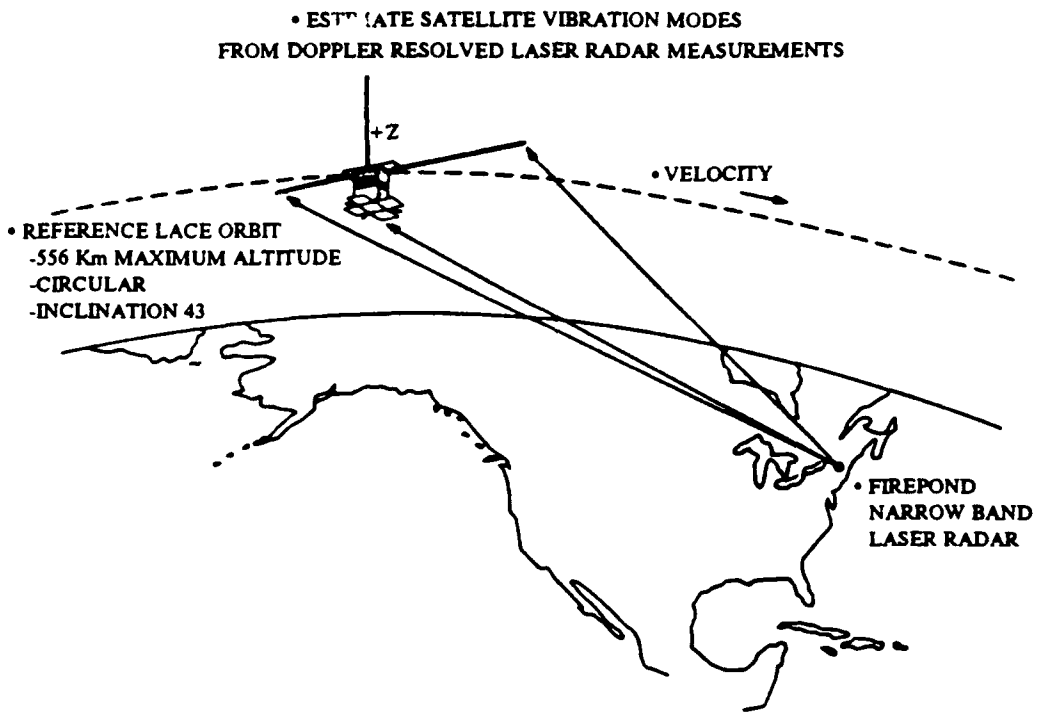

Figure 2. Primary Experiment [Fig 1b from Ref. 1]

facility. The reflected signals from the satellite contained Doppler frequency components proportional to the relative velocity of the germanium retroreflectors projected along the radar line-of-sight. The Doppler separation is the difference between the Doppler frequencies of the retroreflectors. The procedure employed in the extraction of this Doppler separation from the reflected signal consisted of obtaining the power spectrum of the heterodyne signal envelope. This method greatly reduced the tolerance requirement of the equipment and calculations needed to extract the Doppler separation data from the measurements.

One major drawback to this method is that noise rejection for this system is poor. The Doppler separation history obtained from the pulse-by-pulse processing had a significant amount of clutter. Lincoln Laboratory used a histogram filtering based technique for further noise rejection to obtain a Doppler separation history. Another approach to this problem, explored in the following chapters, will consist of the use of a Kalman Filter in conjunction with the Rauch-Tung-Striebel fixed interval optimal smoother. The Kalman filter is used to estimate the Doppler frequency and control a dynamic tracking window. This method of extracting the information components from the data showed a remarkable improvement in the resolution of the LACE satellite's Doppler separation history over the histogram filtering-based technique.

The basic theory and the equations used in the LACE signal processing are discussed in the second chapter. In Chapter III the Kalman filter equations and performance characteristics are discussed. The Rauch-Tung-Striebel fixed interval smoother is discussed in Chapter IV. The remaining two chapters are devoted to the description of how the Kalman filter and the Rauch-Tung-Striebel fixed interval optimal smoother were utilized in this application and the results obtained.

A. THE LACE DYNAMICS EXPERIMENT

Lincoln Laboratory discusses the experiment and the procedures employed for the analysis of the narrowband IR measurements obtained for the 13 and 18 July 1990 [Ref. 2]. On both days, the LACE satellite's configuration had the leading boom's extension at 4.6 meters (15 feet) and the trailing boom's extension at 46 meters (150 feet). This configuration had been set up for a significant time prior to the illumination to eliminate any vibrational modes that might have been excited by the boom's movement. On both tracking runs, the Firepond laser radar had a peak transmit power of 780 watts, a pulse duration of 1.5 milliseconds, and a pulse repetition frequency (PRF) of 62.5 Hz. The maximum elevation that the LACE satellite achieved relative to the Firepond laser radar site on 13 July was 82 degrees and on 18 July was 77 degrees. Due to the transmission beam's footprint of 12 meters at the minimum range of 547 kilometers, only the leading boom's retroreflector and body's retroreflector were illuminated.

The procedure that was used in the analysis of the received data first consisted of digitizing 3.4 milliseconds of the in-phase and quadrature(IQ) data at 1.2 MHz (generating 4080 complex IQ samples). These complex IQ samples were then squared to yield the IQ envelope as shown in Figure 3. This figure shows the first radar return obtained from the tape for 18 July 1990. The power spectral density from the IQ envelope

was then obtained. Figure 4 depicts the power spectral density of Figure 3. Observation of this radar return indicates a nominal 12 kHz Doppler separation.

The Doppler separation history obtained from the pulse-by-pulse processing had a significant amount of clutter. Consequently, a filtering-based technique had to be employed to extract the information component from the processed data.

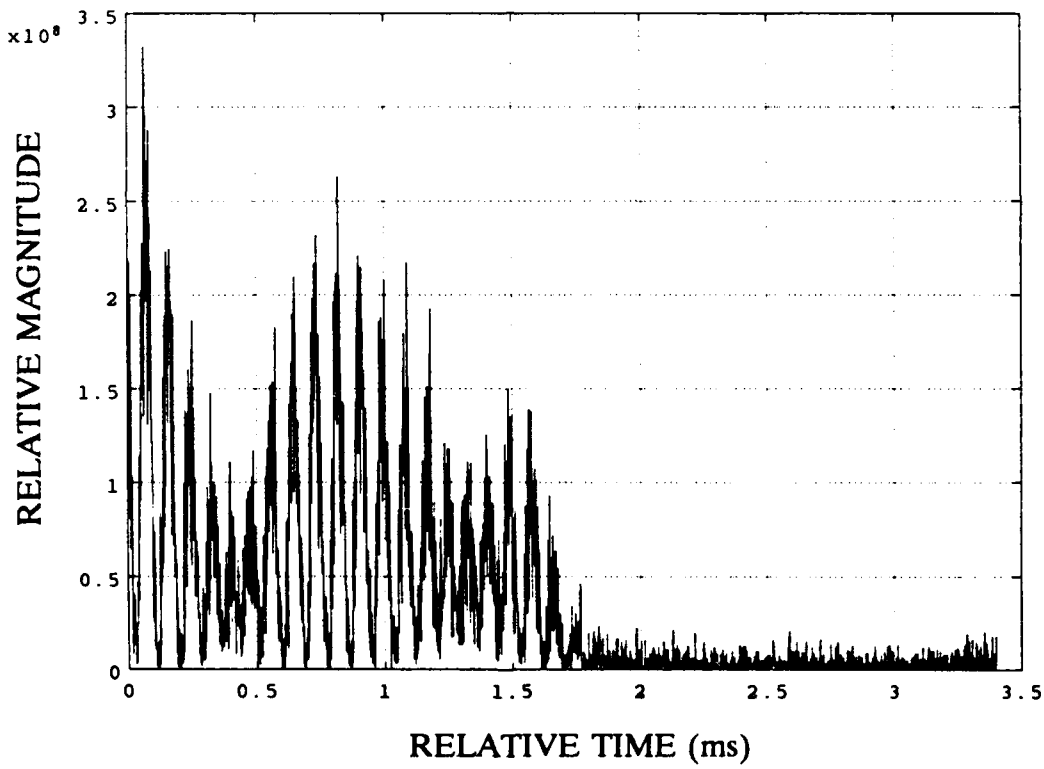

Figure 3. LACE IQ Envelope, GMT DAY 200, 7603.875 Seconds After Midnight

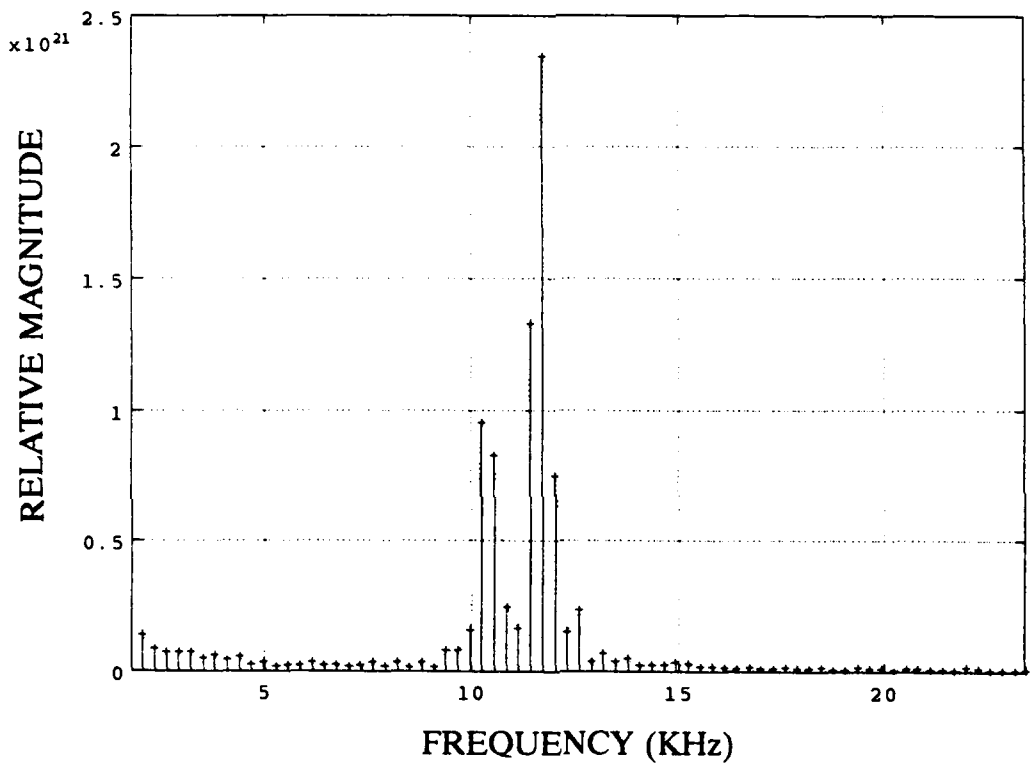

Figure 4. Power Spectrum of IQ Envelope, GMT DAY 200, 7603.875 Seconds After Midnight

This filtering-based technique consisted of taking the maximum value of a 2 second (125 point) moving histogram to determine the most likely Doppler separation track. Figure 5 for 13 July 1990 and Figure 6 for 18 July 1990 illustrates the

results obtained by using this method. In analyzing Figures 5 and 6, it is evident that both histories are parabolic in nature. The biggest difference between the two Doppler separation histories is the existence of a flatter track for 13 July. Obtaining the power spectrum of these Doppler separation histories will reveal the frequency components resulting from the structural dynamics of the craft.

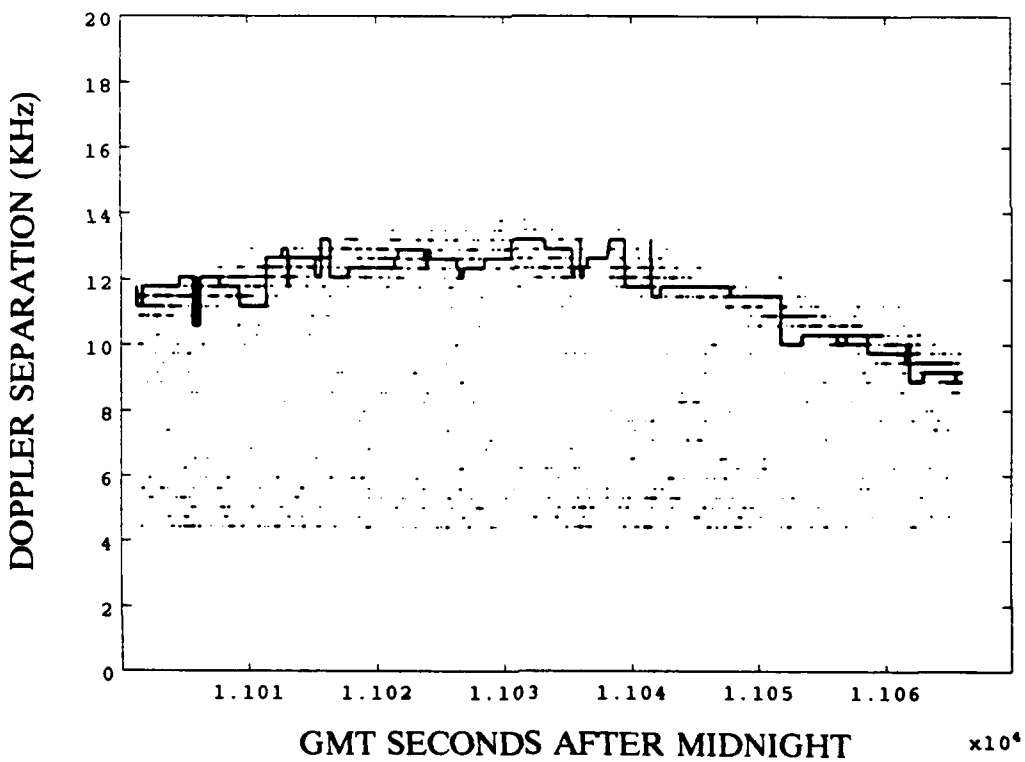

Figure 5. Doppler Separation, GMT DAY 195, 2 Second Iteration [Fig 7a from Ref. 2]

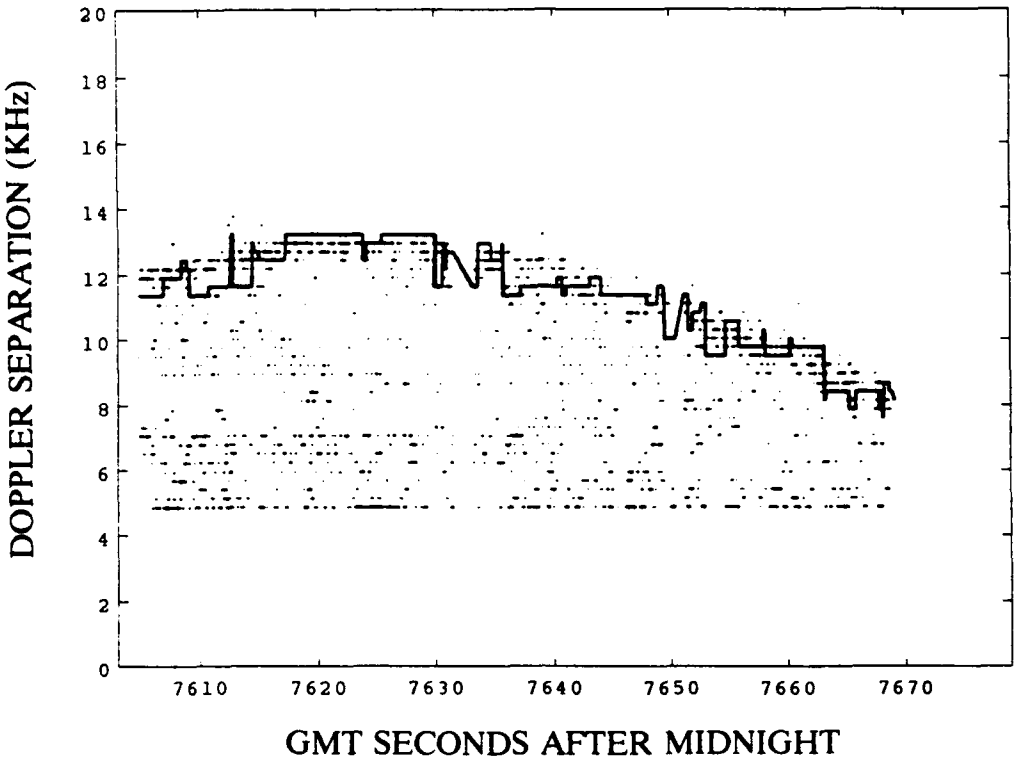

Figure 6. Doppler Separation, GMT DAY 200, 2 Second Iteration
[Fig 7b from Ref. 2]

II. LACE SIGNAL PROCESSING

LACE Signal Processing refers to the equations and procedures required to describe the signal processing of the received signal and the calculation of the power spectrum of this received signal. This process is represented in Figure 7, where *LO* represents the local oscillator and *LPF* represents the low pass filter. *A/D* is the analog to digital converter. FFT represents the fast Fourier transform. $S_I^2(n)$ is the squared in-phase component and $S_Q^2(n)$ is the squared quadrature

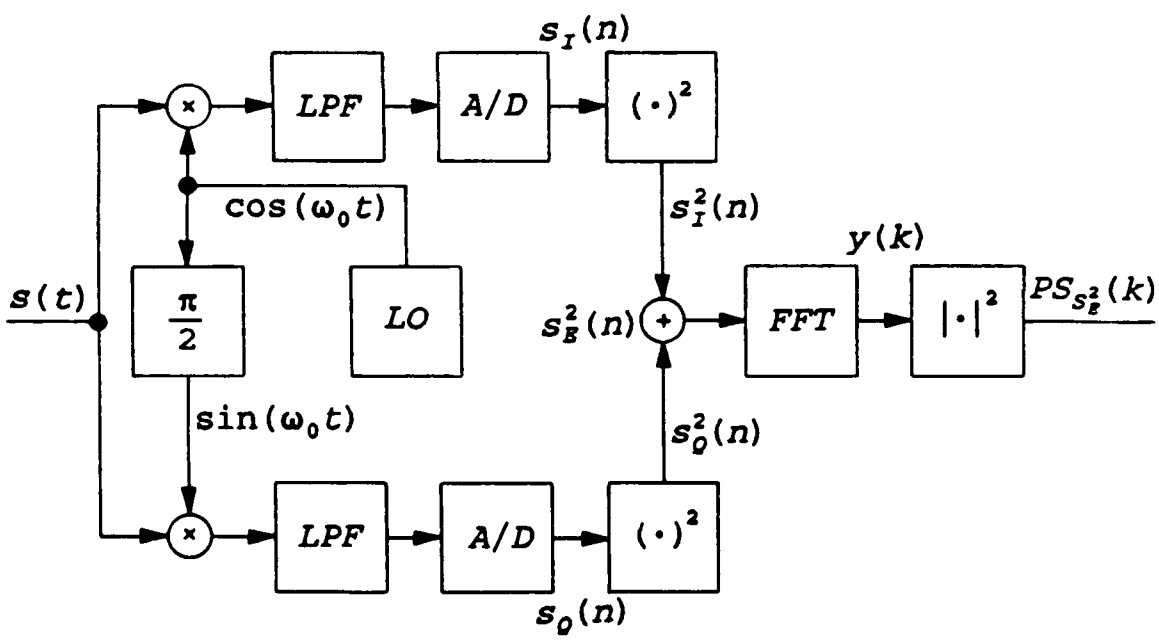

Figure 7. LACE Signal Processing Block Diagram

component of the signal. The other terms in this figure are described in Equations (1) through (6). The equations used to describe the LACE Signal Processing were derived in References 3 and 4.

The signal received from the LACE satellite consisted of the components from both the forward boom's retroreflector and the body's retroreflector. Equation (1) is a mathematical representation of a noise free signal:

$$s(t) = a_1 \cos[(\omega_0 + \omega_{d1})t + \phi_1] + a_2 \cos[(\omega_0 + \omega_{d2})t + \phi_2] \qquad (1)$$

where the following terms apply to this equation:

$s(t)$-is the received signal,

a_1-is the amplitude of the leading retroreflector,

a_2-is the amplitude of the body retroreflector,

ω_0-is the carrier frequency,

ω_{d1}-is the Doppler frequency of the leading retroreflector,

ω_{d2}-is the Doppler frequency from the body retroreflector,

ϕ_1-is the phase return of the leading retroreflector, and

ϕ_2-is the phase return of the body retroreflector.

This signal is then processed by the laser radar to yield the in-phase and quadrature components of the signal. The in-phase and quadrature components are passed through a *LPF* and an *A/D*

converter. The two resulting signals are described mathematically by Equations (2) and (3):

$$s_I(n) = \frac{a_1}{2}\cos(\omega_{d1}nT+\phi_1) + \frac{a_2}{2}\cos(\omega_{d2}nT+\phi_2) \tag{2}$$

$$s_Q(n) = \frac{a_1}{2}\sin(\omega_{d1}nT+\phi_1) + \frac{a_2}{2}\sin(\omega_{d2}nT+\phi_2) \tag{3}$$

where the following terms apply to these equations:

$s_I(n)$-is the in-phase component and

$s_Q(n)$-is the quadrature component.

The IQ envelope was obtained by adding together the squared in-phase and squared quadrature components generated by the radar. This signal is represented by Equation (4):

$$s_E^2(n) = \frac{(a_1^2+a_2^2)n}{4} + \frac{(a_1a_2)n}{2}\cos[(\omega_{d1}-\omega_{d2})n+\phi_1-\phi_2] \tag{4}$$

where the following term applies to this equation:

$s_E^2(n)$-is the IQ envelope.

Equations (5) and (6) are used to calculate the power spectrum of the IQ envelope:

$$y(k) = \sum_{n=0}^{N-1} s_E^2(n)\, e^{-j\left(\frac{2\pi}{N}\right)nk} \tag{5}$$

$$PS_{S_E^2}(k) = y(k)\, y(k)^* \tag{6}$$

where the following terms applies to these equation:

$y(k)$-is the complex signal (Fourier transform of IQ envelope),

$y(k)^*$-is the complex conjugate of the complex signal,

N -is the transform length, and

$PS_{S_E^2}(k)$-is the power spectrum.

This results in one peak located at the Doppler separation frequency as was indicated in Figure 4.

III. KALMAN FILTER

The Kalman filter was developed in 1960 by Dr R. E. Kalman as an optimal recursive filter for the estimation of a state vector from measurement data corrupted by noise. It offered advantages over other filters such as the Wiener filter in that it reduces the mathematical complexity of the processing of large data strings. A block diagram of the Kalman filter is depicted in Figure 8, where *DELAY* represents one discrete-time delay. The rest of the terms in this figure are discussed in Equations (7) through (16).

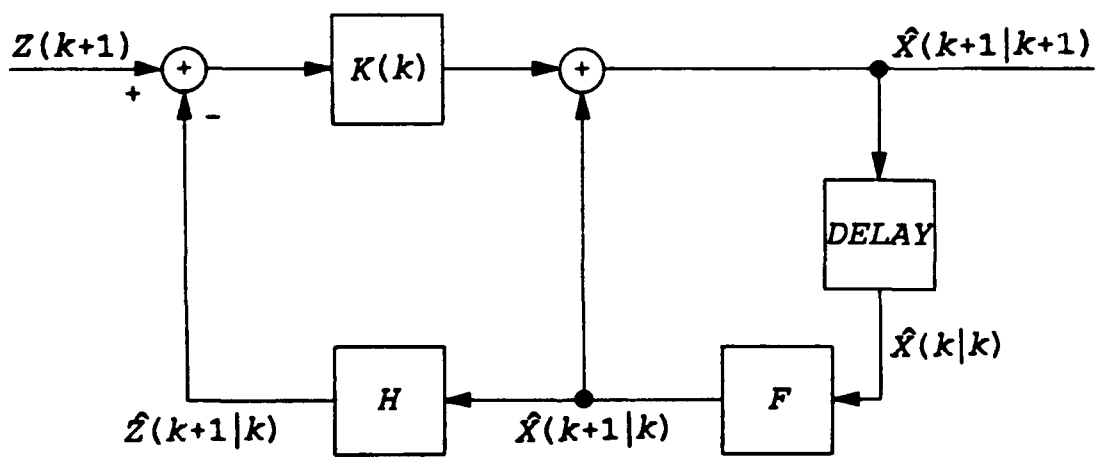

Figure 8. Kalman Filter Block Diagram

The Kalman filter equations are derived in References 5 and 6. It is assumed that the process is time invariant. A mathematical model of the system is given below. The discrete-time state equation is represented by Equation (7):

$$X(k+1) = FX(k) + Gv(k) \tag{7}$$

The measurement equation is represented by Equation (8):

$$Z(k+1) = HX(k+1) + Dw(k) \tag{8}$$

where the following terms apply to the two preceding equations:

$X(k+1)$ -is the updated state vector,

$X(k)$ -is the state vector,

F -is the state transition matrix,

$v(k)$ -is the discrete time process noise assumed to be a zero-mean, white, random sequence,

$Z(k+1)$ -is the measurement vector,

H -is the gain through which the output leaves the system,

$w(k)$ -is the discrete time measurement noise assumed to be a zero-mean, white, random sequence,

G -is the gain through which the process noise enters the system, and

D -is the gain through which the measurement noise enters the system.

The equations involved specifically in the Kalman filter algorithm are discussed by dividing then into the prediction, innovation, gain and correction parts. In the prediction section, the conditional mean of the state vector (Equation 9), the conditional state error covariance matrix (Equation 10), and the predicted measurement vector (Equation 11) are computed:

$$\hat{X}(k+1|k) = F\hat{X}(k|k) \tag{9}$$

$$P(k+1|k) = FP(k|k)F^T + GQG^T \tag{10}$$

$$\hat{Z}(k+1|k) = H\hat{X}(k+1|k) \tag{11}$$

where the following terms apply to these three equations:

$\hat{X}(k+1|k)$ -is the conditional state estimate,

$\hat{X}(k|k)$ -is the previous state estimate,

$P(k+1|k)$ -is the conditional predicted state error covariance matrix,

Q -is the covariance of the process noise,

$P(k|k)$ -is the previous predicted state error covariance matrix, and

$\hat{Z}(k+1|k)$ -is the estimated measurement.

The innovation section calculates the error between the measurement equations and the innovation covariance. Equation (12) calculates the measurement residual or innovation between Equations (8) and (11):

$$e(k+1|k) = Z(k+1) - \hat{Z}(k+1|k) \tag{12}$$

Equation (13) determines the innovation covariance:

$$S(k+1|k) = HP(k+1|k)H^T + DRD^T \tag{13}$$

where the following terms apply to these two equations:

$e(k+1|k)$ -is the innovation or measurement residual,

$S(k+1|k)$ -is the innovation covariance, and

R-is the measurement noise covariance.

The Kalman filer gain or weighting factor is found by Equation (14):

$$K(k) = P(k+1|k)H^T S(k+1|k)^{-1} \tag{14}$$

where:

$K(k)$ -is the Kalman filter gain.

The Kalman gain is the weighting factor that is placed on the measurement residual and the covariance prediction in the correction phase. Equation (15) corrects the old estimated state and the covariance correction is found by Equation (16):

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k)\, e(k+1|k) \tag{15}$$

$$P(k+1|k+1) = [I - K(k)\, H]\, P(k+1|k) \tag{16}$$

where the following terms apply to these two equations:

$\hat{X}(k+1|k+1)$ -is the updated state estimate,

$P(k+1|k+1)$ -is the updated predicted state error covariance, and

I -is the identity matrix.

The preceding equations are then used recursively in the discussed order to obtain estimates of the state at a given k.

There are two major factors that can affect the performance of the Kalman filter [Ref. 7], the first being the Kalman filter parameters such as process noise covariance, measurement noise covariance, and the initial conditions. These parameters are the fine tuning mechanisms of the filter.

It is seen in the Kalman filter equations that the gain is dependent on the prediction covariance and the measurement noise covariance. The prediction covariance is also dependent on the process noise covariance. If the process noise covariance in Equation (10) is increased, the prediction covariance increases. Therefore, the Kalman filter gain in Equation (14) can be considered as a trade off between the covariance of the process noise to the covariance of the measurement noise. With this in mind, as the process noise covariance increases, the Kalman filter gain increases and consequently, the bandwidth increases. This forces a faster transient response which leads to more noise in the estimates generated by Equation (11). By decreasing the measurement noise covariance in Equation (13), the same effect can be achieved. If the process noise covariance is decreased then the opposite effect will occur, which means that less noise will be present in the estimated states. With respect to the initial conditions chosen, the only part of the algorithm affected will be the transient part. As more data is processed the initial conditions fade eventually reaching a steady state value. By choosing a large prediction covariance more emphasis will be put on the measurements and less on the model in the transient phase. The second major factor affecting the Kalman filter performance is the model type. Since the model is used to generate the estimated states it should be as close to the physical phenomenon as possible. If the type of model chosen

for a particular process is correct with only time constants slightly off, some degeneration will occur. On the other hand, if the system is modeled incorrectly, a model mismatch will result. A model mismatch can cause the estimate states to diverge from the actual states. It can be seen through the preceding discussion of the filter parameters and the choice of the model, the importance of correct modeling in the achievement of optimal performance from the Kalman filter [Ref. 8].

IV. FIXED INTERVAL OPTIMAL SMOOTHER

The Rauch-Tung-Striebel fixed interval optimal smoother was designed to be a post processing algorithm to be used in conjunction with a Kalman filter. This algorithm will improve the results obtained from the Kalman filter by utilizing the future information not available during the Kalman filtering process. The fixed interval optimal smoothing algorithm recalculates each estimate generated from the Kalman filter based on the information obtained for the entire set of data analyzed. This procedure generates what is called the smoothed estimates as seen in the block diagram of Figure 9, where the term *ADV* represents one discrete-time advance. The other terms are discussed in Equations (17) and (18).

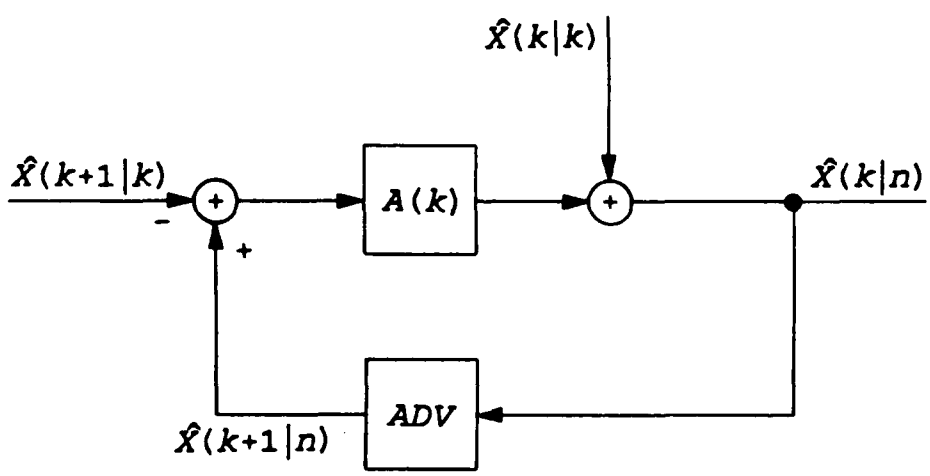

Figure 9. Fixed Interval Optimal Smoother Block Diagram

The Rauch-Tung-Striebel fixed interval optimal smoothing algorithm was derived in Reference 5. Equation (17) calculates a weighting factor:

$$A(k) = P(k|k)\, F^T P(K+1|k)^{-1} \tag{17}$$

where:

$A(k)$-is the smoothing algorithm gain,

F -is the state transitional matrix from the Kalman filter,

$P(k|k)$ -is the prediction covariance from the Kalman filter, and

$P(K+1|k)$ -is the conditional prediction covariance from the Kalman filter.

This weighting factor does not depend on past gains, just the conditional prediction error covariance and the previous prediction error covariance from the Kalman filter for a particular discrete-time. If more uncertainty exists in the forward filter, the weighting factor becomes larger. Equation (18) is the second equation involved in this algorithm which calculates the smoothed estimates:

$$\hat{X}(k|n) = \hat{X}(k|k) + A(k)\,[\hat{X}(k+1|n) - \hat{X}(k+1|k)] \tag{18}$$

where:

n -is the final time,

$\hat{X}(k|n)$ -is the smoothed state estimate,

$\hat{X}(k|k)$ -is the state estimate from the Kalman filter,

$\hat{X}(k+1|n)$ -is the previous smoothed state estimate, and

$\hat{X}(k+1|k)$ -is the conditional state estimate from the Kalman filter.

The next equation is not involved in the algorithm, but is useful in determining how well the smoothing is being accomplished.

$$P(k|n) = P(k|k) + A(k)\,[P(k+1|n) - P(k+1|k)]\,A(k)^T \qquad (19)$$

where:

$P(k|n)$ -is the smoothed state error covariance

$P(k+1|n)$ -is the previous smooth state error covariance

These equations, excluding Equation (19), are used recursively in the discussed order to obtain the smoothed estimates.

It is seen from the preceding equations that this algorithm has no performance parameter that can be adjusted. Consequently, it depends solely on the accuracy of the Kalman filter's parameters. The gain in Equation (17) is dependent on the covariance error of the previous and conditional values. This gain is then applied to the difference between the smooth estimate calculated for the previous data point and the

corresponding predicted value generated by the Kalman filter. The Kalman filter estimate is then adjusted by this weighted difference as in Equation (18) [Ref. 9]. This means that the predicted states, corrected states, predicted covariance, and corrected covariance for each data point must be saved during the forward processing operation.

V. KALMAN SMOOTHER APPLIED TO LACE

The Doppler separation history for the LACE satellite is determined by utilizing the Kalman filter and the Rauch-Tung-Striebel fixed interval optimal smoother. Numerous preliminary steps were required to extract the Doppler separation from each pulse and ultimately obtain the Doppler separation history. The data first had to be read from the tape and converted into data files to be loaded into the Matlab environment for processing [Ref. 10]. An algorithm had to be developed to perform the preliminary processing of the data as discussed in the second chapter. The Kalman filter and the Rauch-Tung-Striebel fixed interval optimal smoother algorithms had to be implemented. A dynamic tracking window controlled by the Kalman filter had to be designed to track the desired frequency contained in the power spectrum for each radar pulse. Bad data had to be identified and eliminated during the processing phase. Adequate plotting routines had to be developed so that a good visual pulse-by-pulse analysis of selected data could be performed to ensure that the results were, in fact, what was expected. The following paragraphs discuss the previously-mentioned items in more detail with respect to the subroutines that were developed to perform these tasks.

The data received for 13 and 18 July 1990 consisted of approximately 9200 records per tape (120 seconds of data in binary form). Only 4500 records from each tape were processed in order to retain continuity with respect to the amount of data processed with the histogram filtering based method. Each record represents one radar pulse, which is 3.4 milliseconds of IQ data digitized at 1.2 MHz. Subroutine DFSIOC found in the appendix, performs the task of extracting this information from the tape and coordinating the other subroutines to process the data. This subroutine enables radar pulses or records to be read from any specified run time to any other specified run time by adjusting the skip and count on the tape drive control line (!rsh srv1 "dd if=/dev/nrmto ibs=16384 count=100 skip=0 > bin.dat).

Subroutine BIN2INT found in the appendix was developed using Fortran [Ref. 11] to convert the binary in-phase, quadrature, and timing information from the tape into an integer format capable of being loaded into the Matlab environment. Fortran was used in this application because Matlab does not have this conversion capability. The binary data is converted by BIN2INT, one record (16384 bytes or characters) at a time, into an integer format. Of the 16384 characters, 1 to 16360 characters contained the IQ data for the radar pulse, 16361 to 16367 characters contained the timing information for the radar pulse, and the remaining 18 characters are not used. The storage of the IQ data on the

tape alternated between a byte of in-phase and a byte of quadrature data. The timing information was stored in consecutive nibbles. The IQ information that subroutine BIN2INT converted is then stored in a data file called IQ_DAT.dat and the timing information is stored in a data file called TimDat.dat. These data files are then sequentially loaded into the Matlab environment.

Once the IQ data was converted into integer format and loaded into the Matlab environment, the next step was to process the data. The first, second, and third steps in subroutine DFSCAL, found in the appendix, performed the procedure discussed in the second chapter. These steps consisted of adding together the squared in-phase and squared quadrature components as in Equation (4) to produce the IQ envelope. The IQ envelope of a sample radar pulse is plotted in Figure 3. The conversion from the time domain to the frequency domain is accomplished by taking the fast Fourier transform of the IQ envelope and then calculating the power spectral density as in Equations (5) and (6). The results are shown in Figure 4 for the sample radar pulse depicted in Figure 3.

The next step involved tracking the particular frequency of interest (the Doppler separation) contained in the power spectrum for each radar pulse. A dynamic tracking window controlled by a Kalman filter was designed to perform this function. In the development of the dynamic window, two

aspects had to be resolved: location and size. The location of the window was determined by centering it around the Doppler separation estimate obtained from the Kalman filter. The size of the window had to be large enough to track the change in frequency, but small enough to reject unwanted frequencies. Tracking these unwanted frequencies would cause the window's location to drift from the desired frequency. Window size control was achieved by setting the window's size equal to the prediction state error covariance from the Kalman filter plus two times the frequency resolution of the fast Fourier transform. The use of the prediction state error covariance plus two times the frequency resolution was empirically determined to be the parameter that produced the best results. The Doppler separation is then determined by obtaining the corresponding frequency of the peak magnitude within the window. This corresponding frequency of the peak magnitude is entered in Equation (12) of the Kalman filter as the measurement. The measurement is processed with the Kalman filter as discussed in the third chapter.

The first obstacle encountered in processing the data was the clutter mentioned in References 1 and 2. This clutter or bad data can cause the tracking mechanism in subroutine DFSCAL to drift from the actual Doppler separation track if not addressed. Figure 10 was obtained by plotting the peak

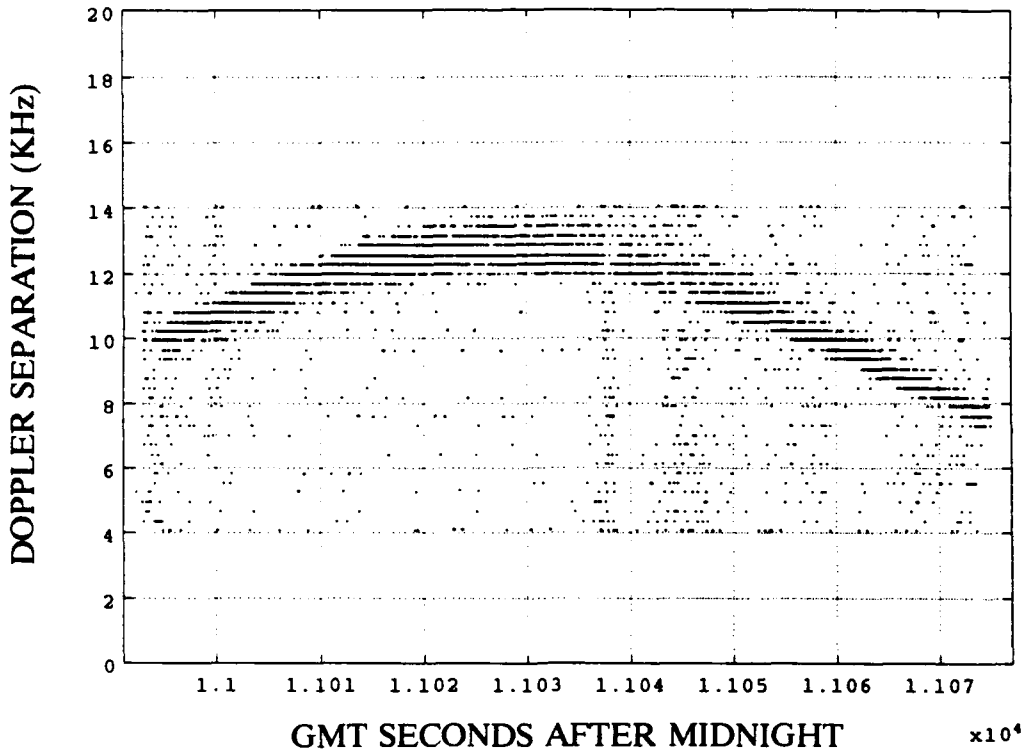

Figure 10. Fixed Window, Unfiltered Data, GMT DAY 195

magnitude within a fixed window from 4 kHz to 14 kHz. Each dot in this figure represents one radar return that could contain either the accurate Doppler separation information or the apparent clutter. In analyzing the power spectrums of selected radar pulses there exists three distinct types of radar returns: ones that contained accurate Doppler separation information; ones that contained accurate Doppler separation information corrupted by noise; and ones that are just noise or an extremely weak signal embedded in noise. The IQ envelope of a sample radar return that contains accurate Doppler separation information is depicted in Figure 11. Figure 12 shows the power spectrum of this radar return and the dynamic tracking window. The dashed lines represent the window. The

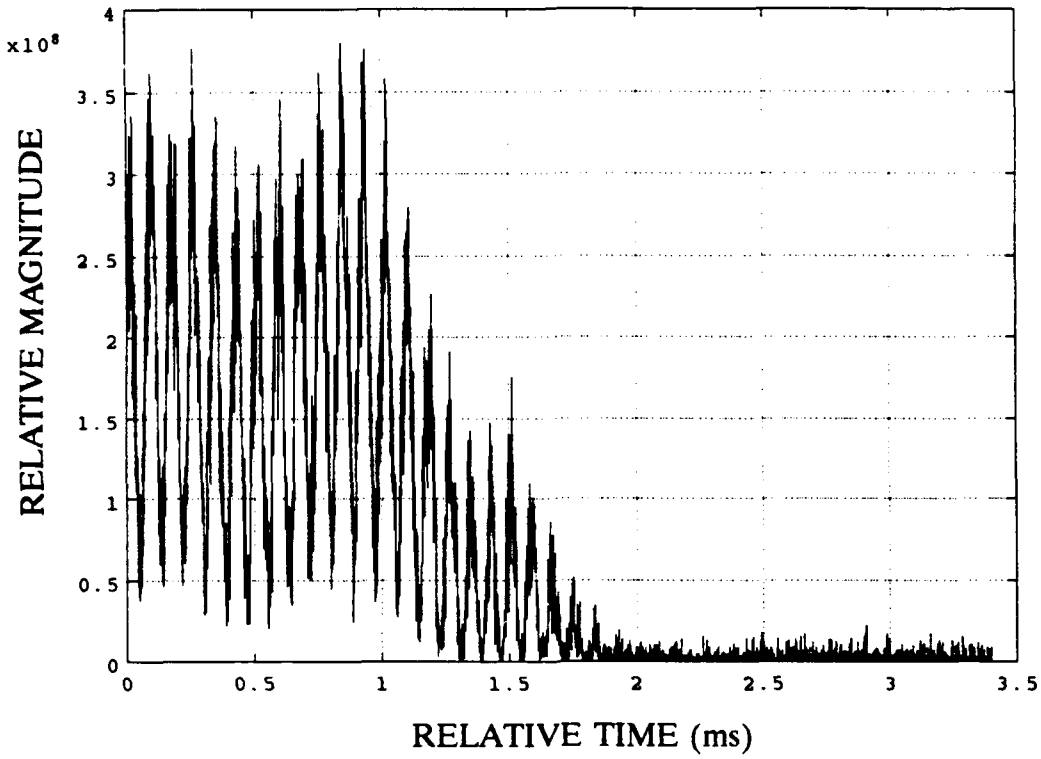

Figure 11. Accurate IQ Envelope Information

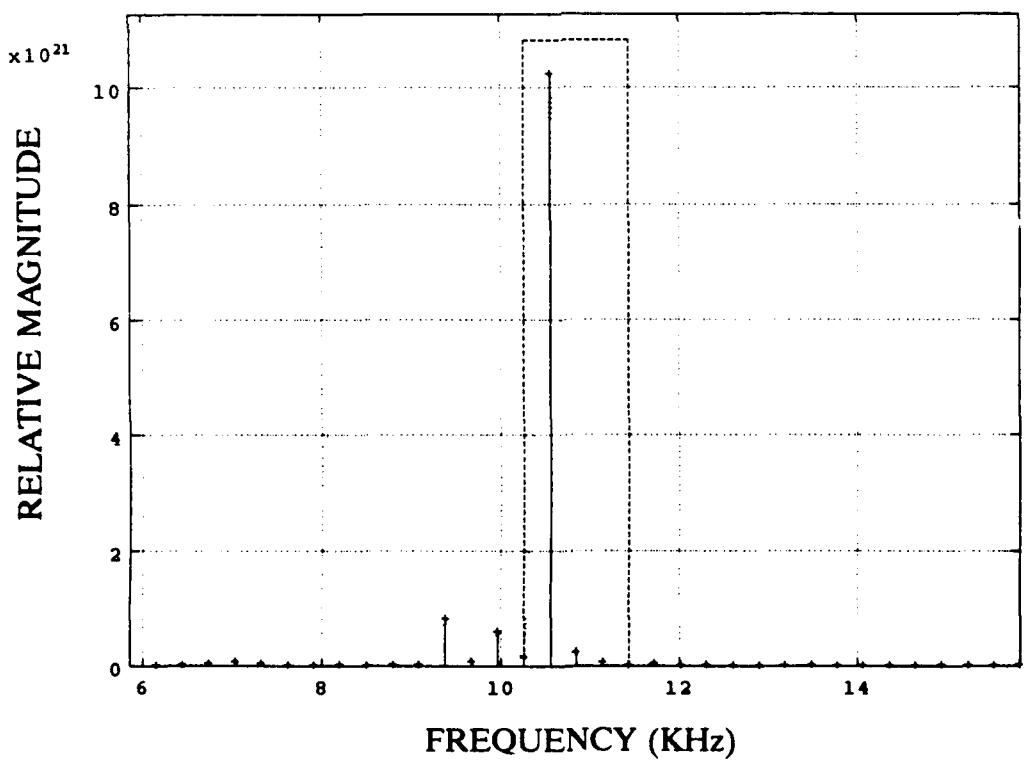

Figure 12. Accurate Power Spectrum Information

solid lines indicate the power spectrum of the IQ envelope. The dotted line traces the outline of the power spectrum. The power spectral density being centered around a single frequency in Figure 12 is a good indication that this radar return contains accurate Doppler separation information. A sample radar return which contains accurate Doppler separation information corrupted by noise is represented by Figure 13 and its power spectrum is shown in Figure 14. In this case, the Kalman filter's filtering mechanism will eliminate any large deviations from occurring in the Doppler separation history. The third type of radar return is when there was just noise or an extremely weak signal embedded in noise. This type is represented in Figure 15. The power spectrum of this indicates the presence of noise as seen in Figure 16. Comparing Figure 16 to Figure 14, a difference in magnitudes is revealed. This type of superfluous data is eliminated by placing a magnitude constraint on the particular frequency of interest within the dynamic window. The incorporation of the conditional statement (if MAGD(K+1)<MAGD(K)+1e2| MAGD(K+1)>MAGD(K)-1e2) into subroutine DFSCAL performs this task. This conditional statement differentiates between the magnitude of the last good radar return and the present radar return to determine if the relative magnitude is within a fixed distance of the last radar return. If this condition is met, the Kalman filter algorithm is used as described in the third chapter. On the

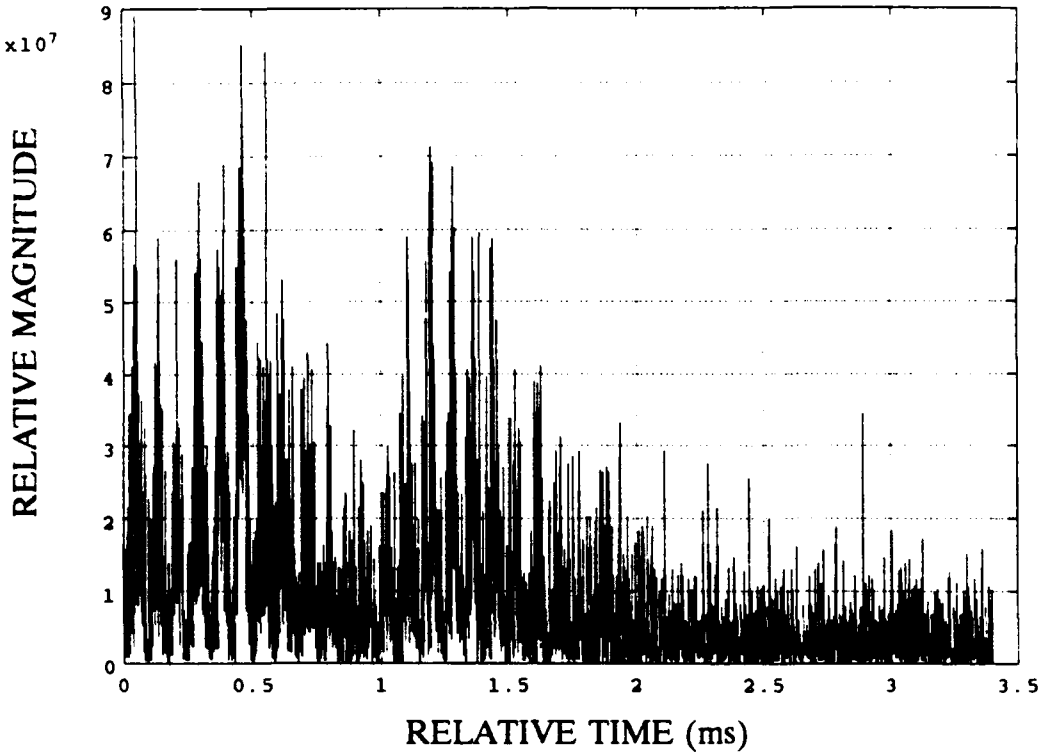

Figure 13. IQ Envelope Corrupted by Noise

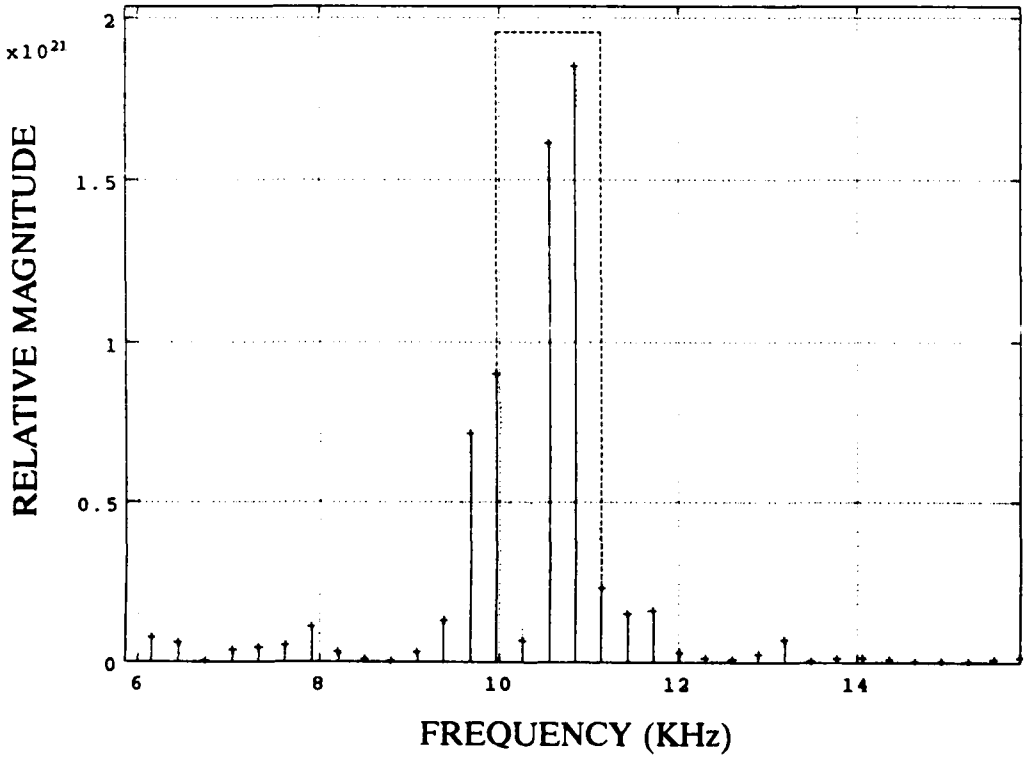

Figure 14. Power Spectrum Corrupted by Noise

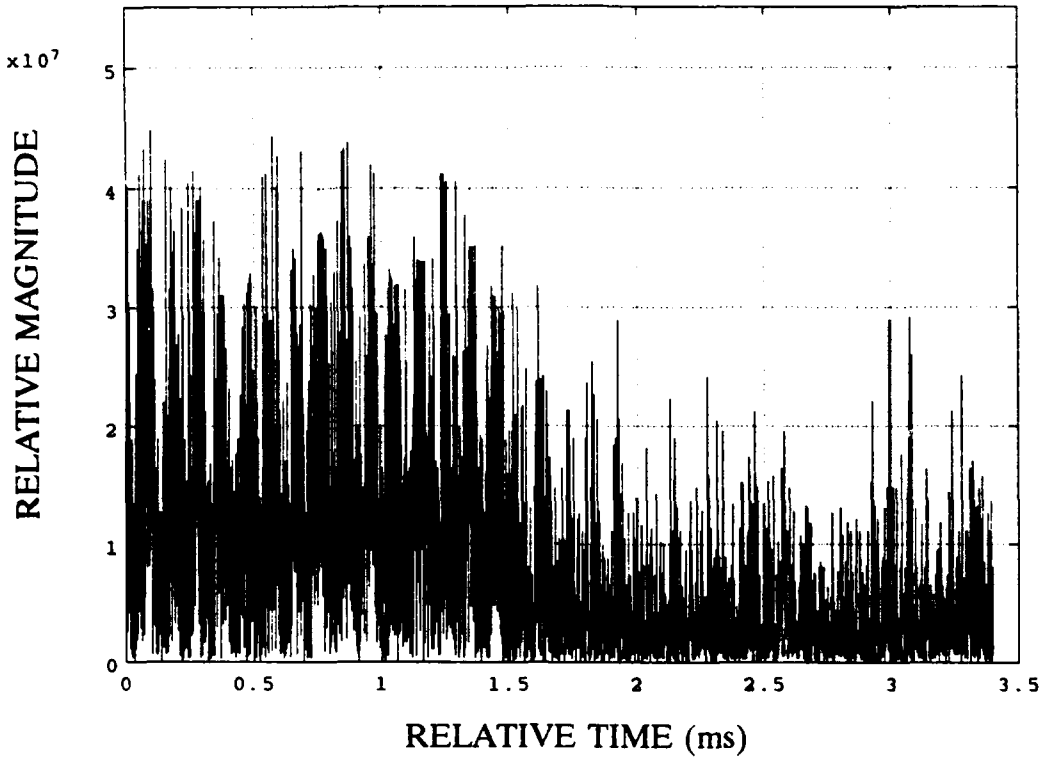

Figure 15. Amplified Noise in Time Domain

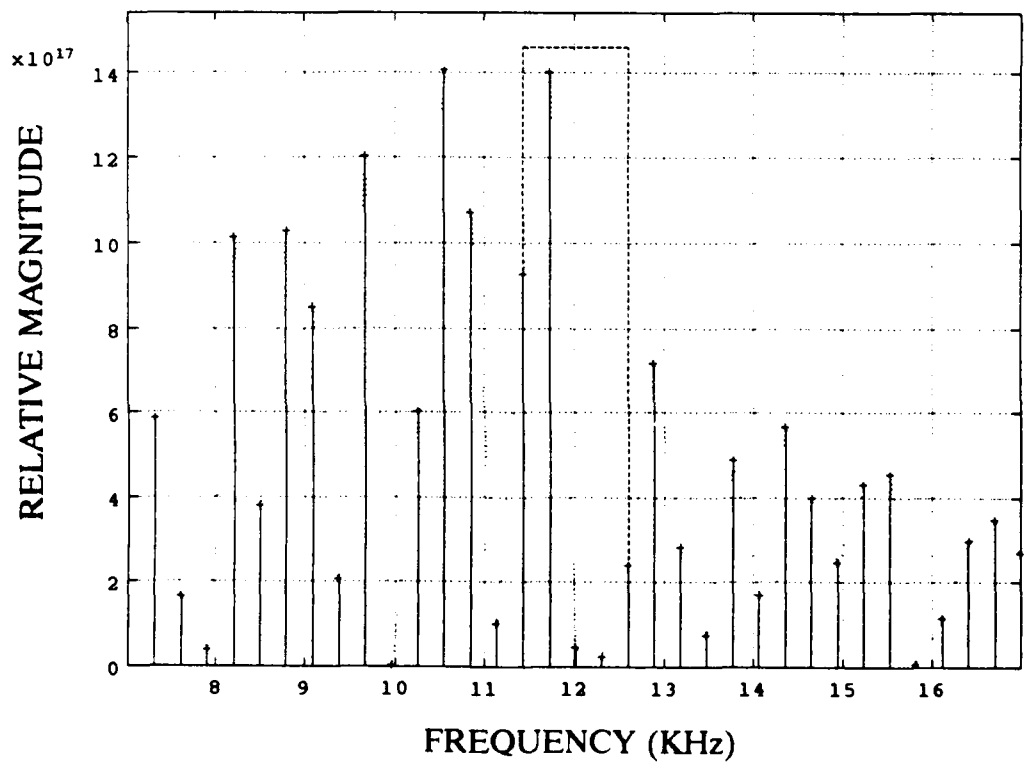

Figure 16. Amplified Noise in Frequency Domain

other hand, if this condition is not met, then the estimate of the Doppler separation frequency is not updated and the prediction covariance is increased by setting it equal to the conditional predicted state error covariance. This will cause the dynamic window to increase in size and place more emphasis on the next valid measurement within the window.

The next obstacle encountered in the implementation of the Kalman filter, was determining the type of model best suited for the data. In developing the model for the Kalman filter, the histogram-based tracks for 13 and 18 July in the first chapter were analyzed to determined if a differential equation could be applied to the trajectory of the LACE satellite. After examining these figures, the development of an elaborate model to simulate the parabolic trajectory of the LACE satellite was virtually impossible to realize. The reason was due to the different elevations and azimuths the satellite could incur each time it was tracked. This leads to the slightly different trajectories as noted earlier with Figures 5 and 6. A very simple model approach was then used. Adequate results were obtained using a first order model whose state is the Doppler separation in the frequency domain. Having the state of the model be a scalar quantity greatly simplifies the algorithm. To determine the value for the state transition matrix F, the parabolic nature of the histogram based tracks were taken into account. This meant that a frequency increase would occur at the start of a tracking run. After a maximum

value was reached during the tracking run, there would occur a frequency decrease. The only eigenvalue for the F that would not cause degeneration to occur, is the value one. If an eigenvalue greater or less than unity is used, the estimated frequency will diverge either for the increasing Doppler or the decreasing Doppler, depending on which value was chosen.

When the preliminary processing of the data and the Kalman filtering had been accomplished, the Doppler separation estimates were ready to be smoothed. Subroutine DFSBFS found in the appendix was developed to perform this task. This subroutine implements the Rauch-Tung-Striebel fixed interval optimal smoother equations as discussed in the fourth chapter. Subroutine PLOT4 found in the appendix is utilized to plot the results obtained from this subroutine.

The final step was to determine the value for the measurement noise covariance (R), the process noise covariance (Q), and the initial conditions that would optimize the processing algorithm. The initial conditions consisted of the prediction estimate covariance ($P(0)$), the Doppler separation estimate ($XFREQ(0)$), and the relative magnitude of the Doppler separation ($MAGD(0)$). Subroutine DFSVAL found in the appendix is used to initialize the parameters for all the subroutines. The initial Doppler separation frequency is determined by the point at which the analysis is started for a particular tracking run. The first radar return to be processed for 18 July 1990 was depicted in Figure 3. The frequency of 12.012

kHz obtained from this figure is used to initialize the Doppler separation estimate for the filter. The prediction estimate covariance was set to 292.969 Hz, since this was the frequency resolution obtained. The initial magnitude is determined to be 100 by checking the beginning data. The value for the process noise covariance (Q) was varied from 292.969 Hz to 7.324 Hz. The measurement noise covariance (R) was varied between 292.969 Hz and 2929.690 Hz. In empirically obtaining the optimal values for the process noise covariance and the measurement noise covariance, they were initially set at 292.969 Hz. By setting the process noise covariance and the measurement noise covariance to this value, the Kalman filter tracked every deviation no matter how off track they were. Figure 17 shows this undesirable result. The process noise covariance was when reduced to 146.485 Hz with the measurement noise covariance still set at 292.969 Hz in an attempt to gain more filtering from the Kalman filter. This showed some improvement in the noise rejection performance of the filter as indicated by Figure 18. Increasing the measurement noise ten times to 2929.960 Hz with the process noise covariance left at 146.485 Hz vastly improved the performance of the filter, which is seen in Figure 19. Based upon the previous results, the process noise covariance was further reduced to 7.324 Hz leaving the measurement noise covariance at 2929.690 Hz. A good track of the Doppler separation history was

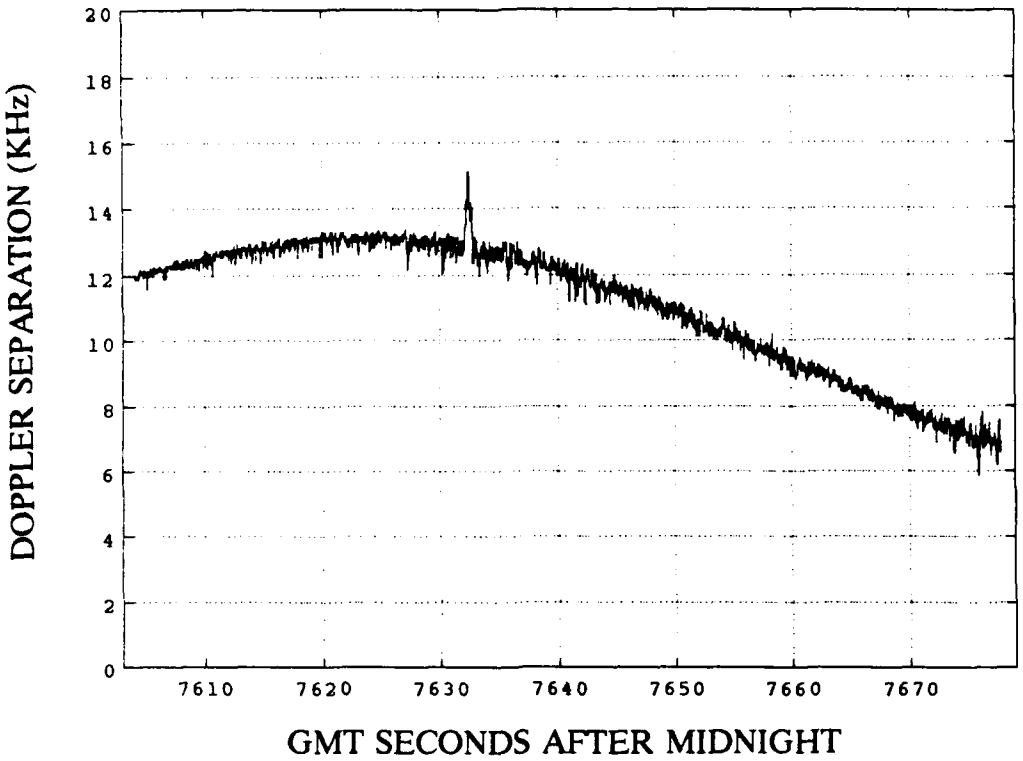

Figure 17. Doppler Separation, GMT DAY 200, Q = 292.969, R = 292.969

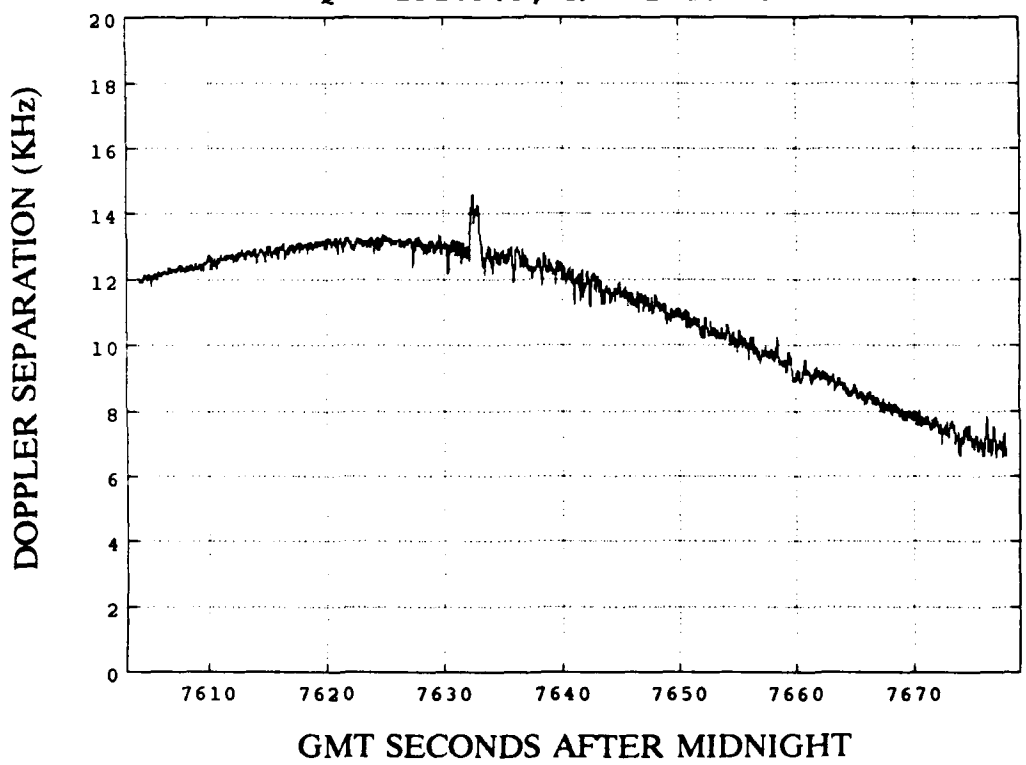

Figure 18. Doppler Separation, GMT DAY 200, Q = 146.485, R = 292.969

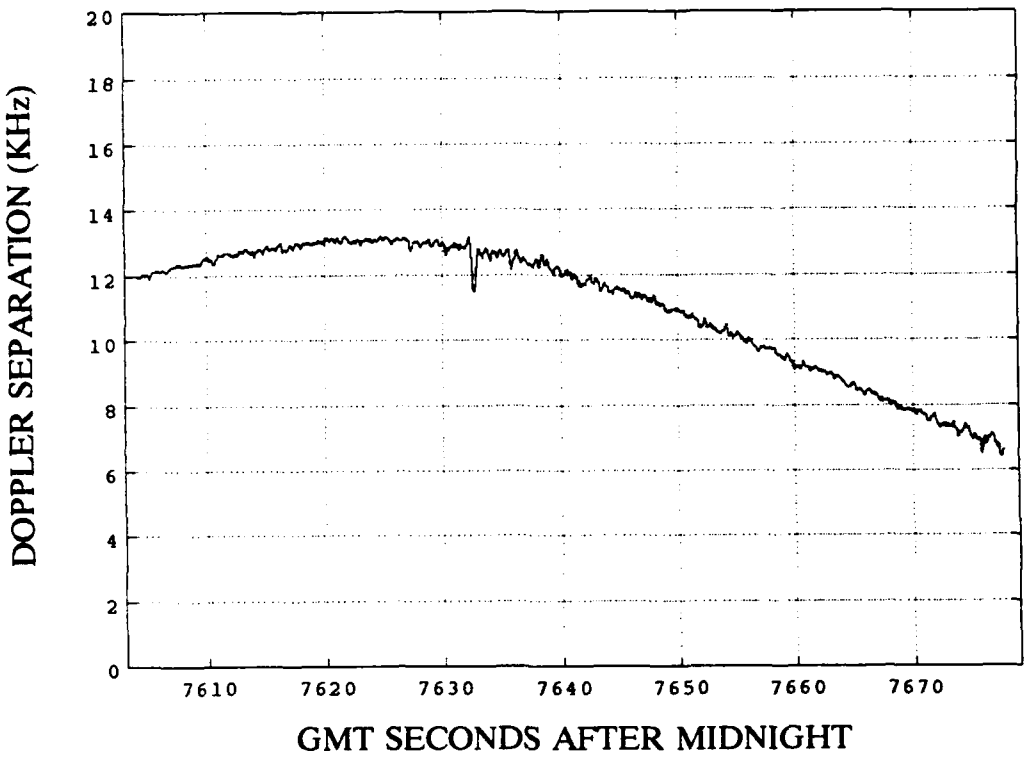

Figure 19. Doppler Separation, GMT DAY 200, Q = 146.485, R = 2929.690

obtained as indicated in Figure 20 for 18 July 1990. Figure 21 shows the Doppler separation history for 13 July 1990.

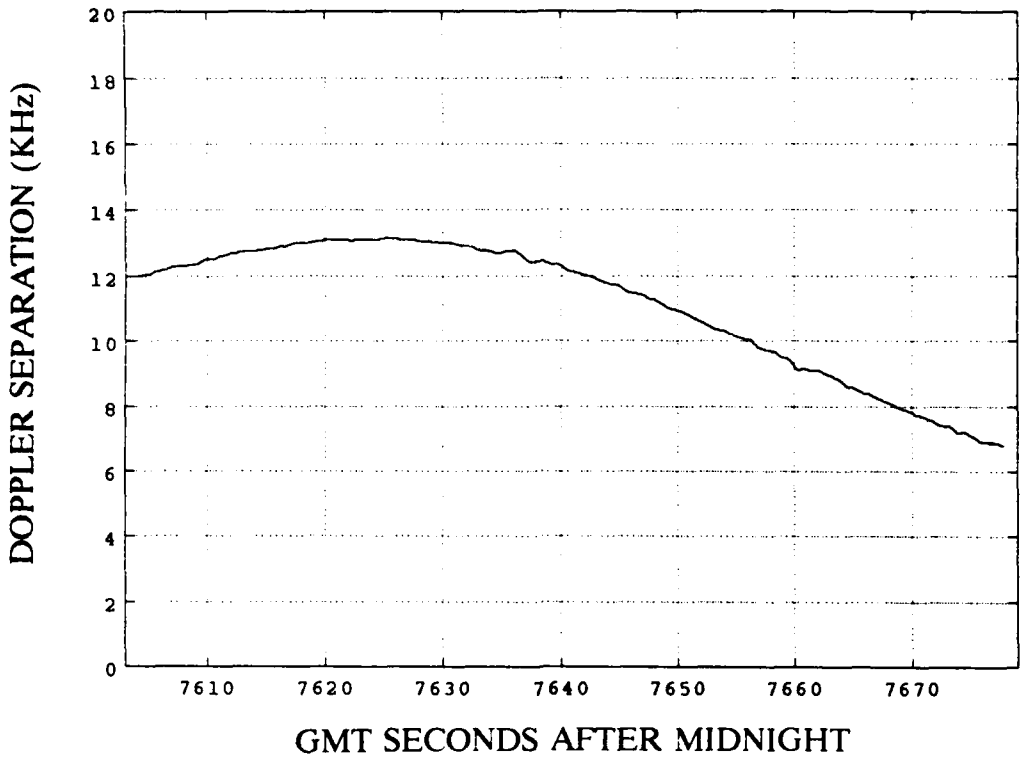

Figure 20. Doppler Separation, GMT DAY 200, Q = 7.324, R = 2929.690

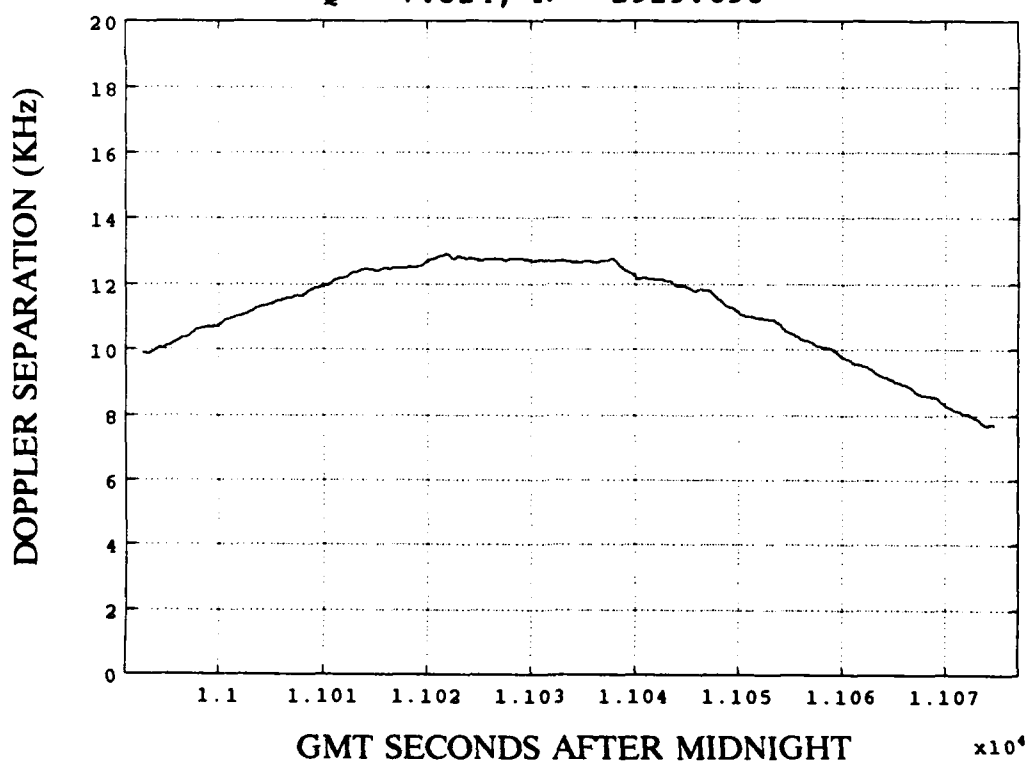

Figure 21. Doppler Separation, GMT DAY 195, Q = 7.324, R = 2929.690

VI. CONCLUSION

The results obtained in the previous chapter have demonstrated the effectiveness of the Kalman filter in conjunction with the Rauch-Tung-Striebel fixed interval optimal smoother as a post processing utility in determining the Doppler separation history from the data. To compare the effectiveness of the two filtering-based techniques, Figure 22 for 13 July 1990 and Figure 23 for 18 July 1990, were developed. Figure 22 is the combination of Figures 5 and 21, where Figure 23 is the combination of Figures 6 and 20.

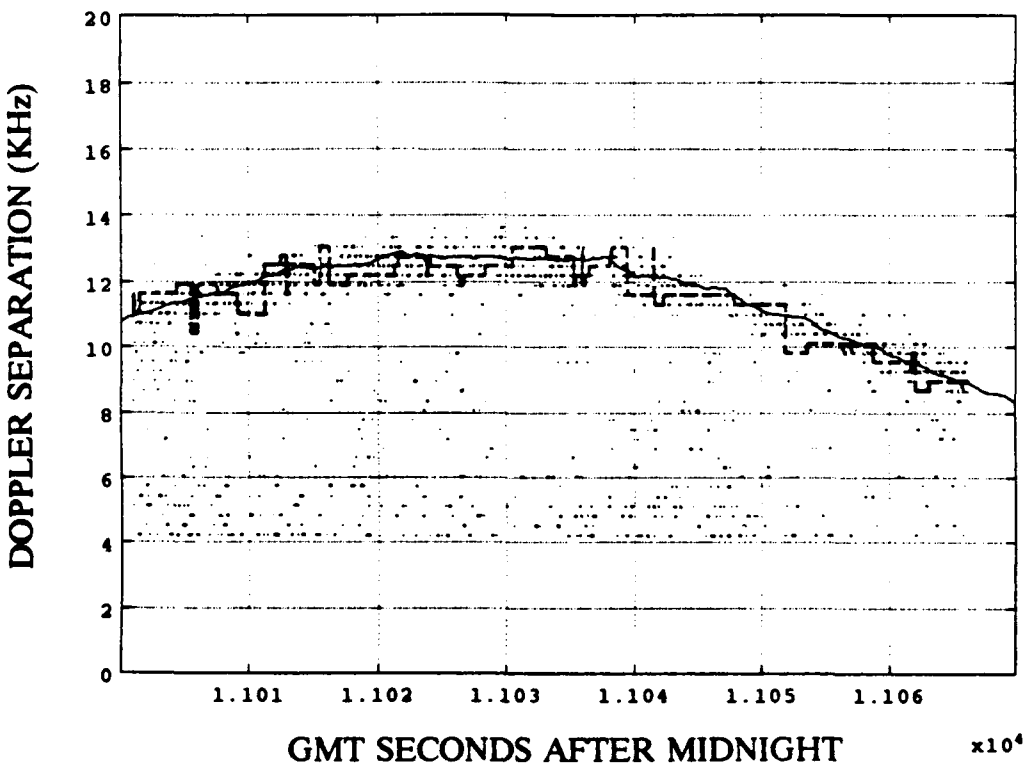

Figure 22. Doppler Separation Composite, GMT DAY 195 [Fig 7a from Ref. 2]

In both figures, the dashed lines are the results obtained from the histogram filtering-based method, the solid lines are for the method utilized in this thesis, and the dots represent the unfiltered data. Analysis of these figures demonstrates that the method used in this thesis of determining the Doppler separation history for both days is far superior to that of the histogram filtering based technique.

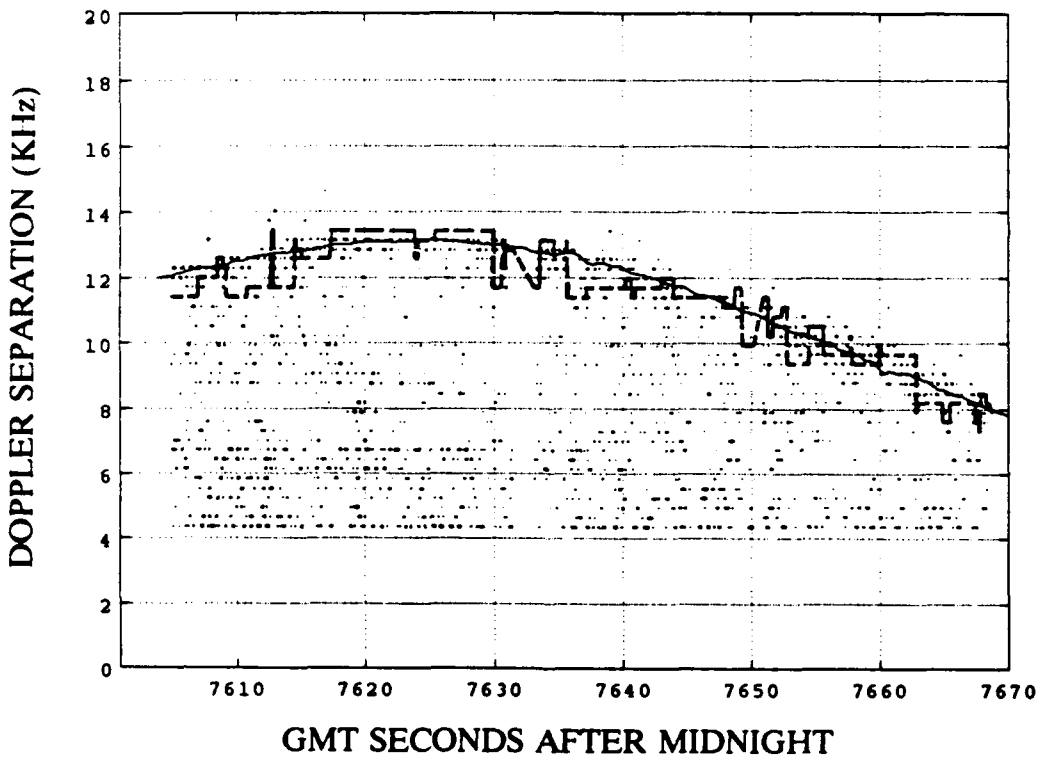

Figure 23. Doppler Separation Composite, GMT DAY 200 [Fig 7b after REf. 2]

APPENDIX
SUBROUTINES DEVELOPED TO PROCESS THE DATA

The following subroutines are designed to function together with subroutine DFSIOC as the primary subroutine. The data is read from the tape in equally-sized blocks that are consecutively processed. The block sizes can vary from 1 to whatever size the computer is capable of handling. The number of blocks or groups can be varied from 1 to the amount of data to be processed. To use this package, the following must reflect the same values: the number of groups (GRNO) in subroutines DFSIOC and DFSVAL; the terminal count in the main loop of BIN2INT and the number of records per group (PSNO) in DFSVAL; the count in the tape drive control line of DFSIOC and the number of records per group (PSNO) in DFSVAL. These subroutines at present are set up to process 4500 records in groups of 100 (GRNO=45, PSNO=100).

- DFSIOC

To initiate this subroutine to process the data, just enter DFSIOC in the Matlab environment. DFSIOC will perform the task of extracting the information from the tape and coordinating subroutines DFSVAL, BIN2INT, DFSCAL, DFSBFS, PLOT3, and PLOT4. The data extracted from the tape is stored in a temp file called Bin.dat. It loads the data that was converted (binary to integer) from the temp file (Bin.dat) by BIN2INT into the Matlab environment for further processing. PSNO in this subroutine must reflect the same end count as PSNO in subroutine DFSVAL. In the tape drive control line, count is the number of records to be read and skip is the number of records to be passed over. Count must reflect the same value as PSNO in subroutine DFSVAL. The other options available are: to plot the data using PLOT3 and PLOT4; to save the data with the save function.

```
%                              SUBROUTINE DFSIOC
 dfsval;                                          %CALLS SUBROUTINE DFSVAL
 for GRNO=1:1:45                                  %MAIN LOOP TO PROCESS DATA
    !rsh srv1 "dd if=/dev/nrmt0 ibs=16384         %TAPE DRIVE CONTROL (READS)
     count=100 skip=0" >bin.dat
    !bin2int                                      %CALLS SUBROUTINE BIN2INT
    !rm /staff/thorngre/bin.dat                   %CLEARS TEMPORARY DATA FILE
    load IQ_DAT.dat                               %LOADS IQ DATA FILE
    load TimDAT.dat                               %LOADS TIMING INFORMATION
    TimDAT1=[TimDAT1;TimDAT(:,6)];                %SAVES TIMING INFORMATION
    DAY=TimDAT(1,1);                              %SAVES GMT DAY OF TRACK
    dfscal;                                       %CALLS SUBROUTINE DFSCAL
    clear IQ_DAT TimDAT                           %CLEARS UNNECESSARY DATA
 end                                              %END OF THE MAIN LOOP
 dfsbfs                                           %CALLS SUBROUTINE DFSBFS
 !rsh srv1 mt rw                                  %REWINDS TAPE TO BEGINING
 plot3;                                           %CALLS SUBROUTINE PLOT3-OPT
```

```
 plot4;                                          %CALLS SUBROUTINE PLOT4-OPT
%save DSF.dat  DSF -ascii                        %SAVES DATA IN ASCII FORMAT
%save XDSF.dat XMFREQ -ascii
%save BXDSF.dat BXDSF -ascii
%save TimDAT1.dat TimDAT1 -ascii
%save DAY.dat DAY -ascii
```

- DFSVAL

Declares and sets the parameters used in the subroutines DFSIOC, DFSCAL, PLOT1, PLOT2, PLOT3, and PLOT4. PSNO is the number of records in the group (GRNO). These must reflect the values used in DFSIOC and BIN2INT. Subroutine DFSVAL is called by subroutine DFSIOC.

```
%                          SUBROUTINE  DFSVAL
 GRNO=45;                                        %NO OF GROUPS OF PULSES
 PSNO=100;                                       %NO OF PULSES PER GROUP
 RECLN=8160;                                     %LENGTH OF DATA PER PULSE
 Freq=292.969.*(0:2047);                         %CREATES FREQUENCY VECTOR
 Time=(0:4079)/(1.2e6);                          %CREATES TIME VECTOR FOR IQ^2
 TimDAT1=[];                                     %UNDETERMINED TIMIG VECTOR
 G=zeros(PSNO*GRNO+1,1);                         %SETS UP KALMAN GAIN VECTOR
 P=zeros(PSNO*GRNO+1,1);                         %SETS UP COV PRED VECTOR
 PP=zeros(PSNO*GRNO+1,1);                        %SETS UP CON COV PRED VECTOR
 DSF=zeros(PSNO*GRNO+1,1);                       %SETS UP DSF FIXED WINDOW VET
 DSFW=zeros(PSNO*GRNO+1,1);                      %SETS UP DYNAMIC WINDOW VET
 XDSF=zeros(PSNO*GRNO+1,1);                      %SETS UP ESTIMATED DSF VET
 XMFREQ=zeros(PSNO*GRNO+1,1);                    %SETS UP CON EST STATE VET
 XFREQ=zeros(PSNO*GRNO+1,1);                     %SETS UP ESTIMATED STATE VET
 BXDSF=zeros(PSNO*GRNO+1,1);                     %SETS UP SMOOTH EST DSF
 MAGD=zeros(PSNO*GRNO+1,1);                      %SETS UP MAG VECTOR FOR D.W.
 DAY=(1,1);                                      %GMT DAY OF TRACKING RUN
 magfi=(1,1);                                    %MAX MAG IN FIXED WINDOW
 freqfi=(1,1);                                   %INDEX OF MAX FREQ IN F.W.
 magdi=(1,1);                                    %MAX MAG IN DYNAMIC WINDOW
 freqdi=(1,1);                                   %INDEX OF MAX FREQ IN D.W.
 freqli=(1,1);                                   %INDEX OF LOWER LIMIT D.W.
 frequi=(1,1);                                   %INDEX OF UPPER LIMIT D.W.
 F=[1];                                          %TRANSITIONAL STATE MATRIX
 G=[1];                                          %INPUT WEIGHTING FACTOR
 D=[1];                                          %INPUT WEIGHTING FACTOR
 C=[1];                                          %OUTPUT WEIGHTING FACTOR
 Q=[292.969*0.025];                              %PROCESS NOISE COVARIANCE
 R=[292.969*10];                                 %MEASUREMENT NOISE COVARIANCE
 P(1)=[292.969*1];                               %INITIAL COV PRED ESTIMATE
 MAGD(1)=[2.5e21];                               %INITIAL MAGNITUDE OF PSD
 XFREQ(1)=[292.969*41];                          %INITIAL FREQUENCY ESTIMATE
```

- BIN2INT

Has been written in Fortran to convert the binary data obtained from the tape into integer format. The input format is in binary byte form from the temp file Bin.dat. There are

16384 bytes of data in a record of which 16360 bytes are inphase and quadrature data, 6 bytes are the timing information, and the rest are not used. There are no delimiters between values. The output format consists of I6 format for the data and is stored in IQ_DAT.dat. The output for the timing information is F4.4 format and is stored in TimDAT.dat. The main loop in this subroutine must specify the numbers of records that are going to be processed per group (PSNO) in subroutine DFSVAL. BIN2INT at present is set up to process 100 records. This subroutine is called by subroutine DFSIOC.

```
c                                SUBROUTINE BIN2INT
c       DECLARES VARIABLES
        character*1 cdat(16384)
        integer*2 jdat(8192)
        integer*2 nibbles(0:11)
        integer*2 tcdat(0:2)
        real day,hour,min,sec,mil,TIME
        equivalence (cdat,jdat)
c       OPENS THE DATA FILE CALLED 'BIN.DAT' FOR DIRECT, UNFORMATTED READ
c       AND OUTPUTS TWO FILES CALLED 'IQ_DAT.DAT' AND 'TIMDAT.DAT'.
        open(1,file='/staff/thorngre/bin.dat', access='direct',
     &  recl=16384,form='unformatted')
        open(2,file='IQ_DAT.dat')
        open(3,file='TimDAT.dat')
c       LOOP TO READ DATA INFORMATION AND OUTPUT IT TO A FILE
 15     do 60, i=1,100
        read(1, rec=i)cdat
           do 20, j=1,8192
              write(2,1020)jdat(j)
 1020         format(i6)
 20           continue
c       LOOP TO READ TIMING INFORMATION
           do 30, k=0,2
              tcdat(k)=jdat(8181+k)
 30        continue
c       LOOP TO CALCULATE TIME FROM TIMING INFORMATION AND OUTPUT IT
c       TO A FILE
           do 40 l=0,11,4
              nibbles(l+0)=and(rshift(tcdat(l/4),12),15)
              nibbles(l+1)=and(rshift(tcdat(l/4),8),15)
              nibbles(l+2)=and(rshift(tcdat(l/4),4),15)
              nibbles(l+3)=and(tcdat(l/4),15)
 40        continue
           day=nibbles(0)*100+nibbles(1)*10+nibbles(2)
           hour=nibbles(3)*10+nibbles(4)
           min=nibbles(5)*10+nibbles(6)
           sec=nibbles(7)*10+nibbles(8)
           mil=nibbles(9)*100+nibbles(10)*10+nibbles(11)
           time=60*60*hour+60*min+sec+(mil/1000)
           write(3,1030)day,hour,min,sec,mil,time
 1030      format(f4,1x,f4,1x,f4,1x,f4,1x,f4,1x,f10.4)
 60     continue
        end
```

- DFSCAL

Processes records that contain the inphase and quadrature data to obtain the IQ envelope, power spectrum of the IQ envelope, and the estimated Doppler separation history. The Kalman Filter algorithm is implemented to perform two functions. The first is to provide control for the dynamic window's location and size. The second is to provide a good estimation of the Doppler separation from the measured data. The conditional statement is used to check the magnitude of the data. The fixed window in this subroutine provides an observation of the unfiltered data. The options available in DFSCAL are to view IQ envelope and the power spectrum of the IQ envelope subroutine PLOT1 or subroutine PLOT2. This subroutine is called by DFSIOC.

```
%                                SUBROUTINE  DFSCAL
 for K=(GRNO*PSNO-PSNO+1):1:(GRNO*PSNO)          %LOOP TO CALCULATE DSF/PULSE
    IQS_NO=1;                                    %INITIAL COUNTER FOR IQ_DAT
    for k=1:2:RECLN                              %LOOP TO CALCULATE IQ^2
       IQS_DAT(IQS_NO)=(IQ_DAT(k+(IQS_NO-1) ...
                       *8192))^2+(IQ_DAT(k+ ...
                       1+(IQS_NO-1)*8192))^2;
       IQS_NO=IQS_NO+1;
    end                                          %END OF LOOP TO CAL IQ^2
    IQS_FFT=fft(IQS_DAT,4096);                   %CALCULATES FFT OF IQ^2
    IQS_PSD=IQS_FFT.*conj(IQS_FFT);              %CALCULATES POWER SPECTRUM
    [magfi freqfi]=max(IQS_PSD(15:49));          %FIXED WINDOW 4KHz AT 14KHz
    freqfi=freqfi+14;                            %INDEX CORRECTION
    DSF(K)=Freq(freqfi);                         %DETERMINE DSF FROM INDEX
    XMFREQ(K+1)=F*XFREQ(K);                      %UPDATES ESTIMATED STATE
    XDSF(K)=C*XMFREQ(K+1);                       %CALCULATE ESTIMATED DSF
    PP(K)=(F*P(K)*F')+(G1*Q*G1');                %UPDATE CON PRED COVARIANCE
    G(K)=PP(K)*C'*inv(C*(PP(K)*C'+D*R*D'));      %CALCULATE KALMAN FILTER GAIN
    freqli=round((XDSF(K)-PP(K))/292.969);       %LOWER INDEX FOR WINDOW
    frequi=round(((XDSF(K)+PP(K))/292.969)+2);   %UPPER INDEX FOR WINDOW
    [magdi freqdi]=max(IQS_PSD(freqli...         %INDEX MAX MAG WITHIN WINDOW
                   :frequi));
    freqdi=freqli+freqdi-1;                      %INDEX CORRECTION
    DSFW(K)=Freq(freqdi);                        %DETERMINES FREQ FROM INDEX
    MAGD(K+1)=magdi;                             %SAVES MAG FOR EACH LOOP
    if MAGD(K+1)<MAGD(K)+1e2 |...                %CONDITIONAL TEST FOR PULSE
       MAGD(K+1)>MAGD(K)-1e2,                    %PULSE CONTAINS USEFUL DATA
       XFREQ(K+1)=XMFREQ(K+1)+G(K) ...           %UPDATES STATE
                 *(DSFW(K)-XDSF(K));
       P(K+1)=(eye(1)-G(K)*C)*PP(K);             %UPDATES COVARIANCE PREDICTION
    else                                         %PULSE HAS NO USEFUL DATA
       XFREQ(K+1)=XMFREQ(K+1);                   %STATE EQUALS CON EST STATE
       P(K+1)=PP(K);                             %SET COV PRED TO CON COV PRED
       DSFW(K)=0;                                %NO MEASUREMENT OBTAINED
    end                                          %END OF CONDITIONAL STATEMENT
%   plot1                                        %CALLS SUBROUTINE PLOT1-OPT
%   plot2                                        %CALLS SUBROUTINE PLOT2-OPT
 end                                             %END OF LOOP TO CALCULATE DSF
```

- DFSBFS

Was developed based on the Rauch-Tung-Striebel fixed-interval optimal smoother algorithm to provide a smoothed estimate Doppler separation history. This subroutine is called by DFSIOC.

```
%                                SUBROUTINE DFSBFS
 BXDSF(1:PSNO*GRNO+1)=XFREQ(1:PSNO*GRNO+1);      %INITIALIZES 1ST VALUE BDSF
 for n=PSNO*GRNO:-1:1;                           %LOOP TO CALCULATE BACKWARDS
   A=P(n+1)*F'*(inv(PP(n)));                     %CALCULATES SMOOTHING GAIN
   BXDSF(n)=XFREQ(n)+(A*(BXDSF(n+1) ...          %UPDATS SMOOTH ESTIMATE DSF
            -XMFREQ(n+1)));
 end                                             %END OF LOOP CAL BACKWARDS
```

- PLOT1

Plots the IQ envelope for the radar pulse to the monitor. The option in PLOT1 is to save the plot by using the meta function. This subroutine is called by DFSCAL.

```
%                                SUBROUTINE  PLOT1
 title1=(['SECONDS AFTER MIDNIGHT ',...          %TIMING INFORMATION HEADER
        num2str(TimDAT1(K,1))]);
 title2=(['GMT DAY ',num2str(DAY)]);             %TIMING INFORMATION HEADER
 plot(Time(1:4080),IQS_DAT(1:4080)),grid         %PLOTS IQ ENVELOPE DATA
 title(['LACE IQ ENVELOPE,',title2])             %PLOT TITLE
 text(1,1,title1,'sc');                          %PLOTS TIMING INFORMATION
 ylabel('RELATIVE MAGNITUDE'),                   %PLOTS Y-AXIS LABEL
 xlabel('TIME (ms)'),                            %PLOTS X-AXIS LABEL
%meta sxs1                                       %SAVES PLOT-OPTIONAL
```

- PLOT2

Plots the power spectrum of the IQ envelope and the dynamic window for the radar pulse to the monitor. The option in PLOT2 is to save the plot by using the meta function. This subroutine is called by DFSCAL.

```
%                                SUBROUTINE  PLOT2
 title1=(['SECONDS AFTER MIDNIGHT ',...          %TIMING INFORMATION HEADER
        num2str(TimDAT1(K,1))]);
 title2=(['GMT DAY ',num2str(DAY)]);             %TIMING INFORMATION HEADER
 plot(Freq(freqli-15:frequi+15),IQS_PSD ...      %PLOTS OUTLINE PWR SPECTRUM
     (freqli-15:frequi+15),':r'),hold on         %RETAINS PLOT IN MEMORY
 Y1GRA=0:magi:magi;Y11GRA=[magi magi];           %SETS UP DYNAMIC WINDOW VET
 X1GRA=Freq(freqli)*(ones(1:length(Y1GRA)));
 X2GRA=(Freq(frequi)*(ones(1:length(Y1GRA))))';
 Y2GRA=Freq(freqli):Freq(frequi)-Freq ...
       (freqli):Freq(frequi);
 plot(X1GRA,Y1GRA,'--g',X2GRA,Y1GRA,'--g'),      %PLOTS SIDES DYNANIC WINDOW
 plot(Y2GRA,Y11GRA,'--g'),                       %PLOTS TOP OF DYNAMIC WINDOW
 magdis=[zeros(freqli-15:frequi+15);...          %SETS UP POWER SPECTRUM
```

```
        IQS_PSD(freqli-15:frequi+15)...
        ;zeros(freqli-15:frequi+15)];
 indexs=[Freq(freqli-15:frequi+15);Freq ...
        (freqli-15:frequi+15);Freq(freqli ...
        -15:frequi+15)];
 magdis=[magdis(:)'];
 indexs=[indexs(:)'];
 plot(indexs,magdis,'-r'),grid                    %PLOTS POWER SPECTRUM
 title(['POWER SPECTRUM OF IQ ENVELOPE,',title2])
 text(1,1,title1,'sc');                           %PLOTS SECONDS AFTER MIDNIGHT
 xlabel('FREQUENCY (KHz)'),                       %PLOTS X-AXIS LABEL
 ylabel('RELATIVE MAGNITUDE'),                    %PLOTS Y-AXIS LABEL
%meta sxs2                                        %SAVES PLOT-OPTIONAL
 hold off                                         %RELEASES PLOT FROM MEMORY
```

- PLOT3

Plots the unfiltered data in the fixed window and the estimated Doppler separation history to the monitor. The option in PLOT3 is to save the plot by using the meta function. This subroutine is called by DFSIOC.

```
%                              SUBROUTINE PLOT3
 title2(['GMT DAY ',num2str(DAY)]);               %TIMING INFORMATION HEADER
 title3(['Q = ',num2str(Q)]);                     %PROCESS NOISE COVARIANCE
 title4(['R = ',num2str(R)]);                     %MEASUREMENT NOISE COV
 axis([TimDAT1(1,1) TimDAT1(length(TimDAT1)... %DETERMINES AXIS FOR PLOT
     ,1) 0 20000])
 plot(TimDAT1,DSF(1:length(TimDAT1),1),'.r',...%PLOTS DATA
     TimDAT1,XDSF(1:length(TimDAT1),1),'-g');
 title(['DOPPLER SEPARATION,',title2])            %PLOTS TITLE
 text(2,1,title3,'sc'),                           %PLOTS PROCESS NOISE COV
 text(3,1,title4,'sc'),                           %PLOTS MEASUREMENT NOISE COV
 xlabel('SECONDS AFTER MIDNIGHT'),                %PLOTS X-AXIS LABEL
 ylabel('DOPPLER SEPARATION (KHz)'),              %PLOTS Y-AXIS LABEL
% meta sxs4                                       %SAVES PLOT-OPTIONAL
```

- PLOT4

Plots the smoothed Doppler separation history to the monitor. The option in PLOT4 is to save the plot by using the meta function. This subroutine is called by DFSIOC.

```
%                              SUBROUTINE PLOT4
 title2(['GMT DAY ',num2str(DAY)]);               %TIMING INFORMATION HEADER
 title3(['Q = ',num2str(Q)]);                     %PROCESS NOISE COVARIANCE
 title4(['R = ',num2str(R)]);                     %MEASUREMENT NOISE COV
 axis([TimDAT1(1,1) TimDAT1(length(TimDAT1)... %DETERMINES AXIS FOR PLOT
     ,1) 0 20000])
 plot(TimDAT1,BXDSF(1:length(TimDAT1),1),...      %PLOTS SMOOTH EST DSF
     '-r'),grid;
 title(['DOPPLER SEPARATION,',title2]),           %PLOTS TITLE
```

```
 text(2,1,title3,'sc'),                    %PLOTS PROCESS NOISE COV
 text(3,1,title4,'sc'),                    %PLOTS MEASUREMENT NOISE COV
 xlabel('SECONDS AFTER MIDNIGHT'),         %PLOTS X-AXIS LABEL
 ylabel('DOPPLER SEPARATION (KHz)'),       %PLOTS Y-AXIS LABEL
% meta sxs4                                %SAVES PLOT-OPTIONAL
```

LIST OF REFERENCES

1. Naval Research Laboratory communication, 24 June 1991.

2. Schultz, K.I., "Analysis of Narrowband IR Lace Measurements Acquired During GMT Day 195 and 200", Massachusetts Institute of Technology Lincoln Laboratory, 6 August 1990.

3. Burl, J.B., "Lace Signal Processing", memo, Naval Postgraduate School, Monterey, CA, 1991

4. Strum, R.D. and Kirk, D.E., *Discrete System and Digital Signal Processing*, Addison-Wesley Publishing Company, Inc., 1988.

5. Gelb, Arthur (ed.), Applied Optimal Estimation, M.I.T. Press, 1974.

6. Bar-Shalon, Y. and Fortman, T.E., Tracking and Data Association, Academic Press, Inc., 1988.

7. Candy, James V., *Signal Processing, The Model Based Approach*, McGraw-Hill, Inc., 1986.

8. Bozic, S. M., Digital and Kalman Filtering, Edward Arnold (Publishers) Ltd, 1981.

9. Nicklas, R. B., "An Application Of Kalman Filter Fixed Interval Smoothing Algorithm To Underwater Target Tracking," Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1989.

10. "Pro-Matlab User's Guide," The Math Works, Inc., 1990

11. Wagener, Jerrold L., Fortran 77 Principles of Programming, John Wiley & Sons, Inc., 1980.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Prof. Jeff Burl, Code EC/Bl Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Prof. Ralph Hippenstiel, Code EC/Hi Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Naval Research Laboratory ATTN: Shalom Fisher Concept Development Section (Code 8241) 4555 Overlook Ave., S.W. Washington, DC 20375	1
Frank R. Thorngren, LT, USN Philadelphia Naval Shipyard Philadelphia, Pennsylvania 19112	1