1

9

CANADIAN THESES ON MICROFICHE

I.S.B.N.

THESES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

55618

NAME OF AUTHOR/NOM DE L'AUTEUR CRAIG ANTHONY MACKELL

TITLE OF THESIS/TITRE DE LA THÈSE AN EXTENDED KALMAN FILTER INTEGRATED
NAVIGATION SYSTEM FOR THE NAIF COMBAT SSC
RESEARCH AIRCRAFT

UNIVERSITY/UNIVERSITÉ CARLETON U, OTTAWA

DEGREE FOR WHICH THESIS WAS PRESENTED/
GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE PHD MASTER OF ENGINEERING

YEAR THIS DEGREE CONFERRED/ANNÉE D'OBTENTION DE CE GRADE 1982

NAME OF SUPERVISOR/NOM DU DIRECTEUR DE THÈSE DR B.W. LEACH, NATIONAL ACRO. EST.
NAC

Permission is hereby granted to the NATIONAL LIBRARY OF
CANADA to microfilm this thesis and to lend or sell copies
of the film.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈ-
QUE NATIONALE DU CANADA de microfilmer cette thèse et
de prêter ou de vendre des exemplaires du film.

The author reserves other publication rights, and neither the
thesis nor extensive extracts from it may be printed or other-
wise reproduced without the author's written permission.

L'auteur se réserve les autres droits de publication; ni la
thèse ni de longs extraits de celle-ci ne doivent être imprimés
ou autrement reproduits sans l'autorisation écrite de l'auteur.

DATED/DATE 30 April 1982 SIGNED/SIGNÉ Craig Mackell

PERMANENT ADDRESS/RÉSIDENCE FIXE 134-2111 MONTRÉAL ROAD, OTTAWA, ONT
K1J 8M8

AN EXTENDED KALMAN FILTER
INTEGRATED NAVIGATION SYSTEM
FOR THE
NAE CONVAIR 580 RESEARCH AIRCRAFT

by

Craig Anthony Maskell , BSc(EE)

A THESIS
PRESENTED TO
THE FACULTY OF GRADUATE STUDIES AND RESEARCH
CARLETON UNIVERSITY

IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF ENGINEERING

Department of Mechanical and Aeronautical Engineering
Faculty of Engineering
Carleton University
OTTAWA, CANADA
April 1982

The undersigned recommend to the Faculty of Graduate Studies and Research, Carleton University, for acceptance, the thesis, entitled

AN EXTENDED KALMAN FILTER INTEGRATED NAVIGATION
SYSTEM FOR THE NAE CONVAIR 580-RESEARCH AIRCRAFT

by

Craig Anthony Maskell

in partial fulfillment of the requirements for the degree of
Master of Engineering.

B.W. Leach

Thesis Supervisor
National Aeronautical Establishment

[Signature]
Thesis Supervisor, Department of
Mechanical and Aeronautical
Engineering

[Signature]
Chairman, Department of
Mechanical and Aeronautical
Engineering

May 7/82
DATE

ABSTRACT

A practical integrated navigation system suitable for use on the NAE Convair 580 Aeromagnetics research aircraft is proposed. An extended, "quasi-linear" ten-state Kalman filter scheme using Doppler radar, magnetic compass, and VLF phase data is developed, with emphasis on intuitive concepts and the basic errors that are likely to occur in the navigation sensors. A simple algorithm to allow the filter to adapt to certain known VLF anomalies is described, and the accuracies and advantages of using such a filtering scheme are proven through numerous off-line simulation runs. The application of the proposed system in an on-line environment is shown to be viable through several filter runs using real flight navigation data. Position errors of less than 0.4 NM are observed and these are considerably smaller than errors observed when other navigation schemes are used with the same data.

ACKNOWLEDGEMENTS

I wish to express my sincere and unreserved appreciation to Dr. Barrie Leach of the National Aeronautical Establishment, who has offered many suggestions, comments and criticisms during the course of my research.

I am thankful for the various forms of assistance given by Mr. C.D. Hardwick and the Aeromagnetics Group of the NAE Flight Research Laboratory. I am also indebted to the Heads of the Flight Research Lab, Mr. A.D. Wood and, currently, Dr. S.R.M. Sinclair, who have provided the facilities and resources necessary for the evolution of this project.

My academic supervisor, Professor R.J. Kind, has been very helpful throughout my stay as a Graduate student at Carleton University. Financial support during this period has been generously arranged by the Chairmen of the Department of Mechanical and Aeronautical Engineering, Professor H.R. Saravanamutto and, currently, Professor J.A. Goldak, through the Faculty of Graduate Studies and Research.

Mrs. A.M. Gorman, Head of the CISTI Uplands Library, has made reference searching seem very easy and I wish to thank her for the support she has given to my academic and personal careers.

Mssrs. D.H. Carter and S.J. Rielly of the Computer Analysis Group at the Flight Research Lab have been most helpful in the development of the simulation code. Steve has also provided expert advice on the computer graphical plotting routines.

The excellent typing of this thesis has been accomplished by Miss L. Cossette and Mrs. P. Clyne, who have done a remarkable job in following my handwritten original.

The encouragement necessary for the completion of this research has come from many sources. Among others, Mike Parent, and especially Nicole Fournier, have followed the ups and downs of my Graduate studies and have been quick to add a note of optimism when needed.

Finally, my deepest appreciation and love is to my parents, Ken and Anne Maskell. Their guidance, reassurance and inspirations have been profound.

TABLE OF CONTENTS

	PAGE
ABSTRACT	i
ACKNOWLEDGEMENTS	iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
NOMENCLATURE	xi
 CHAPTER	
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Outline of the Current Study	3
1.3 Review of Previous Work	5
 2 NAVIGATION AND AIRCRAFT SYSTEMS	11
2.1 Introduction	11
2.2 General Navigation Concepts	11
2.3 Accuracy Requirements	12
2.4 Dead Reckoning: Doppler Radar	13
2.4.1 Dead Reckoning Measurements	13
2.4.2 Doppler Radar Systems	13
2.4.3 Doppler Radar Errors	14
2.4.4 Compass Heading Measurements	15
2.4.5 Effects of Uncorrected Errors	16
2.5 Position Fixing: VLF Communications Stations	17
2.5.1 VLF Communications Stations	17
2.5.2 Navigation Using VLF Signals	18
2.5.3 VLF Propagation and Errors	19
2.6 NAE Convair 580 Systems	23
2.7 Summary	24
 3 KALMAN FILTER THEORY	25
3.1 Introduction	25
3.2 Basic Minimum Variance Estimation	25
3.2.1 Single Dimension Optimal Estimation	26
3.2.2 Optimal Estimation using Measurement Data	28
3.2.3 Multi-dimensional Optimal Estimation	30
3.3 Physical Dynamic System Representation (State Space)	32
3.3.1 System Description	32
3.3.2 Measurement Description	33
3.4 Derivation of Kalman Filter Equations for Dynamic Systems	34
3.5 Quasi-linearization of the Kalman Filter	36

CHAPTER		PAGE
4	A KALMAN FILTER FOR AN INTEGRATED AIRCRAFT NAVIGATION SYSTEM	39
	4.1 Introduction	39
	4.2 State Variables Chosen	40
	4.3 Plant Update Equations	41
	4.4 Plant Noise Processes	42
	4.5 Quasi-linear State Transition Matrix	44
	4.6 VLF Observation Equations	46
	4.7 Quasi-linear Observation Matrix	47
	4.8 The Kalman Filtering Process	48
	4.8.1 Initialization	48
	4.8.2 Adaptive Filtering	50
	4.8.3 Practical Considerations	51
	4.8.4 Sub-optimality of the Kalman Filter	52
	4.9 Summary	53
5	KALMAN FILTER SIMULATION STUDY : FORTRAN PROGRAM	57
	5.1 Introduction	57
	5.2 FORTRAN Kalman Filter Simulator	58
	5.3 Flight Simulation Algorithms	60
	5.4 Adaptive Filter Options	62
	5.5 Automatic Contingency Compensation	63
6	KALMAN FILTER SIMULATION STUDY : RESULTS	69
	6.1 Introduction	
	6.2 Basic Operation of the Filter	70
	6.3 Results from using VLF Data of Various Qualities	72
	6.4 State Size Reduction Through Elimination of VLF Stations	74
	6.5 VLF Anomalies : A Priori Knowledge	77
	6.5.1 VLF Bias Drifting	77
	6.5.2 VLF Bias Jumps	79
	6.5.3 VLF Noise Increase	80
	6.6 VLF Anomalies : Automatic Contingency Compensation	82
	6.7 Variable Update Interval	85
	6.8 Exponential Data Weighting	86
	6.9 Summary and Conclusions	88
7	REAL FLIGHT DATA RESULTS	91
	7.1 Introduction	91
	7.2 Algorithms for Real Flight Data	91
	7.3 Convair 580 Navigation Flight	92
	7.4 Kalman Filter Runs using Real Flight Data	95
	7.4.1 Filter Runs with Nominal Initial Parameters	95
	7.4.2 Refinements in the Parameter Specifications	98

CHAPTER		PAGE
7 (con't)		
	7.4.3 Refinement in the System Model: Bias Noise Augmentation	99
	7.4.4 Automatic Contingency Algorithm	105
	7.5 Summary and Conclusions	106
8	CONVAIR 580 ON-LINE IMPLEMENTATION	108
	8.1 Introduction	108
	8.2 Kalman Filter On-line Implementation Sequence	109
	8.3 General Real-time Considerations	110
9	SUMMARY AND CONCLUSIONS	113
	9.1 Summary	113
	9.2 Future Work	114
	9.3 Conclusions	115
	REFERENCES	116
	APPENDIX	
A	DOPPLER VELOCITY AND DRIFT ANGLE MECHANIZATION	202
B	PLANT NOISE AND UPDATE EQUATIONS	204
C	STATISTICS OF AN INTEGRATED DISCRETE RANDOM NOISE PROCESS	209
D	AUTOMATIC VLF CONTINGENCY DETECTION ALGORITHM	214
E	DOCUMENTATION FOR KALMAN FILTER SIMULATION PACKAGE	219

LIST OF TABLES

TABLE		PAGE
2.1	U.S. Navy VLF Communication and Navigation Stations	125
2.2	VLF Navigation Receivers	126
5.1	Kalman Filter Modes	127
6.1	Simulator and Filter Parameters: Basic Filter Operation	128
6.2	Simulator and Filter Parameters: Variable Quality Data	129
7.1	Visual Reference Points for Real Flight Data Navigation Study	131
7.2	Real Flight Data Noise Variances	132
7.3	Nominal Initial Parameters for Real Flight Data Kalman Filter Run	133
7.4	Refined Initial Parameters for Real Flight Data Kalman Filter Run	134
7.5	Sequence of Events for On-line VLF Bias Jump Detection	135
8.1	Memory Requirements for the On-board Kalman Filter Task	136

LIST OF FIGURES

FIGURE		PAGE
2.1	Dead Reckoning Nomenclature	137
2.2	Doppler Radar: 3 Beam Janus Configuration	138
2.3	Position Error Due to Uncorrected Doppler/Heading Biases	139
2.4	VLF Communication and Navigation Stations	140
2.5	Locations of VLF Navigation Stations Relative to Ottawa	141
2.6	Radio Position Fixing Methods	142
5.1	Main Sections of the Kalman Filter Integrated Navigation System Simulator	144
5.2	Block Diagram of the Kalman Filter Integrated Navigation System Simulator	145
5.3	Typical Simulated Flight Trajectories	146
5.4	Representation of the Measurement Residual	147
5.5	Typical Innovations Sequence for a VLF Station with a Drifting Bias	148
5.6	Typical VLF Bias Drift and Jump Detection Criteria	149
5.7	Flowchart of Automatic On-line VLF Anomaly Detection Algorithm	150
6.1	Kalman Filter State Estimates-Large Initial Bias Errors and Large Initial Error Covariances	151
6.2	Kalman Filter State Error Responses- Large Initial Bias Errors with Small, Medium, and Large Initial Error Covariances	152
6.3	Propagation of P_{33} for Small, Medium and Large P_0	153
6.4	Propagation of P_{55} for Small, Medium, and Large P_0	154
6.5	Propagation of $\ P\ $ for Small, Medium, and Large P_0	155
6.6	Mean Square Errors of Kalman Filter Estimates for Small, Medium, and Large P_0	156
6.7	Kalman Filter Error Responses- Poor Quality Doppler, Heading, and VLF Data	157
6.8	Kalman Filter Error Responses- Good Quality Doppler, Heading, and VLF Data	158

FIGURE		PAGE
6.9	Kalman Filter Error Responses- Very Good Doppler Heading and VLF Data	159
6.10	Kalman Filter Error Responses- Very Good Quality Data, $R_{\text{ASSUMED}} \gg R_{\text{ACTUAL}}$	160
6.11	Kalman Filter Error Responses- Poor Quality Data, $R_{\text{ASSUMED}} \ll R_{\text{ACTUAL}}$	161
6.12	Kalman Filter Error Responses- Reduced Order Filter with Three VLF Stations and Good Quality Data	162
6.13	Kalman Filter Error Responses- Reduced Order Filter with Three VLF Stations and Poor Quality Data	163
6.14	Kalman Filter Error Responses- Reduced Order Filter with Two VLF Stations and Good Quality Data	164
6.15	Kalman Filter Error Responses- Reduced Order Filter with Two VLF Stations and Poor Quality Data	165
6.16	Kalman Filter Error Responses- Reduced Order Filter with One VLF Station and Good Quality Data	166
6.17	Kalman Filter Error Responses- Reduced Order Filter with One VLF Station and Poor Quality Data	167
6.18	Propagation of P Diagonal Elements for 5 th Order Filter	168
6.19	Naive Kalman Filter Error Responses- Effects of a Linear VLF Bias Drift in One Station	169
6.20	Smart Kalman Filter Error Responses- P_{55} Fixed for Duration of a Linear VLF Bias Drift	170
6.21	Naive Kalman Filter Error Responses- Effects of a Sudden VLF Bias Jump in One Station	171
6.22	Smart Kalman Filter Error Responses- P_{77} set to a Large Value at the Occurrence of a Sudden VLF Bias Jump	172
6.23	Naive Kalman Filter Error Responses- VLF Noise Increase at 2000 Seconds	173
6.24	Smart Kalman Filter Error Responses- VLF Noise Increase at 2000 Seconds, R adjusted	174
6.25	Effect of Errors in Measurement Noise Variance on Steady State RMS Error in State Estimation	175
6.26	Kalman Filter Error Responses- On-line Anomaly Algorithm used to Detect Bias Drifts in all VLF Stations	176

FIGURE		PAGE
6.27a	Innovations Sequence for Drifting VLF Bias Situation: Automatic On-line Detection Scheme In Use	177
6.27b	Propagation of $P_{55_{k+1}}$ When On-line Anomaly Detection is used	177
6.28	Kalman Filter Error Responses- Exponential Data Weighting; $S=1.05$, No Anomaly	178
6.29	Kalman Filter Error Responses- Exponential Data Weighting; $S=1.05$, VLF Bias Drift in Station 0	179
7.1a	Doppler/VLF Kalman Filter Navigation Scheme	180
7.1b	Flowchart of Real Flight Data Kalman Filter Integrated Navigation System	181
7.2	Flight Pattern of the Real Flight Data Navigation Study	183
7.3	Raw Doppler/Heading Data from the Real Navigation Data Flight	184
7.4	Positions of VLF Stations used in the Real Flight Data Navigation Study	185
7.5	Kalman Filter State Estimates- Real Flight Data with Nominal Filter Parameters	186
7.6	Kalman Filter Position Errors (with Nominal Parameters)	187
7.7	Kalman Filter State Estimates- Real Flight Data with Refined Initial Filter Parameters	188
7.8	Kalman Filter Position Errors (with Refined Initial Filter Parameters)	189
7.9	Innovations Sequence Corresponding to VLF Station L; Real Flight Data Study with Refined Parameters	190
7.10	Innovations Sequence Corresponding to VLF Station R; Real Flight Data Study with Refined Parameters	190
7.11	Kalman Filter State Estimates- Real Flight Data with VLF Bias Noise Modelling	191
7.12	Kalman Filter Position Errors (with VLF Bias Noise Modelling)	192
7.13	Innovations Sequence Corresponding to VLF Station L; Real Flight Data Study with VLF Bias Noise Modelling	193
7.14	Kalman Filter State Estimates- Real Flight Data with VLF and Doppler Bias Noise Modelling	194

FIGURE		PAGE
7.15	Kalman Filter Position Errors (with VLF and Doppler Bias Noise Modelling)	195
7.16	Innovations Sequence Corresponding to VLF Station L; Real Flight Data Study with Doppler and VLF Bias Noise Modelling	196
7.17	Kalman Filter State Estimates- Real Flight Data with Simulated Bias Jump and Bias Noise Modelling	197
7.18	Kalman Filter State Estimates- Real Flight Data with Simulated Bias Jump and Inaccurate VLF Bias Noise Modelling	198
7.19	Kalman Filter State Estimates- Real Flight Data with Simulated Bias Jump and On-line Anomaly Detection	199
8.1	Project Development Leading to On-board Integrated System Implementation	200
8.2	Timing Sequence for Doppler/VLF Data Acquisition and Kalman Filter Updates	201
B.1	Nomenclature for Plant Noise and Update Equations	208
E.1	Flowchart for KFSIM	234

The NAE Convair 580 also has numerous other facilities ; included in its navigation complex: such as a radar altimeter, DME system, Loran-C receiver and video cameras for visual "on-top" data recording. In addition to the GNS-200, a GNS-500 VLF receiver has recently been installed and is currently undergoing performance evaluation and data quality checks. It is proposed that the GNS-500 will eventually replace the GNS-200 VLF system.

2.7 Summary

This chapter has presented details of the basic navigation problem and the two types of navigation methods. The specific dead reckoning and position fixing schemes to be combined in the integrated system have been described with emphasis placed on the development of appropriate error models. Although the Doppler and VLF systems could provide fairly accurate position estimates by themselves in most instances, their inclusion in an integrated system provides redundant information which allows greater accuracy and anomaly compensation.

The actual navigation units installed on the NAE Convair 580 have been briefly described. Additional information is available in the cited references.

P_{k+1}	A posteriori state error covariance matrix
$P'_{LAT_{k+1}}, P'_{LONG_{k+1}}$	Plant position update
P_0	Initial state error covariance matrix
P_s	Start position of aircraft
P_T	True position of aircraft
Q_k	Plant noise autocovariance matrix
R	Measurement noise autocovariance matrix
$\underline{U}_k$	Plant noise vector
V	Aircraft velocity
VAR	Magnetic variation
V_{DOP}	Doppler measured aircraft velocity
$\underline{V}_{k+1}$	Measurement noise vector
V_s	Speed of propagation of VLF signal
V_{true}	True aircraft velocity
V_x	Along-track aircraft velocity (earth-referenced)
V_y	Across-track aircraft velocity (earth-referenced)
$\underline{X}$	n-dimensional state vector
$\underline{X}_1, \underline{X}_2$	Independent vector estimates of $\underline{X}$
$\underline{X}_{k+1}$	True state at time t_{k+1}
$\hat{\underline{X}}_k$	Optimal state estimate at time t_k
$\underline{X}'_{k+1}$	Best plant update of the state at time t_{k+1}
$\hat{\underline{X}}_0$	Initial optimal state estimate vector
$\hat{\underline{X}}$	Optimal estimate of the weighted average of $\underline{X}_1$ and $\underline{X}_2$
$\underline{Y}_{k+1}$	State measurement vector at time t_{k+1}
$\underline{Y}_2$	m-dimensional measurement vector

b_0, b_1	Linear regression coefficients
e_k	Error in the state estimate
f_D	Doppler frequency shift
f_{k+1}	Nonlinear plant update at time t_{k+1}
f_a, f_b, f_c	Doppler radar beam frequencies
f_{rec}	Apparent frequency of Doppler return signal
f_{trans}	Frequency of transmitted Doppler signal
f^*	Continuous-time plant nonlinearity
h	Scale factor inherent in single dimension measurement process
$\underline{h}$	Nonlinear measurement vector function of the state
k_1, k_2	Weights on x_1, x_2
k	Optimal estimate of the single dimension weighting factor
s	Exponential data weighting factor
u	Forward aircraft velocity (fixed-body referenced)
u_{B_v}	Random noise in Doppler velocity bias
$u_{B_{\alpha\beta}}$	Random noise in track angle bias
$u_{B_{\phi\epsilon}}$	Random noise in VLF phase bias
v	Lateral aircraft velocity (fixed-body referenced)
w	Vertical aircraft velocity (fixed-body referenced)
x	Single dimension random variable
x_1, x_2	Independent estimates of x
$\bar{x}$	Weighted average estimate of x
$\hat{x}$	Optimal estimate of x
y_c	Measured μ SEC phase of VLF station i

y_2	Single dimension measurement of x_2 ($y_2 = hx_2$)
y_1, y_2	End points of linear regression curve
$\Delta P_N, \Delta P_E$	North and east components of distance travelled
Φ	State transition matrix (linear and quasi-linear)
α	Aircraft drift angle
α_{DOP}	Doppler measured drift angle
α_{true}	True aircraft drift angle
β	Aircraft heading angle with respect to true north; Magnetic compass measured heading angle
β_{true}	True heading with respect to true north
ϵ_N, ϵ_E	North and east errors in calculated aircraft position
θ	Aircraft pitch angle
λ	Wavelength of transmitted Doppler signal
σ_v^2	Variance of raw Doppler velocity data
σ_v^{*2}	Variance of integrated Doppler velocity data
σ_x^2	Variance of $\bar{x}$
$\hat{\sigma}_x^2$	Optimal estimate of the variance of x
σ_1^2, σ_2^2	Variances of x_1 and x_2
σ_{α}^2	Variance of raw track angle data
σ_{α}^{*2}	Variance of integrated track angle data
$\sigma_{\phi_t}^2$	Variance of VLF phase data for station t
ϕ	Aircraft roll angle
ϕ_t	True μ SEC phase of VLF station t
$\phi_{(p)}$	True theoretical VLF phase based on plant updated position
ψ	Doppler antenna depression angle
ψ_{true}	True heading with respect to magnetic north

Vector and Matrix Notation: Vectors are indicated by one underscore, for example, $\underline{x}$ or $\underline{f}$.

Matrices are indicated by two underscores, for example, $\underline{\underline{A}}$ or $\underline{\underline{K}}$.

Matrix Operations

$E[\cdot]$ is the expectation operator

$[\cdot]^T$ is the matrix transpose of $[\cdot]$

$[\cdot]^{-1}$ is the matrix inverse of $[\cdot]$

$\| \cdot \|$ is the matrix norm of $[\cdot]$

$[1]$ is the unit matrix

Units: Depending in what context the particular variable is being used, the following units are assumed:

Velocity	KNOTS or NM/Sec
Angle	DEGREES or RADIANS
VLF Phase	μ SEC
Aircraft Position	DEGREES Latitude and Longitude
Doppler radar freq.	Hz
Velocity noise variance	KNOTS ² ; (NM/Sec) ²
Angle noise variance	DEG ² ; RADIANS ²
VLF phase noise variance	μ SEC ²

CHAPTER 1

INTRODUCTION

1.1 Introduction

Precise navigation has been required by man since the beginning of civilization. Regardless of the nature of a voyage, exploration, or trip, the traveller has always had to answer three basic questions: "Where am I now?", "Where is my destination?" and "What is the best direction to follow?" For many years these questions were addressed and answered sub-consciously: the hunter or tribesman searching for food or an enemy tribe simply travelled in the direction he thought was best without even realizing that he was in fact basing his decision on the answers to the three questions, or in other words, "navigating". The explorers in the Middle Ages used the sun and stars as guides and by the time of Columbus, the magnetic compass was in widespread use. By this time "navigation" (from the Latin *navis*, a ship, and *agere*, to drive: "ship driving") had become established as more of an art than a science. With the advent of increased travel frequency and distance, it became necessary to have more precise navigation techniques: the art was slowly being replaced by a science. Of course, with the introduction of rapid aircraft and countless passenger flights, precise navigation is not just a simple requirement or nicety; it is a prerequisite. The nature of current scientific (and military) missions dictate that the research vehicle be adequately instrumented with high quality and accurate navigation systems: one cannot travel to the moon by relying solely on Polaris!

In the modern context, navigation is concerned with establishing the "best" route to follow when undertaking a journey. Although this statement may seem all-encompassing and very general, it is in fact the basic reason for the existence of accurate navigation systems. In determining the best route, however, the following factors must be considered:

- determining the direction or heading of the voyage (or legs of the voyage)
- avoiding collision with other vehicles
- fuel consumption and fuel remaining
- time

Closely associated with the first factor is the determination of the present position of the vehicle and the position of the destination. Collision

avoidance is obviously important. Fuel consumption should be minimized, which implies taking as little time as possible in order to complete the journey. The last two factors are largely economic ones, although few people would enjoy a three day trip to travel 100 miles, regardless of economics!

Of the four factors mentioned above, perhaps the most important is the determination of present position. Establishment of exact position in terms of earth coordinates is really an impossibility, even with today's extremely reliable and accurate systems. Therefore, we are not concerned if we do not obtain perfect position estimation; but, rather, we attempt to make our estimate as good as possible given the conditions at hand. The development of the Kalman filter integrated navigation system detailed in this report attempts to fulfill this requirement.

The NAE Convair 580 (CF-NRC-X) is a twin engine medium range aircraft currently being used primarily in aeromagnetics research. The maximum take off weight is approximately 60,000 pounds and, depending on the particular research flight requirements, the maximum range is about 1600 NM. Additional details of the aircraft may be found in [1]. The nature of aeromagnetic surveys is such that very accurate position indications are required since the aircraft instrumentation is essentially mapping the earth's magnetic field. In a typical mapping flight, grid lines are flown with 2-5 NM spacing. Aeromagnetic data are plotted as a function of aircraft position. Hence, the more accurate the position estimates, the better the aeromagnetic map.

The Convair 580 currently has several navigation systems on board: an LTN-51 inertial system, GNS-200/GNS-500 Very Low Frequency (VLF) receivers, a Decca Doppler radar and an Internav Loran-C receiver, plus numerous associated support systems. An Interdata 7/32 minicomputer is used for data processing and recording. The integrated navigation system developed in this study combines data from the GNS-200 and Doppler radar in an "optimal" fashion in order to provide a position estimate. The data is processed by a Kalman filter operating in discrete time with a position update generated every ten seconds. The Kalman filter is basically a set of five recursion equations which produce optimal estimates (in the minimum variance sense) of the latitude and longitude positions and several other quantities. For reasons explained later, the filter is actually a sub-optimal one, but the estimates are accurate nonetheless.

1.2 Outline of the Current Study

The development of the integrated navigation system employing a Kalman filter involves five basic steps, given in order as follows:

- identification of Doppler/heading and VLF error models,
- formulation of the Kalman filter equations applicable to the specific navigation systems under consideration,
- development of a Monte Carlo simulation program and evaluation of the proposed filter through off-line simulation runs,
- evaluation of the filter using real flight data,
- on-line implementation on the aircraft computer.

Chapter 2 of this report describes the Doppler dead reckoning and VLF position fixing systems with emphasis on the errors that can be expected under certain operating conditions. The formulation of appropriate error models is detailed, under the assumptions of constant bias errors and Gaussian random noise. An example of the effect of uncorrected Doppler and heading errors is given. Position fixing using VLF communications signals is described, again with emphasis on errors caused by propagation anomalies, and the predictable and unpredictable error sources are detailed. The final section in Chapter 2 briefly describes the physical systems on board the Convair 580.

Chapter 3 attempts to develop the Kalman filter equations from basic statistical and intuitive concepts rather than from rigorous probability theory. The practical applications of Kalman filtering are kept in mind throughout the explanations. Starting with single dimension optimal estimation, the minimum variance estimation equations are developed into the general multi-dimensional case using measurement data. The concept of state space and physical dynamic system representation is introduced, which leads to the definition of the five Kalman filter equations for a linear dynamic process. Because of the nonlinear nature of the navigation process, however, the equations must be further refined ("extended") through a quasi-linearization procedure. The end result is the definition of an extended Kalman filter applicable to the integrated navigation task.

Given the navigation systems and models described in Chapter 2 and the Kalman filter of Chapter 3, the actual practical filter is developed in Chapter 4. The plant update equations and noise processes are defined, and the appropriate filter matrices are given. The actual Kalman filtering process is detailed in Section 4.8 with emphasis on initialization, adaptive filtering, practical considerations, and the sub-optimality of the proposed filter. Finally, the last section of Chapter 4 summarizes the extended Kalman filter.

Much of the study involved evaluation of the filter under simulated flight conditions. Chapter 5 details the off line computer simulation package developed for use in the IBM 3033 TSS (Time Sharing System) environment. The algorithms are described, as are the various options available to the user. The general nature of the routines allows considerable flexibility and choice when selecting flight and sensor conditions. An on-line automatic anomaly detection scheme for the VLF data is proposed in Section 5.5.

The results of the simulation study are given in Chapter 6. Numerous simulations were carried out and some interesting results are described. These simulations show that acceptable filter operation is possible, and, furthermore, the numerical problems that have been reported in numerous other papers apparently are not cause for alarm in the present filter. Compensation of VLF anomalies is shown to be possible using an on-line detection algorithm which seems to circumvent possible practical problems inherent in theoretical algorithms suggested by other authors.

Chapter 7 describes the filter runs using real data from a recent navigation flight. The filter model and parameters are refined to the point where fairly accurate position estimates are produced by the filter. Visual "on-tops" serve as references for establishing the filter accuracy.

The actual on-line real time version of the filter is described in Chapter 8. Although not yet fully operational, the details given in this chapter indicate that acceptable performance will be attained.

1.3 Review of Previous Work

The sheer bulk of material published on Kalman filtering and navigation in the last twenty years precludes a thorough and complete review of the previous work in this area. Since the publication of the famed Kalman paper in 1960 (with due respect to Swerling, 1958), there have been hundreds (thousands?) of papers, conferences and texts dealing with optimal recursive filtering, and a large proportion of these pertain to applications in the aerospace and aeronautical fields (particularly earth navigation). An entire project could be devoted to a comprehensive literature review on this topic.

The material presented here does not attempt to review every paper written on Kalman filter integrated navigation systems. It is hoped, however, that the more pertinent (and certainly the definitive) works that are applicable to the present study will be included. The review includes four categories: (i) basic Kalman filtering theory coinciding with Chapters 3 and 4 of this study; (ii) integrated aircraft navigation systems employing Kalman filters; (iii) developments in parameter estimation techniques; and (iv) fault detection and bias jump identification. Several texts are briefly mentioned at the end of the review.

The foundation for Kalman filtering actually preceded Kalman by about 150 years. Karl Gauss published a paper in 1809 [2] which detailed his work on least squares estimation applied to the orbit and motion of planets and comets. In his paper, Gauss refers to concepts (although not by the same names) such as observability, dynamic modelling, and probabilistic representation of unknown quantities. Putting it into perspective, Kalman filtering is "simply" an efficient computational solution of the least squares method. Sorenson [3] gives a rather interesting overview of the development of Kalman filtering with respect to the original work by Gauss and others.

In the 20th century, Fisher [4] introduced the concept of maximum likelihood estimation. In the 1940's, and independent of each other, Kolmogorov [5] and Weiner [6] pioneered the linear minimum mean square estimation technique using integral equations. The so-called Weiner filter has certain limitations (see [7]) which precludes, to a large extent, its application to many practical problems.

The disadvantages of the Weiner filter were overcome by Kalman in his 1960 paper [7] which generalized the results of Weiner. By using a state space approach, it was possible to solve the linear estimation problem in a recursive fashion. Furthermore, the Kalman filter utilizes difference equations rather than integral equations. The publication of this now "classical" paper coincided very nicely with the expanding use of computers for numerical computations, and also with the new stringent navigation requirements of space programs. The recursive Kalman filter was a natural task for the digital (and discrete) computer. Kalman and Bucy [8] combined their previously independent work to give extensions to the 1960 Kalman solution. The Kalman and Kalman and Bucy papers are regarded as the starting point to the recursive optimal estimation applications that have been so abundant since then. It should be noted that although the basic Kalman filtering theory is clearly laid out in References [7] and [8], very few details are given on the practical application of the recursive equations. Much of the subsequent research by other authors has dealt with the very important aspect of implementing the basic theory in the "real world".

Two major problem areas that were encountered early in the "Kalman filter era" were computational or numerical difficulties and non-stationary noise processes. The first problem has been effectively solved by numerous approaches which are numerically more attractive than the basic Kalman filter recursion equations, such as square root filtering (Ref. [9], for example). The second problem necessitates some sort of adaptive filtering scheme in which the statistics of the plant and measurement noise processes can be accurately estimated. In addition, another problem somewhat similar to the non-stationary noise situation is that of sudden (or even slow) shifts in certain filter state variables when the filter is operating under steady state conditions.

The result of not accounting for these changes is a phenomenon known as filter divergence, which is characterized by an increasing discrepancy between the theoretical error in the state estimated by the filter and the actual state error. The problem and analysis of filter divergence has been examined by Schlee, et al [10], who cites the causes of divergence as model errors, bias errors and computer or machine errors. Modified filters are proposed and results of simulation studies show that divergence is more severe for sensors with increased *a priori* accuracy and for increased data

rates. Price [11] shows the desirability of having process noise in the model to aid in the control of divergence and stability. Fitzgerald [12] describes "apparent" divergence (finite degradations due to modelling inaccuracies) and "true" divergence (errors may become unbounded), and concludes that manipulation of the process noise covariance matrix can partially compensate for filter divergence due to bias shifts. Farrell [13] treats filter divergence in the navigation context and gives general solutions to the problem. Several methods for handling divergence are suggested by Gelb [14], such as exponential data weighting and the inclusion of a fictitious noise process.

The correct identification of the various system parameters (i.e. noise covariance matrices) is important because the Kalman filter is not optimal if the filter variances do not match the actual noise variances. In practical applications the noise processes are very seldom stationary and, in addition, the statistics may not be accurately known *a priori*. Numerous authors have proposed schemes which attempt to adapt the filter to the changing conditions. Kailath [15] introduces the so-called "innovations" property of the Kalman filter, which is the basis for most adaptive algorithms. Briefly, the innovations property states in part that if the filter is operating in an optimal sense then the innovations (or residual) sequence will be a Gaussian, zero mean, white noise process. This result is given by Mehra [16]. Mehra computes the autocorrelation function of the innovations sequence to decide if the sequence is white under steady state filter operation. If the filter is sub-optimal, the plant and measurement covariances are estimated. One drawback of the proposed method is the necessity to have large sample sizes, although a reference is cited for small sample size cases. Another problem, as pointed out by Boozer and McDaniel [17], is that the whiteness of the steady state filter innovation sequence is not a sufficient condition for optimality when the state transition matrix of the system is not known. The problem of unknown first and second order noise statistics is treated by Chin and Kaufmann [18] who apply an identification scheme to an inertial navigation system. The noise means and variances are independently estimated and the filter adapted accordingly. Simulation results indicate that, although the adaptive system eventually converges on the true statistics, a large number of observations are required.

A useful algorithm for detecting manoeuvres in an aircraft flight trajectory and adapting the Kalman filter is proposed by McAulay and Denlinger [19]. The assumption that a white, zero mean innovations process will be produced by the optimal filter during straight line, constant velocity tracks allows the detection of a manoeuvre if a bias is observed in the innovations process. The generalized likelihood ratio test is used in a theoretical detection scheme, but a modified approach is suggested for practical applications.

Sudden or gradual shifts in certain state variables during steady state filter operation can be severely detrimental to the estimation accuracy. Detection and estimation of these anomalies is therefore particularly important. As in the case of the parameter estimation problem, numerous authors have proposed methods to deal with this situation. The solutions are variably known as fault detection or detection of bias shifts or jumps. Mehra and Peschon [20] give a general approach to fault detection, diagnosis and prognosis. The innovations sequence is tested in several different ways to allow detection of anomalies; several applications are given. A more detailed approach is taken by Wilsky and Jones [21] who propose an adaptive scheme based on generalized likelihood ratio techniques to detect jumps of unknown magnitudes occurring at unknown times. The accuracy of the filter under normal conditions is not degraded and simulation results show that the proposed method is highly reliable in tracking jumps and very few false alarms are apparent. The problem of estimating the state in the presence of a constant but unknown bias vector is considered by Friedland [22] and three recent studies by the same author [23, 24, 25] consider procedures for estimating sudden changes of biases in linear dynamic systems. The detection strategies are based on nonlinear extensions to the basic maximum likelihood principle. Simulation results show [25] that the detection algorithm is quite useful in adapting the Kalman filter to changes in biases, although several "false alarms" are observed. The problem of accurately setting a threshold level is discussed: there is a tradeoff between detecting all bias shifts and inducing too many false alarms. In addition, the results show the effects of increased observation noise on the bias detection accuracy: small bias shifts tend to be masked (and hence undetectable) in very noisy observations. A concise summary of fault detection possibilities is provided by Gregson [26].

A different approach to controlling filter divergence is to place more weight on the most recent observations while dewatering or even discarding past observations. This method attempts to eliminate the effects of a small and overly optimistic error covariance matrix, and is referred to as limited memory optimal filtering or exponential age fading filtering. Jazwinski [27] introduces a limited memory filter which combines batch processing with recursive filtering. This filter requires that a certain number of past observations be stored in memory and processed in a batch procedure. The filter has applications in problems where divergence occurs due to modeling or computational errors. A true recursive filter which exponentially weights the observation data is formulated by Tarn and Zaborszky [28]. In effect, past observations are weighted by a progressively small number. The approach is numerically attractive in that only one single additional multiplication by a fixed scalar is required at each observation time. Simulation results indicate the usefulness of this method. A basic method of choosing the so-called fading factor is suggested by Sorenson and Sacks [29] who also give asymptotic properties and bounds for the error covariance matrix. An adaptive procedure for selecting values for the fading factor is based on the behaviour of the observation residuals. An extension of the fading memory filter to nonlinear situations is given by Sacks and Sorenson [30].

Practical Kalman filters must usually account for a nonlinear dynamic system. The so-called extended Kalman filter is basically a linear filter with slight modifications which make the recursive equations suitable for estimation of nonlinear systems. Richman and Friedland [31] formulate a "quasi-linear" filter which is applied to an integrated inertial/Doppler/Loran navigator, and is based on the extended filter introduced by Cox [32], Freidland and Bernstein [33], and others. The plant and observation models are developed and computer simulations show significant improvement in accuracy over a straight inertial navigator. Cox [32] considers the problem of estimating the state of a discrete time, noisy nonlinear system. Generalizations and extensions are given and emphasis is placed on the method's suitability for real time implementation on a digital computer. Comprehensive treatments of the nonlinear filtering problem are given by Jazwinski [34], Gelb [14] and Anderson and Moore [35].

Examples of Kalman filter applications in aircraft navigation are abundant and the literature contains numerous reports of integrated systems. Zimmerman [36] points out several advantages of an optimally integrated system such as reliability, low cost, failure detection and fully automated operation. An inertial/Doppler/Omega navigator is considered and its performance is analyzed. An interesting account of a flight test program for a Doppler/inertial system is given by Gaines [37]. A twelve element state error vector is used and position accuracies during 3-4 hour flights are shown to be quite acceptable. Schmidt, et al [38] details the development and testing of the Lockheed C-5 inertial/Doppler/Loran navigation system. The Panavia Tornado integrated system is described by Schwegler [39] and practical considerations such as software proving, computation requirements and modeling problems are discussed. For a perspective of integrated navigation systems and Kalman filtering see Brown [40].

The above review attempts to present an overview of the literature applicable to the topics included in this study. As mentioned previously, many papers are necessarily left out. Although cited in the review, several texts and collections of papers on the subjects of Kalman filtering and navigation are given below. These represent a summary of the hundreds of studies that have advanced the usefulness of the basic Kalman filter theory.

The Advisory Group for Aerospace Research and Development (AGARD) publishes excellent collections of papers dealing with the state of the art in numerous topics. For comprehensive detail on Kalman filter theory and applications, see [42-b] and [41]. Integrated navigation systems (with appropriate Kalman filter coverage) are discussed in [42] and [43]. Finally, [13], [14], [34] and [35] may be consulted for additional tutorial information on the stochastic estimation problem and its applications in various disciplines.

For the sake of completeness, I will mention two Bibliographies that have appeared in recent years. For an excellent review of linear filtering theory Kailith [44] should be consulted - some 400 references are cited and a summary (both analytical and descriptive) of the progress of linear filtering during the last three decades is given. Mendel and Gieseking [45] do not include an actual review of the literature, but the 900 references that are given are divided into numerous categories including state estimation, stochastic control, computational analysis and applications (five sections on Aerospace applications).

CHAPTER 2

NAVIGATION AND AIRCRAFT SYSTEMS

2.1 Introduction

Current aircraft missions generally require a highly accurate navigation system. Of particular importance is the necessity to have a precise knowledge of aircraft position especially in scientific research missions, aeromagnetic surveying, and military applications. There are a number of expensive navigation systems currently available which can provide, under normal operating conditions, excellent determination of position. However, each of these systems, if used alone, is prone to errors due to various reasons, especially when the aircraft is operating in a "difficult" environment. One method commonly used to reduce the effects of individual system errors is to combine the available navigation information in some way so that a so-called integrated navigation system is obtained. In this manner, redundant information is available and, in essence, one sub-system can help correct errors present in a second sub-system. For example, trans-Atlantic passenger aircraft carry two or even three inertial navigation systems for improved accuracy and reliability. One interesting point to note is the increased use of computerization in aircraft systems. Time sharing with a small onboard minicomputer, or the use of a dedicated microcomputer, makes navigation system integration possible. Indeed, as will be seen later, the use of a computer not only makes the integration possible, but optimal blending of the information from several systems can be effected.

This chapter briefly describes the navigation problem and introduces some navigation nomenclature used in future sections. The two navigation sub-systems used in this study, Doppler radar and VLF communication stations, are described with emphasis placed on the basic operating principles and errors that are likely to occur. An indepth discussion of airborne navigation and the associated systems is not necessary here since the quoted references provide a solid treatment of the subjects.

2.2 General Navigation Concepts

The underlying purpose of navigation is to provide an indication of the position of a moving vehicle, be it a ship, aircraft, or whatever.

This position indication must be accurate and must be frequently available during the movement of the vehicle. The actual requirements for an aircraft mission may vary, depending upon the nature of the task, (see [46], Ch. 1) but the fundamental quantity considered is the aircraft geographical latitude and longitude. The method of determining this position depends on the navigation system being used, but all position determining methods can be classified as either dead reckoning or position fixing. The dead reckoning scheme uses measurements of aircraft speed and direction in conjunction with a previously known (or estimated) position. An updated position is obtained by extrapolation of the previous position via the measured aircraft motion and its known dynamics.

Position fixing, on the other hand, does not depend on a previously known position. The position fix is usually effected via a combination of measurements with respect to ground based radio navigation stations. Lines of position are established which identify the position of the aircraft [46].

2.3 Accuracy Requirements

The required accuracy of position estimation again is a function of the aircraft mission. For trans-Atlantic passenger flights, for example, the accuracy of the position estimate need not necessarily be extremely good when the aircraft is over the middle of the ocean, providing that the position can be accurately determined (say by some other means) as soon as the aircraft reaches land. Some other missions, however, require continuous and very accurate position estimations. Consider an aerial survey for example. It suffices to say that the accuracy of any navigation system must meet the requirements of the mission. For economical reasons, though, a particular navigation system should not exceed the required accuracy excessively: system cost is often a function of accuracy (to a point). Obviously, a highly accurate and expensive inertial navigation system would not be installed if a less accurate and cheaper unit would suffice. One of the justifications for sub-system integration is that the overall cost of combining poor quality sensors with a small minicomputer is often less than using more accurate navigation systems by themselves. An intelligent integration of the cheaper sensors can often match the accuracy of the more expensive units.

2.4 Dead Reckoning Navigation: Doppler Radar

2.4.1 Dead Reckoning Measurements

As mentioned earlier, dead reckoning navigation requires knowledge of the aircraft motion in terms of speed and direction. Figure 2-1 shows the nomenclature used in describing a dead reckoning navigation scheme. The groundspeed of the aircraft is the vector sum of the along track and across track ground velocities (V_x and V_y). The heading angle, β , is measured by a compass, with the magnetic variation accounted for. The drift angle, α , is caused by a component of the wind speed acting in the across-track direction. By measuring the groundspeed and heading and drift angles, the true track may be computed, and hence dead reckoning navigation is possible.

2.4.2 Doppler Radar Systems

Measurement of aircraft groundspeed and drift angle is possible by using a Doppler radar navigation system which is based upon the Doppler effect. Simply stated, the Doppler effect is the apparent increase or decrease in frequency of a received signal resulting from relative movement between the signal source and the receiver. For aircraft applications, a Doppler radar is mounted on the aircraft and a transmitted microwave signal is beamed to the ground. The signal is then reflected back to the aircraft, where the received signal is measured for a shift in frequency from the original transmitted frequency. In the most basic configuration (i.e. one microwave beam only), the Doppler shift is given by

$$f_D = f_{\text{rec}} - f_{\text{trans}} = \frac{2V \cos \Psi}{\lambda} \quad (2-1)$$

where V = forward velocity of aircraft

Ψ = antenna depression angle (from the horizontal)

λ = wavelength of the transmitted signal.

For a fixed Ψ and λ , the aircraft forward velocity is easily computed after the Doppler shift frequency is measured. In practical arrangements, more than one beam is required in order to measure aircraft along track and across track velocities, and drift angles. The three beam Janus configuration, shown in Figure 2-2, is currently installed onboard the Convair 580 aircraft.

Other possible configurations are described in numerous texts. (See [46], [47], [48]).

Appendix A shows the mechanics of converting the three beam measurements into the quantities required for dead reckoning navigation.

Since the underlying reason for implementing an integrated navigation system is to reduce errors, it is important that errors possibly present in the Doppler system be examined. As shall be seen later, a good understanding of these system errors is important if a reasonable error model is to be specified for use in the Kalman filter.

2.4.3 Doppler Radar Errors

The errors present in the Doppler output quantities (ground-speed and drift angle) can be classified into three categories: [46], [49].

- Random errors
- Bias errors with constant speed and drift angle offsets
- Bias errors with scale factors (percent of speed/angle)

The random errors are caused by sources such as the Doppler fluctuation (noiselike nature of the Doppler signal) and attitude stabilization (See Appendix A). These errors have zero means, and are assumed to be statistically independent from each other.

The net effect of the two types of bias errors is an error in either the Doppler velocity or drift angle, or both, which is primarily a function of system installation, antenna calibration errors, land terrain variations and electronic calibration inaccuracies. The net bias error is often quoted as a percentage (based on time of flight) plus a constant term. For example, the velocity bias error after 100 miles of flying could be $0.25\% + 0.1$ knots. At a groundspeed of 200 knots, this translates into an error of $0.5 + 0.1 = 0.6$ KNOTS. The percentage term is assumed to be a function of distance flown, and presumably increases with the distance. Similar bias errors are present in the drift angle measurement. According to [49], the most significant errors in a Doppler radar are those which occur during over-water operation, due to the state of the sea surface. Although these error producing mechanisms are well known, the actual Doppler radar system is not designed to account for them; this is usually the task of the onboard navigation computer. Despite the numerous other error sources, the errors of modern Doppler systems tend to be relatively low. Tables 2 and

3 of [49] indicate a typical groundspeed error of less than 0.1% of the aircraft velocity.

An appropriate expression for the groundspeed output of a Doppler radar system can be given by:

$$V_{DOP} = V_{TRUE} + B_V + N_V \quad (2-2)$$

where V_{DOP} $\equiv$ the quantity obtained from the Doppler radar system.

V_{TRUE} $\equiv$ true velocity (i.e. no error).

B_V $\equiv$ lumped bias error term.

N_V $\equiv$ random noise at the time of the Doppler output.

- Of course, the true noise quantity would not be known at any given instant, but the statistics of the overall noise process would likely be available.

In a similar fashion, the Doppler drift angle can be represented by

$$\alpha_{DOP} = \alpha_{TRUE} + B_{\alpha} + N_{\alpha} \quad (2-3)$$

Again, the true noise quantity is not known, but its statistical properties are*.

2.4.4 Compass Heading Measurements

The heading angle with respect to true north, necessary for dead reckoning navigation, is obtained from a magnetic compass, or alternatively, a gyrocompass. The magnetic compass, used in this study, measures the heading of the aircraft with respect to magnetic north. Subtracting the magnetic variation gives a heading with respect to true north.

Without going into detail, it suffices to point out that the compass heading also has some inherent errors, and the output of the compass can be assumed to have the form

$$\beta = \psi_{TRUE} + B_{\psi} + N_{\psi} - VAR$$

* Gaussian distribution is assumed.

where $\psi_{\text{TRUE}} \equiv$ true compass heading with respect to magnetic north

$B_{\psi} \equiv$ bias in heading

$N_{\psi} \equiv$ noise in heading

$\text{VAR} \equiv$ magnetic variation

or alternatively,

$$\beta = \beta_{\text{TRUE}} + B_{\beta} + N_{\beta} \quad (2-4)$$

The true track angle of the aircraft (Fig. 2-1) can then be determined by summing the heading and drift angles:

$$\text{TRACK ANGLE} = (\alpha + \beta) \quad (2-5)$$

2.4.5 Effects of Uncorrected Errors

The importance of knowing the biases in Equations 2-2, 2-3 and 2-4 can be readily emphasized by a simple example. Consider an aircraft flying along a straight line track at roughly a constant groundspeed (See Fig. 2-3). After a certain flying time the aircraft position is to be determined.

For this example, assume the following conditions:

$$V_{\text{TRUE}} = 200 \text{ KTS } (0.0555 \text{ NM/sec})$$

$$(\alpha + \beta) = 45^\circ$$

$$B_V = 5 \text{ KTS } (0.001388 \text{ NM/sec})$$

$$B_{\alpha\beta} = 5^\circ$$

After 30 minutes of flying; the true distance of the aircraft from its start position (P_S) is given by

$$P_T - P_S = (0.0555 \frac{\text{NM}}{\text{S}}) (1800) = 100 \text{ NM}$$

and the North and East components are

$$\Delta P_N = 100 \cos (45^\circ) = 70.7 \text{ NM}$$

$$\Delta P_E = 100 \sin (45^\circ) = 70.7 \text{ NM}$$

However, if the velocity and track biases are unknown (i.e. unaccounted for), the calculated distances would be:

$$P_C - P_S = (0.0555 + 0.001388) 1800 = 102.4 \text{ NM}$$

$$\Delta P_{N_C} = 102.4 \cos (45^\circ + 5^\circ) = 65.8 \text{ NM}$$

$$\Delta P_E = 102.4 \sin (45^\circ + 5^\circ) = 78.4 \text{ NM}$$

The errors in the calculated position are:

$$\epsilon_N = 65.8 - 70.7 = -4.9 \text{ NM}$$

$$\epsilon_E = 78.4 - 70.7 = 7.7 \text{ NM}$$

The total (RMS) position error is thus

$$\epsilon_T = (4.9^2 + 7.7^2)^{\frac{1}{2}} = 9.13 \text{ NM}$$

The errors obviously would increase as the flight progressed. Even an approximate knowledge of the biases would allow a reduction in the errors.

2.5 Position Fixing: VLF Communications Stations

There are numerous position fixing systems currently in use, some of the more popular ones being Loran-C, Decca, VOR/DME, Omega and VLF communications signals. The commonality in these schemes is that ground-based radio transmissions are used to determine the position of the aircraft. Various operating frequencies are used and each system has its own characteristics along with advantages and disadvantages. The position fixing method used in this research is the VLF Communications signals approach. For an overview of the other schemes, [46] and [50] may be consulted.

2.5.1 VLF Communications Stations

The U.S. Navy operates a number of Very Low Frequency (VLF) stations to provide a global, highly reliable communications service to ships and submarines [50]. Although not originally intended for navigation use, the VLF signals are phase stabilized with atomic clocks and, hence, can be used to obtain fairly accurate position fixes. The operating frequencies range from 16 to 24 KHz, and station powers tend to be on the order of 500 KW.

(See Table 2-1 for VLF station information and Figure 2-4 and 2-5 for locations). The siting of the nine VLF stations assures adequate reception of at least three signals throughout most of the world at any given time.

Several manufacturers are currently marketing VLF navigation receivers for general aviation applications. Table 2-2 lists some of the more notable units. (From [51]).

2.5.2 Navigation using VLF Signals

There are four basic types of position fixing methods using ground based radio stations. These are shown in Figure 2-6. The rho-theta and theta-theta methods are generally much less accurate than the rho-rho (also known as range-range) or hyperbolic methods, mainly due to the errors incurred in the measurement of the heading angles. The rho-rho method is usually quite accurate, although ambiguities in position can result if only two stations are used. In the hyperbolic scheme, two pairs of stations (i.e. a minimum of three stations) are usually sufficient for an accurate position fix. The last two methods are the ones most commonly used and we consider the hyperbolic method in the following description.

In the VLF hyperbolic scheme, position fixing is accomplished by measuring the differences in reception times of pairs of phase synchronized signals. In effect, at a given (unknown) aircraft position, the difference in the distances from the position to two known VLF stations is measured. This difference in distance then places the aircraft on a so-called hyperbolic line of position (LOP), along which the distance difference is constant. By using two such station pairs (i.e. two separate distance difference measurements), two intersecting lines of position can be determined and the aircraft position is then at the intersection point. References [47], [48] and [52] provide additional extensive coverage on this topic.

In using the VLF signals by themselves for position fixing purposes, the time or distance differences between pairs of individual stations and the aircraft are measured. In anticipation of the Kalman filter integrated system to be described later, it is not the differences that are used, but rather the absolute "distance" in a time sense from the

aircraft to each of the individual stations is the required quantity. For given propagation conditions, a "signal" emitted from a VLF station will take a specific amount of time to reach the aircraft, and this is given by

$$T = \frac{d}{V} \quad (2-6)$$

where T = time for the "signal" to reach the aircraft, seconds.

d = distance in NM from the station to the aircraft.

V = propagation velocity of the signal in NM/S.

Note that if the "signal" was a single pulse of very short duration, we could measure the time interval that it took the pulse to reach the aircraft by knowing when the pulse was transmitted. This requires having a clock on the aircraft which is synchronized with the VLF station's clock. However, the VLF stations transmit continuous carriers, so instead of dealing with short duration pulses, the aircraft VLF receiver is used along with the synchronized aircraft clock (a Rb frequency standard) to measure phase differences of the received signals with the clock. Knowing the frequency of transmission for each station, the "distances" in μ sec from the aircraft to the stations can be determined. Reference [52] details the actual mechanics involved in obtaining these phase difference measurements. It should be emphasized that the phase differences are really measurements of the time it takes the VLF signals to reach the aircraft.

2.5.3 VLF Propagation and Errors

Even though the U.S. Navy VLF communications stations usually provide reliable navigation information and excellent coverage, certain anomalous conditions may occur from time to time which, if not accounted for, will cause significant errors to appear in the phase information and ultimately in any position fix. These anomalies are direct results of changes in the propagation characteristics of the VLF signals. A basic understanding of the fundamental concepts of VLF radio propagation allows one to grasp how and when the various anomalies affect navigation accuracy.

The theory presented here has been condensed from material appearing in [52], [54] and [55]. For additional information, the reader is directed to the considerable number of papers by Swanson, some of which are cited in [53].

At the VLF frequencies between approximately 10 and 25 KHz, the earth behaves like a good electrical conductor, and has a reflection coefficient of approximately +1. Similarly, the ionosphere is also a good reflector with a reflection coefficient approaching -1. The earth and ionosphere can be treated as the boundaries of a spherical waveguide. Propagation of VLF signals depends on the characteristics of these waveguide boundaries and furthermore, at distances greater than about 1000 Km from a transmitter, it is possible to describe the propagation in terms of waveguide modes. Long range propagation beyond the line of sight is achieved through repeated reflection of the transmitted wave between the earth and ionosphere. If one mode is dominant, which is usually the case, then signal phase and amplitude should vary regularly as a function of distance from the transmitter and fluctuations due to interference between various modes should not occur. The phase of the signal will be a function of not only distance but also of the ground conditions and the characteristics of the lower D-region of the ionosphere (the two waveguide boundaries). By far the more important cause of anomalous conditions are the various changes that occur in the D-layer, whether they are predictable or not.

The two classes of anomalies (i.e. changes from the nominal conditions) due to ionospheric variations are:

- (1) predictable seasonal and diurnal changes, and
- (2) unpredictable variations related to solar flares.

The so-called diurnal effect is basically a change in the apparent height of the D-layer. This is caused by a darkness-daylight transition occurring somewhere along the signal path. A typical day-to-night height variation is on the order of 20 Km, which is a considerable change in the dimensions of the earth-ionosphere waveguide (the average height of the D-layer is approximately 70 Km above the earth's surface). This change in height translates into a phase shift in the VLF signal. Hardwick has observed [52] that the average μ sec shift due to diurnal effects for both OMEGA and VLF communications signals can be on the order of 10 - 30 μ sec.

The rate of shift depends on whether the daylight - darkness terminator sweeps along the VLF reception path, or across the path. In addition, the magnitude of the diurnal phase shift varies with distance from the VLF transmitter [54]. The result of a diurnal phase shift which is not accounted for can be a significant error in a given position fix.

> Fortunately, however, the diurnal effect is well known by now, and published skywave corrections allow appropriate compensation of phase shifts. In fact, most current VLF navigation receiver systems automatically account for most diurnal variations.

The second major cause of anomalous conditions are the unpredictable variations due to solar flares. These are characterized by either Sudden Phase Anomalies (SPA's), caused by Sudden Ionospheric Disturbances (SID's), or Polar Cap Absorptions (PCA's). An SID is produced primarily by an increase in the energy and quantity of x-ray flux received from the sun as a result of eruptions, or flares, near sunspots. This causes severe changes in the ionization of the daytime D-region and lowers the effective height of the sunlit portion of the ionosphere. [55]. During the quiet portion of the solar cycle, SID's are probably not too troublesome. When SID's are occurring, a certain amount of forewarning is available to navigators via radio stations WWV and WWVH, operated by the American National Bureau of Standards.

Polar Cap Absorptions are caused by the precipitation of solar particles (primarily protons) in the vicinity of the earth's magnetic poles shortly after large solar flares. These are rare events, and are important principally for signals propagating into or through the Arctic. They may cause very large and irregular perturbations persisting for several days to a week.

Two other sources of error are worthy of mention. Local or distant lightning can be quite disturbing if it lasts for any length of time, and tends to be the major source of signal noise in the VLF region. One other source occurs when a long path signal is stronger than the short path signal for a particular VLF station. Since the position fix assumes short paths for all VLF signals, very severe errors may occur if a long path phase difference is used in the solution.

The anomalies discussed above can cause significant errors in the phase measurement of a VLF signal. As in the case of the Doppler/heading signals, it is necessary to formulate an error model for the VLF signals. A simple model could include three components: a true μ sec value, a non-random bias term, and a random noise term. Hence, a VLF measurement for station i can be expressed as follows:

$$y_i = \text{True VLF } \mu \text{ sec distance} + \text{True VLF Bias} + \text{True VLF noise}$$

or,

$$y_i = \phi_i + B_{\phi_i} + N_{\phi_i} \quad (2-7)$$

If no anomalies are occurring and the noise is zero the VLF measurement will equal the true μ sec distance. Generally, however, the true distance will be corrupted by bias and noise terms, due in part to the sources discussed above. In addition to ionospheric anomalies, several other error sources may be present. The bias and noise terms can consist of the following:

BIAS ERRORS • diurnal shifts

- SPA's
- PCA's
- receiver biases
- Drifting due to a non-stabilized transmitter

NOISE ERRORS • atmospheric disturbances - lightning

- receiver noise (considered negligible, however)
- local disturbances (industrial and electrical noise)

The bias errors are not necessarily constant, but may be characterized by considerable drifting. The diurnal shift is the most obvious example of a drifting bias, but another source of bias drift is a non-stabilized VLF station. For example, at the time of this writing (January 1982) VLF station R has been observed to exhibit a bias drift of approximately 18 μ sec/hour, due to the fact that this station is not stabilized to an atomic clock as the other stations are.

Errors due to noise appear as non-deterministic, zero mean random noise terms, usually with a known variance level. Of course for any one given measurement the exact magnitude of the noise is unknown. The statistical

variance of the noise can change during a flight, but at all times the noise is considered to have a zero mean, since any non zero mean component can be lumped in with the bias term.

2.6 NAE Convair 580 Navigation Systems

The integrated navigation system developed in this report has three main constituent sub-systems: a Doppler radar, a magnetic compass, and a VLF radio receiver. A brief description of the particular units installed on the NAE Convair 580 aircraft will be given here. Additional information may be found in [1] and [52].

(i) Decca Doppler Radar (Type 72)

- three-beam Janus configuration; two forward, one rearward looking beams.
- sample rate of 2 Hz.
- velocity accuracies of 0.3% of true total velocity vector for along and across track velocities.
- usable for altitudes 0-50,000 feet.
- Doppler groundspeed and drift angle are computed by using the conversions detailed in Appendix A.

(ii) Sperry C-12 Magnetic Compass

- dual compass heading reference system
- sample rate of 2 Hz
- accuracy of 0.25 degrees (RMS)*

(iii) Global Navigation GNS-200 VLF receiver

- 7 VLF station capability, with a maximum of six stations used in the integrated navigation system.
- an onboard Rb frequency standard is used in conjunction with the receiver to measure phase differences.
- sample rate of 0.1 Hz (samples every 10 seconds).
- Flight tests done in 1972 produced average position errors of less than one nautical mile when the GNS200 was used by itself (See [52]).

* this is believed to be a rather optimistic quote since actual flight operations have indicated a slightly poorer performance.

The NAE Convair 580 also has numerous other facilities included in its navigation complex: such as a radar altimeter, DME system, Loran-C receiver and video cameras for visual "on-top" data recording. In addition to the GNS-200, a GNS-500 VLF receiver has recently been installed and is currently undergoing performance evaluation and data quality checks. It is proposed that the GNS-500 will eventually replace the GNS-200 VLF system.

2.7 Summary

This chapter has presented details of the basic navigation problem and the two types of navigation methods. The specific dead reckoning and position fixing schemes to be combined in the integrated system have been described with emphasis placed on the development of appropriate error models. Although the Doppler and VLF systems could provide fairly accurate position estimates by themselves in most instances, their inclusion in an integrated system provides redundant information which allows greater accuracy and anomaly compensation.

The actual navigation units installed on the NAE Convair 580 have been briefly described. Additional information is available in the cited references.

CHAPTER 3

KALMAN FILTER THEORY

3.1 Introduction

Since the publication of the initial Kalman filter paper in 1960 [7], there has been a veritable explosion of studies in this area, both in the purely analytical, as well as in the applied aspects of Kalman filtering and least squares estimation. The aerospace and aeronautics fields have been the main focal points for this research, although the theory has been applied to the solution of many other problems.

Quite often, descriptions of Kalman filtering tend to involve a fairly extensive presentation of probability and stochastic theory. Unfortunately, this mathematically rigorous approach can easily mask or prevent the gaining of an intuitive insight into the practical use of Kalman filtering techniques, particularly for those less mathematically inclined. The theory presented here will attempt to explain Kalman filtering in a straightforward and elementary manner, with attention paid to practical implications rather than to strict mathematical rigor. This will hopefully lay the groundwork for the practical Kalman filter navigation system to be detailed later.

This chapter has three main sections. First, general Kalman filter theory will be developed from basic statistical concepts, without reference to any particular system or application. The multi-dimensional state variable formulation of the Kalman filter equations for a dynamic system follows a simpler, more easily understood description of the scalar minimum variance estimation problem. Secondly, the equations for the extended Kalman filter used in the current study will be developed, with references being made to the navigation systems detailed in Chapter 2. Finally, extensions of the basic Kalman filter theory will be described, once again with emphasis on the practical applications to the navigation system at hand.

3.2 Basic Minimum Variance Estimation

In its most basic form, Kalman filtering is nothing more than a process for obtaining an optimal estimate of the value of a quantity when given independent and imperfect estimates of that quantity. In the

simple scalar case, two independent estimates of a quantity are combined in such a manner as to produce a weighted average or mean. The combined average is called an "optimal" estimate because its variance is the minimum possible variance obtainable, given that each independent estimate has a variance associated with it.

Now, one point should be noted. In basic statistical theory, the "variance" of a measurable quantity is defined as the average of the squared deviations from the mean value of the observations of the quantity [56]. However, it is important to bear in mind that the variance can also be intuitively regarded as an indication of the quality of a measured variable or process. For example, the larger the variance, the noisier a signal is, and consequently, we would tend to place less confidence or "weight" on that signal than for a signal with a small variance. Hence, we introduce the concept of using the variance as some sort of weighting factor. This idea will be important throughout the description and development of the Kalman filter.

3.2.1 Single Dimension Optimal Estimation

Assume we are given two independent estimates of a certain quantity or variable x . Call these estimates x_1 and x_2 . Without specifying how we have obtained these estimates, assume also that each has a variance associated with it: σ_1^2 and σ_2^2 respectively. Because neither of the estimates is perfect (i.e. both variances are non-zero), we wish to combine them in such a way as to give us the best possible estimate of x . From the intuitive concept of using the variance as a confidence indicator, we would like the final estimate of x to have the smallest possible variance.

We define the weighted average of the two independent estimates as

$$\bar{x} = k_1 x_1 + k_2 x_2 \quad (3.1)$$

where k_1 and k_2 are weights to be determined. The statistical mean value, or expected value, of $\bar{x}$ is given by

$$\begin{aligned} E(\bar{x}) &= E(k_1 x_1 + k_2 x_2) \\ &= k_1 E(x_1) + k_2 E(x_2) \end{aligned} \quad (3.2)$$

where $E(\cdot)$ is the expectation operator.

We now define the variance of $\bar{x}$ as

$$\sigma_{\bar{x}}^2 = E[(\bar{x} - E(\bar{x}))^2] \quad (3.3)$$

Substituting Equations 3.1 and 3.2 into Equation 3.3 and expanding, we can derive an expression for the variance of the weighted average:

$$\begin{aligned} \sigma_{\bar{x}}^2 &= E[(\bar{x} - E(\bar{x}))^2] \\ &= E[(k_1 x_1 + k_2 x_2 - k_1 E(x_1) - k_2 E(x_2))^2] \\ &= E[(k_1 x_1 - k_1 E(x_1)) + (k_2 x_2 - k_2 E(x_2))]^2 \\ &\quad + 2(k_1 x_1 - k_1 E(x_1))(k_2 x_2 - k_2 E(x_2))] \\ &= k_1^2 E[(x_1 - E(x_1))^2] + k_2^2 E[(x_2 - E(x_2))^2] \\ &\quad + 2k_1 k_2 E[(x_1 - E(x_1))(x_2 - E(x_2))] \end{aligned}$$

Since x_1 and x_2 are independent estimates, x_1 and x_2 are uncorrelated, and the third term in the above equation is zero.

Hence,

$$\sigma_{\bar{x}}^2 = k_1^2 \sigma_1^2 + k_2^2 \sigma_2^2 \quad (3.4)$$

Now we wish to determine k_1 and k_2 to give a minimum value of the variance of $\bar{x}$. We can express k_2 in terms of k_1 by considering that if, say, x_2 had zero variance, we would use only this perfect estimate of x and forget about x_1 . Hence from Equation 3.1, let $k_2 = k$ and $k_1 = 1-k$. Equation 3.4 can then be written as

$$\sigma_{\bar{x}}^2 = (1-k)^2 \sigma_1^2 + (k)^2 \sigma_2^2 \quad (3.5)$$

To determine the value of k to give a minimum variance solution, partially differentiate Equation 3.5 with respect to k and set to zero. Hence,

$$\frac{\partial \sigma_{\bar{x}}^2}{\partial k} = 2k\sigma_2^2 - (1-k)2\sigma_1^2 = 0; \quad (3.6)$$

and solving for $\hat{k}$ results in,

$$\hat{k} = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \quad (3.7)$$

The "hat" (^) above k is used to indicate an optimal value or estimate of the weighting factor. Substitution of $\hat{k}$ into Equations 3.1 and 3.5 gives expressions for the optimal estimates of x and the variance of $\hat{x}$

$$\begin{aligned} \hat{x} &= (1-\hat{k})x_1 + \hat{k}x_2 \\ &= x_1 - \hat{k}(x_1 - x_2) \\ &= \frac{\sigma_2^2 x_1 + \sigma_1^2 x_2}{\sigma_1^2 + \sigma_2^2} \end{aligned} \quad (3.8)$$

$$\begin{aligned} \sigma_{\hat{x}}^2 &= (1-\hat{k})^2 \sigma_1^2 + \hat{k}^2 \sigma_2^2 \\ &= \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} = \hat{k} \sigma_2^2 = \sigma_1^2 (1-\hat{k}) \end{aligned} \quad (3.9)$$

To summarize, two independent estimates of a quantity can be combined to give an optimal estimate of the quantity as follows:

$$\begin{aligned} \hat{x} &= x_1 - \hat{k}(x_1 - x_2) \\ \sigma_{\hat{x}}^2 &= \sigma_1^2 (1-\hat{k}) \\ \hat{k} &= \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \end{aligned} \quad (3.10)$$

Of course, the optimal estimate is not perfect, but it is the best possible estimate we can obtain from the available data.

3.2.2 Optimal Estimation Using Measurement Data

In Section 3.2.1 we assumed that two independent estimates of x were available, without making mention of how these quantities were obtained. As shall be seen later, in the formulation of a practical Kalman filter applied to a dynamic system one of these estimates is usually based upon the

last optimal estimate, used in conjunction with the known dynamics of the system. A second estimate is obtained through measurement of x_2 , i.e. x_2 is estimated by a measurement $y = hx_2$, where h is a scale factor inherent in the measurement process. The variance of the measurement is then $\sigma_y^2 = h^2 \sigma_{x_2}^2$. The optimal estimate of x can now be written as

$$\begin{aligned} \hat{x} &= x_1 - k(x_1 - \frac{y}{h}); \\ &= x_1 - \frac{k}{h}(hx_1 - y). \end{aligned} \quad (3.11a)$$

Now rewrite the expression for the optimal gain k in terms of the variances σ_1^2 and σ_y^2 :

$$\text{Let } \hat{k} = \frac{k}{h} = \frac{\sigma_1^2}{(\sigma_1^2 + \sigma_{x_2}^2)h};$$

$$\text{but } \sigma_{x_2}^2 = \frac{\sigma_y^2}{h^2};$$

$$\text{so } \hat{k} = \frac{\sigma_1^2}{(\sigma_1^2 + \frac{\sigma_y^2}{h^2})h} = \frac{h\sigma_1^2}{h^2\sigma_1^2 + \sigma_y^2} \quad (3.11b)$$

Hence, if we have two estimates of x , x_1 based on the system dynamics, and $y_2 = hx_2$, a measured estimate of x , then

$$\hat{x} = x_1 - \hat{k}(hx_1 - y) = x_1 + \hat{k}(y - hx_1) \quad (3.11c)$$

is the optimal estimate of x , where the optimal weighting is

$$\hat{k} = \frac{h\sigma_1^2}{h^2\sigma_1^2 + \sigma_y^2} \quad (3.12)$$

From a practical viewpoint, we can consider x_1 as an initial rough estimate of x , and the measurement y is used to improve this estimate through the term $\hat{k}(y - hx_1)$, giving the final optimal estimate, $\hat{x}$. $\hat{k}$ is the optimal gain, and the quantity $(y - hx_1)$ is variably known as the residual or innovation quantity. The residual is the difference between the theoretical estimate and a measure-

ment estimate, and represents new information brought into the solution by the measurement.

3.2.3 Multi-Dimensional Optimal Estimation

In the preceding section a single dimension optimal estimate was formulated. In general, estimation problems, however, we are usually interested in estimating not simply one quantity or variable but, rather, we seek optimal estimates of several different quantities at the same time. The derivation of optimal estimation equations for a multi-dimensional problem is best done by using matrix and vector notation. The material presented here follows that found in [57].

Consider an n -dimensional vector quantity $\underline{X}$. We wish to obtain an optimal estimate of this vector, based on two independent and imperfect estimates of $\underline{X}$. Hence, we seek estimates of $\{x_1, x_2, \dots, x_n\}$, given the independent vector estimates $\underline{X}_1 = \{x_{11}, x_{12}, \dots, x_{1n}\}$ and $\underline{X}_2 = \{x_{21}, x_{22}, \dots, x_{2m}\}$.

In accordance with the idea of using a measurement as one of the estimates, assume the second estimate vector is not just $\underline{X}$, but rather a measurement vector, $\underline{Y}_2$, based on $\underline{X}_2$.

$$\underline{Y}_2 = \underline{H}\underline{X}_2,$$

where $\underline{H}$ is known as the measurement matrix, and operates on $\underline{X}_2$ in much the same manner as does the scalar h in Equation 3.11c. Note that we consider the general case here - the dimensions of $\underline{X}_1$ and $\underline{Y}_2 = \underline{H}\underline{X}_2$ are not necessarily identical. The dimensions of the vectors and matrices are: $\underline{X}_1(n \times 1)$, $\underline{X}_2(n \times 1)$, $\underline{Y}_2(m \times 1)$, $\underline{K}(n \times m)$, and $\underline{H}(m \times n)$. We assume that $\underline{X}_1$ and $\underline{Y}_2$ have covariance matrices $\underline{P}$ and $\underline{R}$ respectively. The optimal estimate of the weighted mean of the independent estimates can be expressed as

$$\bar{\underline{X}} = \underline{X}_1 - \underline{K}(\underline{H}\underline{X}_1 - \underline{Y}_2) \quad (3.13)$$

where $\underline{Y}_2 = \underline{H}\underline{X}_2$.

The covariance of $\bar{\underline{X}}$ is given by

$$\bar{\underline{P}} = E\{[\bar{\underline{X}} - E(\bar{\underline{X}})][\bar{\underline{X}} - E(\bar{\underline{X}})]^T\}; \quad (3.14)$$

where $[\cdot]^T$ indicates the matrix transpose of $[\cdot]$. Similarly, the covariance matrices of the initial independent estimate vectors are defined as

$$\underline{P}' = E\{[\underline{x}_1 - E(\underline{x}_1)][\underline{x}_1 - E(\underline{x}_1)]^T\}$$

$$\underline{R} = E\{[\underline{y}_1 - E(\underline{y}_1)][\underline{y}_1 - E(\underline{y}_1)]^T\}$$

Assuming $\underline{x}_1$ and $\underline{y}_1$ are independent (i.e. uncorrelated), Equation 3.14 can be written as

$$\underline{\bar{P}} = (\underline{I} - \underline{K}\underline{H}) \underline{P}' (\underline{I} - \underline{K}\underline{H})^T + \underline{K}\underline{R}\underline{K}^T. \quad (3.15)$$

As in Section 3.2.1, we wish to minimize the variances of the estimates of $(x_1, x_2, \dots, x_n)$. This can be done by minimizing the diagonal elements of $\underline{\bar{P}}$. In other words, we seek a gain matrix $\underline{K}$ which will cause each diagonal element of $\underline{\bar{P}}$ to be a minimum. Through matrix manipulation, the required gain matrix is shown to be

$$\underline{K} = \underline{P}'\underline{H}^T [\underline{H}\underline{P}'\underline{H}^T + \underline{R}]^{-1}. \quad (3.16)$$

This result is arrived at in slightly different manners in [14] and [57]. Substituting Equation 3.16 into Equation 3.15 we obtain

$$\underline{\bar{P}} = \underline{P}' - \underline{K}\underline{H}\underline{P}' \quad (3.17)$$

In summary, the solution to the multi-dimensional optimal estimation problem is given by

$$\hat{\underline{X}} = \underline{X}_1 - \underline{K}(\underline{H}\underline{X}_1 - \underline{Y}_1)$$

$$\underline{\bar{P}} = \underline{P}' - \underline{K}\underline{H}\underline{P}'$$

$$\underline{K} = \underline{P}'\underline{H}^T [\underline{H}\underline{P}'\underline{H}^T + \underline{R}]^{-1}$$

$$\underline{Y}_1 = \underline{H}\underline{X}_1$$

3.3 Physical Dynamic System Representation

3.3.1 System Description

In describing a complex physical process, it is useful to represent the dynamics of the system at hand by a state-space approach. In such a manner, any physical process may be described in terms of a set of state variables. At any point in time the state of the system is expressed by the state variables that have been chosen to represent the process. See [58] for additional details on state-space representations.

In general, any physical process is not completely deterministic because of certain random perturbations inherent in the system. The propagation of a process may be modelled as having one part which is deterministic (we know the basic equations of motion governing the process), and a second part which consists of the random fluctuations or perturbations. The random "noise" is assumed to have a zero mean, and a non-zero variance.

The state of an n^{th} order linear discrete* process at a given time may be represented by

$$\underline{X}_{k+1} = \underline{\Phi}_k \underline{X}_k + \underline{G}_k \underline{U}_k \quad (3.18)$$

where $\underline{X}_{k+1}$ state at time t_{k+1} , n state variables

$\underline{X}_k$ state at time t_k

$\underline{\Phi}_k$ $n \times n$ matrix representing the known dynamics of the process, derived from the equations of motion

$\underline{U}_k$ ℓ ($\ell \leq n$) noise terms (random perturbations) present at time t_k

$\underline{G}_k$ noise "transition" matrix representing the effects of the noise terms on the state variables of the system. ($\underline{G}_k$ is NOT necessarily identical to $\underline{\Phi}_k$ and in general $\underline{G}_k \neq \underline{I}$).

Two points should be noted. Equation 3.18 represents a Markov process: the present state, the known system dynamics and the nature of the perturbing noise are sufficient to predict the future state; the past has no influence on the future. Secondly, it is not necessary for the transition matrix $\underline{\Phi}$ or the noise "transition" matrix $\underline{G}$ to be constant in a time sense.

* we will consider only discrete systems in this study

Similarly, the nature of the noise may also change (i.e. the noise variances need not be constant). In general, however, these quantities should be constant during any one time step.

The noise process is assumed to be a Gaussian one (most physical noise processes have probability distributions which are very nearly normal). Each noise term has a variance, σ^2 , associated with it. Also, a covariance matrix for the system noise can be defined as

$$Q_k = \begin{bmatrix} u_1 & \dots & 0 \\ \vdots & u & \vdots \\ 0 & \dots & u_{\ell} \end{bmatrix}$$

3.3.2 Measurement Description

Section 3.2.3 indicated that a measurement vector is used in the optimal state estimate procedure. In this section the state-space representation of such a measurement on a physical process is formulated.

Consider the state of a system at time t_{k+1} . Through suitable instrumentation we can make measurements on the system, but these measurements do not necessarily have to be of any of the individual state variables (indeed, any or all of the state variables may not be directly measurable or even available to us). However, the measurements will be some function of the state at the time considered. For example, consider the geographical location of an aircraft in flight (the state) represented by the state variables LATITUDE and LONGITUDE. It is impossible to make direct measurements of LATITUDE and LONGITUDE, but we can make measurements of RF phase information from particular radio stations (which are functions of the aircraft position) and from these deduce the values of the state variables by using appropriate equations.

Each measurement will usually have some Gaussian noise associated with it. From an empirical analysis of the measurement process and instrumentation, a noise variance can usually be assigned to each individual measurement. Note, of course, that the actual contribution of the noise to the measurement is unknown.

From the above reasoning, the measurement process can be represented as

$$\underline{Y}_{k+1} = \underline{H}_{k+1} \underline{X}_{k+1} + \underline{V}_{k+1} \quad (3.19)$$

where $\underline{Y}_{k+1}$ (mx1) measurement vector at time t_{k+1}

$\underline{X}_{k+1}$ true state of the system (n^{th} order)

$\underline{H}_{k+1}$ (mxn) matrix representing the relationship between the state and the measurements

$\underline{V}_{k+1}$ (mx1) noise vector.

A measurement noise covariance matrix can be defined as

$$\underline{R}_{k+1} = \begin{bmatrix} \sigma_{v_1}^2 & & & 0 \\ & \sigma_{v_2}^2 & & \\ & & \ddots & \\ 0 & & & \sigma_{v_m}^2 \end{bmatrix} \quad (\text{mxm}) \text{ matrix.}$$

3.4 Kalman Filter Equations for Dynamic Systems

Sufficient background material has been presented to allow a formulation of a set of equations defining a Kalman filter for a linear dynamic system. The system is assumed to have the form given by Equation 3.18. It is desired to obtain optimal estimates of the state variables given

- a theoretical update of the state based on the last optimal state estimate and the system dynamics, and
- a set of measurements on the system (which are functions of the state).

Because the system noise (referred to as the "plant noise") found in Equation 3.18 is not deterministic, but has a zero mean, the best possible "plant" update of the state is simply

$$\underline{X}'_{k+1} = \underline{\Phi}_k \underline{X}_k; \quad (3.20)$$

where the prime indicates an update based on the plant only. The variance of this plant update is easily defined as follows. The optimal state estimate consists of a true state plus an error associated with the state:

$$\hat{\underline{x}}_{k+1} = \underline{x}_{k+1} + \underline{e}_{k+1}$$

Hence Equation 3.20 may be written as

$$\begin{aligned}\underline{x}'_{k+1} &= \underline{I}_k (\underline{x}_k + \underline{e}_k) \\ &= \underline{I}_k \underline{x}_k + \underline{I}_k \underline{e}_k\end{aligned}$$

but since $\underline{x}_{k+1} = \underline{I}_k \underline{x}_k + \underline{G}_k \underline{u}_k$, we have

$$\underline{x}'_{k+1} = \underline{x}_{k+1} - \underline{G}_k \underline{u}_k + \underline{I}_k \underline{e}_k.$$

Now, $\underline{x}_{k+1}$ is the true state so its variance is zero by definition. The variance of $\underline{e}_k$ is simply the variance of $\hat{\underline{x}}_k$ itself - $\bar{\underline{p}}_k$ or just $\underline{p}_k$, and the covariance of $\underline{u}_k$ has already been defined as $\underline{Q}_k$. Thus the variance of the plant state update is given by:

$$\underline{p}'_{k+1} = \underline{G}_k \underline{Q}_k \underline{G}_k^T + \underline{I}_k \underline{p}_k \underline{I}_k^T, \quad (3.21a)$$

since $\underline{u}_k$ and $\underline{e}_k$ are independent.

In effect, all of the information pertaining to the quantity $\underline{x}_1$ given in Section 3.2.3 has been defined. Realizing that $\underline{x}'_{k+1} = \underline{x}_1$, and using Equations 3.13, 3.16 and 3.17, the optimal estimate of the state at time t_{k+1} is

$$\hat{\underline{x}}_{k+1} = \underline{x}'_{k+1} - \underline{K}_{k+1} (\underline{H}_{k+1} \underline{x}'_{k+1} - \underline{y}_{k+1}), \quad (3.21b)$$

where $\underline{K}_{k+1} = \underline{p}'_{k+1} \underline{H}_{k+1}^T (\underline{H}_{k+1} \underline{p}'_{k+1} \underline{H}_{k+1}^T + \underline{R}_{k+1})^{-1}$. The matrices in these equations have been defined above. The m quantities defined by $(\underline{H}_{k+1} \underline{x}'_{k+1} - \underline{y}_{k+1})$ are known as the measurement residuals, as previously defined in Subsection 3.2.2. The covariance of the optimal estimate is

$$\underline{p}_{k+1} = \underline{p}'_{k+1} - \underline{K}_{k+1} \underline{H}_{k+1} \underline{p}'_{k+1}. \quad (3.21c)$$

In summary, a set of five equations can be used to define a Kalman filter for a linear, discrete dynamic system:

$$\begin{aligned}
 \underline{X}'_{k+1} &= \underline{\Phi}_k \underline{X}_k \\
 \underline{P}'_{k+1} &= \underline{\Phi}_k \underline{P}_k \underline{\Phi}_k^T + \underline{G}_k \underline{Q}_k \underline{G}_k^T \\
 \underline{K}_{k+1} &= \underline{P}'_{k+1} \underline{H}_{k+1}^T [\underline{H}_{k+1} \underline{P}'_{k+1} \underline{H}_{k+1}^T + \underline{R}_{k+1}]^{-1} \\
 \underline{X}_{k+1} &= \underline{X}'_{k+1} - \underline{K}_{k+1} (\underline{H}_{k+1} \underline{X}'_{k+1} - \underline{Y}_{k+1}) \\
 \underline{P}_{k+1} &= \underline{P}'_{k+1} - \underline{K}_{k+1} \underline{H}_{k+1} \underline{P}'_{k+1}
 \end{aligned} \tag{3.22}$$

The first two equations in the set of Equations 3.22 "extrapolate" between the measurement times. The last three equations represent the optimal updating of this extrapolated estimate after a measurement has been made.

3.5 Quasi-Linearization: Extended Kalman Filter

Up to this point, the dynamic system has been assumed to be linear. In Equation 3.18, for example, the propagation of the state is governed in part by a linear state transition matrix, $\underline{\Phi}$. A problem arises, however, if we are faced with a system whose motion is a nonlinear one. This is often the case in real life situations and navigation plant dynamics and measurements rarely follow a pattern definable by a linear equation. Hence, in the context of the navigation problem, Equation 3.22 is not completely accurate or usable. This section defines a variation of the basic linear Kalman filter, which can be applied to a nonlinear system. This so-called "quasi-linear" Kalman filter follows from the theory of the extended Kalman filter [14], [34], [35].

In very basic terms, the "quasi-linear" approach uses the best available estimates of the state variables as references in defining a quasi-linear state transition matrix, and a quasi-linear observation matrix. In effect, approximate first order linearization takes place about the nominal (or reference) state trajectory defined by the best estimates of the state variables.

Now, consider a nonlinear system defined by the following dynamics:

$$\underline{x}_{k+1} = \underline{f}(\underline{x}_k) + \underline{G}_k \underline{u}_k \quad (3.23)$$

The $(n \times 1)$ vector $\underline{f}$ has elements which are nonlinear functions of the state $\underline{x}$ (as opposed to the linear transition matrix). Similarly, a nominal measurement equation may be defined as

$$\underline{y}_{k+1} = \underline{h}(\underline{x}_{k+1}) + \underline{v}_{k+1} \quad (3.24)$$

Here $\underline{h}$ is an $(m \times 1)$ vector whose elements are also nonlinear functions of the state at time t_{k+1} . The similarity between these nonlinear equations and the linear system defined by Equations 3.18 and 3.19 is readily apparent.

The formulation of the quasi-linear Kalman filter will now be detailed. In Section 3.4, the best "plant" update of the state was given by $\underline{x}'_{k+1} = \underline{\Phi}_k \underline{x}_k$. In the nonlinear case, the best plant update is

$$\underline{x}'_{k+1} = \underline{f}_{k+1}(\underline{x}_k) = \underline{x}_k + \int_{t_k}^{t_{k+1}} \underline{f}^*(\underline{x}_k) dt \quad (3.25)$$

$\underline{x}_k$ is the optimal state estimate at time t_k . The second term represents the extrapolation from t_k to the current update time, t_{k+1} . Note that $\underline{f}^*$ is not the same as $\underline{f}(\underline{x}_k)$, but rather, represents the original continuous-time plant nonlinearity. Integration of this nonlinear function over the update interval gives the theoretical extrapolation of the state from t_k to t_{k+1} .

A quasi-linear state transition matrix may be defined as one having elements [34]

$$\Phi_{ij} = \left. \frac{\partial f_i(\underline{x})}{\partial x_j} \right|_{\underline{x} = \underline{x}_k} \quad \begin{matrix} i = 1, \dots, n \\ j = 1, \dots, n \end{matrix} \quad (3.26)$$

This "transition matrix" represents the change in the nonlinear functions of $\underline{x}$ with respect to the state $\underline{x}$, and is linearized at each measurement time by evaluating the partial derivative at the current best state estimate.

Similarly, a quasi-linear observation or measurement matrix can be defined as having elements

$$H_{k+1, i j} = \left. \frac{\partial h_i(\underline{x})}{\partial x_j} \right|_{\underline{x} = \hat{\underline{x}}_k} \quad \begin{matrix} i = 1, \dots, m \\ j = 1, \dots, n \end{matrix} \quad (3.27)$$

These new matrices bear close resemblance to those found in Equation 3.22. A set of equations can now be formulated which will define a Kalman filter applicable to a nonlinear system:

$$\begin{aligned} \underline{x}'_{k+1} &= \underline{f}(\underline{x}_k) = \underline{x}_k + \int_{t_k}^{t_{k+1}} \underline{f}^*(\underline{x}_k) dt. \\ \underline{P}'_{k+1} &= \underline{\Phi}_k \underline{P}_k \underline{\Phi}_k^T + \underline{G}_k \underline{Q}_k \underline{G}_k^T \\ \underline{K}_{k+1} &= \underline{P}'_{k+1} \underline{H}_{k+1}^T [\underline{H}_{k+1} \underline{P}'_{k+1} \underline{H}_{k+1}^T + \underline{R}_{k+1}]^{-1} \quad (3.28) \\ \hat{\underline{x}}_{k+1} &= \underline{x}'_{k+1} + \underline{K}_{k+1} [\underline{y}_{k+1} - \underline{h}(\underline{x}'_{k+1})] \\ \underline{P}_{k+1} &= \underline{P}'_{k+1} - \underline{K}_{k+1} \underline{H}_{k+1} \underline{P}'_{k+1} \end{aligned}$$

Note that the $\underline{\Phi}$ and $\underline{H}$ matrices are those defined by Equations 3.26 and 3.27. Also, it is important to realize that the nonlinear plant motion is used in the plant update* but $\underline{\Phi}$ and $\underline{H}$ are quasi-linear, being based on the nonlinear vectors $\underline{f}$ and $\underline{h}$.

A more detailed presentation of the quasi-linear extended Kalman filter may be found in [33], [34] and [35].

* and also in the calculation of $\underline{h}$

$$R = \begin{bmatrix} \phi_1 & & & & & \\ & \phi_2 & & & & \\ & & \phi_3 & & 0 & \\ & & & \phi_4 & & \\ 0 & & & & \phi_5 & \\ & & & & & \phi_6 \end{bmatrix} \quad (4.7)$$

4.7 Quasi-linear Observation Matrix

As given by Equation 3.27, a quasi-linear observation matrix is defined by the elements

$$H_{k+1,ij} = \left. \frac{\partial h_i(\underline{X})}{\partial X_j} \right|_{\underline{X} = \underline{X}_k} \quad \begin{matrix} i = 1, \dots, 6 \\ j = 1, \dots, 10 \end{matrix}$$

The elements of $\underline{h}$ cannot be expressed explicitly in an analytic fashion but the partials defining the elements of H_{k+1} can be determined through use of the Sodano inverse routine and the method formulated by Leach [59]. The partials are defined by

$$\left. \frac{\partial h_1(\underline{X})}{\partial X_1} \right|_{\underline{X} = \underline{X}_k} = (-60/V_S) \cdot \cos(\hat{A}z) \quad (4.8)$$

$$\left. \frac{\partial h_1(\underline{X})}{\partial X_2} \right|_{\underline{X} = \underline{X}_k} = (60/V_S) \cdot \sin(\hat{A}z) \cdot \cos(X_1 \cdot \pi/180) \quad (4.9)$$

where V_S is the speed of propagation of the VLF signal

$\hat{A}z$ is the azimuth (heading) from the aircraft position
to VLF station i

4.2 State Variables Chosen

The state of the integrated navigation system considered here is defined by a set of ten state variables, with the discrete Kalman filter updating the state estimate every ten seconds. The variables are defined as:

$$\begin{aligned}
 x_1 &= \text{PLAT} && \text{geographical latitude of the aircraft (degrees)} \\
 x_2 &= \text{PLONG} && \text{geographical longitude of the aircraft (degrees)} \\
 x_3 &= B_V && \text{bias in Doppler groundspeed (NM/Sec)} \\
 x_4 &= B_{\alpha\beta} && \text{bias in aircraft track angle (radians)} \\
 x_5 &= B_{\phi_1} && \text{bias in VLF station 1 phase measurement (\mu SEC)} \\
 x_6 &= B_{\phi_2} && \text{bias in VLF station 2 phase measurement (\mu SEC)} \\
 &\vdots && \vdots \\
 x_{10} &= B_{\phi_6} && \text{bias in VLF station 6 phase measurement (\mu SEC)}
 \end{aligned}$$

The choice of aircraft latitude and longitude as state variables is obvious. The other variables were chosen as a consequence of the assumed Doppler/compass and VLF biases described in Chapter 2. The fourth state variable, $B_{\alpha\beta}$, is the sum of the drift angle and heading angle biases: it was felt that because these two biases act together in the plant dynamics to form an overall true track bias, it would be unnecessary to separate them in the state definition. This approach thus allows ten variables to be used rather than eleven. Another state formulation that was considered would have included the basic Doppler radar mechanics. The three individual Doppler beam frequencies would have been modelled in terms of their biases and noise, giving a 12th order state formulation. This was felt to be unnecessarily complicated, and the Doppler velocity bias is expressed solely by one variable, B_V . In the computer simulation routine, the three beam frequencies are used, however, but simply to define the Doppler radar system simulator.

4.3 Plant Update Equations

The plant position update equations are defined in terms of the Doppler/compass quantities and the state variables chosen. Recall the best plant update, Equation 3.25 :

$$\underline{X}'_{k+1} = \underline{f}(\underline{\hat{X}}_k) = \underline{\hat{X}}_k + \int_{t_k}^{t_{k+1}} \underline{f}^*(\underline{\hat{X}}_k) dt$$

The plant position update is given by :

$$\begin{aligned} P'_{LAT_{k+1}} = X'_{1_{k+1}} &= \hat{X}_{1_k} + \int f^*(\hat{X}_k) dt \\ P'_{LONG_{k+1}} = X'_{2_{k+1}} &= \hat{X}_{2_k} + \int f^*(\hat{X}_k) dt \end{aligned} \quad (4.1)$$

The first term, $\hat{X}_k$, is the state estimate at time t_k . The second term in Equation 4.1 represents the increment in the position of the aircraft based upon the Doppler/compass plant measurements, without including the (unknown) noise terms. The plant update equations are defined as follows, with their actual derivations given in Appendix B.

$$\begin{aligned} f_1(\hat{X}_k) = X'_{1_{k+1}} &= \hat{P}_{LAT_k} + [A1_k - A3_k \cdot \hat{B}_{V_k}] \sin(\hat{B}_{\alpha\beta k}) \\ &\quad + [A2_k - A4_k \cdot \hat{B}_{V_k}] \cos(\hat{B}_{\alpha\beta k}) \\ &= \hat{X}_{1_k} + [A1_k - A3_k \cdot \hat{X}_{3_k}] \sin(\hat{X}_{4_k}) \\ &\quad + [A2_k - A4_k \cdot \hat{X}_{3_k}] \cos(\hat{X}_{4_k}) \\ &\quad + \frac{[A1_k - A3_k \cdot \hat{B}_{V_k}] \cos(\hat{B}_{\alpha\beta k})}{\cos(\hat{P}_{LAT_k} \cdot \pi / 180)} \\ &\quad + \frac{[A2_k - A4_k \cdot \hat{B}_{V_k}] \sin(\hat{B}_{\alpha\beta k})}{\cos(\hat{P}_{LAT_k} \cdot \pi / 180)} \end{aligned} \quad (4.2)$$

Equation 4.2 (continued) :

$$f_2(\hat{x}_k) = \hat{x}_{2k} - \frac{[A1_k - A3_k \cdot \hat{x}_{3k}] \cos(\hat{x}_{4k}) - [A2_k - A4_k \cdot \hat{x}_{3k}] \sin(\hat{x}_{4k})}{\cos(\hat{x}_{1k} \cdot \pi/180)}$$

where $A1_k = 1/60 \int V_{DOP} \sin(\alpha+\beta) dt$

$A2_k = 1/60 \int V_{DOP} \cos(\alpha+\beta) dt$

$A3_k = 1/60 \int \sin(\alpha+\beta) dt$

$A4_k = 1/60 \int \cos(\alpha+\beta) dt$

The bias updates are simply given by :

$$f_3(\hat{x}_k) = \hat{B}_{v_k} = \hat{x}_{3k}$$

$$f_4(\hat{x}_k) = \hat{B}_{\alpha\beta k} = \hat{x}_{4k}$$

$$f_5(\hat{x}_k) = \hat{B}_{\phi k} = \hat{x}_{5k}$$

$$\vdots$$

$$f_{10}(\hat{x}_k) = \hat{B}_k = \hat{x}_{10k}$$

since dB/dt is assumed zero (the biases are assumed to be constant).

4.4 Plant Noise Processes

The nonlinear system as defined by Equation 3.23 is given by

$$\underline{x}_{k+1} = \underline{f}(\underline{x}_k) + \underline{G}_k \underline{u}_k$$

The plant updates have already been defined in

the previous section. We now require a definition for the plant noise matrix, G_k . In Appendix B, the plant equations assume noise terms present in the Doppler/heading quantities. It is easy to see from Equations B.6 and B.7 that the noise matrix is given by

$$G_k = \begin{bmatrix} A6_k & A5_k \\ A8_k & A7_k \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \quad (4.3)$$

$$\text{where } A5_k = 1/60(V_{DOP} - \hat{B}_{v_k}) \sin(\alpha + \beta - \hat{B}_{\alpha\beta k}) \Big|_{t=t_k+5}$$

$$A6_k = -1/60 \cos(\alpha + \beta - \hat{B}_{\alpha\beta k}) \Big|_{t=t_k+5}$$

$$A7_k = \frac{(V_{DOP} - \hat{B}_{v_k}) \cos(\alpha + \beta - \hat{B}_{\alpha\beta k})}{60 \cos(\hat{P}_{LAT_k} \cdot \pi/180)} \Big|_{t=t_k+5}$$

$$A8_k = \frac{\sin(\alpha + \beta - \hat{B}_{\alpha\beta k})}{60 \cos(\hat{P}_{LAT_k} \cdot \pi/180)} \Big|_{t=t_k+5}$$

The integrated noise processes, $U_{v_k}^*$ and $U_{\alpha\beta k}^*$, have zero means, and variances σ_v^{*2} and $\sigma_{\alpha\beta}^{*2}$ respectively.

These variances are not equal to the variances of the raw noise processes, and must be determined if the plant covariance matrix is to be accurately set. The noise processes are integrated via the common trapezoidal digital method. Appendix C develops expressions for the statistics of a noise process that has been integrated in this fashion. In summary, the integrated noise variances are given by:

$$\sigma_V^{*2} = (\Delta t/2)^2 (n-1)^2 (4) \cdot \sigma_V^2/n$$

where Δt is the interval between Doppler data samples (0.5 sec)

n is the number of data samples in each update interval (21)

σ_V^2 is the variance of the raw Doppler velocity noise

Similarly,

$$\sigma_{\alpha\beta}^{*2} = (\Delta t/2)^2 (n-1)^2 (4) \cdot \sigma_{\alpha\beta}^2/n$$

The Doppler/heading noise autocovariance matrix is defined by

$$Q_k \equiv \begin{bmatrix} \sigma_V^{*2} & 0 \\ 0 & \sigma_{\alpha\beta}^{*2} \end{bmatrix} \quad (4.4)$$

4.5 Quasi-linear State Transition Matrix

As defined in Section 3.5, the Φ matrix is defined as

$$\Phi_{kij} \equiv \left. \frac{\partial f_i(X)}{\partial X_j} \right|_{\hat{X} = \hat{X}_k} \quad \begin{matrix} i=1, \dots, n \\ j=1, \dots, n \end{matrix} \quad (n=10)$$

Each of the elements of $f(\hat{X}_k)$ given in Section 4.3 may be partially differentiated with respect to the ten state variables, and evaluated at the last optimal state estimate. Fortunately, the effort involved in calculating these 100 partial derivatives is not as great as one might imagine: most of the terms are either 0 or 1, with only five other

partials actually needing evaluation. These are given by:

$$\begin{aligned}
 \partial f_1 / \partial X_{3k} &= -A3_k \cdot \sin(\hat{X}_{4k}) - A4_k \cdot \cos(\hat{X}_{4k}) \\
 \partial f_1 / \partial X_{4k} &= (A1_k - A3_k \cdot \hat{X}_{3k}) \cos(\hat{X}_{4k}) - (A2_k - A4_k \cdot \hat{X}_{3k}) \sin(\hat{X}_{4k}) \\
 \partial f_2 / \partial X_{1k} &= -[(A1_k - A3_k \cdot \hat{X}_{3k}) \cos(\hat{X}_{4k}) + (A4_k \cdot \hat{X}_{3k} - A2_k) \sin(\hat{X}_{4k})] \\
 &\quad \cdot \frac{\pi \cdot \sin(\hat{X}_{1k} \cdot \pi/180)}{180 \cos^2(\hat{X}_{1k} \cdot \pi/180)} \\
 \partial f_2 / \partial X_{3k} &= [A3_k \cdot \cos(\hat{X}_{4k}) - A4_k \cdot \sin(\hat{X}_{4k})] \cdot \frac{1}{\cos(\hat{X}_{1k} \cdot \pi/180)} \\
 \partial f_2 / \partial X_{4k} &= [(A1_k - A3_k \cdot \hat{X}_{3k}) \sin(\hat{X}_{4k}) - (A4_k \cdot \hat{X}_{3k} - A2_k) \cos(\hat{X}_{4k})] \\
 &\quad \cdot \frac{1}{\cos(\hat{X}_{1k} \cdot \pi/180)}
 \end{aligned} \tag{4.5}$$

where A1, A2, A3 and A4 are identical to the coefficients defined in Section 4.3.

The quasi-linear state transition matrix is defined as:

$$\Phi_k = \begin{bmatrix} 1 & 0 & \partial f_1 / \partial X_{3_k} & \partial f_1 / \partial X_{4_k} & 0 & \dots & 0 \\ \partial f_2 / \partial X_{1_k} & 1 & \partial f_2 / \partial X_{3_k} & \partial f_2 / \partial X_{4_k} & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 \ 1 \end{bmatrix} \quad (4.6)$$

4.6 VLF Observation Equations

The measurement or observation process was previously defined in Section 3.5 as $\underline{Y}_{k+1} = \underline{h}(\underline{X}_{k+1}) + \underline{V}_{k+1}$, and the optimal state update equation is

$$\hat{\underline{X}}_{k+1} = \hat{\underline{X}}_{k+1} - \underline{K}_{k+1} [\underline{h}(\hat{\underline{X}}_{k+1}) - \underline{Y}_{k+1}]$$

The observations, $\underline{Y}_{k+1}$, consist of a set of six VLF time "distance" measurements (see Section 2.5) which are the measured μ SEC quantities obtained from the aircraft VLF system. The six elements of the $\underline{h}$ vector are the theoretical μ SEC distances from the aircraft to the six VLF stations. These theoretical distances are obtained via a computer algorithm known as the Sodano inverse routine([60],[61]), in which the plant update elements $f_1(\hat{\underline{X}}_k)$ and $f_2(\hat{\underline{X}}_k)$ are used as the aircraft geographical position. The Sodano inverse gives the theoretical distance between any two points on the earth's surface, and yields very accurate results.

The VLF measurement process is assumed to contain random noise elements ($\underline{V}_{k+1}$). Each of these noise quantities has an associated variance and the measurement noise autocovariance matrix, $\underline{R}$, may be defined as :

$$R = \begin{bmatrix} \sigma_{\phi_1}^2 & & & & & \\ & \sigma_{\phi_2}^2 & & & & \\ & & \sigma_{\phi_3}^2 & & 0 & \\ & & & \sigma_{\phi_4}^2 & & \\ 0 & & & & \sigma_{\phi_5}^2 & \\ & & & & & \sigma_{\phi_6}^2 \end{bmatrix} \quad (4.7)$$

4.7 Quasi-linear Observation Matrix

As given by Equation 3.27, a quasi-linear observation matrix is defined by the elements

$$H_{k+1}{}_{ij} = \left. \frac{\partial h_i(\underline{X})}{\partial X_j} \right|_{\underline{X} = \hat{\underline{X}}_k} \quad \begin{matrix} i = 1, \dots, 6 \\ j = 1, \dots, 10 \end{matrix}$$

The elements of $\underline{h}$ cannot be expressed explicitly in an analytic fashion but the partials defining the elements of H_{k+1} can be determined through use of the Sodano inverse routine and the method formulated by Leach [59]. The partials are defined by

$$\left. \frac{\partial h_i(\underline{X})}{\partial X_1} \right|_{\underline{X} = \hat{\underline{X}}_k} = (-60/V_s) \cdot \cos(Az) \quad (4.8)$$

$$\left. \frac{\partial h_i(\underline{X})}{\partial X_2} \right|_{\underline{X} = \hat{\underline{X}}_k} = (60/V_s) \cdot \sin(Az) \cdot \cos(X_1 \cdot \pi/180) \quad (4.9)$$

where V_s is the speed of propagation of the VLF signal
 Az is the azimuth (heading) from the aircraft position
to VLF station i

Note that the elements of H_{k+1} represent the change in the station's μ Sec. distance with respect to a given change in the aircraft latitude and longitude. H_{k+1} is then formulated as:

$$H_{k+1} = \begin{bmatrix} \partial h_1 / \partial X_{1k} & \partial h_1 / \partial X_{2k} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \partial h_2 / \partial X_{1k} & \partial h_2 / \partial X_{2k} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \partial h_6 / \partial X_{1k} & \partial h_6 / \partial X_{2k} & 0 & \vdots & \vdots & \vdots & 0 & 1 & \vdots & \vdots \end{bmatrix} \quad (4.10)$$

4.8 The Kalman Filtering Process

At this point, all of the matrices found in Equation 3.28 have essentially been defined and, in effect, the formulation of the Kalman filter for the integrated aircraft navigation system has been completed. Some important factors remain to be discussed, however.

4.8.1 Initialization

The Kalman filter is defined by five equations which are processed in an iterative fashion. That is, at each update time, the appropriate matrix manipulations are performed and the state-estimate is obtained. Before these iterations can be carried out, however, the filter requires a certain amount of initialization, i.e. at the 'zero-time' point the filter must be initialized. The vectors and matrices involved in this first step are described in the following paragraphs.

- (i) Initial State Vector, $\hat{X}_0$ - The ten state variables are set to values which represent the 'best' known estimates at the 'zero-time' point. These estimates may in fact be only rough approximations

to the true initial state, especially the bias estimates ($\hat{x}_{3_0} \rightarrow \hat{x}_{10_0}$). Generally, the initial latitude and longitude positions should be quite accurate, since the position of the aircraft at initialization is frequently within the airport take-off area (eg. runway intersection, hangar apron, or whatever), and the latitude and longitude coordinates are easily obtained from a map or airport publication. The initial bias estimates are not as easily or accurately determined. One possible solution would be to simply set the initial bias estimates to zero 'default' values, and let the Kalman filter do a little more work, so to say, in estimating these biases via the iterative process.

(ii) Initial State Covariance Matrix, P_0 - This is a 10x10 matrix with diagonal elements representing the variances in the initial state variable estimates. In essence, these diagonal elements indicate the amount of weight or trust you are willing to place on your initial state estimate. If the initial position estimate, for example, is very accurate, the values of P_{11_0} and P_{22_0} should be set to 'small' values. Alternatively, if the initial bias estimates are felt to be rather 'ballpark' figures, then the appropriate elements of P_0 should reflect your uncertainty by being set to 'large' values. In the extreme case a zero diagonal element would indicate a perfect initial estimate, and in the case of constant biases, no additional estimation would be required or attempted by the filter.

(iii) Initial Measurement Noise Autocovariance Matrix, R_0 - The diagonal elements of R_0 represent the variances to be expected in the VLF signal measurements; also, by the intuitive reasoning of Section 3.2, these elements reflect the amount of weight you are willing to place on the actual VLF measurements. For example, if you know from past experience that a certain VLF signal tends to be noisier, or less reliable, than others, then the corresponding R_0 diagonal should be set to a higher value than the other elements. Since the VLF signal noise processes are assumed to be independent, the off-diagonal elements of R_0 are fixed at zero.

(iv) Initial Plant Noise Autocovariance Matrix, Q_0 . The setting of Q_0 is done in a manner similar to that for R_0 . Through previous knowledge of the Doppler/Heading systems, one can usually have a pretty good idea of the reliability, accuracy, and noise levels to be found in the plant. Again, the off-diagonal elements of Q_0 are fixed at zero.

4.8.2 Adaptive Filtering

Once the initialization process has been accomplished, presumably the Kalman filter can be left to run on its own, i.e. the five filter equations are processed at each update time. If the initial covariance matrices have been suitably (i.e. accurately) chosen, and if the plant and VLF measurement signals are statistically stationary, then one would assume that the filter is operating at, or very close to, its optimal point. This is, in fact, a correct assumption, but if the initialization process was one of rather rough approximations, or if the signal statistics change, the Kalman filter must be informed of this via appropriate changes in the covariance matrices. This so-called adaptive filtering method involves some sort of preprocessing of the measurement data (or more often of the innovations sequence) before the Kalman filter equations are sequenced. Based upon a decision algorithm, one or more of the covariance elements (i.e. in the P , Q , or R matrices) would be adjusted to reflect the new weighting desired on any of the plant or observation signals. In this manner, signal anomalies or contingencies may be accounted for, and the filter can continue to operate as close to its optimal point as possible.

Some sort of adaptive filtering is almost always used or required when a Kalman filter is being considered. As will be seen in detail in Chapters 5 and 6, formulating a good adaptive filtering scheme is quite a difficult task, and handling the various contingencies which can occur is a practical problem which cannot be overlooked if the filter is expected to perform with any degree of accuracy.

Section 5.5 details the adaptive filtering scheme proposed and tested for the integrated navigation system developed here.

4.8.3 Practical Considerations

In addition to the problems of initialization and contingency handling, there are a number of other practical problems which must be considered during the development phase of the filter. These are outlined below, along with the solutions proposed for the filter at hand.

- Update interval - The filter update interval must be chosen so that all of the necessary matrix operations and data processing can be accomplished by the aircraft computer within the corresponding time interval. The interval should not be too long on the other hand, because the estimation accuracy could degrade with increasing time between updates. The update interval used here is based on the VLF measurement rate - one update every ten seconds. As shall be seen later, simulation studies compare update intervals of 10, 20, and 40 seconds. The basic ten second interval is used in the actual aircraft implementation.
- Numerical accuracy - The filter equations are defined by a sequence of matrix operations, and the numerical accuracy is generally a function of the number of operations to be performed per iteration, as well as the computer limitations. In the filter developed here, single precision (4 byte) operation was chosen, and no special numerical formulations, such as square root filtering [9], were used. At least for the large-computer simulation studies, it was felt that most of the filter inaccuracy would not result from the computer limitations, but rather from the quasi-linear approximations and the initialization errors due to the operator's decisions regarding initial covariances. In addition, because of the relative low order state, it was decided to keep the formulation as numerically simple as possible, and not to resort to methods that would have been automatically chosen for higher order filters.

4.8.4 Sub-optimality of the Kalman Filter

The Kalman filter defined by Equation 3.22, for a linear, discrete process, if initialized correctly, will operate in an optimal fashion. That is, the state estimates will be of a minimum variance nature. Equation 3.22 defines an optimal Kalman filter.

The quasi-linear formulation, however, does not produce an optimal filter, simply because of the approximations that were incurred in the linearization process. In addition, for the practical filter implementation, the $\underline{P}_0$, $\underline{Q}_0$, and $\underline{R}_0$ matrices will likely not perfectly reflect the system signal variances, and this will obviously detract even more from the optimal filtering condition. The degree of optimality inherent in the filter cannot be directly specified; that is, we cannot state whether the filter is, say, 95% optimal or 80% optimal, or whatever. However, through simulation studies and sensitivity analyses, the appropriate parameters may be discovered which will allow a good degree of optimality, i.e. we can stipulate which conditions should be met in order for the given sub-optimal filter to operate with the greatest possible accuracy. Reference [14] gives examples of such sensitivity analyses.

Chapter 6 deals with the simulation runs performed in the current study.

4.9* Summary

This section summarizes the quasi-linear Kalman filter. The recursion equations are given along with the various matrices.

The five Kalman filter equations are as follows:

$$\underline{X}'_{k+1} = \underline{f}(\underline{\hat{X}}_k) = \underline{\hat{X}}_k + \int_{t_k}^{t_{k+1}} \underline{f}^*(\underline{\hat{X}}_k) dt$$

Plant update

$$\underline{P}'_{k+1} = \underline{\Phi}_k \underline{P}_k \underline{\Phi}_k^T + \underline{G}_k \underline{Q}_k \underline{G}_k^T$$

A priori covariance matrix of plant update

$$\underline{K}_{k+1} = \underline{P}'_{k+1} \underline{H}_{k+1}^T [\underline{H}_{k+1} \underline{P}'_{k+1} \underline{H}_{k+1}^T + \underline{R}_{k+1}]^{-1}$$

Kalman Gain matrix

$$\underline{\hat{X}}_{k+1} = \underline{X}'_{k+1} + \underline{K}_{k+1} [\underline{y}_{k+1} - \underline{h}(\underline{X}'_{k+1})]$$

Kalman optimal update

$$\underline{P}_{k+1} = \underline{P}'_{k+1} - \underline{K}_{k+1} \underline{H}_{k+1} \underline{P}'_{k+1}$$

A posteriori covariance matrix of the Kalman state update

The vectors and matrices found in the recursion equations are defined as follows:

$$\underline{X}'_{k+1} \equiv [f_1(\underline{\hat{X}}_k) \ f_2(\underline{\hat{X}}_k) \ f_3(\underline{\hat{X}}_k) \ \cdots \ f_{10}(\underline{\hat{X}}_k)]^T$$

Plant update vector

$$\underline{G}_k \equiv \begin{bmatrix} A6 & A5 \\ A8 & A7 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$$

Plant noise matrix
(10 x 2)

$$\Phi_k \equiv \begin{bmatrix} 1 & 0 & \partial f_1 / \partial x_{3k} & \partial f_1 / \partial x_{4k} & 0 & \dots & 0 \\ \partial f_2 / \partial x_{1k} & 1 & \partial f_2 / \partial x_{3k} & \partial f_2 / \partial x_{4k} & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & \vdots & \vdots & \vdots & 0 & 1 \end{bmatrix}$$

Quasi-linear
State transition
Matrix
(10 x 10)

$$Q_k \equiv \begin{bmatrix} \sigma_v^{*2} & 0 \\ 0 & \sigma_{\alpha\beta}^{*2} \end{bmatrix}$$

Plant noise
Autocovariance
Matrix (2 x 2)

$$H_{k+1} \equiv \begin{bmatrix} \partial h_1 / \partial x_{1k} & \partial h_1 / \partial x_{2k} & 0 & 0 & 1 & 0 & \dots & 0 \\ \partial h_2 / \partial x_{1k} & \partial h_2 / \partial x_{2k} & 0 & 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \partial h_6 / \partial x_{1k} & \partial h_6 / \partial x_{2k} & 0 & \dots & \dots & \dots & 0 & 1 \end{bmatrix}$$

Quasi-linear
observation
Matrix (6 x 10)

$$y_{k+1} \equiv [y_{1k+1} \quad y_{2k+1} \quad \dots \quad y_{6k+1}]^T$$

observation
vector

$$\underline{R}_{k+1} \equiv \begin{bmatrix} \sigma_{\phi_1}^2 & & & & & \\ & \sigma_{\phi_2}^2 & & & & \\ & & \sigma_{\phi_3}^2 & & & \\ & & & \sigma_{\phi_4}^2 & & \\ & 0 & & & \sigma_{\phi_5}^2 & \\ & & & & & \sigma_{\phi_6}^2 \end{bmatrix}$$

Observation noise
Autocovariance matrix

$$\underline{h}(\underline{X}'_{k+1}) \equiv [h_1(\underline{X}'_{k+1}) \quad h_2(\underline{X}'_{k+1}) \quad \dots \quad h_6(\underline{X}'_{k+1})]^T$$

theoretical
observation
vector

$$\underline{\hat{X}}_{k+1} \equiv [\hat{X}_{1k+1} \quad \hat{X}_{2k+1} \quad \dots \quad \hat{X}_{10k+1}]^T$$

Kalman optimal
state estimate vector

$$\begin{bmatrix} \hat{P}_{LAT} \\ \hat{P}_{LONG} \\ \hat{B}_V \\ \hat{B}_{\alpha\beta} \\ \hat{B}_{\phi_1} \\ \hat{B}_{\phi_2} \\ \hat{B}_{\phi_6} \end{bmatrix}_{k+1}$$

The initial error covariance matrix is defined as follows:

$$P_0 = \begin{bmatrix} \sigma_{PLAT_0}^2 & & & & & & \\ & \sigma_{PLONG_0}^2 & & & & & \\ & & \sigma_{BV_0}^2 & & & & \\ & & & \sigma_{B\alpha\beta_0}^2 & & & \\ & 0 & & & \sigma_{B\phi_{10}}^2 & & \\ & & & & & \ddots & \\ & & & & & & \sigma_{B\phi_{60}}^2 \end{bmatrix}$$

CHAPTER 5

KALMAN FILTER SIMULATION STUDY: FORTRAN PROGRAM

5.1 Introduction

During the course of practically all Kalman filter application developments, some sort of large-scale computer simulation study has been performed, and the literature contains many examples of such off-line analyses. The reasons for doing these simulation studies are various, but the main purpose is to determine if the proposed system model/Kalman filter combination will in fact yield acceptable results. Very often, the mathematical complexity of the Kalman filter defies the formulation of a closed form analytical solution in which the effects of changing various parameters could be studied. A computer simulation program, however, allows many filter "runs" to be executed, and the filter performance for different input conditions can be easily checked via graphical display of the quantities under consideration.

By "simulation" we mean that the navigation sensor data that one would normally acquire during a real aircraft flight are modelled and simulated by computer, and these data are then processed by the proposed Kalman filter algorithm. The term "Monte Carlo simulation" is often used to refer to this type of simulation because all of the random noise components are generated by a random number generator usually resident in the computer software library. The advantages of this type of simulation are:

- ability to control the input data to the filter and, hence, any flight condition or trajectory can be examined without conducting expensive aircraft flights.
- immediate filter error checking since the true system parameters have been specified and are known.
- many different simulated flights can be programmed and, hence, an overall assessment of the filter may be made.
- once the simulation package has been programmed and debugged, it takes very little time to run a set of experiments.
- the researcher does not have to involve himself in deep mathematical analyses; the simulation study route falls intuitively in line with the real-time practical running of the filter. It is felt that a better understanding of the filtering mechanics is obtained through simulation experiments.

The expected end result of a simulation study is that, hopefully, the filter will be running at its best possible condition, i.e. as optimal as possible: the filter will be "fine-tuned", so to say. Throughout the simulation study performed here, the main "philosophy" followed was: "given the following simulated flight conditions, how will the filter, in its current configuration, respond?"

There are two main parts to the simulation analysis: the development of the necessary software, and the actual running of the simulation experiments. Each segment necessitated careful planning and execution, and it was realized early in the research that great care must be taken when writing a fairly involved computer program.

The software development began during the author's first engagement as a summer research assistant at NAE. During that summer the basic simulation package was put together, albeit not without several bugs that went unnoticed for some time! Throughout the course of the research, the simulation package was refined and numerous plotting routines were developed. Of course, whenever a new type of experimental condition was decided upon, the computer code had to be written and completely debugged - often this caused more headaches than the actual analysis of the results of program output itself!

This chapter describes the simulation study performed during the filter development, and gives details of the algorithms used. The following chapter gives the results and conclusions of this study.

5.2 Fortran Kalman Filter Simulator

Appendix E gives complete documentation for the actual Kalman filter FORTRAN IV simulation package as developed for use on the NRC IBM 3033. This section will outline in brief terms the philosophy and structure of the package.

One of the major considerations was to have the simulator as flexible as possible. With this in mind, the user is given numerous options which may be invoked during any simulation run. These include:

- choice of either a straight-line flight trajectory, circular trajectory, or a combination. Total flexibility was designed for the trajectory: aircraft groundspeed, heading and radius of turn, for example, are user specified.

- total flight time and filter update interval are variable (up to 5000 seconds flight; 10-40 second updates).
- any level of system bias and signal noise can be programmed.
- any operating location is possible (except north of approximately 85°N).
- VLF or Doppler/heading bias drifting or bias jumps can be easily prescribed by the user, and the Kalman filter can be programmed to operate in either a "naive" or "smart" mode.
- an automatic contingency compensation algorithm can be invoked, with variable coefficient settings for trigger thresholds.
- system and filter outputs may be written to various data sets, if desired, for debugging or analysis.
- simulated plant/VLF quantities may be appropriately converted and written to a magnetic tape. This creates a "control" set of data which may be used as input on the Interdata 7/32 mini-computer version of the Kalman filter (this is described in Chapter 8).
- a very flexible computer plotting routine was developed to allow graphical representation of the filter outputs. This proved to be invaluable when analyzing the results of a Kalman filter run.

Because the package was developed with the above points in mind, the graphical results of a particular filter run can be obtained in about six minutes from the time the user starts the program. This, of course, allows sets of experiments to be run rather quickly, and results can be analyzed within a short time. If the user wishes to change any parameters, this is easily done in the next program run.

The main sections of the simulation package are shown in Figure 5-1 and a representative block diagram appears in Figure 5-2. Program input parameters come from either a pre-stored data set or from the user's input terminal. The input segment also obtains the various optional information from the user such as bias drifting or circular trajectory parameters. All user defined quantities are in "user-oriented" units - Knots, degrees or Nautical miles. A parameter conversion sequence translates these into "program-oriented" units - NM/sec, radians, etc. Given the parameters which describe the requested flight pattern, the trajectory simulator computes the true aircraft position at each update point. A random noise generator

is used in conjunction with the true plant/VLF quantities (see Sections 2.4.3, 2.4.4 and 2.5.3) and specified biases to give simulated system measurements via use of the Doppler/heading and VLF simulators. Adaptive filtering algorithms are invoked through use of "smart" filter options, or via the automatic contingency algorithm. Various subroutines constitute the actual Kalman filter and these are basically matrix manipulation routines. The errors in the state estimates are determined by comparing the filter outputs with the true, user-specified quantities. Finally, the computer plotting routine is used to display either the state estimates or the true state error responses.

5.3 Flight Simulator Algorithms

As mentioned above, the flight simulator was designed to be as flexible as possible, but without creating an overly complex or unwieldy computer code. The simulator evolved into a fairly comprehensive package, with numerous options being available to the user. Of course, the final simulator version is by no means a complete one: there are many other options that could be included, such as a more detailed flight trajectory simulator. The configuration presented here is felt to be adequate for most test conditions that are likely to be considered. This section describes the various algorithms used in providing the flight simulator options.

(i) Initial Input Data

Most of the initial input parameters such as signal variances, true biases, initial state estimates and initial covariance matrices are obtained from a pre-stored data set that is easily changed via the IBM TSS EDITOR facility. In the normal simulator operating mode, filter runs of the same type use the same basic parameters. By employing an input data set containing these nonchanging parameters, the user is relieved of the job of having to repeatedly give the simulator a large number of inputs via his keyboard. Hence, only the desired optional information need be provided by the users on each run.

(ii) Plant Bias Drifting

To examine the effects of drifting or suddenly jumping Doppler/heading biases, the Doppler plant bias option allows specification of either a velocity bias change, or heading bias change, or both. The user, having selected the plant bias drifting option, inputs the bias drift start time, end time, and magnitude of the bias change. A bias "jump" is selected by using a very short duration drift, i.e. the user specifies,

say, a 5 knot bias change to occur over a 10 second period. Similarly, a very slowly drifting plant bias is selected by having a long interval between the bias drift start time and end time.

(iii) VLF Bias Drifting

As explained in Section 2.5.3, a drift in a VLF signal can occur because of several reasons. This can be modelled by specifying a drift in a particular VLF bias. The user is given complete flexibility in selecting the VLF drifting bias parameters - any or all of the six VLF stations may be chosen to have a bias drift, and the drift parameters themselves can be different for each station. The user simply specifies the bias drift start time, end time and magnitude of the bias drift. As with a plant bias change, a sudden VLF bias jump may be modelled by specifying a very short duration "drift".

(iv) VLF Noise Change

Because of various reasons, the noise statistics of a real VLF signal may change: a signal may become noisier, or conversely, more "reliable". The simulator provides the user with the option to change the variance of any VLF station at any time during the simulated flight. In the current configuration, only one such variance change per station is allowed for each flight. The user specifies the time of the noise change, and inputs the new variance level. Each station may have a different variance change time and new variance level.

(v) Circular Track Trajectory

The true trajectory simulator can give a straight-line track, a circular track, or a combination of the two. The basic trajectory assumed is a simple straight-line but the user may choose a circular track option, at which point the circle start and end times are specified, in addition to the desired radius of the circular track. Some possible trajectories are:

- straight-line track only.
- straight-line track followed by a circular track.
- straight-line followed by circular track, followed by a second straight track.
- circular track only.
- straight/circular/straight track with same start and end position (a closed circuit trajectory requiring use of FORTRAN routine "TRKSET" before the actual simulator run - see Appendix E for details).

Figure 5-3 shows several of these combinations. During a circular manoeuvre, the aircraft pitch and roll angles remain constant and in this regard the circular trajectory is not a completely accurate representation of what would happen in a real aircraft manoeuvre. Still, the circular track simulation is useful in examining the effects of a flight pattern with acceleration (i.e. angular acceleration). This of course is important when considering the Doppler plant quality degradation due to a non-straight trajectory. It is hoped that the Kalman filter would be able to account for any possible plant degradation, and hence counteract the effects of a circular manoeuvre.

(vi) "Last Chance" Options

After the user has specified the desired options, the simulator routine pauses and allows one last change in any of the parameters via use of the IBM TSS "QUALIFY" and "SET" program control system (PCS) commands. This permits changes to any parameter read in from the input data set or the keyboard. For example, the user might wish to change only one of the initial P matrix elements, or perhaps change the desired total flight time. In addition, any of the user's own options may be altered if he has changed his mind, or if a mistake on input has been discovered. The PCS "GO" command is issued to resume the simulator run.

5.4 Adaptive Filter Options

In order to maintain accurate operation of the Kalman filter, certain matrix elements must be adjusted if and when signal anomalies occur, or when signal statistics change. This adaptive filtering process has been described in Section 4-8 and is an important part of the Kalman filter simulation package. The Kalman filter programmed for the simulation studies has two modes of operation - a "naive" mode in which the filter matrices do not adapt themselves to various contingencies, and a "smart" mode in which the adaptive filtering process is allowed to take place. Of course, in a real-life implementation, only the adaptive filter would be used, but in the simulations, the effects of not making the matrix adjustments are analyzed in order to determine just how the adaptive filter should be structured.

In this regard, each of the signal and trajectory options described in the previous section may be invoked with either a naive or smart Kalman filter version. The user is given several options before

the filter is actually run in order to specify which filter mode is desired. At the same time, the user may set various coefficients which determine the "level" of adaptive filtering desired (by "level" we mean by how much the matrix elements are to be adjusted in relation to the actual simulated signal anomalies). This permits a "fine tuning" procedure by which the optimal adaptive filter coefficients may be determined.

Table 5-1 outlines the naive/smart modes of filter operation that are possible.

For a given simulation run, any or all of the options listed in Table 5-1 may be selected along with a corresponding naive/smart filter arrangement. For the smart mode of operation, the Kalman filter is explicitly "informed" of the various contingencies through the adjustment of the appropriate matrices at the time the contingency takes place. In a real flight situation, these contingency occurrences would not normally be known *a priori* and in order to provide an adaptive filtering approach some sort of automatic in-flight check must be performed. The following section describes one such method of anomaly detection for the VLF signals.

5.5 Automatic Contingency Compensation

During the normal steady state operation of the Kalman filter, the diagonal elements of the various covariance matrices will have attained relatively constant values. The propagation of the P matrix diagonals is such that they tend to become quite small compared to their initial values, and a steady state level is reached after a certain time from the start of the filter run. The Q and R matrix diagonals will also be constant under normal conditions, where the plant and measurement noises are statistically stationary. These two factors cause the gain matrix, K , to also become fairly constant with small elements after the transient phase of the filter run is over. Intuitively, one can see that less and less weight will be put on the VLF measurements as the filter continues to process the given information since the K matrix decreases in magnitude. The last best state estimate is given more and more weight. One notable, and very significant consequence of this feature of the Kalman filter is that sudden, or even moderately slow changes in either the plant or VLF biases or noise can cause the filter estimates to diverge because the filter assumes one thing but in reality the incoming data is behaving

quite differently. Prevention of this possible divergence can be accomplished in basically three different ways:

- augment the plant noise vector in order to model the added uncertainty in the shifting quantities,
- explicitly prevent the P matrix diagonal elements from getting smaller than a specified level,
- let P propagate normally until an anomalous or divergent condition is detected, then increase the appropriate element(s) of P .

The third method seems most attractive since the filter would be allowed to run in an "optimal" fashion until a divergent condition is suspected at which point the filter adapts itself to the changing conditions. Such a scheme has been developed for the VLF measurement data. As will be discussed later, the detection of Doppler anomalies is somewhat more difficult; fortunately, as practical observation has shown, the Doppler data tends to be extremely reliable under most operating conditions and, hence, Doppler anomaly detection is deemed unnecessary. Tracking minor fluctuations in the Doppler/heading biases could be accomplished through plant noise modelling, which will be discussed in Chapter 7.

In order to compensate for possible changes in either the VLF biases or noise, it is necessary to have an algorithm which will automatically detect such changes, and adjust appropriate matrix elements. Some of the practical VLF signal anomalies that are known to occur are the following: (see Section 2-5 for more detail)

- a sudden or slow change in the VLF bias level, i.e. a bias jump or bias drift (due to SID's or the diurnal effect).
- an increase (or decrease) in the noise level of one or more VLF signals.
- complete loss of one VLF signal.
- complete loss of all VLF signals.

Each of the above anomalies requires a specific countermeasure if the filter is to continue providing accurate state estimates. An in-flight algorithm must be able to:

- (i) quickly detect and identify an anomalous condition.
- (ii) correctly adjust appropriate matrix elements.
- (iii) detect when the anomaly has ended, and

(iv) re-set matrix elements to reflect the non-anomalous condition.

Several authors have proposed methods to accurately set the $\underline{Q}$ and $\underline{R}$ covariance matrices in order to ensure optimal filter operation (See [14], [15], [16] for example). All of these methods use the so-called innovations or residuals sequence (see Section 3-2) to identify a non-optimal filter operation. Unfortunately, these theoretical procedures require either a large number of innovations samples, or assume stationary noise processes. In the practical VLF contingency identification problem, using a large sample size precludes the required rapid detection of an anomaly and, furthermore, the expected VLF contingencies are characterized by non-stationary noise processes. In addition, an anomalous condition in one of the actual state variables (i.e. one of the VLF biases) necessitates adjustment of the $\underline{P}$ matrix, and not the $\underline{Q}$ or $\underline{R}$ matrix. A fairly simple adaptive filter algorithm which attempts to circumvent these problems has been designed for use with the integrated VLF/Doppler navigation system.

The basic quantity used in the detection scheme is again the innovations process, $\underline{y}_{K+1} = \underline{h}(\underline{x}_{K+1})$, which is determined during the execution of the extended Kalman filter equations (Equation 3-28). The two terms appearing in an innovations sample at time t_{K+1} can be defined as having the following components:

$$\underline{y}_{K+1} \equiv \text{True VLF } \mu \text{ sec distance} + \text{true VLF bias} + \text{true VLF noise}$$

$$\underline{h}(\underline{x}_{K+1}) \equiv \text{"plant theoretical" } \mu \text{ sec distance} + \text{estimated VLF bias}$$

Thus, the innovations sample can be expressed as (for station i)

$$\begin{aligned} \underline{y}_{K+1} - \underline{h}_i(\underline{x}_{K+1}) &\equiv (\text{true VLF } \mu \text{ sec} - \text{plant theoretical } \mu \text{ sec}) \\ &\quad + (\text{true VLF bias} - \text{estimated VLF bias}) \\ &\quad + \text{true VLF noise} \end{aligned} \quad (5-1)$$

It is proposed that the innovations process may be used to estimate the VLF noise and to identify VLF bias contingencies. The reasoning is as follows:

Under steady state conditions of a filter run, the theoretical

μ sec distance based on an accurate plant update position should be quite precise, as should be the filter estimated VLF bias. In that case, from Equation 5-1 the innovations process will be an accurate approximation of the true VLF noise process. The argument may be further refined. The theoretical plant μ sec estimate may be considered to consist of three components:

$$h_i(\underline{x}'_{K+1}) \equiv \text{True theoretical plant } \mu \text{ sec} + \text{plant } \mu \text{ sec error} + \text{estimated VLF bias.} \quad (5-2)$$

To simplify notation, define the following:

$\phi_i \equiv$ true VLF μ sec distance

$B_{\phi_i} \equiv$ true VLF μ sec bias

$N_{\phi_i} \equiv$ true VLF noise

$\phi_{i_p} \equiv$ true theoretical plant μ sec

$B_{\phi_p} \equiv$ plant μ sec error (i.e. μ sec bias due to plant position error)

$\hat{B}_{\phi_i} \equiv$ estimated μ sec bias

Hence we can write (at time t_{K+1})

$$y_i - h_i \equiv \phi_i + B_{\phi_i} + N_{\phi_i} - \phi_{i_p} - B_{\phi_p} - \hat{B}_{\phi_i} \quad (5-3)$$

Figure 5-4 shows a representation of a VLF measurement compared to the theoretical μ sec distance based on a plant updated position.

But $\phi_{i_p} = \phi_i$, so Equation 5-3 can be rewritten as

$$y_i - h_i = B_{\phi_i} - \hat{B}_{\phi_i} - B_{\phi_p} + N_{\phi_i} \quad (5-4)$$

In the steady state, $\hat{B}_{\phi_i} \approx B_{\phi_i}$ and B_{ϕ_p} would be small.

Thus

$$y_i - h_i \approx N_{\phi_i}$$

Hence, the innovations process can be used to accurately estimate the VLF noise process. If at any time the true VLF bias, B_{ϕ_i} , begins to drift, $\hat{B}_{\phi_i}$ will no longer be an accurate estimate of B_{ϕ_i} and this will be reflected in the innovations sequence as a drift as well. The VLF noise process will then appear to be superimposed on the drifting innovations process. Figure 5-5 shows a typical time sequence of an innovations process for a simulated uncompensated drifting VLF bias situation.

The proposed automatic detection algorithm is based upon a least squares linear curve fit of the innovations samples for each VLF station. Because a small number of samples are used for each curve fit, a tolerance level must be set in order to account for the possibility of statistical errors in the estimated linear curve parameters. Appendix D contains the details of the algorithm. The main features are described here.

The algorithm uses twelve stored samples of the innovations sequence at a time for each VLF station (i.e. the algorithm is invoked every 110 seconds based on a ten second Kalman filter update interval). A least squares linear curve is fitted to the samples, a sample variance is calculated (which estimates the VLF noise variance), and a VLF bias drift is suspected if the end point of the linear curve falls outside a tolerance band which is based on a 99% confidence interval (approximately) for the regression coefficients. In addition, if several successive innovations samples fall outside the $\pm 3\sigma$ band, as determined for the particular VLF innovations process up to that point in time, the appropriate R matrix element is immediately set to a very large value. This eliminates the VLF station from the Kalman filter solution until a sufficient number of innovations samples are gathered to allow exact identification of the VLF contingency. Once the contingency is identified either the R matrix is adjusted (in the case of a noise change) or the P matrix element is increased (reflecting a VLF bias jump or drift). After each adjustment of the P matrix element, it is allowed to propagate normally until a drift is detected again. Similarly, if an R matrix diagonal element has been previously set to a very large value, it is then reset to a value of twice

the current VLF noise sample variance once the VLF contingency has been determined. (twice the sample variance in order to be on the safe side to allow for possible errors in the variance calculations).

The VLF bias drift rate is approximated by the slope of the fitted linear curve. The greater the slope, the greater the increase in the appropriate P matrix element.

Figure 5-6 shows a typical VLF bias drift detection criterion based on the numerical example given in Appendix D. A flowchart of the automatic detection and compensation algorithm is given in Figure 5-7. Section 6-6 of the next chapter gives results of simulation studies using the algorithm.

As indicated above, it is possible that one or more VLF signals may become so noisy or unreliable that it would be desirable to completely eliminate those stations from the Kalman filter solution. This can be accomplished by setting a flag to indicate that a station is "down". One criterion for the "station down" contingency is the amplitude of the actual VLF signal. If it falls below some prescribed minimum level, then the station is weighted out of the filter by setting the R matrix element to a large value. This element remains large until the station is deemed usable again.

CHAPTER 6

KALMAN FILTER SIMULATION STUDY: RESULTS

6.1 Introduction

Well before the practical on-line Kalman filter was even considered, it was necessary to perform many simulation runs to confirm that the proposed routines would in fact yield acceptable results. Similar simulations have been performed by many other authors (see, for example, [39] for an interesting narrative on the development of the Panavia Tornado integrated navigation system). The results obtained and published, however, can be used as rough guidelines only. Each application of the Kalman filtering techniques to the navigation problem is unique - the particular VLF/Doppler system developed here has not been previously considered or analyzed. The simulation program carried out in the course of the filter development allowed examination of many aspects of the running of the filter, and permitted conclusions to be drawn that would not have been possible simply by doing a literature search or survey of previous filter applications.

The main goal in performing the simulation study was to determine how the filter would yield the "best" results under various flight and data conditions. The specific simulation studies performed can be classified into the following categories:

- basic filter operation - transient study and accuracy
- VLF data of varying quality
- variable update interval
- state size reduction
- VLF anomalies - a priori knowledge
- VLF anomalies - automatic on-line detection and compensation
- exponential data weighting

Each of these specific sets of simulations will be described in the following sections. It should be noted that other simulation results have been previously reported by Leach [59] based upon the same Kalman filter simulation routines. Since the present author was involved with the development of the computer algorithms from the beginning of the project, certain obvious similarities will exist between some of the material presented here and that appearing in [59].

6.2 Basic Operation of the Filter

Following the initial creation and debugging of the basic simulation package, it was desired to verify that the filter would accurately estimate the state variables under very simple flight conditions. A set of three runs was designed not only to demonstrate that the filter would work, but also to show the effects of different initial conditions. In this case, the initial $\underline{P}$ matrix was set to different levels on each run. Relatively "poor" data was simulated, in that the noise levels of the plant and VLF signals were set to abnormally high levels. A simple straight-line trajectory was chosen. Table 6-1 summarizes the simulated parameters. Note that the diagonal elements of the Doppler/heading autocovariance matrix, $\underline{Q}$, reflect the variances of the integrated Doppler/heading data, as explained in Appendix C. For all three runs, the initial velocity, heading and VLF bias estimation errors were set to 10 knots, 10 degrees and 10 μ sec, respectively. These values would normally be much lower in a real flight situation, but, in order to emphasize the capability of the filter, they were chosen to be large in these runs. As indicated in Chapter 4, the $\underline{P}$ matrix diagonal elements represent the variances of state estimation errors. In other words, the square root of each $\underline{P}$ matrix diagonal should be an indication of how accurate the state estimate is. Alternatively, the initial $\underline{P}$ matrix elements should reflect the estimated error in the initial state estimates. In the three runs considered here, the initial errors in the bias estimations are known to be 10 knots, 10 degrees and 10 μ sec. Hence the appropriate $\underline{P}$ matrix diagonals should initially be set to 100 knots², 100 deg² and 100 μ sec², i.e.

$$\begin{aligned} P_{033} &= 100 \text{ knots}^2 \\ P_{044} &= 100 \text{ deg}^2 \\ P_{055} \rightarrow P_{010,10} &= 100 \mu \text{ sec}^2 \end{aligned}$$

Figures 6-1 and 6-2 show the results of using different initial $\underline{P}$ matrix levels. The state variable estimates corresponding to a fairly accurate $\underline{P}_0$ are shown in Figure 6-1. The speed-of-response, particularly for the

VLF bias estimates, is very fast and a steady state condition is reached within a relatively short period. Figure 6-2 shows the actual errors in the state estimates for the three different cases. The solid line corresponds to the responses in Figure 6-1, i.e. an accurate $\underline{P}_0$. The second run, whose results are shown by the dotted line in Figure 6-2, had initial $\underline{P}$ matrix diagonals which were much smaller than those in the first case. It is seen that the filter is more "sluggish" in accurately estimating the state. The latitude and longitude errors are large during the first 1000 seconds or so, due not only to the poor quality of the plant/VLF data, but also due to the poor estimation of the plant biases during that period. The third run has a very small $\underline{P}_0$ and the filter is only able to accurately estimate some of the state variables. Even at that, it requires a substantial amount of time to bring the errors down to acceptable levels. The responses for the third run are shown by the dashed traces in Figure 6-2.

The results of these runs correspond with what one would intuitively conclude. A large initial $\underline{P}$ matrix indicates to the filter that the initial state estimates are poor, and hence the filter places little weight on these estimates, whether they are actually very poor or not. As the initial $\underline{P}$ matrix diagonals are decreased in size, the filter places more and more weight on the initial state estimate. As seen in the third run, the initial state was poorly estimated, but the initial $\underline{P}$ matrix indicated the opposite of this. Hence the filter "accepted" the initial state as being very accurate, even though it was not.

An examination of the time history of the various $\underline{P}$ matrix diagonals shows that the state variable responses closely follow the propagations of the corresponding $\underline{P}$ matrix elements. Figure 6-3 shows the time history of the P_{33} element for each of the runs. Comparing these responses with those for the Doppler velocity bias error responses, it is seen that the two are very much alike in nature. (Compare Figure [6-3c] with solid trace in Figure 6-2, Figure [6-3b] with dotted trace and Figure [6-3a] with dashed trace.) Figure 6-4 shows the response of the P_{55} element, corresponding to the VLF station 0 bias error response. If the filter is working properly then the error covariance responses should follow the actual state error responses themselves. In other words, the error covariance matrix diagonals will indicate the squares of the estimated errors in the state estimates.

Figure 6-5 shows the matrix norm, $\|\underline{P}\|$ of the error covariance

of the sum of the squares of the matrix diagonal elements). The Mean Square state errors for the corresponding runs are shown in

Figure 6-6 (i.e. MS error is given by $[\sum_{i=1}^{10} \tilde{x}_i^2]$, where $\tilde{x}_i$ is the error in the i^{th} state variable).

6.3 Results from Using VLF Data of Various Qualities

As seen in the previous section, the accuracy of the state estimate is a function of the initial estimates of the state variables as well as the initial $\underline{P}$ matrix setting. A second factor which affects the accuracy of the filter is, of course, the quality of the actual VLF and Doppler/heading data, i.e. the amount of noise present on the measurement signals. It would be impossible for the filter to give extremely accurate state estimates if the data it is using are highly corrupted by nondeterministic noise, even if the noise covariance matrices accurately reflect the data quality.

To investigate the consequences of using signals with varying noise levels, three Kalman filter runs were designed, each using either "poor", "good" or "very good" simulated Doppler plant and VLF data. Table 6-2 gives the noise specifications of these signals. Note that, in all three cases, the initial state estimate errors are identical. It was felt that, in a practical sense, the estimates of the initial state variables, particularly the plant and VLF biases, will generally be independent of the actual noise levels of the signals. Hence, a very poor quality VLF signal, for example, does not imply that the initial VLF bias estimate for that station would be a poor one.

The state variable error responses for the poor quality data condition are shown in Figure 6-7. The steady state accuracy in the position and plant bias estimations is quite good, but the initial transient errors tend to be large. The VLF bias estimates are never much better than their initial estimated values. When good data is simulated, on the other hand, the plant/bias and latitude/longitude estimates improve considerably, not only during the transient period, but also in the steady state sense. In this case, the VLF bias errors are at least showing signs of moving closer to zero (Figure 6-8). Figure 6-9 shows the responses when very good data is simulated. The improvement in accuracy is substantial,

particularly in the Doppler velocity bias estimate. The difference between using poor and very good data is dramatic: this reflects the 100X improvement in the quality of the very good data over the poor data. The filter places much more weight on the very good incoming measurement signals and less weight on the initial state estimates themselves.

In connection with the experiments using data of different qualities, two further runs were designed to demonstrate the consequences of having severe mismatches between the true VLF noise variances and the assumed filter measurement covariance matrix ($\underline{R}$). It has been seen that the initial $\underline{R}$ matrix diagonal elements should be set to levels approximately equal to the true VLF signal variances. In a real flight situation, these variances may not be accurately known and, thus, the $\underline{R}$ matrix would only be a rough approximation to the correct or optimal situation. The first mismatch situation has very good VLF data simulated, with noise variances of $0.1 \mu \text{ sec}^2$ for all six stations. The $\underline{R}$ matrix diagonals are set to values of 100X these variances, (i.e. $R_{11} \rightarrow R_{66} = 10.0 \mu \text{ sec}^2$). This simulates the situation where the true VLF signals are of very good quality, but we do not know this and approximate the optimal $\underline{R}$ matrix by specifying large variances. In effect, we are not "allowing" the filter to take full advantage of the very good quality VLF measurement data. As seen in Figure 6-10, the velocity/heading bias errors are relatively large during the first 1000 seconds of the run, and the velocity bias error seems to be increasing slowly after about the 2000 second mark. The VLF bias error responses are very sluggish because we have told the filter to place more weight on the initial VLF bias estimates and less weight on the incoming VLF measurements. Comparing these responses with those in Figure 6-9 (very good data and no variance mismatch), it is observed that the position errors are practically identical in both cases, likely due to the fact that the VLF biases are small to begin with and the plant data is of good quality.

The second run in this set simulates poor Doppler/heading and VLF data. ($\sigma^2 = 10.0 \mu \text{ sec}^2$) The initial $\underline{R}$ matrix is set, however, to a small level, ($R_{ii} = 0.10 \mu \text{ sec}^2$) indicating to the filter that the VLF measurement weighting should be large. This is considered to be a more severe mismatch than in the first case. Comparing the

responses in Figure 6-11 (the mismatch situation) with those in Figure 6-7 (poor data, but $\underline{R}$ properly set), it is seen that the mismatched $\underline{R}$ matrix causes the state errors to increase. Note, especially, the severe transients in the velocity and heading biases. The latitude/longitude error responses are worse than in the matched $\underline{R}$ matrix case, and the VLF bias errors tend to be much greater.

These experiments verify that better quality data will indeed yield more accurate state estimates. The importance of setting the $\underline{R}$ matrix elements to values accurately reflecting the true VLF noise variances is obvious. In a real flight situation, the VLF noise statistics may not be perfectly known but, hopefully, some *a priori* knowledge about the VLF stations will be available, allowing fairly accurate setting of the $\underline{R}$ matrix. An adaptive filter approach would allow the $\underline{R}$ matrix to be adjusted while in-flight.

6.4 State Size Reduction through Elimination of VLF Stations

One of the adverse situations that can arise during the real flight operation of the Kalman filter navigation system is that not all six VLF stations may be usable throughout the entire flight. In fact, only a few stations may be available at any one time. Several simulation runs using a reduced number of VLF stations were made, and the results turned out to be rather interesting.

In the runs which simulated four and five VLF stations available, very little difference in the error responses was observed over the response for the case when all six VLF stations were used. This result was seen when either "good" or "poor" plant/VLF data was simulated. This observation leads one to believe that accurate operation is possible with a reduced-order filter, providing of course that the Doppler plant/VLF data is reasonably good. No simulations were made with extremely poor quality VLF data, but it is believed that the accuracy of the state estimates would suffer considerably if fewer than five or six VLF stations were used in a very poor data scenario.

Figure 6-12 shows the state error responses when three VLF stations are available. In this run, good quality data is simulated, again with a simple straight-line flight trajectory. Comparing the responses with those in the run using six VLF stations (Figure 6-8) it is seen that the only difference lies in the ability of the filter

to reduce the errors in the VLF bias estimates (particularly station M). This problem is not serious, however, because the bias errors are small to begin with in both runs. The latitude/longitude accuracies are pretty much alike, as are the velocity/heading angle bias estimates. In a similar run using poor quality data, the Doppler velocity bias errors in the three station run actually tend to be slightly smaller than those in the six station case. The same thing is seen in the heading angle bias responses. (Compare Figure 6-13 with 6-7). This unpredicted result is felt to be a consequence of the so-called linear dependency problem as conjectured by Leach ([59], Section 4-3). Basically, a problem can arise if, under certain conditions, the Kalman filter attempts to resolve all six of the VLF biases simultaneously. The linear dependency problem is believed to have two possible sources: the actual propagation of the filter gain matrix and the geographical locations of the VLF stations themselves relative to the aircraft location. Correlations among VLF bias estimates due to pairwise similarities of the VLF geometries can occur when two or more stations are practically in-line with the aircraft position. The map in Figure 2-5 shows several of these poor geometries: Stations W and M, D and M, A and O for example. Under certain conditions these linear dependencies can cause significant estimation errors which may last for a long period of time. Investigation of this problem has shown that the main factors affecting the degree of linear dependency are: magnitude of the initial error covariances, flight trajectory, geographical location of the aircraft, and the number of biases attempted to be accurately tracked. Reference [62] summarizes this linear dependency problem. The relationship between linear dependency and the aircraft location has been described by Leach [59] who has shown that, for identical filter parameters in both cases, a change in the geographical location of the aircraft flight area can either induce or eliminate a linear dependency situation.

In the runs considered here, the linear dependency situation is likely appearing in the six-station run, but not in the three-station run. Slightly improved accuracy thus seems possible for the reduced VLF station operation. In a real flight context, however, practicality dictates using as many VLF stations as are available since some stations could become severely degraded in quality during a flight.

Further runs were made using only two VLF stations. As seen in Figure 6-14, the two-station, good data situation results in a substantial degradation of the position estimates, especially during the first 3000 seconds of operation. The VLF biases remain at practically their initial estimated values: the filter is not able to improve these estimates. The linear dependency problem is not occurring here, but the small number of VLF stations is not sufficient to provide enough redundant phase information. When poor data is simulated in the two station case (Figure 6-15), the state estimates are even less accurate.

The most interesting responses are observed in the runs using only one VLF station. (Figures 6-16 & 6-17). In both the good quality and poor quality runs, the latitude/longitude position errors grow with time and are highly dependent, of course, on the plant bias errors. The single VLF station does not provide the "damping" that was apparent in the other runs with two or more stations.

The time histories of the $\underline{P}$ matrix diagonal elements for the good quality, one VLF station run are displayed in Figure 6-18. The first two plots show the propagation of P_{11} and P_{22} , which represent the variances in the latitude and longitude error estimates, respectively. Note that the shape of the time history curve of P_{11} is somewhat similar to that for the latitude error, except for a portion after about the 4600 second mark. Similarities also exist for the P_{22} and longitude error curves. The VLF bias error curve is practically flat in Figure 6-16, and this reflects the very slowly changing P_{55} element. In this run, the initial values of the $\underline{P}$ matrix elements corresponding to the initial plant bias estimates were set at ten times their nominal values (i.e. $P_{033} = P_{044} = 10.0$ instead of 1.0 for initial bias errors of 1.0 knots and 1.0 deg.). For this reason, it is concluded that the time histories for P_{33} and P_{44} would not appear similar to the plant bias error responses. Another run using $P_{033} = P_{044} = 1.0$ yielded bias error responses that were essentially identical to those seen in Figure 6-16, but the time history trace of those elements were more similar to the actual bias error responses. A similar comparison between state estimation errors and the corresponding state covariance matrix elements may be found in Gelb ([14], Example 6-1-2). In the simulation runs performed in the current study, it has been observed on numerous occasions that the

square roots of the P matrix diagonals may be taken as fairly good estimates of the state estimation errors.

6.5 VLF Anomalies: A priori Knowledge

The necessity to include some sort of scheme to adapt the Kalman filter to various signal contingencies was emphasized in Section 4.8, while Section 5.6 outlined the "naive" and "smart" filters programmed in the simulation package. This section details the simulation runs which were performed in order to show the effects of using either a smart or naive filter when these VLF anomalies occur. It is important to point out the difference between the smart filter version used here and the automatic VLF Contingency routine described in Section 5.5. In these "smart" filter runs, the filter is explicitly "told", through use of several "flags" in the computer program, to adjust the appropriate matrices at the time the anomaly begins to occur. If the user elects to simulate a particular anomaly, then the Kalman filter routine will provide the appropriate matrix adjustment. In the automatic contingency routine, however, another algorithm is invoked which attempts to detect, by itself, any anomalies that occur. The simulation runs using this automatic method are described in Section 6.6.

6.5.1 VLF Bias Drifting

As numerous simulation runs have demonstrated, the most critical VLF anomaly, next to a complete signal outage, is the bias drifting or jumping situation. If the bias shift is not accounted for through the adjustment of the appropriate matrix element, severe errors will develop in the state estimates. The two simulations described below have been programmed with a bias drift in VLF station "0". The drift rate of 18 μ sec/hours is typical of what might be found in a diurnal shift situation. Good quality Doppler/heading and VLF data was used and, again, the flight trajectory is one with constant heading and groundspeed.

In the first run, with state error responses shown in Figure 6-19, the naive version of the filter is not aware of the bias drift in station 0. The bias error in that station grows to approximately 10 μ sec because the filter bias estimate is practically zero, whereas the true bias has drifted from zero μ sec to 10 μ sec. The effects on the

other state estimates are obvious. Significant and growing state errors are evident in all the state variables except the heading angle bias and the VLF bias for station W. Other simulation runs have verified that the heading angle bias estimate is quite well insulated against VLF anomalies (so is the groundspeed bias estimate, but to a lesser extent). The error in the bias estimate of station W does not increase, due to the geometry of the six stations relative to the Ottawa operating location. Considering the map in Figure 2-5, it is seen that the line connecting Ottawa and station W is approximately perpendicular to the line connecting Ottawa and station "O". This is a very favourable geometry, since an anomaly occurring in one of these two stations will do very little harm to the bias estimate of the other. (It is interesting to note that in a scheme using only hyperbolic navigation based on VLF stations, minimum errors will occur if a similar perpendicular geometry is used). The geographical position of station D relative to Ottawa and station O also seems to be good. Even though the error in the station D bias estimate is drifting, it is doing so at a rate much lower than that found in stations G, M, and A, which are in an unfavourable position relative to station O.

The smart version of the filter accounts for the bias drift by adjusting the $\underline{P}$ matrix element corresponding to the 5th state variable (i.e. P_{55}). Through numerous other simulation runs and analyses, it was determined that the best method of handling a VLF bias drift is to fix the $\underline{P}$ matrix element at some large value for the duration of the drift, and then allow the $\underline{P}$ matrix to propagate normally as soon as the drift has stopped. It is worth noting several other attempts that were made to account for a VLF bias drift. These were:

(i) adjust the $\underline{P}$ matrix element only at the start of the drift and allow normal propagation thereafter. This method was not successful: the $\underline{P}$ matrix element would quickly decrease in size but the bias would still be drifting.

(ii) adjust the $\underline{R}$ matrix diagonal element as soon as the bias begins to drift. By increasing the value of the $\underline{R}$ matrix element, the

2

VLF station itself can be weighted out of the Kalman filter and, hence, any quantity included in that VLF signal (true μ sec distance, VLF bias and noise) is no longer used. This method is able to negate the effects of a drifting VLF bias on the other state variables, but in an overall context is not a good solution. Firstly, if a bias drift occurs at the same time in several stations, then weighting out all of the drifting stations would incur a substantial reduction in the amount of VLF information available, which can be detrimental as seen in Section 6.4. In addition, if a VLF station is weighted out, then its bias cannot be estimated accurately by the filter. When the bias drift has stopped, the R matrix element would have to be decreased in size and, at the same time, the corresponding P matrix diagonal element would have to be increased in order to tell the filter that a significant bias error is present.

Through empirical analysis, it was felt that the best way to adjust the P matrix element is to set it to some value which is a function of the actual bias drift rate. In the smart filter used here, P_{55} is set at a value 5000 times the drift rate, or $5000 \times 0.005 \mu \text{ sec} = 25 \mu \text{ sec}^2$. With this element held constant for the entire 2000 second drift period, the estimates of the other state variables remain accurate, as seen in Figure 6-20. The bias estimate for station 0 itself is also fairly accurate, although noisy. At least the bias error is kept to within a reasonably bounded level (about $\pm 2 \mu \text{ sec}$).

Similar results were observed in the simulated situation of several biases drifting at the same time. Fixing the appropriate elements of P counteracted any effect that the bias drifts might have on the other state variable estimates.

6.5.2 VLF Bias Jumps

An extension to the VLF bias drifting contingency is the situation of a bias jump, i.e., a very rapid "drift". Such anomalies are not as common as a drifting bias, but may occasionally occur due to various ionospheric disturbances. Two simulation runs were programmed to show the effects of a fairly significant bias jump. In Figure 6-21 the responses for a naive filter run are shown. Station M has a $10 \mu \text{ sec}$

bias jump at the 2000 second mark, but the filter is not informed of this. Hence, at the 2000 second point, the seventh diagonal element of the $\underline{P}$ matrix (P_{77}) is very small, and the VLF bias for station M is accurately estimated up to that time. As soon as the bias jumps to 10 μ sec, the error in the bias estimate is also approximately 10 μ sec, but the much smaller value of P_{77} does not reflect this. The accuracies of the other state variable estimates subsequently degrade as the filter continues to run in the naive mode. It is interesting to note the similarity between the response of station M in this naive filter run and that which appears in Figure 6-2 (the small $\underline{P}_0$ situation). Both runs have a $\underline{P}$ matrix diagonal element which does not accurately reflect the error in the bias estimate.

When the smart version of the filter was used, the effects of the bias jump were eliminated. The responses shown in Figure 6-22 indicate that the 10 μ sec bias jump in station M does not produce an increase in the errors of the other state variable estimates. Even though the bias jump is large, the filter is able to track this reasonably well, and the error in the bias estimate for station M is kept to within $\pm 1 \mu$ sec. The time history of the bias error response reveals a slight "ringing" effect just after the 2000 second mark. This is probably due to the fact that P_{77} was set to an extremely large value at 2000 seconds ($2500 \mu \text{ sec}^2$). It is felt that better results would have been observed if P_{77} had been set to a smaller value, say, in the 500 - 1000 $\mu \text{ sec}^2$ range.

6.5.3 VLF Noise Increase

A third type of VLF anomaly occurs when the noise "level" of the VLF signal(s) increases; that is, the variance of the VLF random noise increases. This could be due to any of the various reasons described in Chapter 2. It has already been shown (Section 6.3) that a severe mismatch between the actual VLF noise variances and the assumed filter observation autocovariance matrix, $\underline{R}$, will cause significant state estimation errors to develop. It is exactly this condition which occurs when the observation noise increases during a flight (or in the middle of a simulation run). To prevent large estimation errors from

developing, the filter must adapt itself to the new noise level: this is accomplished through adjustment of the appropriate elements of $\underline{R}$.

Several simulation runs were made in order to investigate the effects of this noise increase. It was observed that moderate increases (by a factor of 50 - 100X) in the variances of only one or two stations would not seriously degrade filter performance if the $\underline{R}$ matrix was left unchanged (i.e. a "naive" filter). If, however, one VLF noise variance increased by a very large amount, say by a factor of 250-500, then the accuracy of the filter estimates was decreased considerably. This is to be expected because of the very large magnitude of the noise component that is unaccounted for in that one station. The simulation runs have shown that the filter accuracy depends upon several factors:

- the number of stations which experience noise increases - moderate increases in one or two variances can be left unaccounted for without serious degradation in filter accuracy. Estimation errors will dramatically increase if all six VLF noise variances increase without a corresponding adjustment of $\underline{R}$.
- the magnitude of the new noise level - small increases are not as detrimental as large increases.
- the magnitude of $\underline{R} - \underline{R}$ should be adjusted to reflect the new noise variances: severe mismatches between the true noise variances and the corresponding elements of $\underline{R}$ (i.e. the assumed noise variances) can seriously affect filter accuracy.

Results of two filter runs which simulated the extreme case of noise increases in all six VLF signals are given here. Good VLF data ($\sigma^2_{\phi_i} = 1.0 \mu \text{ sec}^2$) is initially simulated, with the $\underline{R}_{ii}$ set to the same values. At the 2000 second mark the true noise variances of all six stations are increased to $100.0 \mu \text{ sec}^2$. Figures 6-23 and 6-24 show the responses for a naive filter and smart filter, respectively. In the naive filter mode, the assumed variance is 100X less than the

true variance - a similar situation to that seen in Figure 6-11. The smart filter accounts for the noise increase, however, by increasing the R matrix diagonals to $100.0 \mu \text{ sec}^2$ at the 2000 second point in the run. As seen in Figure 6-24, this has a "damping" effect on the VLF bias estimates in that the estimates are more or less constant after the time the noise increased. This is explained by the fact that the Kalman gain matrix elements which correspond to the VLF bias state variables have been decreased and hence less weight is put on the incoming observation data. Since, by the 2000 second mark, the state error covariance matrix elements P_{55} and $P_{10,10}$ are small, little further VLF bias estimation can occur. This is similar to the situation observed in Figure 6-7 where poor VLF data and correspondingly large R diagonals prevent the initial VLF biases estimates from being further refined.

The results obtained in these simulation runs with varying noise levels/ R matrix settings can be compared with a study done by Mehra [63]. Figure 6-25 (from [63]) shows the effect of errors in the observation noise autocovariance matrix relative to the true noise. Quite obviously, it is better to be pessimistic in setting the assumed measurement variances.

6.6 VLF Anomalies: Automatic Contingency Compensation

The preceding section described several experiments in which the Kalman filter was explicitly "informed" of a contingency situation through the adjustment of the P or R matrix during the occurrence of the anomaly. In most real flight operations the anomalies would not be known *a priori* and the filter would have to automatically detect such anomalies. The algorithm described in section 5.7 was used in several simulation runs to confirm that the proposed automatic detection and compensation scheme would in fact provide satisfactory results. These simulation runs deal mainly with the drifting VLF bias contingency, which is considered to be the most severe anomaly, next to a complete loss of VLF signal(s).

One of the first things to be examined was the nature of the innovations sequence during a drifting bias contingency. Figure 5-5 shows a typical innovations sequence for a bias drift in station "0".

6 The drift rate is $10 \mu \text{ sec/hour}$ and good quality VLF data has been simulated as in the runs described in section 6.5. The true VLF noise sequence has also been plotted for comparison purposes. Note that the innovations sequence accurately represents the true noise process up to the point where the bias drifting starts (at the 1000 second mark). After that point the bias drift is very apparent, and it is observed that the innovations sequence still includes a true noise component.

Further simulation runs verified that the innovations sequence could be consistently used to detect the bias drift anomaly. The most severe case of this anomaly is when all six biases are drifting simultaneously. Figure 6-26 shows the responses when this is the case and the automatic scheme is used. Again, good quality VLF data has been simulated and the biases are drifting in the 2000 - 4000 second segment of the "flight". All six anomalies have been detected and appropriate action has been initiated by the algorithm in order to compensate for the drifts. Consequently, the errors in the state variable estimates are kept to low levels throughout the contingency period. This response can be compared to that of the smart filter responses shown in Figure 6-20.

It should be noted at this point that a certain amount of empirical or experimental trial-and-error was involved in setting the two coefficients used in the anomaly detection and $\underline{P}$ matrix adjustment equations. Section 5.5 mentioned that a 99% confidence interval is used in the detection scheme which translates into a theoretical value of 3.169 for the Student t distribution (see Appendix D). However, during several simulation runs it was observed that this value was actually too low, resulting in numerous "false alarms" in which a bias drift was falsely detected by the algorithm when in fact no drift was occurring. A slightly larger value of 3.4 was used in the simulation described in this section. Of course, this coefficient would likely have to be adjusted if poorer or better quality data is available (or simulated).

Appendix D gives details on the adjustment of the $\underline{P}$ matrix elements corresponding to the drifting VLF biases. Again, experimental results indicated that the coefficient in the adjustment equation (Equation

[D-6] in Appendix D) should be on the order of 250 or so. Significantly smaller coefficients resulted in sluggish responses and compensation, and large coefficients tended to produce large, unwanted transient spikes in the VLF bias estimations at the occurrence and detection of bias shifts. The responses observed in Figure 6-26 are felt to be quite good, with the VLF bias errors being reduced quickly as soon as the drift has been detected. The actual innovations sequence of station 0 for the simulation run described above is shown in Figure 6-27(a), and the corresponding P_{55} trajectory is shown in Figure 6-27(b). Note that Figure 6-27(b) shows the values of P_{55} after the Kalman filter has sequenced the five filter equations, i.e. $(P_{55})_{K+1}$ is shown rather than $(P_{55})_K$. Of course, upon detection of the bias drift in station 0 the value $(P_{55})_K$ would be much larger than $(P_{55})_{K+1}$ calculated at the end of the filter update operation.

Other simulation runs have shown that a complete loss of one or more VLF stations is not severely detrimental to accurate filter operations, providing that the station loss is quickly detected (through the station acquisition flag mentioned in section 5.5) and the corresponding $\underline{R}$ matrix element is set to a value of approximately 10,000 or so. During the course of the simulations, it was discovered that severe numerical ill-conditioning may occur if one or several of the $\underline{R}$ matrix diagonals are set to values exceeding roughly 20,000. Recall the definition of the Kalman gain matrix (see Equation 3-28). In the determination of $\underline{R}$ an inverse of a 6 X 6 matrix must be determined, i.e. $[\underline{H}_K + 1 \quad \underline{P}_K + 1 \quad \underline{H}_K^T + 1 \quad \underline{R}_K + 1]^{-1}$. The diagonals of the quadratic term $\underline{H}^T \underline{P} \underline{H}$ will usually be fairly small if no severe anomalies are present in the data. In order for an inverse to be found, most practical matrix inverse routines require a fairly well conditioned matrix, with all pivots approximately the same magnitude. In the case of the simulations performed here, the matrix inverse routine failed if the above mentioned ill conditioning was present. Fortunately, for the given filter configuration, a particular VLF station can be entirely weighted out of the solution if its $\underline{R}$ matrix element is set to 10,000. Chapter 5 of [13] gives some numerical considerations to be borne in mind in relation to this practical inversion problem.

It has also been observed that the filter can account for an increase in the VLF signal noise simply by adjusting the $\underline{R}$ matrix.

The automatic detection algorithm is able to accurately estimate the signal noise variance even in the presence of a drifting VLF bias for the particular station. Of course, slight increases in VLF noise levels may be left uncompensated, as shown in Section 6.5, because the filter tends to be rather insensitive to small variations in noise. However, recall that the simulations described in Section 6.3 indicate that severe mismatches between the R matrix elements and the actual noise variances can lead to degraded accuracy.

6.7 Variable Update Interval

When performing the simulations on the large IBM 3033 computer, run times and memory availability were of no concern since such a large machine, when used in an "off-line" sense, can provide practically unlimited resources. In a real time, on-line flight operation, however, processing time and memory space are two very important considerations. One set of simulation runs was performed in order to evaluate the filter accuracy when update intervals of greater than the basic 10 second interval were used (see subsection 4-8-3). The objective in these experiments was to determine if the filter could give adequate results when 20 or 40 second update intervals were used. When these simulation runs were made, we had no idea of just how much real time would be used by the aircraft computer in processing the filter algorithms. It was later realized, however, that the cycle time for the onboard Kalman filter would be only about two seconds, well within the VLF data acquisition rate of 10 seconds. Even though this verified that the 10 second update interval could be used, very brief details of the results of the 20 and 40 second update simulations are given here for completeness. It also should be noted that a 20 second update interval was used in some of the simulations in order to cut down processing time and computer costs.

The results of the various simulations indicated that accurate and acceptable performance is possible even if a 40 second update interval is used during severe flight and signal conditions (such as a very tight turn with VLF anomalies present). A slight degradation in accuracy was observed when very poor Doppler/heading plant data was simulated, but the filter performance under this condition was still considered acceptable.

Although not totally confirmed through simulations, it is believed that severely degraded accuracy could result if the geographical operating location is higher than about 85° N latitude. The main reason for this problem is that at the extreme north latitudes, changes in geographical longitude are very sensitive to distance flown. Consider the plant update equation for longitude:

$$P'_{LONG_{K+1}} = \hat{P}_{LONG_K} - \frac{[(A1 - A3 \cdot \hat{B}_V \cos(\hat{B}_{\alpha\beta})) - [(A2 - A4 \cdot \hat{B}_V) \sin \hat{B}_{\alpha\beta}]]}{\cos(\hat{P}_{LAT_K} \cdot \pi/180)}$$

As the update interval increases, the integration interval used in determining A1-A4 increases. This causes increased errors in A1-A4, i.e. an increase in the error in the numerator of the update term above. Now, because $P_{LAT} \rightarrow 90^\circ$, $\cos(P_{LAT} \cdot \pi/180) \rightarrow 0$ and any errors in the numerator are greatly amplified.

For example,

$$\text{for } P_{LAT} = 45^\circ, P'_{LONG_{K+1}} = \hat{P}_{LONG_K} - 1.41 [\cdot]$$

$$\text{but for } P_{LAT} = 80^\circ, P'_{LONG_{K+1}} = \hat{P}_{LONG_K} - 5.76 [\cdot]$$

where $[\cdot]$ represents the numerator in the update equation. This degraded accuracy would also likely appear even if the basic 10 second update was used. The errors would simply be amplified if a greater update interval was selected during flights at extreme north latitudes.*

6.8 Exponential Data Weighting

A review of the literature reveals that several fairly simple methods have been proposed which attempt to counteract possible filter divergence due to anomalous conditions such as bias drifting. One such approach is known as exponential data weighting or data age fading [27], [28] and [29]. The basic concept is that past observations are given less and less weight as new measurements are taken. The measurements taken long in the past are effectively completely weighted out of the Kalman filter solution at the current time, in so doing the current

* The error in NM would not increase significantly, however, due to the fact that the number of NM per degree longitude decreases as the

measurements are given more relative weight and hopefully this would allow easy "detection" of anomalies occurring in the recent observation data ([28] gives more detail). The implementation of this idea is very-simple: only one single additional multiplication by a fixed scale factor appears at each observation time; as follows:

$$P_{K+1} = S \Phi P_K \Phi^T + G Q G^T$$

where S is a chosen scalar. ($S \geq 1$)

The simplicity and computational attractiveness of this method suggested that a simulation study be made in order to determine the effectiveness of exponential weighting in the context of the Kalman filter developed here. It was hoped that the more complex automatic contingency algorithm discussed previously could be discarded if the simpler approach was adequate.

The results of the simulation runs were disappointing and it was eventually concluded that this simple approach was not well conditioned to the proposed system. Typical examples of the simulation runs are shown in Figure 6-28 and 6-29. In the first case a scale factor of 1.05 is used in a no-anomaly situation with good quality data. The estimation accuracies are good up until about the 3000 second point when obvious divergence occurs (these responses can be compared with those shown in Figure 6-8 which uses $S=1.00$). A linear dependency condition is apparent in the VLF bias estimates and the error increases. A second run uses the same scale factor, but in this case a VLF bias drift has been programmed in station "0" (10 μ sec drift 2000 - 4000 second segment). The other VLF bias errors increase by a very significant degree, as do the latitude and longitude errors. Further simulations resulted in similar responses, except of course when the scale factor was set to a value very near 1.0 (i.e. in the range 1.001-1.005) in which case the filter behaved almost identically to one with S set equal to 1.0. In those runs practically no exponential data weighting occurred.

The state estimation errors increased dramatically with small increases in the scale factor. When S was set at 1.5, for example,

position estimation errors were on the order of 15-20 NM with VLF bias errors reaching 200 μ sec even when no anomalies were programmed. A reduction in the number of VLF stations used did not improve matters (it was thought that a lower order filter might eliminate a linear dependency situation) and, furthermore, similar divergent results were observed when either poor or very good quality data was simulated. These results do not agree with those obtained in other studies ([28], [29], for example). One reason for the unacceptable performance of the current filter is proposed as follows. The state variables include six VLF biases which are modelled as non-dynamic quantities (i.e. biases are assumed constant in the system model). Very simply explained, the basic linear dependency problem created by trying to accurately estimate too many VLF biases in a poor station geometry configuration may likely be amplified by the exponential data weighting due in part to the nature of the $\underline{P}$ matrix propagation. ~~The~~ exponentially weighted filter seems very sensitive to even small changes in the scale factor, suggesting possible ill conditioning (again, poor VLF geometry/linear dependency).

As a final observation, the simulation results recorded in both [28] and [29] are for very low order systems in which only dynamically changing state variables are considered. Those simulated systems do not include biases in the state formulation. The theoretical aspects of the exponential data weighting concept seem solid enough, but as shown in this study, may be useless in a practical application. A survey of the Kalman filter application literature seems to indicate that this method is not commonly used, as pointed out in [29].

6.9 Summary and Conclusions

The simulation study has attempted to show the conditions under which the filter will give accurate and acceptable state estimations. By simulating the actual sensor data it is possible to determine the errors in the filter estimates, and hence define the best filter parameters and system models. In summary, the following conclusions are made regarding the filter operation:

- (i) The initial error covariance matrix ($\underline{P}_0$) should have diagonal elements which accurately reflect the estimated errors in the initial state. Ideally, the diagonals should be set to the squares of the

actual errors in the initial state variable estimates. An overly optimistic $\underline{P}_0$ causes sluggish responses while a pessimistic $\underline{P}_0$ can induce large spikes in the estimates during the initial transient period.

(ii) The time history traces of the $\underline{P}$ matrix diagonals are very similar in most cases to the time histories of the actual state estimate errors. The square roots of the diagonals can be used as indications of the probable errors in the filter estimates.

(iii) The accuracy of the filter estimates is partly a function of the quality of the sensor data. Very noisy process and observation data will result in relatively poor state estimates even if the corresponding noise covariance matrices are accurately defined.

(iv) The ratio of the assumed noise variances to the corresponding true or actual noise variances affects the accuracy of the filter estimates. It is preferable to be pessimistic in setting the elements of the noise covariance matrices so that the assumed variances are larger than the actual variances, not smaller.

(v) The filter will give accurate estimates of latitude and longitude even if fewer than six VLF stations are used, although in a practical sense it is desirable to have as many stations as possible on line despite the possibility of a linear dependency situation. Significant reduction in accuracy is observed when only one or two stations are used even if the quality of the observation data is good.

(vi) Anomalies in the observation process are classified as noise level increases, bias jumps, bias drifts and complete loss of signal. The simulations show that these anomalies can be properly accounted for by the following methods:

- noise increase - adjust the appropriate element(s) of the observation noise covariance matrix, $\underline{R}$.
- bias jumps, drifts - increase the appropriate elements of $\underline{P}$ and fix the elements for the duration of the bias shift
- loss of station - set the $\underline{R}$ matrix element(s) to a very large value.

(vii) Detection of an anomalous situation can be effected by an automatic scheme based on the innovations sequences which contain information on the noise and bias processes.

(viii) It is observed that an increase in the basic 10 second update

interval to 20, 30 or 40 seconds will not seriously degrade the filter accuracy.

(ix) An exponential age weighting procedure was attempted but the simulation results indicate that this type of anomaly compensation is not well suited to the particular filter proposed in this study. A likely cause of filter divergence is the linear dependency situation induced by a poor VLF station geometry.

CHAPTER 7

REAL FLIGHT DATA RESULTS

7.1 Introduction

The simulation experiments detailed in Chapter 6 all used rather well-conditioned data in that contingencies such as VLF bias drifting or noise increases were well defined and behaved. In this regard, the simulations by themselves do not truly verify that the proposed Kalman filter algorithms will perform in an adequate fashion when implemented on the aircraft computer and using actual flight data. It is a well known fact that conditions in the "real world" are often very different from those simulated in the laboratory. The navigation filter proposed here can only be really proven under actual flight conditions, and this can be accomplished either in-flight using real-time processing or, alternatively, on the ground in an off-line environment using real flight data previously stored on magnetic tape. The latter approach is obviously more practical in the early implementation stages, since various filter parameters may be specified for the same flight data allowing numerous computer runs to be made for comparison purposes. Another obvious, but important, advantage of the off-line approach is that a particular set of flight data may be run through the filter in a fraction of the real flight time.

This chapter describes the algorithms used in the real flight data version of the filter. Results of the filter runs using the flight data are detailed, and various refinements to both the system model and filter parameters are described. It will be observed that fairly accurate position estimates are obtained during a flight of about one hour's duration.

7.2 Algorithms for Real Flight Data

The algorithms used in the real-flight data version of the filter are essentially the same as those applied in the simulation experiments. An obvious departure is that the data simulators are no longer required since flight data are acquired directly from the aircraft magnetic tape. In addition, the *a priori* adaptive filter options (see Section 6.5) are not usable simply because we usually have little or no

a priori knowledge about the nature of the navigation data and anomalies that can occur during a flight. This, of course, implies that the contingencies must be detected by the automatic scheme proposed in Section 6.6, or handled in some other "on-line" manner. It is possible, however, to examine the navigation data on the ground prior to its processing by the Kalman filter and, hopefully, detect the more apparent anomalies. This would give some usable *a priori* knowledge with which we could ensure that these anomalies are accounted for during any on-ground Kalman filter runs.

Figure 7-1a shows the Kalman filter navigation scheme and Figure 7-1b shows a flowchart representing the real flight data Kalman filter mechanization. The present configuration includes very little pre-processing of the VLF measurement data before being used by the filter. The automatic contingency algorithm is included as an option at this point because of development still envisaged for the filter in the future. It will be later seen that this algorithm still requires some improvement in order to be completely acceptable in the real flight, real-time environment.

Several references ([13], [64] among others) on Kalman filter navigation routines suggest that editing or pre-processing of the data should be employed for various reasons. One is that the noise characteristics of the sensor signals (Doppler velocity, for example) may not be accurately represented by the proposed system model. A second possibility is that of spurious observations, or "glitches" caused by system faults or other sources. It would be desirable to remove these serious data faults or eliminate the severely corrupted observations from the filter solution. An on-line scheme would allow such editing during a real-time flight. Alternatively, on-ground analysis of the data would possibly point to any "glitches" mentioned above and these could be removed or smoothed prior to an off-line Kalman filter run. This type of editing will be left for the future in the filter development considered here.

7.3 Convair 580 Navigation Flight

A recent navigation flight of the NAE Convair 580 resulted in about an hour's worth of usable navigation data being collected and analyzed. The Kalman filter was not used during the actual flight but

the data collected on magnetic tape was processed afterwards on the ground. The quantities recorded were the GNS-200 phase data for six VLF stations, aircraft pitch and roll angles, Doppler radar beam frequencies and magnetic bearing angles. During the hour of flying, the aircraft performed a series of "racetrack" patterns in the Ottawa vicinity, frequently passing over well known visual landmarks such as runway intersections, bridges, and airport radio beacons. An "on-top" discrete marker signal was recorded each time the aircraft flew over one of these visual sites, which provided reference points for later establishing the accuracy of the Kalman filter position estimates. The farthest visual on-top was approximately forty miles from the Ottawa airport. Table [7-1] shows the on-tops that were made as well as their geographical coordinates. The coordinates of the VOR's and beacons were obtained from published information on the Ottawa Airport while the other coordinates were determined from topographical maps with scales of 1:25,000. All of the visual reference points are felt to be accurate to within approximately 0.10 NM.

The flight pattern for this mission is shown in Figure 7.2. The simulation studies have clearly shown the importance of accurately setting the initial filter parameters; namely, the three covariance matrices and the initial state variable estimate. Of particular significance is the accuracy of the initial state error covariance matrix, P_0 . It has also been observed, through the simulations described in section 6.3, that the R matrix diagonal elements should accurately reflect the true VLF noise variances. With the entire data record from the flight available to us, a certain amount of pre-filtering analysis may be undertaken, especially on the VLF phase data.

Statistics for the VLF phase measurements were obtained in two ways. Groundrun statistics were determined by using 42 phase samples (7 minutes' worth) for each station which were recorded while the aircraft was stationary just prior to takeoff. Mean and variance calculations were done in the standard manner using the phase measurements directly except for station R in which case a linear regression was applied (since the aircraft is stationary, the natural μ sec variations due to aircraft flight are not present in the data). The means of the phase data were very close to zero and certainly negligible in comparison to any expected

biases. The ground run variances are given in Table 7-2, and are observed to be very small except for station R. As noted previously, this station is not phase stabilized so it is intuitively felt that the variance of its signal should be larger than the others. A comparison of this variance to that, say, for station M might lead one to conclude that station M should be given much more weight in the filter run than station R. This is, of course, a logical assumption, but only if we are willing to accept the groundrun variances as appropriate throughout the entire flight of the aircraft. Further analysis of the VLF flight data has provided meaningful approximations for the in-flight variances. When considering the in-flight situation in general, the phase observations cannot be directly used in mean and variance calculations, even if only rough approximations are desired. However, if the flight trajectory is such that there is an adequate straight-line segment, the VLF phase trajectories should be fairly linear with time, providing of course that no serious anomalies are present. A simple linear regression may be performed on the observations in order to remove the natural phase drift due to the aircraft movement. The variance of the phase data can be calculated directly from the coefficients used in the regression analysis (see Equations D-1 and D-2 in Appendix D). Of course, the statistical means of the linearly varying phase data are meaningless.

The aircraft flight described above included several straight-line segments, three of which were used for phase variance calculations. As seen in Table 7-2 most of the in-flight variances are only slightly larger than their ground run counterparts. The absolute accuracy of these variances is questionable because of the assumptions made in calculating them. We have no idea if in fact VLF anomalies are occurring during the selected straight-line trajectories and, hence, the calculated variances should be taken as rough guidelines only. It is concluded, however, that the R matrix diagonals should be set to values corresponding to the orders of magnitudes of the estimated in-flight variances. Diagonal elements of $0.05\text{--}0.20 \mu \text{ sec}^2$ would seem to be appropriate, except for station R in which case a larger variance is intuitively preferred because of the known unstabilized nature of that transmitter. Not only is its bias known to drift, but analyses of previous navigation data has indicated that station R tends to be noisier than the other VLF stations. The

a priori knowledge that the bias for station R exhibits a drift of approximately $-18 \mu \text{ sec/hour}$ suggests that the appropriate $\underline{P}$ matrix diagonal element can be fixed at some value for the entire real flight Kalman filter run. This of course forces the filter to accurately track the drifting bias.

7.4 Kalman Filter Runs using Real Flight Data

This section describes the pertinent observations and results of the analysis using the real flight data. It should be first noted that many filter runs were made in this part of the program development but only the more relevant details are presented here. Numerous runs had incorrect or inaccurate initial parameters and, in these cases, the filter operated in sort of a "naive" mode. Most of these runs were discounted if, for example, the position errors tended to be larger than an acceptable level. It was noted, however, what initial conditions or modelling structure caused larger errors. On the other hand, some of the filter runs resulted in favourably low error levels, and it will be shown that the Kalman filter can provide accurate navigation information, and at the same time, be prepared for possible observation data anomalies. A comparison of the Kalman filter scheme with other navigation algorithms has been presented by Leach [59] and it has been observed that significant improvements in accuracy are attainable with this Kalman filter routine.

7.4.1 Filter run with "Nominal" Parameters

From the pre-filtering analysis described in the previous section, and a similar analysis of the raw Doppler/heading data, the Doppler plant, VLF observation and initial error covariances could be readily defined. The Doppler and heading data were felt to be reasonably accurate and, thus, the elements of the plant noise covariance matrix, $\underline{Q}$, were set to "nominal" values, namely 5 KNOTS^2 and 0.5 deg^2 . Similarly, the observation covariance matrix, $\underline{R}$, reflects the in-flight VLF signal variances ($0.1 \rightarrow 0.2 \mu \text{ sec}^2$ for all stations except R, whose assumed variance was set to $0.5 \mu \text{ sec}^2$). The initial error covariance matrix, $\underline{P}_0$, had diagonal elements with values felt to represent the likely covariances in the initial state estimate. The initial position estimate was believed to be accurate to within 0.1 NM ($\approx 0.002^\circ$); to be on the "safe" side

(although perhaps violating the strict definition of the error covariance) P_{11_0} and P_{22_0} were both set to 0.001 deg^2 (instead of $(0.002)^2$). The initial bias estimates ($\hat{x}_{3_0} \rightarrow \hat{x}_{10_0}$) were all zero.

Normally one would have some idea of the biases expected in the Doppler/heading quantities but this information was not available for this particular set of data. A zero bias is simply an arbitrary and "easy" choice; small non-zero plant biases could also have been chosen. The initial true VLF biases should be very close to zero because of the initialization sequence for the VLF data: the initial position estimate is used to obtain initial theoretical μ sec distances to each station in order to set an "arbitrary bias" (see Appendix E) needed in the observation data acquisition routine. Since the initial position estimate is usually very good, the initial theoretical μ sec counts will also be very accurate (i.e. biases ideally zero, but in practice very small) and, hence, any initial bias is accounted for at initialization (we have called this "bias zeroing"). As a consequence, the 6 VLF biases were initially estimated as zero, with error covariances of $0.1 \mu \text{ sec}^2$. In summary, Table 7-3 shows the initial parameters for the "nominal" filter run. One further note of importance concerns the error covariance for the station R bias. As mentioned previously, it is well known that this station exhibits a bias drift and the P matrix element corresponding to this bias should be fixed for the entire filter run if accurate tracking of the bias is desired. This was not done in the run described here, but rather, $P_{10,10}$ was allowed to propagate on its own.

The automatic contingency algorithm described in section 5.5 has not been invoked in this particular run.

Figure 7-5 shows the estimates of the state variables. The responses are pretty much self-explanatory but several comments should be made. It appears as though the Doppler velocity bias estimate has a steady state value of approximately zero while the heading angle bias approaches a level of about 4 degrees. The VLF biases seem fairly small, except for station G and R, in which cases the bias estimates exhibit well defined and significant drifts. It is probable that there is a "linear dependency" between station G and station R: station R is known to drift but station G presumably should have a small, non drifting bias.

In addition, although a bias drift in station R is apparent, the drift rate seems too small (recall that the known bias drift rate in station R is approximately $-18 \mu \text{ sec/hour}$). This is an indication that the initial parameters may not be exactly correct, and the bias error covariance for station R should, of course, be fixed as previously speculated. The position errors for this run are shown in Figure 7-6. At each of the visual on-top points, a comparison of the filter position estimate with the visual landmark coordinates has been made. (Because the on-top times do not usually occur at the exact end of a filter update interval a simple linear interpolation was made between two filter updates to "time-align" the filter position estimates with the on-top positions). Out of the eleven position errors available, the maximum total (RMS) error is 0.67 NM. Ten of the errors are less than 0.45 NM, which is considered to reflect a reasonable position accuracy. Despite the obvious error in the bias estimation for station R (approx. $10 \mu \text{ sec}$ at the end of the run) the position accuracy is surprisingly good. There are two reasons for this. Only two out of six station biases have been poorly estimated (stations R and G) so four other stations are still giving good measurement data. Secondly, the R matrix element corresponding to station R is 3 to 5 times the other R matrix elements. In other words, the weight placed on station R is $1/3$ to $1/5$ of that for the other stations: any errors in the station R data would likely not cause serious problems. Although station G is weighted practically the same as the other stations, the effects of the errors in its bias estimations are deweighted because of the other four good stations. Using this argument, it is probable that no significant improvement in position estimation would occur even if the station R bias had been very accurately tracked. Of course, if only three or four stations (including station R) had been available during the flight, the results would have been quite different: no doubt the inaccurately tracked bias drift for station R would have caused a substantial degradation in position accuracy.

The absolute accuracy of the VLF bias estimations cannot be checked when real flight data is being used. However, examination of the raw VLF data at the times when the aircraft passes over the same visual on-top can give some clues as to just how well the biases are being estimated. For example, the intersection of RW 25-32 at the Ottawa

airport is crossed three times during the flight. If the VLF biases have not changed since the start of the flight, the raw VLF μ sec counts should be identical at each pass-over time, allowing of course for the accuracy of the on-top position (within 0.1 NM) and the random noise inherent in the VLF signals themselves. The raw μ sec count should vary by no more than about 1 μ sec between the three on-top times. Upon examining the data it was observed that this was the case; no appreciable bias changes could be detected. It could be concluded, therefore, that the filter's estimates of the VLF biases are valid (except for station G and R), and at least not grossly in error.

7.4.2 Refinements in the Parameter Specifications

The most important requirement that the filter must fulfill is the achievement of accurate position estimates, but at the same time, the errors in the bias estimates should also be small. The real flight data run described in the last subsection resulted in obviously incorrect VLF bias estimates for stations R and G but fairly accurate position estimates. Another run will now be described for which various parameters were changed and slightly better performance observed: the VLF bias estimates for stations R and G were improved and the maximum position error was reduced.

Table 7-4 indicates the parameters specified for this run. The initial VLF bias error covariances for the first five stations have been reduced to $0.01 \mu \text{ sec}^2$ while the bias error covariance for station R (i.e. $P_{10,10}$) has been fixed at $100 \mu \text{ sec}^2$ for the entire run. The initial Doppler/heading plant bias error covariances have been increased to 100 KTS^2 and 100 deg^2 to reflect a greater uncertainty in the initial plant bias estimates. As well, the plant noise covariances have been adjusted as a result of an examination of the raw Doppler/heading data (see statistics in Table 7-2; a plot of the raw data is given in Figure 7-3). Finally, more weight has been put on the measurement data of station M ($R_{22} = 0.05 \mu \text{ sec}^2$ instead of $0.1 \mu \text{ sec}^2$). The net effect of these changes is observed in Figure 7-7. The filter is able to accurately track the drifting bias in station R, and the linear dependency that was apparent in station G has disappeared. The maximum total position error, as shown in Figure 7-8, is now approximately 0.48 NM compared to 0.67 NM previously.

It is interesting to examine the various innovations sequences for the particular run. As explained in the theory for the Kalman filter, the optimality of the filter can be indicated by the statistical whiteness of the innovations sequence; the closer the sequence is to being perfectly white, the more optimal is the filter. Although no support or proof is given here, the whiteness qualification may possibly be invalidated when certain quantities are linearly dependent. In any case, Figures 7-9 and 7-10 show the innovations sequences for station L and R. The bias in station R is presumably being tracked very accurately as seen in the time history trace (Figure 7-7) and its innovation sequence appears quite random and fairly white in nature. The sequence for station L, however, is obviously not white, and it exhibits several non-random segments, particularly the strong ramp starting at about the 2300 second mark. This may be an indication that the bias estimates for this station are not as accurate as previously speculated. Even so, the position estimates are very good despite the possible inaccuracies in the bias estimates.

7.4.3 Refinement of the System Model: Bias Noise Augmentation

It has been previously noted that exact modelling of the various quantities used in the navigation system is a very difficult task, and often only an approximate model can be formulated even after substantial data analysis. (see, for example, [65]). In the initial development phase of the models used in the system described here, a fairly simple representation of the Doppler/heading plant and observation signals was proposed; that is, each measured quantity was modelled as a true component corrupted by a constant bias and a zero-mean random noise process. After numerous real flight data filter runs, it was felt that perhaps the simple model could be slightly refined by the addition of a bias plant noise process, representing random fluctuations that could exist in the bias terms themselves. Recall that the original Doppler/heading and VLF observation models were assumed to have the following forms:

$$\left. \begin{aligned} V_{\text{DOP}} &= V_{\text{TRUE}} + B_V + N_V \\ \alpha_{\text{DOP}} &= \alpha_{\text{TRUE}} + B_\alpha + N_\alpha \\ \beta &= \beta_{\text{TRUE}} + B_\beta + N_\beta \end{aligned} \right\} \quad (\alpha + \beta) = (\alpha + \beta)_{\text{TRUE}} + B_{\alpha\beta} + N_{\alpha\beta}$$

$$y_i = \phi_i + B_{\phi_i} + N_{\phi_i}$$

A refined, or augmented, model accounts for plant and VLF bias fluctuations by using a modified representation for the bias terms in the above expressions:

$$\begin{aligned} V_{DOP} &= V_{TRUE} + B_V^* + N_V \\ (\alpha + \beta) &= (\alpha + \beta)_{TRUE} + B_{\alpha\beta}^* + N_{\alpha\beta} \\ y_i &= \phi_i + B_{\phi_i}^* + N_{\phi_i} \end{aligned} \quad (7-1)$$

where the starred terms represent constant biases corrupted by random (unknown) fluctuations or shifts. In other words, this model assumes that the biases are no longer necessarily constant, but may shift in magnitude from time to time. To account for these random fluctuations, the plant noise vector would now include ten elements:

$$\underline{u} \equiv \begin{bmatrix} u_V^* \\ u_{\alpha\beta}^* \\ u_{B_V} \\ u_{B_{\alpha\beta}} \\ u_{B_{\phi_1}} \\ \vdots \\ u_{B_{\phi_6}} \end{bmatrix} \quad (7-2)$$

where u_{B_V} , $u_{B_{\alpha\beta}}$, $u_{B_{\phi_i}}$ represent some sort of "noise" present in the biases.

This "random bias" modelling now means that the bias update equations are of the following form:

$$B_{V_K+1} = B_{V_K} + u_{B_{V_K}}$$

$$B_{\alpha\beta_{K+1}} = B_{\alpha\beta_K} + u_{B_{\alpha\beta_K}} \quad (7-3)$$

$$B_{\phi_{i_{K+1}}} = B_{\phi_{i_K}} + u_{B_{\phi_{i_K}}}$$

instead of simply $B_{V_{K+1}} = B_{V_K}$, etc. It can easily be shown that the $\underline{Q}$ and $\underline{G}$ matrices are now augmented the following forms:

$$\underline{Q} \equiv \begin{bmatrix} \sigma_V^{*2} & & & & & & & & & \\ & \sigma_{\alpha\beta}^{*2} & & & & & & & & \\ & & 0 & & & & & & & \\ & & & \sigma^2 B_V & & & & & & \\ & & & & \sigma^2 B_{\alpha\beta} & & & & & \\ & 0 & & & & \sigma^2 B_{\phi_1} & & & & \\ & & & & & & \ddots & & & \\ & & & & & & & \sigma^2 B_{\phi_6} & & \end{bmatrix}; \quad \underline{G} \equiv \begin{bmatrix} A_6 & A_5 & A_6 & A_5 & 0 & \cdot & \cdot & \cdot & 0 \\ A_8 & A_7 & A_8 & A_7 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & 1 & \cdot & \cdot & \cdot & 0 \\ & & & & & \ddots & & & \\ & & & & & & 0 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix}$$

With this formulation, the diagonal elements of the *a priori* covariance matrix, $\underline{P}'_{K+1}$, are as follows:

$$\begin{aligned} \underline{P}'_{K+1} &= \underline{\Phi} \underline{P}_K \underline{\Phi}^T + \underline{G} \underline{G}^T \\ P'_{11_{K+1}} &= P_{11_K} + \phi_{13}^2 P_{33_K} + \phi_{14}^2 P_{44_K} + A_6^2 (Q_{11_K} + Q_{33_K}) + A_5^2 (Q_{22_K} + Q_{44_K}) \\ P'_{22_{K+1}} &= P_{22_K} + \phi_{21}^2 P_{11_K} + \phi_{23}^2 P_{33_K} + \phi_{24}^2 P_{44_K} + A_8^2 (Q_{11_K} + Q_{33_K}) \\ &\quad + A_7^2 (Q_{22_K} + Q_{44_K}) \\ P'_{33_{K+1}} &= P_{33_K} + Q_{33_K} \\ P'_{44_{K+1}} &= P_{44_K} + Q_{44_K} \\ P'_{55_{K+1}} &= P_{55_K} + Q_{55_K} \\ P'_{10,10_{K+1}} &= P_{10,10_K} + Q_{10,10_K} \end{aligned} \quad (7.4)$$

The above elements are now augmented by additional uncertainties in the various biases: the inclusion of a possible plant or VLF bias fluctuation necessarily implies a greater probability of error in any particular state variable estimated by the plant update equation.

The additional diagonal elements in the $\underline{Q}$ matrix can be thought of as having two implications. The pure statistical meaning of these elements is that they are the variances of some unknown Gaussian Doppler plant/VLF bias update noise processes. In a more intuitive sense, however, they represent an uncertainty that we have in the basic accuracy of the defined plant process. For example, the larger the value of $\sigma^2_{B_{\phi_i}}$, the more uncertain we are that the VLF bias,

B_{ϕ_i} , is constant. In other words, this bias could in fact deviate from a constant level, even if the deviations are not purely random. The amount of deviation, and the randomness of the deviation times, are reflected in the magnitude of the specified variances. The purpose of augmenting the $\underline{Q}$ and $\underline{G}$ matrices is to hopefully allow the filter to "track" any deviations in the biases, in much the same way as the automatic contingency algorithm described in Chapter 5 and 6.

Numerous real data filter runs have shown that, in general, more accurate state estimations will be produced by the augmented system than by a non-augmented filter. It has been observed in previous simulation studies that possible ill-conditioning may unknowingly be created if the filter attempts to track too many of the VLF biases. In fact, a divergent situation may occur in some cases if the filter is "forced" to accurately track more than four out of the six biases. Again, this is likely a result of the linear dependency problem. This situation became quite apparent in some of the real data runs with fairly large $\underline{Q}$ matrix elements.

Figure 7-11 shows the filter responses when four VLF biases are estimated. In this case, Doppler/heading bias plant noise modelling has not been included ($Q_{33} = Q_{44} = 0$), but a nominal amount of plant noise is assumed for the VLF biases. The initial $\underline{P}$ matrix is identical with that for the run in the preceding subsection, except for the elements corresponding to station G and W, where no bias estimation is attempted. The $\underline{Q}$ matrix elements for these biases are also zero. The $\underline{Q}$ matrix elements

for the other VLF biases are all $0.1 \mu \text{ sec}^2$.

Although the position errors are of about the same magnitude as those in the non-augmented filter example, the bias responses show an obvious "anomaly" beginning at about the 2300 second mark. The time history of the bias estimate for station L is remarkably similar to the innovations sequence previously observed (Figure 7-9), while the innovations sequence for station L in this run is considerably more random and statistically whiter. See Figure 7-13.

A second filter run was made in which plant noise was assumed for all bias terms ($Q_{33} = 0.01 \text{ KTS}^2$ and $Q_{44} = 0.05 \text{ deg}^2$). In addition, tracking of all six VLF biases was attempted by specifying non zero, but very small, initial error covariances for stations G and W (small values to avoid linear dependencies), and leaving the other four bias covariances the same as before. The responses for this run are shown in Figure 7-14. The time history of the heading bias is somewhat different from that observed previously but the groundspeed bias response is practically the same. The anomaly previously observed (at the 2300 second mark) has now disappeared but the innovations sequence is again quite random. Furthermore, the position errors are smaller. This leads to the conclusion that in previous runs the heading bias estimates were not very accurate. With the heading bias noise modelling the estimates are apparently better resulting in more accurate theoretical $\mu \text{ sec}$ counts (i.e. more accurate h_i), a whiter innovations sequence, and ultimately in greater accuracy in the position estimates.

A final analysis to be performed with the augmented filter was to see if a severe VLF anomaly could be accurately tracked. A "pseudo" real flight data set was created by artificially adding a $-10 \mu \text{ sec}$ bias to the station L raw data after the 2000 second mark, in effect creating a bias jump situation in one of the VLF stations. All other characteristics of the data remained as before, and the Doppler and heading data were not adjusted in any way. The primary point to be determined was just how small Q_{55} could be without position accuracy being degraded. In the first run, $Q_{55} \rightarrow Q_{77}$ were set to $0.01 \mu \text{ sec}^2$ and $Q_{88} = Q_{99} = 0.001 \mu \text{ sec}^2$ (again to prevent a linear dependency condition). The responses shown in Figure 7-17 indicate that the bias jump has been very nicely tracked and accurately estimated. The position errors for this run were also quite

acceptable (max total position error 0.44 NM; ten of the eleven errors below 0.36 NM). A second run had $Q_{55} = 0.001 \mu \text{ sec}^2$ and all other covariances identical to those in the first case. Figure 7-18 clearly shows a degraded performance: the bias jump in station L has not been accurately estimated and errors in the other variables are obvious. The position errors after the 2000 second mark in this run were approximately 1.2 NM. The ability of the filter to handle a VLF bias shift is thus quite dependent upon the level of bias plant noise modelling specified. There is a very clear trade-off between being able to track a bias shift and not exciting a linear dependency problem. If the filter were required to track a severe bias shift in any of the stations then all of the bias covariances would have to be set to relatively large values but, as previously noted, this can easily cause a linear dependency situation and/or detrimental filter divergence. Fortunately, however, the best of both worlds can be obtained by using the automatic contingency scheme in conjunction with the augmented model as briefly described in the next subsection.

In summary, simulations using the augmented model have clearly shown that VLF bias anomalies can be accurately detected and estimated by the filter, without the need for an automatic detection scheme. When no anomalies are present, filter performance and accuracy is not degraded by the inclusion of bias noise covariances, provided that these elements are kept to fairly low values. One major disadvantage of this approach for use with real flight data is that the filter seems very sensitive to the level of bias noise modelled. The values of $Q_{33} \rightarrow 0_{10,10}$ must be carefully chosen, otherwise filter divergence or linear dependencies are inevitable during the particular filter run, especially if the covariances are much larger than the actual bias noise present on the signals. The acceptable responses shown in Figure 7-17 were obtained only after much "fine tuning" of the augmented Q matrix. Very small diagonal elements meant that the filter would run in practically a non-augmented fashion, while severe divergence and subsequently large position errors would occur when the elements were too large (say $> 0.5 \mu \text{ sec}^2$). The filter also seems to be quite sensitive to the magnitude of Q relative to P_0 . Although it has been shown that the augmented system is able to track biases and provide accurate position estimates, it is felt that this

should not be the only method used to handle anomalies in an on-board implementation. This is not a truly adaptive scheme, and significant changes in the VLF noise levels would likely be undetected by the augmented filter. A second problem is that larger bias shifts would not be quickly accounted for if the Q matrix diagonal elements are fairly small, as required. A more practical filter would model a small amount of bias plant noise to account for minor bias changes, and would be combined with the automatic detection and compensation scheme previously described to handle the more severe anomalies. The next subsection describes some of the results obtained when this approach was applied to the real flight data.

7.4.4 Automatic Contingency Algorithm

Several runs using the "pseudo" real flight data described in the preceding subsection were made in order to verify the correct operation of the automatic contingency detection scheme used in conjunction with the augmented model. The filter is required to operate "normally" when no anomalies are present (i.e. no "false alarms") and to quickly detect and estimate a bias shift if and when it occurs.

As in previous runs, the bias noise covariances are chosen so as to prevent filter divergence and linear dependencies (the covariances are the same as those used to produce the responses shown in Figure 7-17). Figure 7-19 shows the responses when the automatic contingency algorithm is invoked, and Table 7-5 indicates the sequence of events for this run. The filter took 230 seconds to completely and accurately detect and estimate the bias jump. During that period, station L was weighted out of the filter solution because of its correspondingly large R matrix element. With some additional fine tuning, the P matrix element (P_{55}) could be set to a value at 2110 seconds which would allow a more accurate estimation of the bias shift at that time, and hence only 110 seconds would be used in the detection and estimation sequences.

Another run was made with $Q_{55} = 0.001 \mu \text{ sec}^2$. In the previous filter run, it was seen that this small covariance was not enough to allow an accurate estimation of the new bias level in station L. With the automatic detection scheme invoked, however, the bias shift was accurately detected and estimated this time.

The filter runs described here verify that VLF bias anomalies can be handled quite well by the filter in the configuration described here. Although not specifically examined, it is felt that increases in the VLF noise level can also be handled by the adaptive filtering scheme when real flight data is used.

7.5 Summary and Conclusions

This chapter has outlined the filter runs using real flight data, and the results indicate that a practical system has been formulated which can be safely implemented on board the research aircraft. Section 7.4 has shown that various refinements in both the specified filter parameters and the signal models themselves improved the accuracies of the state estimates. Although the models likely do not exactly represent the true signals, it appears as though they are adequate. The filter in its current configuration is not an optimal one, but the position errors are small and filter divergence is controlled.

The tracking of VLF anomalies is made practical through use of a combined augmented plant noise model and automatic on-line contingency detection scheme. The automatic algorithm comes into operation for gross bias anomalies and changes in the VLF noise levels. Otherwise, the non-zero Q matrix diagonals in the augmented scheme force the filter to track minor changes and fluctuations in the biases. Some further work could be done on the automatic algorithm in order to reduce the anomaly identification/estimation time, but the current configuration does give acceptable results.

The accuracies of the position estimates are felt to be quite good for the flight trajectory used in this study. The numerous turns no doubt cause decreased accuracies in the Doppler/heading data and the filter has done well in tracking the associated plant biases. In addition, the accuracies of the "on-top" reference positions are not as good as one would like.

One further refinement to the adaptive filter scheme would be to use the known on-top position coordinates to re-set the LAT/LONG position estimated by the filter at the time the aircraft flies over the particular on-top landmarks. In effect, the first two filter state variables would be "re-initialized" periodically during the flight. In

this manner the errors in the LAT/LONG positions estimates at the on-top times would be considerably reduced, resulting in an overall improvement in filter accuracy.

CHAPTER 8

CONVAIR 580 ON-LINE IMPLEMENTATION

8.1 Introduction

The final step in the development of a working Kalman filter is its implementation in a real-time environment. At the time of this writing, the filter described in this report is still in the stages of being proven on-board the Convair 580. It was hoped that the filter would be usable on a recent NAE aeromagnetics survey project in the Caribbean but acceptable operation was not achieved for several reasons. The filter is now close to being run in a real-time mode and several "bugs" that were previously unnoticed have now been corrected. The results of the simulations and off-line real data runs are very encouraging: the actual filter development has been completed and all that remains is its integration into the aircraft computing system. Of course, numerous flight tests will have to be performed in order to fully verify the filter's real-time capability.

Detailed information on the NAE Convair 580 Computer system may be found in [1] and [66]. A brief description is provided here, however.

The research aircraft is a self contained flying digital laboratory with capabilities of processing various sensor information while in the air. Although primarily used for aeromagnetic research, several support systems have been implemented (VLF navigation, for example) which in themselves open up other areas of research such as the Kalman filter development. The heart of the various systems is the aircraft minicomputer.

As indicated in [66] the digital computing system consists of the following:

- Interdata Model 7/32 central processor; 150K byte memory.
32 bit machine with floating point capability.
- two magnetic tape units for information storage.
- 10 Megabyte Disc system.
- portable keyboard/printer.

- Interfaces for - Magnetic anomaly detection counters
- Navigation receivers
- Display systems
- etc.

The airborne operating system ("OBOS": On Board Operating System) allows either ASSEMBLER or FORTRAN V language programs to be run. Various synchronous timing rates have been programmed, including 32, 8, 2, 1 and 0.1 Hz. The 0.1 Hz frequency (10-second cycle) coincides with the GNS-200 VLF data acquisition rate.

8.2 Kalman Filter On-Line Implementation Sequence

The sequence of events leading to the actual loading of the on-line, real time version of the filter on the aircraft computer is outlined in this section.

In order to verify that the 7/32 version of the filter operated properly, the implementation sequence shown in Figure 8-1 was followed. Before the on-line version was even loaded onto the lab 7/32, it was run and checked in the IBM TSS environment. This allowed greater flexibility for debugging, parameter changes, and output analysis (a flatbed plotter was available for TSS use). The 7/32 version of the filter is somewhat simpler than the TSS version since many of the user specified options were eliminated and streamlining was necessary in order to reduce the memory requirements. Use of previously generated simulated flight data provided a "control" data set which made debugging much simpler (the output of the 7/32 version was compared with the output of a previous TSS run using the same data).

After program debugging and accuracy verification, the routines were punched on cards and then loaded into the lab 7/32. A certain amount of debugging was necessary to eliminate subtle differences between the IBM TSS FORTRAN and the FORTRAN code and syntax accepted by the 7/32. After compilation, the program on the 7/32 was run, again with a control data set and then with the real flight data. Results from these runs indicated that the 7/32 filter was practically as efficient and accurate as the TSS version.

The actual routine loaded into the aircraft computer has several segments of ASSEMBLER language code, which is necessary for the acquisition

of data such as the Doppler coefficients (A1-A8) and the raw VLF μ sec data. The off-line version of the filter acquires this data from a pre-stored data set; in real time operation, however, the required data is picked off from a "Navigation Parameter Table" (NPT) resident in OBOS. A Doppler routine currently invoked in the aircraft navigation system calculates the Doppler coefficients while the VLF data is put into the NPT by "VLFNAV", an ASSEMBLER Program also already resident in the system. It should be noted that the integration of the Kalman filter into the existing aircraft navigation system was hampered by several "handshaking" problems: the Doppler coefficients were initially miscalculated and proper initialization of the filter did not seem to occur. These problems have since been identified and corrected.

8.3 General Real-Time Considerations

The three main aspects of the minicomputer filter implementation that need to be considered are:

- timing - can the filter perform all of its functions in the time segment allotted?
- memory - can the minicomputer provide enough storage space; or, alternatively, does the filter program and memory requirements exceed the available space?
- accuracy - is the filter prone to numerical accuracy problems due to round-off errors?

The first two considerations exist because there are numerous other tasks that the computer must process: each task requires a finite amount of processing time and each needs a certain amount of computer memory. The third consideration is independent of the other system tasks, but is equally as important as the first two.

The Convair 580 On Board Operating System (OBOS) operates on a basic 10 second master interrupt cycle, coinciding with the 10 second GNS-200 VLF acquisition rate. The basic timing for the navigation tasks is shown in Figure 8-2. Off-line runs made on the lab 7/32 have shown that the filter requires approximately 2.5 seconds to complete each update process, so the filter cycle time actually appears to present no problem: the filter processing can fit nicely into the 10 second segment as indicated in the timing diagram. The 20 second update interval referred to

in section 6-7 is not necessary. The second problem associated with timing is the apparent mismatch in times between the integrated Doppler availability (at the 10 second interrupt point) and the time that the last raw GNS-200 VLF count is valid (as much as a 2 second difference). On average, there is approximately a one second difference between the Doppler valid time and the VLF valid time. This problem was discovered only well into the filter evaluation phase and must be further analyzed and corrected if necessary. The third timing consideration is that the Kalman latitude/longitude estimate ("KLL" in the timing diagram) is not available until well into a particular 10 second interval, but the KLL is valid at the start of that interval. Because of this, the filter position estimate is updated (extrapolated) by integrated Doppler data which is continuously available on a 2 second cycle. It is this extrapolated position which is then displayed on the navigation console at the time KLL is available. The timing diagram shows a segment labelled "VLFNAV" which is the navigation routine already resident in OBOS. It is proposed that the Kalman filter routine immediately follow VLFNAV.

One of the more critical initial considerations was the memory requirements of the Kalman filter. The main on-line program itself is not overly large, but the subroutines and matrices use a significant portion of the available memory. However, the filter program has been loaded on the aircraft computer with some room to spare, so to say. Table 8-1 indicates the approximate memory requirements of the Kalman filter task. Fortunately, further expansion of the filter, or addition of other related tasks, can be managed by installation of more memory boards in the computer. The third important consideration is the numerical accuracy of the minicomputer and there are several possible problem areas:

- ability of the small computer to accurately invert a 6 X 6 matrix in the Kalman gain determination
- control of the positive-definiteness of P_k
- slow numerical degradation with time due to word length errors

Numerical considerations are of course not limited to just a Kalman filter implementation and they must be addressed in any application involving a minicomputer. Many authors emphasize the importance of maintaining a

positive-definite P matrix and it would seem that the appearance of negative diagonal elements in this matrix is a very possible problem. This was never the case in the IBM computer runs, although very small P matrix diagonals have been observed. The 7/32 minicomputer runs yielded similar results: sometimes very small diagonals, but never negative-valued ones. With the bias noise modelling (all Q matrix diagonals non zero) the diagonal elements of P are kept at a "reasonable" non-zero level, thus being prevented from getting overly small.

The matrix inversion routine used in the minicomputer implementation is the same as that in the larger computer, and it is felt to be an efficient algorithm. It includes several test sequences which prevent an improperly inverted matrix from being used in the filter gain calculation. Improper inversion, as noted previously, can occur depending on the relative sizes of the diagonals of $\underline{HP'H^T}$ and R. Detailed analyses of various matrices generated during several runs on both computers have indicated that the 6 X 6 matrix has consistently been properly inverted.

The slow numerical degradation problem is a definite possibility in both computer implementations. It is felt, however, that even with the single precision operations used here, unaccounted small modelling errors inherent in the filter design would tend to be much larger than any errors associated with word length, even if the filter is run for a long period of time.

Comparisons of output from the lab 7/32 with output from the IBM 3033 (for identical initial filter conditions) indicate that the minicomputer accuracy is very good, and the state variable estimates given by the smaller computer are well within 0.5% of the main computer estimates. It appears that the 7/32 will be sufficiently accurate for on-line filter operation: numerical errors do not seem to be a problem area.

CHAPTER 9

SUMMARY AND CONCLUSIONS

9.1 Summary

This report has detailed the development and implementation of a practical Kalman filter integrated navigation system suitable for use on board the NAE Convair 580 research aircraft. A Doppler/compass system is used to define the process dynamics (i.e. the aircraft motion) while VLF μ sec "distance" measurements define the observations required by the filter. Because of the nonlinear nature of both the process and observations, a "quasi-linear" Kalman filter is used rather than the linear filter originally proposed by Kalman [7].

The integrated system has been formulated from basic principles: fundamental sensor errors have been analyzed in order that simple, yet appropriate, models could be proposed and the Kalman filter itself has been derived from basic statistical and intuitive concepts. Of particular importance was the consideration of the individual sensors being used, since, as seen in Chapter 7, the filter will give acceptable state estimates only once an accurate plant error model is defined.

The system has been proven through numerous off-line simulation runs which have shown the possible deficiencies as well as the usefulness of the Kalman filter approach to system integration. The likely occurrence of VLF anomalies during an aircraft flight necessitated the development of an automatic on-line compensation scheme which, through simulations and use of real navigation data, has been proven to be acceptable, although not completely perfect. The adjustment of the observation covariance matrix allows the filter to account for an increase in the VLF noise level while increasing the state error covariance matrix allows tracking of VLF bias anomalies such as drifts or jumps.

The tenth-order filter uses six VLF stations. A reduced order filter will still display adequate performance providing that at least three VLF stations are available (i.e. the filter would then have seven state variables). Degraded accuracy is expected if only one or two VLF stations are used. Of course, the quality of the VLF signals themselves will also affect the accuracy of the state estimates.

The simulation studies have shown the importance of accurately estimating the initial state vector and initial state error covariance

matrix (i.e. $\hat{X}_0$ and $\hat{P}_0$). An overly optimistic initial error covariance matrix (small $\hat{P}_0$) forces the filter to respond in a rather sluggish manner and hence any errors in the initial state estimate are propagated for a considerable length of time. A pessimistic initial covariance, however, will result in a rapid response but this will often be accompanied by significant transient "spikes" resulting in large state estimation errors for a brief period. It is concluded that a pessimistic initial covariance is preferable over an optimistic one.

The basic update interval for the filter is ten seconds. The interval may be extended to 20, 30, or even 40 seconds without a significant decrease in accuracies. Operation at extreme north latitudes would likely have a detrimental effect on the performance if the longer update intervals are used.

The accuracy of the proposed on-line filter has been analyzed by using real flight navigation data. A maximum total position error of approximately 0.4 NM was observed for a total flight time of about one hour. Other navigation schemes using the same data have given errors of more than 2 NM [59]. The on-line anomaly detection algorithm has been shown to work quite well on "pseudo" real data in which a VLF bias jump was effected halfway into the flight.

The actual on-board filter is currently being implemented on the aircraft Interdata minicomputer. General real time problems have been considered in this report and, currently, the main problem seems to be the integration of the Kalman filter routines with the already existing navigation programs.

9.2 Future Work

Although the navigation system developed in this study is practically complete, there are several suggested topics which could be explored in future work.

- (i) Further evaluation of the filter using real navigation data- the off-line runs using the one hour of real data have indicated that the performance of the filter is good. Additional runs using data from longer flights with different trajectories (e.g. straight-line tracks) would be useful in showing possible

deficiencies in the filter, or, alternatively, further verifying the accuracy attainable by the filter.

- (ii) Assessment of an augmented integrated system - the addition of a third type of navigation sensor other than Doppler and VLF would likely improve the accuracy of the filter. A suggested method of augmenting the current filter configuration is to use an inertial navigation system to provide additional observations. The Convair currently has a Litton LTN-51 installed as part of its navigation complex.
- (iii) Operation at high latitudes - as previously indicated, the filter accuracy would likely be severely reduced when the aircraft operates at northern latitudes (greater than about 80-85°N). An algorithm could be developed to allow the system to perform adequately in this situation.

9.3 Conclusions

Although not yet fully operational in an on-line environment, it is felt that the Kalman filter integrated navigation system developed in this study will provide accurate navigation information when used in a real-time flight situation. Although apparently not as accurate as some current high performance (and high cost) systems, the integrated system provides more accurate position and bias estimates than would be obtained from the individual sub-systems currently installed on the NAE Convair 580 aircraft. Thus, improved performance is possible without having to consider a new and higher priced navigation system.

The usefulness of the Kalman filter approach to navigation system integration is indicated by its ability to provide accurate position estimates and, at the same time, to detect and compensate possible anomalous conditions, thus maintaining fairly accurate navigation performance even under difficult situations. The filter's ability to detect system faults such as a VLF station outage or a sudden and severe bias jump further attest to its utility and worth.

REFERENCES

1. Hardwick, C.D. The NAE Convair 580 research aircraft - An up-date for potential users. NAE Laboratory Technical Report LTR-FR-64, National Research Council, Canada, May 1978.
2. Gauss, K.G. Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientum. Hamburg, 1809 (Translation: Theory of Motion of the Heavenly Bodies. Dover, New York, 1963).
3. Sorenson, H.W. Least-squares estimation: from Gauss to Kalman. IEEE Spectrum, July 1970, 63-68.
4. Fisher, R.A. On an absolute criterion for fitting frequency curves. Messenger of Math., Vol. 41, 1912, P. 155.
5. Kolmogorov, A.N. Interpolation and extrapolation of stationary random sequences. Bull. Acad. Sci. USSR, Math. Ser. Vol. 5, 1941, 3-14, (Translation: RAND Corp., Santa Monica, Calif., Memo RM-3090-PR).
6. Wiener, N. The extrapolation, interpolation, and smoothing of stationary time series with engineering applications. John Wiley and Sons, Inc., New York, 1949. Originally a classified report: MIT Radiation Lab, Cambridge, Mass. Feb. 1942.

7. Kalman, R.E. A new approach to linear filtering and prediction problems. Trans. ASME Journal of Basic Engineering, Vol. 82, March 1960, 34-45.
8. Kalman, R.E., and Bucy, R.S. New results in linear filtering and prediction theory. Trans. ASME, J. Basic Engineering, Vol. 83, Dec. 1961, 95-107.
9. Kaminski, P.G., Bryson, A.E., and Schmidt, S.F. Discrete square root filtering; a survey of current techniques. IEEE Trans. Auto. Control, Vol. AC-16, No. 6, Dec. 1971, 727-736.
10. Schlee, F.H., Standish, C.J., and Toda, N.F. Divergence in the Kalman filter. AIAA Journal, Vol. 5, No. 6, June 1967, 1114-1120.
11. Price, C.F. An analysis of the divergence problem in the Kalman filter. Short paper in IEEE Trans. Auto. Control, Vol. AC-13, No. 6, Dec 1968, 699-702.
12. Fitzgerald, R.J. Divergence of the Kalman filter. IEEE Trans. Auto. Control, Vol AC-16, No. 6, Dec 1971, 736-747.
13. Farrell, J.L. Integrated Aircraft Navigation. Academic Press, Inc., New York, 1976.
14. Gelb, A, editor Applied Optimal Estimation. MIT Press, Cambridge, Mass., 1974.
15. Kailath, T. An innovations approach to least-squares estimation Part I: linear filtering in additive white noise. IEEE Trans. Auto. Control, Vol. AC-13, No. 6, Dec 1968, 646-655.

16. Mehra, R.K. On the identification of variances and adaptive Kalman filtering. IEEE Trans. Auto. Control, Vol AC-15, No. 2, April 1970, 175-184.
17. Boozer, D.D., and McDaniel, W.L., Jr. On innovation sequence testing of the Kalman filter. Technical Correspondence in IEEE Trans. Auto. Control, Vol. AC-17, No. 1, Feb. 1972, 158-160.
18. Chin, L., and Kaufmann, R. Improved navigation using adaptive Kalman filtering. Proceedings of ION Bicent. Nat. Aero Symp., April 27-28, 1976, 11-20.
19. McAulay, R.J., and Denlinger, E. A decision-directed adaptive tracker. IEEE Trans. Aero. Elect. Systems, Vol AES-9, No. 2, March 1973, 229-236.
20. Mehra, R.K., and Peschon, J. An innovations approach to fault detection and diagnosis in dynamic systems. Correspondence item in Automatica, Vol. 7, 637-640.
21. Willsky, A.S., and Jones, H.L. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. IEEE Trans. Auto. Control, Vol. AC-21, No.1, Feb. 1976, 108-112.
22. Friedland, B. Treatment of bias in recursive filtering. IEEE Trans. Auto. Control, Vol. AC-14, No. 4, Aug. 1969, 359-367.
23. Friedland, B. Maximum-likelihood estimation of a process with random transitions (failures). IEEE Trans. Auto. Control, Vol. AC-24, No. 6, Dec. 1979, 932-937.

24. Friedland, B. Multidimensional maximum-likelihood failure detection and estimation. IEEE Trans. Auto. Control, Vol. AC-26, No. 2, April 1981, 567-570.
25. Friedland, B., and Grabousky, S.M. Estimating sudden changes of bias in linear dynamic systems. IEEE Trans. Auto. Control, Vol. AC-27, No. 1, Feb. 1982, 237-240.
26. Gregson, F. Fault detection possibilities in a system employing Kalman filtering. Paper No. 25 in AGARD Conference Proceedings CP116, Inertial Navigation Components and Systems, Feb. 1973.
27. Jazwinsky, A.H. Limited memory optimal filtering. IEEE Trans. Auto. Control, Vol. AC-13, No. 5, Oct. 1968, 558-563.
28. Tarn, T.J., and Zaborszky, J. A practical, nondiverging filter. AIAA Journal, Vol. 8, No. 6, June 1970, 1127-1133.
29. Sorenson, H.W., and Sacks, J.E. Recursive fading memory filtering. Information Sciences, Vol. 3, 1971, 101-119.
30. Sacks, J.E., and Sorenson, H.W. Nonlinear extensions of the fading memory filter. IEEE Trans. Auto. Control, Vol. 16, No. 5, Oct. 1971, 506-507.
31. Richman, J., and Friedland, B. Design of optimum mixer-filter for aircraft navigation systems. NAECON '67, Proc. 19th Annual-Nat. Aero. Elect. Conference, Dayton, May 1967, 429-438.
32. Cox, H. On the estimation of state variables and parameters for noisy dynamic systems. IEEE Trans. Auto. Control, Vol. AC-9, No. 1, Jan. 1964, 5-12.

33. Friedland, B. and Bernstein, I. Estimation of the state of a nonlinear process in the presence of nongaussian noise and disturbances. J. Franklin Inst., Vol. 281, July 1966, 455-480.
34. Jazwinsky, A.H. Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.
35. Anderson, B.D.O, and Moore, J.B. Optimal Filtering, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1979.
36. Zimmerman, W. Optimum integration of aircraft navigation systems. IEEE Trans. Aero. Elect. Systems, Vol. AES-5, No. 5, Sept. 1969, 737-747.
37. Gaines, H.T. Flight test of a Kalman filter Doppler aided strapdown inertial navigator. NAECON '71, Proc. Nat. Aero. Elect. Conference, Dayton, 1971, 91-99.
38. Schmidt, S.F., Weinberg, J.D., and Lukesh, J.S. Application of Kalman filtering to the C-5 guidance and control system. Chapter 13 in AGARDograph AG139, Theory and Applications of Kalman Filtering, Feb. 1970.
39. Schwegler, H.F. Development of the integrated all-weather navigation system for Tornado (MRCA). Paper No. 22 in AGARD Conference Proceedings CP220, Applications of Advances in Navigation to Guidance and Control, May 1977.
40. Brown, R.G. Integrated navigation systems and Kalman filtering: a perspective. Navigation (Inst. of Navigation), Vol. 19, No. 4, Winter 1972-73, 355-362.

41. - Practical Aspects of Kalman Filtering Implementation, AGARD Lecture Series No. 82, March 1976.
42. - Hybrid Navigation Systems, AGARD Conference Proceedings CP54, Jan. 1970.
- 42b - Theory and Applications of Kalman Filtering, AGARDograph AG139, Feb. 1970.
43. - Applications of Advances in Navigation to Guidance and Control, AGARD Conference Proceedings CP220, May 1977.
44. Kailath, T. A view of three decades of linear filtering theory. IEEE Trans. Inf. Theory, Vol. IT-20, No. 2, March 1974, 146-181.
45. Mendel, J.M., and Giesekeing, D.L. Bibliography on the linear-quadratic-Gaussian problem. IEEE Trans. Auto. Control, Vol. AC-16, No. 6, Dec. 1971, 847-869.
46. Kayton, M., and Fried, W.R., eds. Avionics Navigation Systems, John Wiley and Sons, Inc., New York, 1969.
47. Beck, G.F., editor. Navigation Systems, Van Nostrand Reinhold, London, 1971.
48. Powell, J. Aircraft Radio Systems, Pitman, London, 1981.
49. Fried, W.R. Modern aircraft navigation sensors for hybrid systems. AGARD Conference Proceedings CP54, Hybrid Navigation Systems, Ch. 10, Jan. 1970.

- 50. Tyler, J.S. Radio navigation systems- current status.
AGARDograph AG245, Principles and Operational Aspects of Precision Position Determination Systems, Ch. 1, July 1979.
- 51. - 1981 Avionics Buyers' Guide, Canadian Aviation, Vol. 54, No. 7, July 1981.
- 52. Hardwick, C.D., and Brownley, T.R. An Evaluation of Aircraft Navigation using VLF Communication Stations with the Global Navigation Inc. GNS-200 VLF Navigation System in the North Star Aircraft.
NAE Laboratory Technical Report LTR-FR-38, National Research Council Canada, Feb. 1973.
- 53. Swanson, E.R. Propagation effects on OMEGA. AGARD Conference Proceedings CP209, Propagation Limitations on Navigation and Positioning Systems, Ch. 15, Feb. 1977.
- 54. Blackband, W.T. The effects of the ionosphere on VLF navigational aids. AGARDograph AG55, Navigation Systems for Aircraft and Space Vehicles, Pergamon Press, London, 1962, pp 71-79.
- 55. Kasper, J.F., Jr. Omega: global navigating by VLF fix.
IEEE Spectrum, May 1979.
- 56. Miller, I., and Freund, J.E. Probability and Statistics for Engineers.
Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1965.
- 57. Barham, P.M., and Humphries, D.E. Derivation of the Kalman Filtering Equations from Elementary Statistical Principles.
Royal Aircraft Establishment, Technical Report No. 69095, May 1969.

58. Ogata, K. Modern Control Engineering. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1970.

59. Leach, B.W. A Kalman Filter Approach to Navigation on the NAE Convair 580 Aeromagnetics Research Aircraft. NAE, Aeronautical Report, LR-604, NRC No. 19271, National Research Council Canada, Feb. 1981.

60. Sodano, E.M. A rigorous non-iterative procedure for rapid inverse solution of very long geodesics. Bulletin Géodésique, No. 47/48, 1958, 13-25.

61. Sodano, E.M. General non-iterative solution of the inverse and direct geodetic problems. Bulletin Géodésique, No. 75, 1965, 69-89.

62. Leach, B.W., and Maskell, C.A. VLF data for aircraft navigation based on an extended Kalman filter design. Paper to be presented at the 6th IFAC Symposium on Identification and System Parameter Estimation, Washington, D.C., June 1982.

63. Mehra, R.K. On optimal and suboptimal linear smoothing. Proc. National Electronics Conference, Vol. XXIV, Dec. 1968, 119-124.

64. Huddle, J.R. Application of Kalman filtering theory to augmented inertial navigation systems. AGARDograph AG139, Theory and Applications of Kalman Filtering, Ch. 11, Feb. 1970.

65. Molnar, D., and Walton, N. A Doppler Markov error model development based on flight test data analysis. NAECON '76, Proc. Nat. Aero. Elect. Conf., Dayton, 1976, 28-31.

66. Hardwick, C.D. Proposed research capabilities of the NAE Convair 580. DME/NAE Quarterly Bulletin, No. 1976(4), National Research Council Canada, Jan. 1977.
67. Bendat, J.S., and Piersol, A.G. Random Data: Analysis and Measurement Procedures. John Wiley and Sons, Inc., New York, 1971.

TABLE 2.1 U.S. NAVY VLF COMMUNICATION AND NAVIGATION STATIONS
(from [59])

LETTER DESIGNATION	CALL SIGN	LOCATION	LATITUDE (deg/min/sec)	LONGITUDE (deg/min/sec)	FREQ. (KHz)	RAD. POWER (KW)
M	NAA	Cutler, Maine	44 38 54.0 N	67 16 54.0 W	17.80	890
W	NLK	Jim Creek, Wash.	48 12 12.0 N	121 55 00.0 W	18.60	250
L	NPM	Hawaii	21 25 30.0 N	158 09 18.0 W	23.40	400
S	NWC	Northwest Cape Australia	21 49 00.0 S	114 09 48.0 E	22.30	260
R	GQD	Anthorne, G.Br.	55 00 00.0 N	03 30 00.0 W	19.00	n/a
G	GBR	Rugby, G.Br.	52 22 10.0 N	01 11 55.0 W	16.00	250
J	NDT	Yosami, Japan	34 58 15.0 N	137 01 19.0 E	17.40	125
O	JXN	Helgeland, Norway	67 04 00.0 N	14 04 00.0 E	16.40	150
A	WSS	Annapolis, Maryland	38 59 06.0 N	76 27 12.0 W	21.40	500
D*	-	Lamore, N.D.	46 21 57.2 N	98 20 08.0 W	13.10	10

n/a : Not available

* : D is actually an OMEGA Navigation station whose side frequency is being used for VLF navigation.

TABLE 2.2 VLF NAVIGATION RECEIVERS

MANUFACTURER	MODEL	WEIGHT	PRICE (\$CAN)
Canadian Marconi Company	CMA-771	40	36-48,000
Collins Canada Division	LRN-80	28	49,500
	LRN-85	29	59,000
Global Navigation Inc.	GNS-500A	38	50-53,000

TABLE 5.1 KALMAN FILTER MODES

FLIGHT/SIGNAL OPTION	NAIVE FILTER	SMART FILTER
(A) Doppler/Heading Bias Drift	$\underline{P}$ not adjusted	P_{33} increased for velocity bias drift P_{44} increased for heading bias drift P_{33} , P_{44} propagate normally when drifting has stopped
(B) VLF Bias Drift	$\underline{P}$ not adjusted	$P_{i+4,i+4}$ increased for VLF station i bias drift. Allowed to propagate normally when drifting stops.
(C) VLF Noise Change	$\underline{R}$ not adjusted	R_{ii} adjusted to reflect increase or decrease in variance of VLF station i noise.
(D) Circular Track	$\underline{Q}$ not adjusted	Q_{11} , Q_{22} increased to reflect the degradation of Doppler data due to acceleration. Reset when straight track resumes.

TABLE 6.1. SIMULATOR AND FILTER PARAMETERS : BASIC FILTER OPERATIONSIMULATED FLIGHT/DATA CONDITIONS

Initial LAT/LONG	45.0° N 75.0° W
Groundspeed	250 KNOTS
Groundspeed noise variance	10 KNOTS ²
Groundspeed Bias	10 KNOTS
Heading Angle	45 DEGREES
Heading Angle Variance	10 DEG ²
Heading Angle Bias	10 DEGREES
Drift Angle	0 DEGREES
VLF Stations used	0 G M A D W
VLF signal variance	10 μSec ² all stations
VLF Phase Bias	10 μSec all stations

INITIAL FILTER PARAMETERS

PLAT₀/PLONG₀ 45.001°/75.001°

Estimated Biases $\hat{B}_V$ 0.0
 $\hat{B}_{\alpha\beta}$ 0.0
 $\hat{B}_{\phi_i}$ 0.0 all i

Initial State Covariances

P_{011} 0.0001 DEG²
 P_{022} 0.0001 DEG²

	<u>SMALL</u>	<u>MEDIUM</u>	<u>LARGE</u>	
P_{033}	0.10	5.0	100.0	KNOTS ²
P_{044}	0.10	5.0	100.0	DEG ²
P_{055}	0.10	5.0	200.0	μSEC ²
$P_{010,10}$	0.10	5.0	200.0	μSEC ²
Q_{011}, Q_{022}		50.0, 50.0		KNOTS ² DEG ²
R_{0ii}		20.0		μSEC ² all i

TABLE 6.2 SIMULATOR AND FILTER PARAMETERS : VARIABLE QUALITY DATA RUNSSIMULATED FLIGHT/DATA CONDITIONS

Initial LAT/LONG	45° N / 75° W
Groundspeed	250 KNOTS
Groundspeed Bias (B_V)	1.0 KNOTS
Heading Angle	45 DEGREES
Heading Angle Bias ($B_{\alpha\beta}$)	1.0 DEGREES
Drift Angle	0.0 DEGREES
VLF Phase Bias (B_ϕ)	0.1 μ SEC, all stations

SIGNAL VARIANCES

	<u>POOR</u>	<u>GOOD</u>	<u>V. GOOD</u>	
Groundspeed noise variance (σ_V^2)	10.0	1.0	0.1	KNOTS ²
Drift Angle noise variance (σ_α^2)	5.0	0.25	0.1	DEG ²
Heading Angle noise variance (σ_β^2)	5.0	0.25	0.1	DEG ²
VLF Phase variance (σ_ϕ^2)	10.0	1.0	0.1	μ SEC ²

INITIAL FILTER PARAMETERS

P_{LAT_0} / P_{LONG_0} 45.001°/75.001°

Estimated Biases 0.0 , all initial bias estimates

Initial Error Covariance Matrix:

P_0

The image shows a log-log plot with data points and a fitted curve. The y-axis has labels 0.0001, 0.0001, 1.0, 1.0, 0.01, 0.01, and 0.01. The x-axis has labels 0.0001, 0.0001, 1.0, 1.0, 0.01, 0.01, and 0.01. A solid line represents the fitted curve, and several data points are plotted as small circles.

(continued)

TABLE 6.2 (continued)

Autocovariance Matrix Diagonal Elements:

		<u>POOR</u>	<u>GOOD</u>	<u>V. GOOD</u>
Q_{11}	$(\hat{\sigma}_{V.}^2)$	50.0	5.0	0.5 KNOTS ²
Q_{22}	$(\hat{\sigma}_{\alpha\beta}^2)$	50.0	2.5	0.5 DEG ²
$R_{11} \rightarrow R_{66}$	$(\hat{\sigma}_{\phi}^2)$	20.0	2.0	0.20 μ SEC ²

TABLE 7.1 VISUAL REFERENCE POINTS FOR REAL FLIGHT DATA NAVIGATION STUDY

ON-TOP REFERENCE	TIME FROM START	GEOGRAPHICAL LOCATION	
Intersection RW 25-32	0 secs	45°18'57"N	75°39'55"W
Ottawa VOR	287	45 26 30	75 53 50
Kinburn Beacon	543	45 25 00	76 08 58
Arnprior Bridge	1134	45 26 06	76 21 00
Kinburn Beacon	1311	45 25 00	76 08 58
Ottawa VOR	1519	45 26 30	75 53 50
Intersection RW 25-32	1795	45 18 57	75 39 55
Uplands Beacon	1978	45 13 45	75 29 36
Uplands Beacon	2238	"	"
Ottawa Beacon	2792	45 21 36	75 33 41
Intersection RW 25-32	2967	45 18 57	75 39 55

TABLE 7.2 REAL FLIGHT DATA NOISE VARIANCES

STATION	GROUND RUN VARIANCE	IN-FLIGHT VARIANCES		
		1639-1989 secs	1760-1910 secs	1760-2000 secs
L	0.011 μSEC^2	0.050	0.023	0.020
M	0.005	0.061	0.004	0.014
D	0.105	0.167	0.053	0.107
G	0.096	0.178	0.033	0.067
W	0.010	0.092	0.042	0.054
R	0.196	0.139	0.106	0.144

DOPPLER
 σ_V^2 1.923 KNOTS²
 σ_α^2 1.048 DEG²
HEADING
 σ_β^2 0.659 DEG²

TABLE 7.3 NOMINAL INITIAL PARAMETERS FOR REAL FLIGHT DATA RUNINITIAL STATE ESTIMATE

$\hat{P}_{LAT_0}, \hat{P}_{LONG_0}$ 45.3158°, 75.6653°
 $\hat{B}_{V_0}$ 0.0 KNOTS
 $\hat{B}_{\alpha\beta_0}$ 0.0 DEGREES
 $\hat{B}_{\phi_i}$ 0.0 μSEC all i

INITIAL ERROR COVARIANCE MATRIX

$$P_0 = \begin{bmatrix} 0.001 & & & & & & & & & \\ & 0.001 & & & & & & & & \\ & & 10.0 & & & & & & & \\ & & & 5.0 & & & & & & \\ & & & & 0.10 & & & & & \\ & & & & & 0.10 & & & & \\ & & & & & & 0.10 & & & \\ & & & & & & & 0.10 & & \\ & & & & & & & & 0.10 & \\ & & & & & & & & & 0.10 \end{bmatrix}$$

PLANT AUTOCOVARIANCE MATRIX

$$Q_0 = \begin{bmatrix} 5.0 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}$$

OBSERVATION AUTOCOVARIANCE MATRIX

$$R_0 = \begin{bmatrix} 0.10 & & & & & \\ & 0.10 & & & & \\ & & 0.15 & & & \\ & & & 0.15 & & \\ & & & & 0.10 & \\ & & & & & 0.50 \end{bmatrix}$$

TABLE 7.4 INITIAL PARAMETERS FOR REFINED REAL FLIGHT DATA FILTER RUN

 $\hat{X}_0$

same as Table 7.3

 P_0

0.001									
	0.001								
		100.0							
			100.0						
				0.01					
					0.01				
						0.01			
							0.01		
								0.01	
									100.0

 Q_0

10.0	0.0
0.0	5.0

 R_0

0.10					
	0.05				
		0.15			
			0.15		
				0.10	
					0.50

TABLE 7.5 SEQUENCE OF EVENTS FOR ON-LINE VLF BIAS JUMP DETECTION

<u>TIME FROM START</u>	<u>ACTION</u>
(a) 0-2000 seconds	<ul style="list-style-type: none"> - filter operates 'normally' - automatic anomaly detection routine monitors VLF noise variance and residuals
(b) 2000	<ul style="list-style-type: none"> - VLF anomaly detected in VLF station L. R_{ij} set to 10,000 μSEC^2
(c) 2110	<ul style="list-style-type: none"> - Bias jump identified - P_{55} set to 39.0 - R_{11} reset to $2 \times S_e^2$
(d) 2120	<ul style="list-style-type: none"> - R_{11} again set to 10,000 - (bias jump was not completely estimated at 2110 secs)
(e) 2230	<ul style="list-style-type: none"> - P_{55} set to 0.12 - bias now accurately estimated - R_{11} reset.
(f) 2230-3000	<ul style="list-style-type: none"> - filter in normal mode

TABLE 8.1 MEMORY REQUIREMENTS FOR THE ON-BOARD KALMAN FILTER TASK

(A) Main ASSEMBLER Kalman Filter Code	2.4 K Bytes
(B) Kalman Filter FORTRAN Subroutines (code)	3 K Bytes
(C) Variable Storage (including matrices)	4 K Bytes
<u>TOTAL</u>	<u>9.4 K Bytes</u>

TN	True North
MN	Magnetic North
VAR	Magnetic Variation
TRK	actual ground track angle ($\alpha + \beta$)
β	heading angle wrt True North
α	heading drift angle
V_x	along track ground velocity
V_y	across track ground velocity
V	aircraft groundspeed

FIGURE 2.1 DEAD RECKONING NAVIGATION NOMENCLATURE

FIGURE 2.2 DOPPLER RADAR: 3-BEAM JANUS CONFIGURATION

FIGURE 2.3 POSITION ERROR DUE TO UNCORRECTED
DOPPLER/HEADING BIASES

FIGURE 2.4 VLF COMMUNICATION AND NAVIGATION STATIONS

FIGURE 2.5 LOCATIONS OF VLF NAVIGATION STATIONS
RELATIVE TO OTTAWA

(a) rho-theta

(b) theta-theta

● ground radio station
○ aircraft position

(c) rho-rho

FIGURE 2.6 RADIO POSITION FIXING METHODS
(from [46])

(d) Hyperbolic scheme showing two pairs of stations and two systems of Lines of Position

FIGURE 2.6 (con't)

FIGURE 5.1 MAIN SECTIONS OF THE KALMAN FILTER
INTEGRATED NAVIGATION SYSTEM AND
SIMULATOR

FIGURE 5.2 BLOCK DIAGRAM OF THE KALMAN FILTER
INTEGRATED NAVIGATION SYSTEM SIMULATOR

FIGURE 5.3 TYPICAL SIMULATED FLIGHT TRAJECTORIES

FIGURE 5.4 REPRESENTATION OF THE MEASUREMENT RESIDUAL

FIGURE 5.5 TYPICAL INNOVATIONS SEQUENCE FOR A VLF STATION,
WITH A DRIFTING BIAS

(a) Bias Drift Situation

(b) Bias Jump Situation

FIGURE 5.6 TYPICAL VLF BIAS DRIFT AND JUMP DETECTION CRITERIA

FIGURE 5.7 FLOWCHART OF AUTOMATIC
ON-LINE VLF ANOMALY
DETECTION ALGORITHM

VLF BIAS ESTIMATE

ESTIMATED VLF PHASE BIAS, μ SEC

POSITION ESTIMATE (Dec Deg.)

VEL/HDG BIAS ESTIMATE

KF.RUN.T 5
SIMULATED DATA

FIGURE 6.1 KALMAN FILTER STATE ESTIMATES - LARGE INITIAL BIAS ERRORS AND
LARGE INITIAL ERROR COVARIANCES

FIGURE 6.3 PROPAGATION OF P_{33} FOR SMALL, MEDIUM, AND LARGE P_0

FIGURE 6.4 PROPAGATION OF P_{55} FOR SMALL, MEDIUM, AND LARGE P_0

FIGURE 6.5 PROPAGATION OF $||P||$ FOR SMALL, MEDIUM, AND LARGE P_0

FIGURE 6.6 MEAN SQUARE ERRORS OF KALMAN FILTER ESTIMATES FOR SMALL, MEDIUM, AND LARGE P_0

ERROR IN VLF PHASE BIAS, μ SEC

LAT/LONG ERRORS (NM)

KF.RUN.T.8
SIMULATED DATA

VEL/HDG BIAS ERRORS

FIGURE 6.7 KALMAN FILTER ERROR RESPONSES -- POOR QUALITY DOPPLER, HEADING, AND VLF DATA

FIGURE 6.8 KALMAN FILTER ERROR RESPONSES — GOOD QUALITY DOPPLER, HEADING, AND VLF DATA

• VLF BIAS ERROR

KF.RUN.T 11
SIMULATED DATA

VEL/HDG BIAS ERRORS

FIGURE 6.9 KALMAN FILTER ERROR RESPONSES – VERY GOOD QUALITY DOPPLER, HEADING, AND VLF DATA

FIGURE 6.10 KALMAN FILTER ERROR RESPONSES - VERY GOOD QUALITY DATA

VLF BIAS ERROR

VEL/HDG BIAS ERRORS

FIGURE 6.11 KALMAN FILTER ERROR RESPONSES - POOR QUALITY DATA

R ASSUMED $\ll$ R ACTUAL

KF.RUN.T 15
SIMULATED DATA

VLF BIAS ERROR

VEL/HDG BIAS ERRORS

FIGURE 6.12 KALMAN FILTER ERROR RESPONSES - REDUCED ORDER FILTER WITH THREE VLF STATIONS AND GOOD QUALITY DATA

VLF BIAS ERROR

VEL/HDG BIAS ERRORS

FIGURE 6.13 KALMAN FILTER ERROR RESPONSES - REDUCED ORDER FILTER WITH THREE VLF STATIONS
AND POOR QUALITY DATA

VEL/HDG BIAS ERRORS

ERROR IN VLF PHASE BIAS, μ SEC

VLF BIAS ERROR

KF. RUN. T 19
SIMULATED DATA

FIGURE 6.14 KALMAN FILTER ERROR RESPONSES - REDUCED ORDER FILTER WITH TWO VLF STATIONS AND GOOD QUALITY DATA

VLF BIAS ERROR

ERROR IN VLF PHASE BIAS, μ SEC

LAT/LONG ERRORS (NM)

KF RUN 1 22
SIMULATED DATA

VEL/HDG BIAS ERRORS

FIGURE 6.15 KALMAN FILTER ERROR RESPONSES - REDUCED ORDER FILTER WITH TWO VLF STATIONS AND POOR QUALITY DATA

FIGURE 6.16 KALMAN FILTER ERROR RESPONSES — REDUCED ORDER FILTER WITH ONE VLF STATION AND GOOD QUALITY DATA

FIGURE 6.17 KALMAN FILTER ERROR RESPONSES - REDUCED ORDER FILTER WITH ONE VLF STATION AND POOR QUALITY DATA

FIGURE 6.18 PROPAGATION OF P DIAGONAL ELEMENTS FOR 5th ORDER FILTER

ERROR IN VLF PHASE BIAS, μ SEC

KF. RUN. T 28
SIMULATED DATA

FIGURE 6.19 NAIVE KALMAN FILTER ERROR RESPONSES - EFFECTS OF A LINEAR VLF BIAS DRIFT IN ONE STATION

ERROR IN VLF PHASE BIAS, μ SEC

KF, RUN, T 29
SIMULATED DATA

FIGURE 6.20 SMART KALMAN FILTER ERROR RESPONSES - P₅₅ FIXED
FOR DURATION OF A LINEAR VLF BIAS DRIFT

VLF BIAS ERROR

ERROR IN VLF PHASE BIAS, μ SEC

LAT/LONG ERRORS (NM)

KF, RUN. T 30
STIMULATED DATA

VEL/HDG BIAS ERRORS

FIGURE 6.21 NAIVE KALMAN FILTER ERROR RESPONSES - EFFECTS OF A SUDDEN
VLF BIAS JUMP IN ONE STATION

VLF BIAS ERROR

ERROR IN VLF PHASE BIAS, μ SEC

FIGURE 6'22 SMART KALMAN FILTER ERROR RESPONSES - P 77 SET TO A LARGE VALUE AT THE OCCURRENCE OF A SUDDEN VLF BIAS JUMP

FIGURE 6.23 NAIVE KALMAN FILTER ERROR RESPONSES - VLF NOISE INCREASE
AT 2000 SECONDS

FIGURE 6.24 SMART KALMAN FILTER ERROR RESPONSES - VLF NOISE INCREASE
AT 2000 SECONDS ; R ADJUSTED

FIGURE 6.25 EFFECT OF ERRORS IN MEASUREMENT NOISE VARIANCE,
ON STEADY STATE RMS ERROR IN STATE ESTIMATION

(from [63])

3

A

FIGURE 6.26 KALMAN FILTER ERROR RESPONSES — ON-LINE ANOMALY ALGORITHM USED TO DETECT BIAS DRIFTS IN ALL VLF STATIONS

FIGURE 6.27(a) INNOVATIONS SEQUENCE FOR DRIFTING VLF BIAS
SITUATION : AUTOMATIC ON-LINE DETECTION
SCHEME IN USE

FIGURE 6.27(b) PROPAGATION OF $P_{55, k+1}$ WHEN ON-LINE ANOMALY DETECTION IS USED

FIGURE 6.28 KALMAN FILTER ERROR RESPONSES - EXPONENTIAL DATA WEIGHTING

S = 1.05 * NO ANOMALY

FIGURE 6.29 KALMAN FILTER ERROR RESPONSES - EXPONENTIAL DATA WEIGHTING

FIGURE 7.1(a) DOPPLER/VLF KALMAN FILTER NAVIGATION SCHEME

(• from [59])

FIGURE 7.1(b) FLOWCHART OF REAL FLIGHT DATA KALMAN FILTER INTEGRATED NAVIGATION SYSTEM

FIGURE 7.1(b) (Con't)

FIGURE 7.2 FLIGHT PATTERN OF THE REAL FLIGHT DATA NAVIGATION STUDY

FIGURE 7.3 RAW DOPPLER/HEADING DATA FROM REAL NAVIGATION FLIGHT

FIGURE 7.4 POSITIONS OF VLF STATIONS USED IN THE REAL FLIGHT DATA NAVIGATION STUDY

FIGURE 7.5 KALMAN FILTER STATE ESTIMATES - REAL FLIGHT DATA
WITH NOMINAL FILTER PARAMETERS

FIGURE 7.6 KALMAN FILTER POSITION ERRORS (WITH NOMINAL PARAMETERS)

VLF BIAS ESTIMATE

VEL/HDG BIAS ESTIMATE

FIGURE 7.7 KALMAN FILTER STATE ESTIMATES - REAL FLIGHT DATA
WITH REFINED INITIAL FILTER PARAMETERS

FIGURE 7.8 KALMAN FILTER POSITION ERRORS
(WITH REFINED INITIAL FILTER PARAMETERS)

FIGURE 7.9 INNOVATIONS SEQUENCE CORRESPONDING TO VLF STATION L;
REAL FLIGHT DATA STUDY WITH REFINED PARAMETERS

FIGURE 7.10 INNOVATIONS SEQUENCE CORRESPONDING TO VLF STATION R
REAL FLIGHT DATA STUDY WITH REFINED PARAMETERS

VLF BIAS ESTIMATE

POSITION ESTIMATE (Dec Deg.)

VEL/HDG BIAS ESTIMATE

FIGURE 7.11 KALMAN FILTER STATE ESTIMATES - REAL FLIGHT DATA
WITH VLF BIAS NOISE MODELLING

FIGURE 7.12 KALMAN FILTER POSITION ERRORS
(WITH VLF BIAS NOISE MODELLING)

FIGURE 7.13 INNOVATIONS SEQUENCE CORRESPONDING TO
VLF STATION L; REAL FLIGHT DATA STUDY
WITH VLF BIAS NOISE MODELLING

VLF BIAS ESTIMATE

POSITION ESTIMATE (Dec Deg.)

VEL/HDG BIAS ESTIMATE

FIGURE 7.14 KALMAN FILTER STATE ESTIMATES - REAL FLIGHT DATA
WITH VLF AND DOPPLER BIAS NOISE MODELLING

FIGURE 7.15 KALMAN FILTER POSITION ERRORS
(WITH VLF AND DOPPLER BIAS NOISE
MODELLING)

FIGURE 7.16 INNOVATIONS SEQUENCE CORRESPONDING TO VLF STATION L;
REAL FLIGHT DATA STUDY WITH DOPPLER AND VLF BIAS NOISE
MODELLING

FIGURE 7.17 KALMAN FILTER STATE ESTIMATES - REAL FLIGHT DATA WITH
SIMULATED BIAS JUMP AND BIAS NOISE MODELLING

FIGURE 7.18 KALMAN FILTER STATE ESTIMATES - REAL FLIGHT DATA WITH SIMULATED BIAS JUMP AND INACCURATE VLF BIAS NOISE MODELLING

FIGURE 7.19 KALMAN FILTER STATE ESTIMATES - REAL FLIGHT DATA WITH
SIMULATED BIAS JUMP AND ON-LINE ANOMALY DETECTION

FIGURE 8.1 PROJECT DEVELOPMENT LEADING TO ON-BOARD INTEGRATED SYSTEM IMPLEMENTATION

FIGURE 8.2 TIMING SEQUENCE FOR DOPPLER/VLF DATA ACQUISITION
AND KALMAN FILTER UPDATES

APPENDIX A

DOPPLER VELOCITY AND DRIFT ANGLE MECHANIZATION

With reference to Figure 2.2, this appendix presents the equations necessary to convert the raw Doppler RF beam frequency data into "straight-and-level" earth-referenced aircraft velocities.

Given the three raw frequencies, f_a , f_b , and f_c (see Fig. 2.2), the components of velocity in each of the three body-fixed aircraft axis directions are given by

$$\begin{aligned} u &= \frac{c}{4f \cos(\psi)} (f_b - f_c) && \text{forward velocity,} \\ v &= \frac{c}{4f \cos(\lambda)} (f_a - f_b) && \text{lateral velocity,} \\ w &= \frac{c}{4f \cos(\Omega)} (f_a + f_c) && \text{vertical velocity} \end{aligned} \quad (A.1)$$

where c is the radiated beam propagation speed,
 f is the frequency of the radiated signal

ψ, λ, Ω are the angles that the beams form with the three aircraft axes (see [47] for more detail)

For the particular system installed on the Convair 580, these velocity components are given by

$$\begin{aligned} u &= \frac{f_b - f_c}{8.925} && \text{KNOTS} \\ v &= \frac{f_a - f_b}{4.4625} && \text{KNOTS} \\ w &= \frac{(f_a + f_c) - 50000}{20.576} && \text{KNOTS} \end{aligned} \quad (A.2)$$

These aircraft axes velocities must be converted to an earth-referenced straight-and-level coordinate system. This conversion is accomplished by accounting for the aircraft pitch and roll angles, and the process is commonly referred to as attitude stabilization.

$$\begin{aligned} V'_x &= u \cdot \cos(\theta) + v \cdot \sin(\theta) \sin(\phi) + w \cdot \sin(\theta) \cos(\phi) \\ V_y &= v \cdot \cos(\phi) - w \cdot \sin(\phi) \\ V_z &= u \cdot \sin(\theta) - v \cdot \cos(\theta) \sin(\phi) - w \cdot \cos(\theta) \cos(\phi) \end{aligned} \quad (A.3)$$

where θ is the pitch angle (nose-up is positive)
 ϕ is the roll angle (starboard wing down is positive)
 V_x is the forward velocity
 V_y is the lateral velocity
 V_z is the upward velocity

The groundspeed and drift angle are then calculated by

$$\begin{aligned} V_{DOP} &= (V_x^2 + V_y^2)^{\frac{1}{2}} \quad \text{KNOTS} \\ \alpha_{DOP} &= \arctan(V_y / V_x) \quad \text{RADIAN} \end{aligned} \quad (A.4)$$

APPENDIX B

PLANT NOISE AND UPDATE EQUATIONS

Appendix C of [59] gives the development of the continuous and discrete time plant noise equations. The discrete versions of these equations are required for the Kalman filter update equations shown in Chapter 4 of the current study. This appendix develops the discrete time equations. Reference can be made to Figure B.1 when following the derivation presented here.

The latitude position update is given by

$$P_{LAT_{k+1}} = P_{LAT_k} + \frac{1}{60} \int_{t_k}^{t_{k+1}} V_T \cos(\alpha+\beta)_T dt \quad (B.1)$$

where the subscript T refers to the true quantities, and the factor of 1/60 converts NM to degrees of position. From Figure B.1, it is seen that

$$V_T = V_{DOP} - B_V - U_V \quad (B.2)$$

$$(\alpha+\beta)_T = (\alpha+\beta) - B_{\alpha\beta} - U_{\alpha\beta}$$

Substitution into Equation B.1 gives

$$P_{LAT_{k+1}} = P_{LAT_k} + \frac{1}{60} \int (V_{DOP} - B_{V_k} - U_V) \cos(\alpha+\beta - B_{\alpha\beta_k} - U_{\alpha\beta}) dt \quad (B.3)$$

Expanding the cosine function, and assuming $U_{\alpha\beta}$ is small (i.e. $\cos(U_{\alpha\beta}) \approx 1$ and $\sin(U_{\alpha\beta}) \approx U_{\alpha\beta}$) the integral in Equation B.3 can now be written as

$$\int = \frac{1}{60} \int (V_{DOP} - B_{V_k} - U_V) [\cos(\alpha+\beta - B_{\alpha\beta_k}) + U_{\alpha\beta} \sin(\alpha+\beta - B_{\alpha\beta_k})] dt$$

Further expansion and collection of like trigonometric functions gives

$$\begin{aligned}
 \int = & \int V_{DOP} \cos(\alpha + \beta - B\alpha\beta_K) dt - \int B_{V_K} \cos(\alpha + \beta - B\alpha\beta_K) dt \\
 & - \int U_V \cos(\alpha + \beta - B\alpha\beta_K) dt + \int V_{DOP} U_{\alpha\beta} \sin(\alpha + \beta - B\alpha\beta_K) dt \\
 & - \int B_{V_K} U_{\alpha\beta} \sin(\alpha + \beta - B\alpha\beta_K) - \int U_V U_{\alpha\beta} \sin(\alpha + \beta - B\alpha\beta_K) dt
 \end{aligned}
 \tag{B.4}$$

The last term is neglected due to the assumed uncorrelated nature of the noises U_V and $U_{\alpha\beta}$. The cosine function in the first two terms can be expanded in terms of cosines and sines, to give, for the first two terms,

$$\begin{aligned}
 & \int V_{DOP} [\cos(\alpha + \beta) \cos(B\alpha\beta_K) + \sin(\alpha + \beta) \sin(B\alpha\beta_K)] dt \\
 \text{and} \quad & \int B_{V_K} [\cos(\alpha + \beta) \cos(B\alpha\beta_K) + \sin(\alpha + \beta) \sin(B\alpha\beta_K)] dt
 \end{aligned}$$

Hence, Equation B.3 may be written as

$$\begin{aligned}
 P_{LAT_{K+1}} = P_{LAT_K} + & \left[\frac{1}{60} \int V_{DOP} \sin(\alpha + \beta) dt - \frac{1}{60} \int B_{V_K} \sin(\alpha + \beta) dt \right] \sin B\alpha\beta_K \\
 & + \left[\frac{1}{60} \int V_{DOP} \cos(\alpha + \beta) dt - \frac{1}{60} \int B_{V_K} \cos(\alpha + \beta) dt \right] \cos(B\alpha\beta_K) \\
 & + \frac{1}{60} \int U_{\alpha\beta} (V_{DOP} - B_{V_K}) \sin(\alpha + \beta - B\alpha\beta_K) dt \\
 & - \frac{1}{60} \int U_V \cos(\alpha + \beta - B\alpha\beta_K) dt
 \end{aligned}
 \tag{B.5}$$

The notation can be simplified by defining the following "Doppler coefficients" :

$$A1_K \equiv \frac{1}{60} \int_{t_K}^{t_{K+1}} V_{DOP} \sin(\alpha + \beta) dt$$

$$A2_K \equiv \frac{1}{60} \int V_{DOP} \cos(\alpha + \beta) dt$$

$$A3_K \equiv \frac{1}{60} \int \sin(\alpha + \beta) dt$$

$$A4_K \equiv \frac{1}{60} \int \cos(\alpha + \beta) dt$$

In addition, define the following integrated random noise processes :

$$U_{\alpha\beta_k}^* \equiv \int_{t_k}^{t_{k+1}} U_{\alpha\beta} dt$$

$$U_{V_k}^* \equiv \int U_V dt$$

Using this notation, Equation B.5 is put into its final form:

$$\begin{aligned} P_{LAT_{k+1}} = P_{LAT_k} &+ (A1 - A3 \cdot BV_k) \sin(B\alpha\beta_k) + (A2 - A4 \cdot BV_k) \cos(B\alpha\beta_k) \\ &+ A5 \cdot U_{\alpha\beta_k}^* + A6 \cdot U_{V_k}^* \end{aligned} \quad (B.6)$$

where

$$A5 \equiv \frac{1}{60} (V_{DOP} - BV_k) \sin(\alpha + \beta - B\alpha\beta_k) \Big|_{t=t_k+5}$$

$$A6 \equiv \frac{-1}{60} \cos(\alpha + \beta - B\alpha\beta_k) \Big|_{t=t_k+5}$$

Note that the average values for V_{DOP} , α , and β over the integration interval have been assumed in the calculations of $A5$ and $A6$. This is a simplification used for the noise terms only.

In a similar manner, the equation for $P_{LONG_{k+1}}$ can also be formulated :

$$\begin{aligned} P_{LONG_{k+1}} = P_{LONG_k} &- \frac{[(A1_k - A3_k \cdot BV_k) \cos(B\alpha\beta_k) - (A2_k - A4_k \cdot BV_k) \sin(B\alpha\beta_k)]}{\cos(P_{LAT_k} \cdot \pi/180)} \\ &+ A7_k \cdot U_{\alpha\beta_k}^* + A8_k \cdot U_{V_k}^* \end{aligned} \quad (B.7)$$

where

$$A7_k \equiv \frac{(V_{DOP} - BV_k) \cos(\alpha + \beta - B\alpha\beta_k)}{60 \cdot \cos(P_{LAT_k} \cdot \pi/180)} \Big|_{t=t_k+5}$$

$$A8_k \equiv \frac{\sin(\alpha + \beta - B\alpha\beta_k)}{60 \cdot \cos(P_{LAT_k} \cdot \pi/180)} \Big|_{t=t_k+5}$$

Note that the conversion from NM to degrees that is actually used in the computer routines is:

$$\text{degrees} = \text{NM} \cdot \frac{6077}{60 \cdot (6077 - 31 \cos(2P_{\text{LAT}}))}$$

This more exact conversion factor would replace the 1/60 factor found in Equations B1 → B7 .

FIGURE B.1 NOMENCLATURE FOR PLANT NOISE AND UPDATE EQUATIONS

APPENDIX C

STATISTICS OF AN INTEGRATED DISCRETE RANDOM NOISE PROCESS

In this study, the statistics of the output of an integrator are analyzed. The input to the integrator is a white noise random sequence whose statistics are known. More specifically, the input represents the random part of the Doppler velocity or heading signal. This random signal is Normally distributed with parameters indicated below. The output of the integrator represents a sort of position error - the integrated velocity noise is a position error with units of NM. The integration interval is ten seconds, and the Doppler noise appears at the input of the integrator at a rate of 2 Hz. Integration is performed via the common trapezoidal method.

C.1 Notation

X	the original random sequence, with elements $x_1, x_2, x_3, \dots, x_N$
x_i	a sample element of X
X_k	the k^{th} 10-second 'batch' of data, $k = 1, \dots, L$
L	total number of 10-second batches of data
N	total number of samples in X (here $N = 4001$)
x_{ki}	the i^{th} element of the k^{th} batch of data
$\bar{x}_k$	the mean value of the k^{th} batch of data
$\sigma^2_{x_k}$	the variance of the k^{th} 10-second batch
Y	the output of the integrator

Other symbols will be defined in the course of the development.

C.2 Definition of the Integration Process

In a continuous time case, the output of an integrator with zero initial conditions is defined as

$$Y = \int_0^T x(t) dt$$

In the discrete case, the output of a digital, trapezoidal integrator is

$$Y = \frac{1}{2} \sum_{i=1}^{n-1} (x_i + x_{i+1}) \Delta t$$

where n is the number of samples used per integration calculation,

Δt is the time interval between data samples.

In our case, $n=21$ and $\Delta t=0.5$ seconds for a 10-second integration process. Hence,

$$Y = 0.25 \sum_{i=1}^{20} (x_i + x_{i+1})$$

In consideration of the large number of 10-second integrations to be performed on the data, let us define the output of the integrator at the end of each 10-second interval by

$$Y_k = 0.25 \sum_{i=1}^{20} (x_{ki} + x_{k,i+1}) \quad (C.1)$$

C.3 Statistics of the Integrated Noise Process

We can rewrite Equation C.1 as follows:

$$Y_k = 0.25 \sum_{i=1}^{20} x_{ki} + 0.25 \sum_{j=2}^{21} x_{k,j} \quad (C.2)$$

Now, the variance of the integrated process is to be determined, i.e. what is σ_Y^2 ? In this case we shall determine S_Y^2 , where

$$S_Y^2 = \frac{1}{L-1} \sum_{k=1}^L (y_k - \bar{y})^2$$

where y_k is the value of the k^{th} output from the integrator

$\bar{y}$ is the mean of the entire integrated process, i.e. the mean of the L y_k values

The mean of the integrated process can be assumed to equal zero, since the mean of the input is zero and the integration process is a linear one. Hence,

$$s_Y^2 = \frac{1}{L-1} \sum_{k=1}^L (y_k)^2 \quad (C.3)$$

but,

$$y_k = 0.25 \sum_{i=1}^{20} x_{ki} + 0.25 \sum_{j=2}^{21} x_{kj}$$

which gives

$$s_Y^2 = \frac{1}{L-1} \sum_{k=1}^L \left(0.25 \sum_{i=1}^{20} x_{ki} + 0.25 \sum_{j=2}^{21} x_{kj} \right)^2$$

Taking the constant factor out of the sum results in

$$s_Y^2 = \frac{0.25^2}{L-1} \sum_{k=1}^L \left(\sum_{i=1}^{20} x_{ki} + \sum_{j=2}^{21} x_{kj} \right)^2 \quad (C.4)$$

We now define

$$\bar{x}_{ki} = \frac{1}{20} \sum_{i=1}^{20} x_{ki}$$

$$\text{and } \bar{x}_{kj} = \frac{1}{20} \sum_{j=2}^{21} x_{kj}$$

that is, $\bar{x}_{ki}$ is the mean of the first 20 sample elements in the k^{th} batch, and $\bar{x}_{kj}$ is the mean of the last 20 sample elements of the same k^{th} batch.

Equation C.4 can be written as

$$\begin{aligned}
 S_Y^2 &= \frac{0.25^2}{L-1} \sum_{k=1}^L \left(\frac{20}{20} \sum_{i=1}^{20} x_{ki} + \frac{20}{20} \sum_{j=2}^{21} x_{kj} \right)^2 \\
 &= \frac{0.25^2}{L-1} 20^2 \sum_{k=1}^L \left(\frac{1}{20} \sum_{i=1}^{20} x_{ki} + \frac{1}{20} \sum_{j=2}^{21} x_{kj} \right)^2 \\
 &= \frac{25}{L-1} \sum_{k=1}^L (\bar{x}_{ki} + \bar{x}_{kj})^2 \quad (C.5)
 \end{aligned}$$

At this point, we make an assumption about the mean values in Equation C.5. $\bar{x}_{ki}$ and $\bar{x}_{kj}$ are calculated from data within the same k^{th} 21-sample batch of random elements. It is probably safe to assume that these two means are practically equal when the variance of the raw random noise process is small. Furthermore, the mean of a 20-sample sequence should be almost equal to the mean of a 21-sample sequence drawn from the same random noise process. Hence, in Equation C.5, let

$$\begin{aligned}
 \bar{x}_{ki} + \bar{x}_{kj} &\approx 2\bar{\bar{x}}_k, \text{ where } \bar{\bar{x}}_k = \frac{1}{21} \sum_{i=1}^{21} x_{ki} \\
 \therefore S_Y^2 &\approx \frac{100}{L-1} \sum_{k=1}^L (\bar{\bar{x}}_k)^2 \\
 &\approx 100 \frac{1}{L-1} \sum_{k=1}^L (\bar{\bar{x}}_k)^2 \quad (C.6)
 \end{aligned}$$

In Equation C.6, $\frac{1}{L-1} \sum_{k=1}^L (\bar{\bar{x}}_k)^2$ is the variance of the L means, $\bar{\bar{x}}_1, \bar{\bar{x}}_2, \dots, \bar{\bar{x}}_L$. The average of these means is zero. Hence,

$$S_Y^2 \approx 100 \sigma_{\bar{\bar{x}}}^2$$

Now, $\sigma_{\bar{\bar{x}}}^2$ is the variance of the L means. From elementary sampling theory (see [67], for example) for a population with N elements, the variance of an n -element sample of the population is given by

$$\sigma_{\bar{x}}^2 = \frac{N-n}{N-1} \frac{\sigma_x^2}{n}$$

where σ_x^2 is the variance of the entire N-element population.

In our particular case, $\bar{x}_k$ is the mean of basically an n-element sample of the original N-element collection of random noise. Hence,

$$\sigma_{\bar{x}}^2 = \frac{N-n}{N-1} \frac{\sigma_x^2}{n} = \frac{4001-21}{4001-1} \frac{\sigma_x^2}{21} = \frac{0.995}{21} \sigma_x^2 = 0.047 \sigma_x^2 \quad (C.9)$$

From Equation C.7, it follows that the variance of the integrated noise process is

$$S_Y^2 \approx 100(0.047 \sigma_x^2) = 4.7 \sigma_x^2 \quad (C.10)$$

In the general case, it can be shown that

$$S_Y^2 \approx \left(\frac{\Delta t}{2}\right)^2 (n-1)^2 (4) \left(\frac{N-n}{N-1}\right) \frac{\sigma_x^2}{n}$$

For very large N, $\frac{N-n}{N-1} \rightarrow 1$, which gives

$$S_Y^2 \approx \left(\frac{\Delta t}{2}\right)^2 (n-1)^2 (4) \frac{\sigma_x^2}{n} \quad (n \neq 1) \quad (C.11)$$

For example, consider the 20-second filter update case. Here $n=41, \Delta t=0.5$, and thus,

$$S_Y^2 = \left(\frac{0.5}{2}\right)^2 (40)^2 (4) \frac{\sigma_x^2}{41} = 9.76 \sigma_x^2$$

APPENDIX D

ON-LINE VLF ANOMALY DETECTION ALGORITHM

The algorithm presented here is used to detect VLF anomalies such as bias drifting, bias jumps, or noise level increases. Twelve innovations samples for each VLF station are used as data points, for a linear least squares curve fit. The end points of the curve are estimated and compared with the theoretical maximum values of the end points that would appear if there were no contingency - with allowances made for the statistical errors that are likely to occur due to the small sample size. If either of the two end points falls outside of its respective tolerance band a bias shift is suspected. The rate of the bias shift is approximated by the slope of the fitted linear curve.

The estimated linear curve (for y a function of x) is given by [56] :

$$\hat{y} = b_0 + b_1(x - x_0) \quad (D.1)$$

where x_0 is the value of x at the start of the particular data sequence.

b_0 and b_1 are the regression coefficients, given by

$$b_0 = \bar{y} - b_1 \bar{x}$$

$$b_1 = S_{xy} / S_{xx}$$

$$\text{where } S_{xy} = N \cdot \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

$$S_{xx} = N \cdot \sum_{i=1}^N (x_i - \bar{x})^2$$

$\bar{x}$ is the mean of the x_i

$\bar{y}$ is the mean of the y_i

The sample variance of the data points is given by

$$S_e^2 = 1/(N-2) \cdot \sum_{i=1}^N [y_i - (b_0 + b_1 x_i)]^2,$$

or, in a more convenient form,

$$S_e^2 = \frac{S_{xx} S_{yy} - S_{xy}^2}{N(N-2) S_{xx}} \quad (D.2)$$

where S_{xx} and S_{xy} are defined above, and

$$S_{yy} = N \sum_{i=1}^N (y_i - \bar{y})^2$$

The so-called standard error of the estimate is the square root of the sample variance,

$$S_e = \left[\frac{S_{xx} S_{yy} - S_{xy}^2}{N(N-2) S_{xx}} \right]^{1/2} \quad (D.3)$$

Now, a $(1-\alpha)$ confidence interval can be defined for any discrete point on the estimated linear curve. The confidence interval for y is given by

$$y = \bar{y} \pm S_e \cdot t_{\frac{\alpha}{2}} \cdot \left[\frac{1}{N} + \frac{(x_0 - \bar{x})^2}{(S_{xx}/N)} \right]^{1/2}$$

where $t_{\frac{\alpha}{2}}$ is the Student-t distribution for $(N-2)$ degrees of freedom.

The foregoing theory applies to any quantities x and y . In the context of the anomaly detection problem the two quantities under consideration are

time (x) and the innovations sequence (y). Using 12 innovations samples for each VLF station, the following algorithm can be used to detect a drifting or jumping VLF bias.

- (i) calculate S_{xx} , S_{xy} , and S_{yy} ; $\bar{x}$ and $\bar{y}$
- (ii) find the standard error of the estimate, S_e , from Equation D.3
- (iii) calculate b_0 and b_1 , the regression coefficients
- (iv) estimate the linear curve end points:

$$\hat{y}_1 = b_0$$

$$\hat{y}_{12} = b_0 + b_1(x_{12} - x_0)$$

- (v) determine the confidence interval for the second end point of the fitted curve:

$$y_{12} = \hat{y}_{12} \pm S_e \cdot t_{\frac{\alpha}{2}} \cdot \left[\frac{1}{12} + \frac{(x_{12} - \bar{x})^2}{(S_{xx}/12)} \right]^{1/2} \quad (D.5)$$

- (vi) If there is no contingency occurring, then y_{12} should equal zero because the VLF noise is assumed to have a zero mean. If the range of y_{12} defined in Equation D.5 goes through zero, then no bias drift is suspected, even though y_{12} may not exactly equal zero. If the confidence interval for y_{12} is such that y_{12} cannot possibly be zero, a bias drift or jump is suspected.

- (vii) some values of the Student-t distribution are:

$$90\% \text{ CI} \quad t_{0.05} = 1.812$$

$$95\% \text{ CI} \quad t_{0.025} = 2.228$$

$$99\% \text{ CI} \quad t_{0.005} = 3.169$$

A typical numerical example taken from a simulation run will show how the algorithm is used. The filter update interval in this case is 20 seconds.

From the innovations sequence for the first VLF station used in the filter simulation run, the following results were obtained:

$$S_e = 0.848$$

$$\bar{x} = (1180+960)/2 = 1070 \text{ secs}$$

$$b_0 = 0.036$$

$$b_1 = 0.007$$

$$\hat{y}_1 = 0.036$$

$$\hat{y}_{12} = 0.036 + 0.007(220) \approx 1.576$$

$$N = 12$$

$$n = N-2 = 10; \quad t_{\frac{\alpha}{2}} = 2.228 \text{ for } 95\% \text{ CI}$$

Hence,

$$y_{12} = \hat{y}_{12} \pm (0.848) \cdot (2.228) \cdot \left[\frac{1}{12} + \frac{(1180-1070)^2}{(686399/12)} \right]^{\frac{1}{2}}$$

$$= 1.576 \pm 1.03$$

resulting in an estimate for y_{12} of $0.546 < y_{12} < 2.61$.

Since $y_{12} > 0$, a bias drift is suspected, with the estimated drift rate being $0.007 \mu\text{SEC}/\text{Sec}$. In the simulation run, a drift rate of $0.005 \mu\text{SEC}/\text{Sec}$ had actually been programmed to occur during the period 1000-2000 seconds. The true VLF bias at 1180 seconds was $1.0 \mu\text{SEC}$ (a drift of $0.9 \mu\text{SEC}$ plus an initial bias of $0.1 \mu\text{SEC}$) which falls nicely inside the tolerance band of $1.576 \pm 1.03 \mu\text{SEC}$.

If a bias drift has been detected, the corresponding P matrix element is increased in value, according to the following equation:

$$P_{ii} = \text{PCOEFF} \times (|y_{12} - \text{VTOL}|)^2 \quad (\text{D.6})$$

where PCOEFF is a scale factor specified by the user

$y_{12} - \text{VTOL}$ is the difference between the actual estimated end point of the linear curve and the theoretical maximum end point value which would appear if there was no anomaly

This is somewhat of an empirical equation but has been shown to be quite effective in setting the $\underline{P}$ matrix element. Additional documentation on the on-line anomaly detection algorithm may be found in Appendix E.

APPENDIX E

DOCUMENTATION FOR KALMAN FILTER SIMULATION PACKAGE

KFSIM is an interactive FORTRAN program which simulates the real time operation of a KALMAN FILTER designed for use in an integrated navigation system employing Doppler radar, magnetic compass and VLF phase data. The program is very flexible in that it allows numerous options in the simulation of the navigation sensor data and in the running of the Kalman filter itself. The documentation which follows attempts to provide enough information to allow a new user to run and understand the program.

E.1 STRUCTURE OF THE PROGRAM

The program has several main sections, indicated as follows

- (i) INITIALIZATION - parameter retrieval from pre-stored input data sets
 - user options (from keyboard)
 - unit conversion
- (ii) Flight trajectory and Navigation Sensor Simulation
 - flight path simulation
 - Doppler data
 - compass data
 - VLF phase data
- (iii) Adaptive parameter processing (if requested by user)
- (iv) Kalman filter processing
- (v) Output of state estimates and errors
 - data storage
 - graphical plotting

E.2 REQUIRED DATA SETS

(i) Input Data Sets

logical unit	data set contents
66	VLF and general filter parameters.
60	VLF station information
64	Doppler/heading parameters

(ii) Output Data Sets

99	state estimates and errors
91	P matrix diagonal elements
94	Innovations/true VLF noise
dsrn	filter matrices
97	LAT/LONG position errors
40	navigation sensor data for future mag tape use.

NOTES: (1) The data sets indicated above must be defined prior to running KFSIM, otherwise the local terminal will be used as the input or output device.

(2) Logical units 90 and 68 are reserved for local terminal output and input, respectively.

(3) output data sets 91, 94, 97, 40, and DSRN are user optional.
Set DSRN = 0 for suppression of filter matrix printing.

E.3 TYPICAL INPUT DATA SETS

The following are typical data sets to be used for the filter input data.

(a) LU 66 VLF and general filter parameters

```

0045 00 00.00      LAT
0075 00 00.00      LONG
0.001 0.001        ERRLT,ERRLG
6
OGMADW
0.0001      EST VARIANCE OF INITIAL STATE EST --> P MATRIX
0.0001
000.01
000.01
000.01
000.01
000.01
000.01
1.0      TRUE VARIANCE OF PHASE --> VARP
1.0
1.0
1.0
1.0
1.0
0001.000    EST. VARIANCE OF PHASE= EVARP= R MATRIX
0001.000    (RESET IN KFSIM IF DESIRED)
0001.000
0001.000
0001.000
0001.000
00.100 00.100 00.100 00.100 00.100 00.100    TRUE PHASE BIAS --> BP
00.000 00.000 00.000 00.000 00.000 00.000    EST. PHASE BIAS --> X(5-10)
125379 963253 785199 326521 152635 662737    IY'S FOR RANDOM NUMBER GEN.
00      DSRN
5000    FLIGHT TIME --> NT
010     PRINTOUT TIMES --> NPRINT (RESET IF DESIRED)
543123* 667799 583427 333333 111235 775001

```

(B) LU 64 DOPPLER/HEADING PARAMETERS

26116.25 26116.25 23885.00
 00.000 0.000
 045.000 00.500 0.250 0.000
 136591 965777 665213 876123
 018.618 008.840 00.000
 0000.000 00.000 00.000
 0015.600 079.980 00.000
 0005.000 01.250 01.250
 10.0
 10.0

TFA,TFB,TFC
 PITCH,ROLL
 BETAT,BBETA,BETAV,EBB
 IY1,IY2,IY3,IY4
 BFA,BFB,BFC
 EBFA,EBFB,EBFC
 EVARV,EVARA,EVARB

(c) LU 60 VLF station information

0
 67 04 00.0 -14 04 00.0 1.0000 60.9756
 G
 52 22 10.0 1 11 55.0 1.0000 62.5000
 H
 44 38 54.0 67 16 54.0 0.9983 56.1798
 A
 38 59 06.0 76 27 12.0 0.9975 46.7290
 P
 9 03 18.0 79 38 54.0 0.9971 41.6667
 D
 46 21 57.2 98 20 08.8 1.0007 76.3350
 W
 48 12 12.0 -121 55 00.0 0.9981 53.7634
 L
 21 25 30.0 158 09 18.0 0.9972 42.7350
 J
 34 58 15.0 -137 01 19.0 0.9986 57.4713
 S
 -21 49 00.0 -114 09 48.0 0.9974 44.8430
 R
 55 00 00.0 03 30 00.0 1.0000 52.63158

E.4 TYPICAL PROGRAM RUN

The following is typical of a KFSIM run

```

runk fvs t224
UNLOAD: CZCDU001 CANCELLED: ARGUMENT SYMBOL FOR DELETE KFSIM NOT FOUND
RUN ID?
AAAA
:t224
  THIS RUN ID IS T224
  DO YOU WANT EIGENVALUES? 1=YES-
:0
  DO YOU WANT FLIGHT DATA OUT TO DSRN 40?
:0
  DO YOU WANT P MATRIX DIAGONALS O/P ?
:0
  DO YOU WANT POSITION ERRORS O/P ?
:0
  DO YOU WANT PLANT BIAS DRIFT?
:1
  ENTER VELOCITY BIAS DRIFT DATA
  START END KTS
  IIII IIII FFF.FF
:1000 2000 10.0
  ENTER HEADING BIAS DRIFT DATA
  IIII IIII FFF.FF
:1000 2000 5.0
  DO YOU WANT P MATRIX RESET FOR PLANT BIAS DRIFT?
:0
  DO YOU WANT VLF BIAS DRIFT? 1=YES
:1
  WHAT VLF BIASES WILL DRIFT? ENTER A '1'
  OGMADW
:1
  HOW MANY BIAS CHANGES PER STATION? MAX OF 4
:2
  ENTER BIAS DRIFT DATA
  START END USEC CHANGE
  IIII IIII FFF.FF (VLFBS(2)=9999, FOR DEFAULT)
  0 1
:1000 2000 -15.0
  0 2
:2000 3000 7.5
  DO YOU WANT SMART FILTER?
:0
  INPUT COEFF FOR INITIAL R MATRIX FIX.....FFFF.FF
:1.00
  VLF NOISE CHANGE? 1=YES
:1
  WHAT VLF PHASE VARIANCES WILL CHANGE?
  OGMADW
:1
  ENTER VARIANCE DATA
  TIME NEW VARIANCE
  IIII FFFFFFFF
  M
:13000 100.0
  WILL R MATRIX BE RE-SET?
:0
  DO YOU WANT AUTO. VLF NOISE/BIAS CHECK?
:1
  INPUT COEFFICIENTS FOR P MATRIX AND TOLERANCE
  FFF.F FF.FF
:250.0 3.4

```

```

DO YOU WANT CIRCULAR TRACK? 1=YES
:0
INPUT VARIANCE FOR RANDOM VLF BIAS FLUCTUATIONS... F6.3
:0
INPUT VALUES FOR Q MATRIX (VLF ELEMENTS)... F6.3
D
:0.001
G
:0.00
M
:0.001
A
:0.001
D
:0.00
W
:0.00
INPUT Q(3,3),Q(4,4)...F5.3
:0.01
:0.01
DO YOU WANT PERIODIC VLF BIAS NOISE? 1=YES
:0
DO YOU WANT GAIN MATRICES PRINTED LOCALLY?
:0
INPUT EXP. DATA WEIGHTING COEFFICIENT... F5.3
:1.000
KFSIM: CHCRW410 PROCEEDING: PAUSE OPTION TO CHANGE ANY PARAMETERS, USE >>QUALIFY<<, >
>SET<<, AND >>GO<<
qualify kfsim
display nt
NT=5000
set nt=4000
display bp
BP =
(1) +.99999964E-01 +.99999964E-01 +.99999964E-01 +.99999964E-01 +.99999964E-01 +.9
9999964E-01
go
GO: CZAFB030 PROCEEDING(PCS): EXECUTION CONTINUES AT CHCRWC.(X'4AB4')
TIME INCREMENT? MULTIPLE OF 10 SECS, I2
:20
CHANGE NPRINT TO MATCH NINC, I2
:20

```

E.5 TYPICAL OUTPUT DATA (LU 99)

2 An example of the data that is written to logical unit 99 appears on the following page. The input parameters are echoed and the Kalman filter estimates and errors follow. Note that only the first five state variables appear in this example of the filter output.

 KALMAN FILTER SIMULATION
 DEC. 07 1981 16.29.30

KFRUN T 5

	SIMULATED TRUE VALUE	VARIANCE TRUE EST. (WEIGHT)	BIAS TRUE EST.	BIAS VARIANCE (WEIGHT)
INIT LATITUDE	45 0 0.00		0.0010	0.0001
LONGITUDE	75 0 0.00		0.0010	0.0001
0 DOPPLER BEAM-FREQS	26116.25	1322.65	181.54	0.00
	26116.25	742.64	80.42	0.00
	23885.00	0.00	0.00	0.00
0 GROUND SPEED	250.00	10.000 100.000	10.00	0.00 100.000
HEADING WRT TRUE N	45.00			
ALPHA (DRIFT ANG)		5.000 50.000	5.00	0.00 100.000
BETA (TAS HDG)		5.000 50.000	5.00	0.00

VLF STATION PHASES	VARIANCE TRUE EST. (WEIGHT)	USEC BIAS TRUE EST.	BIAS VARIANCE (WEIGHT)	SIMULATED BIAS DRIFT START END DRIFT SLOPE
1 D	10.000 20.000	10.000 0.000	200.000	
2 G	10.000 20.000	10.000 0.000	200.000	
3 H	10.000 20.000	10.000 0.000	200.000	
4 A	10.000 20.000	10.000 0.000	200.000	
5 D	10.000 20.000	10.000 0.000	200.000	
6 W	10.000 20.000	10.000 0.000	200.000	

0 INITIAL STATE VECTOR
 45.0010 75.0010 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

0 INITIAL P MATRIX DIAGONALS
 0.0001 0.0001 100.0000 100.0000 200.0000 200.0000 200.0000 200.0000 200.0000

0 INITIAL Q MATRIX DIAGONALS
 100.0000 100.0000

0 INITIAL R MATRIX DIAGONALS
 20.0000 20.0000 20.0000 20.0000 20.0000 20.0000

1
 0 0 LAT: 45.00000 LONG: 75.00000
 45.00099 75.00099 0.00000 0.00000 0.00000
 -0.05951 -0.04208 9.99985 9.99997 10.00000
 0 20 LAT: 45.01636 LONG: 74.97684
 45.01291 74.97322 0.11426 -0.68343 10.20440
 0.20691 0.15338 9.88559 10.68340 -0.20440

E.6 PROGRAM VARIABLE NAMES

An alphabetical list of the major variable names used in KFSIM are listed below.

A	NULF X 1 VECTOR : TEMP. STORAGE OF RESIDUALS
ALPHAD	DOPPLER DRIFT ANGLE
ALPHAT	TRUE DRIFT ANGLE
A1-7 AB	COEFFICIENTS USED IN PLANT UPDATE
BALPHA	TRUE BIAS IN DOPPLER DRIFT ANGLE
BBETA	TRUE BIAS IN HEADING ANGLE
BETAD	SIMULATED DOPPLER HEADING ANGLE
BETAT	TRUE HEADING ANGLE
BETAV	TRUE HEADING ANGLE VARIANCE
BFA,BFB,BFC	TRUE BIAS IN SIMULATED DOPPLER BEAMS
BIY	"SEED" FOR VLF RANDOM NOISE
BP	TRUE BIAS IN VLF PHASE
BV	TRUE BIAS IN DOPPLER VELOCITY
CV	VELOCITY CORRECTION FOR SODANO INVERSE PHASES
DAY,MTN,YEAR	TIME STAMP FOR KF RUN ID
DSRN	DATA SET REFERENCE NUMBER: LOGICAL UNIT NUMBER FOR MATRIX O/P
DVBIAS	MAGNITUDE OF VLF BIAS SHIFT
DX	ERRORS IN STATE ESTIMATES
EBA,EBB	INITIAL ESTIMATES OF DRIFT ANGLE AND HEADING ANGLE BIASES
EBFA,EBFB,EBFC	ESTIMATED BIASES IN DOPPLER BEAM FREQUENCIES
ERRLT, ERRLG	INITIAL ERROR IN LAT/LONG
EVARA,EVARB	INITIAL ESTIMATED VARIANCES IN DRIFT AND HEADING ANGLES
EVARAB	= EVARA + EVARB = Q(2,2)
EVARP	INITIAL ESTIMATED VARIANCE IN VLF PHASE (R0)
EVARX	INITIAL STATE ERROR VARIANCES
F	PLANT UPDATE VECTOR
GAMMA	NOISE "TRANSITION" MATRIX
GS	TRUE GROUND SPEED OF AIRCRAFT
H	MATRIX REPRESENTING RELATIONSHIP BETWEEN THE STATE AND MEASUREMENTS
HBET	HEADING BIAS DRIFT END TIME
HBINC	MAGNITUDE OF HEADING ANGLE BIAS DRIFT
HBST	HEADING BIAS DRIFT START TIME
HEADNG	TRUE AIRCRAFT TRACK ANGLE (= ALPHAT + BETAT)
IBIAS	FLAG FOR OPTIONAL VLF BIAS DRIFTING
IBPER	PERIODIC VLF BIAS SHIFTS
ICIRC	CIRCULAR TRACK

IDTA	DATA O/P
IERRPL	POSITION ERROR O/P
IFLAG1, IFLAG2	FLAGS FOR VLF BIAS DRIFTING CONTROL
INOISE	FLAG FOR OPTIONAL VLF NOISE INCREASE
IPBIAS	DOPPLER BIAS DRIFTING
IPOUT	P MATRIX DIAGONAL O/P
IRSMRT	ADJUSTMENT OF R MATRIX
IPSMRT	P
ISMRT	Q
ISTN	FLAG FOR INDIVIDUAL VLF BIAS DRIFT
IVAR	FLAG FOR ON-LINE VARIANCE CALCULATION IVAR = 0 VARIANCE NOT CALC. YET = 1 VARIANCE HAS BEEN CALC.
IVAUTO	FLAG FOR OPTIONAL AUTOMATIC ON-LINE ANOMALY DETECTION
IVLFN	FLAG FOR INDIVIDUAL VLF STN. NOISE INCREASES
IY	'SEEDS' FOR VLF RANDOM NOISE GENERATOR
IY1, IY2, IY3, IY4	'SEEDS' FOR DOPPLER/HEADING RANDOM NOISE
KAG	VLF STATION ACQUISITION FLAGS
KBIAS	DRIFT RATE FOR VLF BIAS DRIFT
KHBIAS	DRIFT RATE FOR HEADING BIAS DRIFT
KVLF, KVLFF	COUNTER FOR RESIDUALS USED IN VLFFIX
KVBIAS	DRIFT RATE FOR DOPPLER VEL. BIAS DRIFT
K1	KALMAN GAIN MATRIX
LAT, LONG	TRUE GEOGRAPHICAL LAT./LONG. OF AIRCRAFT
LATD, LATM, LATS	TRUE LAT. DEG/MIN/SEC
LONGD, LONGM, LONGS	TRUE LONG. IN DEG/MIN/SEC
M1	A PRIORI ERROR COVARIANCE MATRIX
NBPER	PERIOD OF PERIODIC VLF BIAS SHIFTS
NDOP	NUMBER OF DOPPLER SAMPLES PER UPDATE INTERVAL
NINC	FILTER UPDATE INTERVAL
NN	CURRENT FLIGHT TIME COUNTER (SECONDS)
NPRINT	FLAG FOR FILTER O/P TIMES
NT	TOTAL FLIGHT TIME
NVLF	NUMBER OF VLF STATIONS
NX	NUMBER OF STATE VARIABLE (= NVLF + 4)
OBS	SIMULATED VLF PHASE OBSERVATIONS
OMEGA	RADIAL VELOCITY OF AIRCRAFT IN A CIRC. TRACK
P	A POSTERIORI ERROR COV. MATRIX
PCOEFF	COEFFICIENT FOR ADJUSTING P MATRIX IN VLFFIX
PHI	QUASI-LINEAR STATE TRANSITION MATRIX
PI	3.14159
PITCH	AIRCRAFT PITCH ANGLE
PU3, PU4	P(3,3) AND P(4,4) IN KTS**2 AND DEG**2
P33, P44	COEFFICIENTS USED IN ADJUSTING P(3,3) AND P(4,4) AT THE END OF A CIRC. TRACK
Q	PLANT NOISE COV. MATRIX
Q11, Q22	COEFFICIENTS USED IN ADJUSTING Q(1,1) AND Q(2,2) WHILE IN A CIRC. TRACK

R	VLF NOISE COV. MATRIX
RAD	CONVERSION FACTOR : DEGREES->RADIANS
RADIUS	RADIUS OF CIRCULAR TRACK
RCOEFF	FACTOR FOR ADJUSTING R MATRIX FOR VLF NOISE INCREASE
ROLL	AIRCRAFT ROLL ANGLE
RSET1	FACTOR FOR SETTING INITIAL R MATRIX
SMALLH	THEORETICAL VLF PHASE
STBETA	HEADING ANGLE AT START OF CIRC. TRACK
STHDG	TRUE TRACK ANGLE AT START OF CIRC. TRACK
STNID	ONE-LETTER VLF STATION DESIGNATION
TBX,TBY	TRUE BIAS IN VX AND VY
TFA,TFB,TFC	TRUE DOPPLER BEAM FREQUENCIES
TOLCF	ON-LINE DETECTION THRESHOLD COEFF. IN VLFFIX
TURN1,TURN2	TIMES OF STRAIGHT-CIRC. TRACK TRANSITIONS
TVX,TVY	TRUE ALONG TRACK AND ACROSS TRACK VELOCITIES
VARA	TRUE VARIANCE IN DRIFT ANGLE
VARB	TRUE VARIANCE IN HEADING ANGLE
VARFA,VARFB,VARFC	TRUE VARIANCE IN DOPPLER BEAM FREQS.
VARP	TRUE VLF MEASUREMENT NOISE COVARIANCE
VARPN	NEW VLF NOISE VARIANCE
VARV	TRUE VARIANCE IN DOPPLER VELOCITY
VBET	DOPPLER VELOCITY BIAS DRIFT END TIME
VBINC	MAGNITUDE OF DOPPLER VEL. BIAS DRIFT
VBLEV	MAGNITUDE OF PERIODIC VLF BIAS FLUCTUATIONS
VBST	DOPPLER VELOCITY BIAS DRIFT START TIME
VDOP	SIMULATED DOPPLER VELOCITY
VLFBET,VLFBST	VLF BIAS DRIFT END AND START TIMES
VLFN	MEASUREMENT RESIDUAL (USED IN VLFFIX)
VLFT	TIME OF VLF NOISE INCREASE
VX,VY	ALONG TRACK AND ACROSS TRACK VELOCITIES USED IN OPTIONAL FLIGHT DATA O/P. (FOR MAG TAPE DATA SET)
WL	WAVELENGTH OF VLF SIGNAL
X	STATE VECTOR
XU3,XU4	X(3), X(4) IN KNOTS AND DEGREES

6.7 SUBROUTINE DESCRIPTIONS

THE FOLLOWING SUBROUTINES ARE CALLED BY KFSIM.

ASUB	- CALCULATES DOPPLER COEFFICIENTS (A1 -> A8)
DEGRAD	- CONVERTS POSITION IN DEG/MIN/SEC FORMAT TO DECIMAL DEGREES OR RADIANS
DOPIN	- READS AND RETURNS INITIAL DOPPLER/HEADING PARAMETERS
DOP1	- PROCESSES INITIAL DOPPLER/HEADING PARAMETERS
DOP2	- SIMULATES DOPPLER/HEADING DATA
EMULM	- MULTIPLIES TWO MATRICES
FIGUB	- CALCULATES QUASI-LINEAR STATE TRANSITION MATRIX (PHI) AND GAMMA
FSUB11	- CALCULATES PLANT UPDATE VECTOR
GI	- RETURNS THEORETICAL PHASE DISTANCE FROM A GIVEN GEOGRAPHICAL POSITION TO A VLF STATION
HSUB11	- CALCULATES H MATRIX
KISUB1	- CALCULATES KALMAN GAIN MATRIX
MADD	- ADDS TWO MATRICES
MOUTS	- WRITES A MATRIX IN A 'NICE' FORMAT
MISUB1	- RETURNS A PRIORI ERROR COVARIANCE MATRIX
NRAN	- RANDOM NUMBER GENERATOR (IBM LIBRARY SUBROUTINE)
PBIAS	- RETURNS PARAMETERS FOR OPTIONAL DOPPLER/HEADING-BIAS DRIFT
PLOTID	- RETURNS KF RUN SEQUENCE NUMBER FOR ID
PSUB11	- CALCULATES A POSTERIORI ERROR COV. MATRIX
STAT	- CALCULATES MEAN AND STANDARD DEVIATION OF N SAMPLES OF RANDOM SIGNAL. DATA IS IN MATRIX FORM
TRU	- RETURNS TRUE GEOGRAPHICAL POSITION OF AIRCRAFT
VBIAS	- RETURNS PARAMETERS FOR OPTIONAL VLF BIAS DRIFT
VLFFIX	- AUTOMATIC ON-LINE VLF ANOMALY DETECTION SCHEME

VLFSIM - SIMULATES TRUE VLF PHASE DATA
VNOISE - RETURNS PARAMETERS FOR OPTIONAL VLF NOISE INCREASE

E.7.2 OTHER SUBROUTINES

- INVER - SODANO INVERSE ROUTINE. RETURNS GEODESIC DISTANCE
BETWEEN TWO POINTS ON EARTH'S SURFACE
- INVERT - INVERTS A SQUARE MATRIX
- LINREG - PERFORMS LINEAR REGRESSION ON X,Y DATA
- MTPRDA - MULTIPLIES ONE MATRIX BY THE TRANSPOSE OF A
SECOND MATRIX
- VSTAT - SIMILAR TO STAT, DATA IS IN VECTOR FORM

FIGURE E.1 FLOWCHART FOR KFSIM

FIGURE E.1 (con't)

FIGURE E.1 (con't)

FIGURE E.1 (con't)

```

0000100 C      THIS IS THE MAIN OR CONTROL PROGRAM FOR THE KALMAN FILTER
0000200 C      SIMULATION.
0000300 C
0000400 C      CRAIG MASKELL  FLIGHT RESEARCH LABORATORY
0000500 C
0000600 C
0000700 C      THE MATRIX PRINTING ROUTINE 'MOUTS' HAS BEEN ADAPTED
0000800 C      FROM THE UNIVERSITY OF NEW BRUNSWICK LIBRARY.
0000900 C      SOURCE: D.G. MACNEIL,  UNB COMPUTING CENTRE
0001000 C
0001100 C  ** FOR DOCUMENTATION , ISSUE THE COMMAND 'SCRIPT,KALDOC', OR PROCDIF 'TERSCP KALDOC'
0001200 C
0001300 C
0001400 C
0001500      REAL LAT, LONG, VDOP(81)/81*0.0/, LATS, LONGS
0001600      REAL ALPHAD(81)/81*0.0/, BETAD(81)/81*0.0/
0001700      REAL VARX(10), VARP(10), BP(6), EVARX(10), EVARP(6)
0001800      REAL PHI(10,10), P(10,10)/100*0.000/, M1(10,10), GAMMA(10,10)
0001900      REAL K1(10,6), R(6,6), X(10), F(10), OBS(6,1), TDTA(6), TTDTA(6)
0002000      REAL D(10,10)/100*0.000/, H(6,10), MTH, DAY, YEAR
0002100      REAL DX(10), VLFSD(6), RAND(10), VECTOR(1,10)
0002200      REAL DVBIAS(6,4), KBIAS(6,4), VBINC, KVBINC, HBINC, KHBINC
0002300      REAL RR(6,6), TURN1/0.0/, TURN2/0.0/, SMALLH(6,1), VARP(6)
0002400      REAL A(6,1), B(10,1), VLFN(30,6), VMEAN(6), VSTD(6), MTEMP(5), STEMP(6), DMEAN(6)
0002500      REAL INNOV(6,1), VX(81), VY(81), BBB(81), OBS1(6,1), OBS2(6,1), RAND(6), VBLEV(6)
0002600 C
0002700 C
0002800 C
0002900      REAL*8 T
0003000 C
0003100      INTEGER DEGS, MINS, LATD, LATM, LONGD, LONGM, IY(6), BIY(6)
0003200      INTEGER STNID(6), ISRN, ISTN(6), VLFBS(6,4), VLFBET(6,4)
0003300      INTEGER IVLFN(6), VLFVT(6), IVAR(6), KAQ(6), KVLF(6)
0003400      INTEGER VBST, VBET, NBPER(6)
0003500 C
0003600      INTEGER IFLAG1(6)/6*0/, IFLAG2(6)/6*0/, IPFLAG
0003700 C
0003800 C
0003900      COMMON /COM1/ A1, A2, A3, A4
0004000      COMMON /COM2/ VARV, VARA, BETAV, GS, HEADNG
0004100      COMMON /COM3/ X
0004200      COMMON /COM4/ NX, NVLF
0004300      COMMON /COM5/ R, RR, TDTA, TTDTA
0004400      COMMON /COM6/ TFA, TFB, TFC, BFA, BFB, BFC, EBFA, EBFB, EBFC,
0004500      VARFA, VARFB, VARFC, STDA, STDB, STDC, TVX, TVY, TBX, TBY
0004600 C
0004700 C
0004800 C*****
0004900 C*****
0005000 C
0005100 C
0005200 C      *** PROGRAM RUN IDENTIFICATION NUMBER IS 'ID' ***
0005300      CALL PLOTID(ID)
0005400 C
0005500      WRITE(99,9010)
0005600      WRITE(99,900)
0005700 900  FORMAT(' ', 10X, 'KALMAN FILTER SIMULATION')
0005800      CALL DATE(MTH, DAY, YEAR)
0005900      CALL TIME(T)

```



```

0006000      WRITE(99,901) MTH, DAY, YEAR, T , ID
0006100 901  FORMAT(' ', 10X, 3A4, 4X, A8, 5X, 'KFRUN', 2X, A4)
0006200 C
0006300      WRITE(99,9010)
0006400 9010 FORMAT('0', 10X, '*****')
0006500 C
0006600      PI=3.14159
0006700      RAD=PI/180.
0006800 C
0006900 C
0007000 C      *****
0007100 C      *
0007200 C      * READ INITIAL DATA.
0007300 C      * USER INPUTS DATA IN USER ORIENTED
0007400 C      * QUANTITIES, I.E. KNOTS, DEGREES, ETC.
0007500 C      *
0007600 C      *****
0007700 C
0007800 C
0007900      READ(66,600) LATD, LATH, LATS
0008000 600  FORMAT(I4, 1X, I2, 1X, F5.2)
0008100      READ(66,600) LONGD, LONGH, LONGS
0008200      READ(66,6010) ERRLT, ERRLB
0008300 6010  FORMAT(F5.3, 1X, F5.3)
0008400 C
0008500      CALL DOPIN(PITCH, ROLL, BETAT, BBETA, EBB, BETAV, EVARV, EVARA, EVARB
0008600      &      , VARA, IY1, IY2, IY3, IY4, EVARX(3), EVARX(4))
0008700 C
0008800      CALL DOP1(STDD, BETAV, GS, ALPHAT, BETAT, HEADNG, BV, BALPHA, X(3), EBA, ROLL, PITCH,
0008900      &      , VARV, VARA)
0009000 C
0009100      X(4)= EBA+EBB
0009200      EVARAB= EVARA+EVARB
0009300 C
0009400 C
0009500 601  FORMAT(F7.3)
0009600 602  FORMAT(F4.1, 2X, F5.1)
0009700 603  FORMAT(F6.3, 5(2X, F6.3))
0009800 C
0009900      READ(66,6035) NVLF
0010000 6035  FORMAT(I1)
0010100      READ(66,607) (STNID(I), I=1, NVLF)
0010200 607  FORMAT(6A1)
0010300      NX=NVLF+4
0010400 C
0010500 C
0010600      READ(66,6016) EVARX(1)
0010700      READ(66,6016) EVARX(2)
0010800 C
0010900      DO 11 I=5, NX
0011000      READ(66,6016) EVARX(I)
0011100 6016  FORMAT (F9.5)
0011200 11  CONTINUE
0011300 C
0011400 C      * VARIANCES OF THE PHASE MEASUREMENTS *
0011500 C
0011600 6015  FORMAT(F8.3)
0011700 C
0011800 C
0011900      DO 20 I=1, NVLF

```

```

0012000      READ(66,6015) VARP(I)
0012100 20    CONTINUE
0012200 C
0012300 C
0012400      DO 21 I=1,NVLF
0012500      READ(66,6015) EVARP(I)
0012600 21    CONTINUE
0012700 C
0012800 605    FORMAT(F6.3,2(2X,F6.3))
0012900 C
0013000      READ(66,603) (BF(I),I=1,NVLF)
0013100      READ(66,603) (X(I+4),I=1,NVLF)
0013200 C
0013300      READ(66,606) (IY(I),I=1,NVLF)
0013400 606    FORMAT(I6,5(2X,I6))
0013500      READ(66,6070) DSRN
0013600 6070   FORMAT(I2)
0013700      READ(66,608) NT
0013800 608    FORMAT(I4)
0013900      READ(66,6071) NPRINT
0014000 6071   FORMAT(I3)
0014100 C
0014200 C      *** OPTION FOR EIGENVALUE COMPUTATION ***
0014300      WRITE(90,904)
0014400 904    FORMAT(' ',2X,'DO YOU WANT EIGENVALUES? 1=YES')
0014500      READ(68,6035) IEIG
0014600 C
0014700 C      ** OPTION TO OUTPUT FLIGHT DATA **
0014800 C
0014900      WRITE(90,968)
0015000 968    FORMAT(' ',2X,'DO YOU WANT FLIGHT DATA OUT TO DSRN 40?')
0015100      READ(68,6035) IQTA
0015200 C
0015300      WRITE(90,9680)
0015400 9680   FORMAT(' ',2X,'DO YOU WANT P MATRIX DIAGONALS O/P?')
0015500      READ(68,6035) IPOUT
0015600 C
0015700      WRITE(90,9681)
0015800 9681   FORMAT(' ',2X,'DO YOU WANT POSITION ERRORS O/P?')
0015900      READ(68,6035) IERRPL
0016000 C
0016100 C      *****
0016200 C      * OPTION FOR PLANT BIAS DRIFT *
0016300 C      *****
0016400 C
0016500      WRITE(90,925)
0016600 925    FORMAT(' ', DO YOU WANT PLANT BIAS DRIFT?')
0016700      READ(68,6072) IPBIAS
0016800      IF(IPBIAS.EQ.1) CALL PBIAS(VBST,VBET,VBINC,KVBIAS,HBST,HBET,HBINC,KHBIAS,IPSHRT)
0016900 C
0017000 C
0017100 C      *****
0017200 C      * OPTION FOR VLF BIAS DRIFTING *
0017300 C      *****
0017400 C
0017500      WRITE(90,910)
0017600 910    FORMAT(' ', DO YOU WANT VLF BIAS DRIFT? 1=YES')
0017700      READ(68,6072) IBIAS
0017800 6072   FORMAT(I1)
0017900 C

```

```

0018000      IF(IBIAS.EQ.1) CALL VBIAS(STNID,ISTN,VLFBS,VLFBE,DVBIAS,KBIAS,NVLF,IFLAG2)
0018100 C
0018200 C
0018300 C
0018400 C      *****
0018500 C      * SET R MATRIX COEFF FOR *
0018600 C      * INITIAL NOISE LEVEL *
0018700 C      *****
0018800 C
0018900      WRITE(90,920)
0019000 920    FORMAT(' ',2X,'INPUT COEFF FOR INITIAL R MATRIX FIX.....FFFF.FF')
0019100      READ(68,620) RSET1
0019200 620    FORMAT(F7.2)
0019300 C
0019400 C      *****
0019500 C      * OPTION FOR VLF NOISE CHANGE *
0019600 C      *****
0019700 C
0019800      WRITE(90,911)
0019900 911    FORMAT(' ',2X,'VLF NOISE CHANGE?,1=YES')
0020000      READ(68,6072) INOISE
0020100      IF(INOISE.EQ.0) GO TO 27
0020200      CALL VNOISE(STNID,NVLF,IVLFN,VLFVT,VARPN,ID)
0020300      WRITE(90,912)
0020400 912    FORMAT(' ',2X,'WILL R MATRIX BE RE-SET?')
0020500      READ(68,6072) IRSMRT
0020600 C
0020700      IF(IRSMRT.EQ.0) GO TO 27
0020800      WRITE(90,913)
0020900 913    FORMAT(' ',2X,'INPUT COEFF FOR R MATRIX RESET.....FFFF.FF')
0021000      READ(68,613) RCOEFF
0021100 613    FORMAT(F7.2)
0021200 C
0021300 27     CONTINUE
0021400 C
0021500 C      *****
0021600 C      * OPTION FOR AUTO. ONLINE ANALY *
0021700 C      * DETECTION ALGORITHM *
0021800 C      *****
0021900 C
0022000 C
0022100      WRITE(90,924)
0022200 924    FORMAT(' ',2X,'DO YOU WANT AUTO. VLF NOISE/BIAS CHECK?')
0022300      READ(68,6072) IVAUTO
0022400      IF(IVAUTO.EQ.1) WRITE(90,990)
0022500 990    FORMAT(' ',2X,'INPUT COEFFICIENTS FOR P MATRIX AND TOLERANCE',/,3X,'FFF.F FF.FF')
0022600      IF(IVAUTO.EQ.1) READ(68,690) PCOEFF,TOLCF
0022700 690    FORMAT(F5.1,2X,F5.2)
0022800 C
0022900 C      DEFINE R MATRIX
0023000 C
0023100      DO 60 I=1,NVLF
0023200      R(I,I)=VARP(I) *RSET1
0023300      EVARP(I)=VARP(I) * RSET1
0023400      RR(I,I)=R(I,I)
0023500 60     CONTINUE
0023600 C
0023700 C      *****
0023800 C      * OPTION FOR CIRCULAR TRACK *
0023900 C      *****

```

```

0024000 C
0024100 WRITE(90,902)
0024200 902 FORMAT(' ',2X,'DO YOU WANT CIRCULAR TRACK? 1=YES')
0024300 READ(68,6035) ICIRC
0024400 IF(ICIRC.EQ.0) GO TO 26
0024500 WRITE(90,903)
0024600 903 FORMAT(' ',2X,'INPUT CIRCLE START AND END TIME AND RADIUS',/,2X,
0024700 'FFFF,FF FFFF,FF FFF,FFFF')
0024800 READ(68,610) TURN1,TURN2,RADIUS
0024900 610 FORMAT(F7.2,1X,F7.2,1X,F8.4)
0025000 C
0025100 C ** HEADER FOR BETAT OUTPUT **
0025200 WRITE(50,500)
0025300 500 FORMAT(' ',2X,'SECONDS BETAT')
0025400 C
0025500 25 CONTINUE
0025600 C
0025700 C *** OPTION FOR SMART FILTER FOR CIRC. FLIGHT ***
0025800 WRITE(90,905)
0025900 905 FORMAT(' ',2X,'DO YOU WANT SMART FILTER FOR DOPPLER, 1=YES')
0026000 READ(68,6035) ISMRT
0026100 IF(ISMRT.EQ.0) GO TO 26
0026200 WRITE(90,906)
0026300 906 FORMAT(' ',2X,'INPUT SCALE FACTORS FOR Q MATRIX FIX',/,2X,'FF.FFFF FF.FFFF')
0026400 READ(68,612) Q11,Q22
0026500 612 FORMAT(F7.4,2X,F7.4)
0026600 C
0026700 WRITE(90,907)
0026800 907 FORMAT(' ',2X,'INPUT SCALE FACTORS FOR P MATRIX FIX',/,2X,'FF.FFFF FF.FFFF')
0026900 READ(68,612) P33,P44
0027000 26 CONTINUE
0027100 C
0027200 C *****
0027300 C * VLF BIAS NOISE STATISTICS *
0027400 C *****
0027500 C
0027600 WRITE(90,933)
0027700 933 FORMAT(' ',2X,'INPUT VARIANCE FOR RANDOM VLF BIAS FLUCTUATIONS... F6.3')
0027800 C
0027900 READ(68,631) VARB
0028000 631 FORMAT(F6.3)
0028100 VARBT=VARB
0028200 C
0028300 WRITE(90,934)
0028400 934 FORMAT(' ',2X,'INPUT VALUES FOR Q MATRIX (VLF ELEMENTS)... F6.3')
0028500 DO 29 I=1,NVLF
0028600 WRITE(90,937) STNID(I)
0028700 READ(68,631) Q(I+4,I+4)
0028800 29 CONTINUE
0028900 C
0029000 WRITE(90,938)
0029100 938 FORMAT(' ',2X,'INPUT Q(3,3),Q(4,4)...F5.3')
0029200 READ(68,638) Q(3,3)
0029300 READ(68,638) Q(4,4)
0029400 638 FORMAT(F5.3)
0029500 C
0029600 632 FORMAT(F5.2)
0029700 READ(66,633) (BIY(I),I=1,NVLF)
0029800 633 FORMAT(I6,5(2X,I6))
0029900 C

```

```

0030000      WRITE(90,935)
0030100 935  FORMAT(' ',2X,'DO YOU WANT PERIODIC VLF BIAS NOISE? 1=YES')
0030200      READ(68,6072) IBPER
0030300      IF(IBPER.EQ.0) GO TO 22
0030400 C
0030500      WRITE(90,936)
0030600 936  FORMAT(' ',2X,'INPUT PERIOD AND LEVEL',/,2X,'IIII FF.FF')
0030700 C
0030800      DO 28 I=1,NVLF
0030900      WRITE(90,937) STNID(I)
0031000 937  FORMAT(' ',A1)
0031100      READ(68,634) NBPER(I),VBLEV(I)
0031200 28    CONTINUE
0031300 634  FORMAT(I4,2X,F5.2)
0031400 C
0031500      DO 23 I=1,NVLF
0031600 23    RANDR(I)=VBLEV(I)
0031700 C
0031800 22    CONTINUE
0031900 C
0032000 C
0032100      WRITE(90,940)
0032200 940  FORMAT(' ',2X,'DO YOU WANT GAIN MATRICES PRINTED LOCALLY?')
0032300      READ(68,6072) NMAT
0032400 C
0032500 C
0032600 C      ***** OPTION FOR EXP. DATA WEIGHTING *****
0032700 C
0032800      WRITE(90,908)
0032900 908  FORMAT(' ',2X,'INPUT EXP. DATA WEIGHTING COEFFICIENT...F5.3')
0033000      READ(68,614) S
0033100 614  FORMAT(F5.3)
0033200 C
0033300 C*** OPTION TO LOCALLY CHANGE DATA (LAST CHANCE CHANGES) ***
0033400 C
0033500      PAUSE 'OPTION TO CHANGE ANY PARAMETERS, USE >>QUALIFY<<, >>SET<<, AND >>GO<<'
0033600 C
0033700 C      *****
0033800 C      *
0033900 C      * END OF DATA INPUT SEQUENCE*
0034000 C      *
0034100 C      *****
0034200 C
0034300 C      *****
0034400 C      *
0034500 C      * ECHO INPUT DATA. THIS SERVES *
0034600 C      * AS A CHECK OF YOUR INPUT *
0034700 C      * DATA AND DATA IS WRITTEN TO *
0034800 C      * THE OUTPUT DATA SET FOR *
0034900 C      * LATER REFERENCE. *
0035000 C      *****
0035100 C
0035200 C
0035300      WRITE(99,800)
0035400 800  FORMAT('0',25X,'SIMULATED',9X,'VARIANCE',12X,'BIAS',10X,'BIAS',/,
0035500      &      26X,'TRUE VALUE',5X,'TRUE',4X,'EST.',8X,'TRUE',4X,'EST.',
0035600      &      5X,'VARIANCE',/,49X,'(WEIGHT)',21X,'(WEIGHT)')
0035700 C
0035800 C
0035900      WRITE(99,8010)

```

```

0036000 B010 FORMAT(' ',3X,90(' '))
0036100 WRITE(99,801) LATD,LATM,LATS,ERRLT,EVARX(1),LONGD,LONGM,LONGS,
0036200 & ERRLG,EVARX(2)
0036300 B01 FORMAT(' ',3X,'INIT LATITUDE',8X,I3,1X,I2,1X,F5.2,24X,
0036400 & F6.4,12X,F8.4,/,9X,'LONGITUDE',7X,I3,1X,I2,1X,F5.2,24X,F6.4,12X,F8.4)
0036500 C
0036600 WRITE(99,802) TFA,VARFA,BFA,EBFA,TFB,VARFB,BFB,EBFB,TFE,VARFE,BFE,EBFE
0036700 B02 FORMAT('0',3X,'DOPPLER BEAM FREQS',5X,3(F8.2,6X,F7.2,1X,F6.2,2X,
0036800 & F6.2,/,27X))
0036900 C
0037000 WRITE(99,803) GS,VARV,EVARV,BV,X(3),EVARX(3)
0037100 B03 FORMAT(' ',3X,'GROUNDSPEED',12X,F6.2,8X,F7.3,1X,F7.3,5X,F6.2,2X,F6.2,4X,F7.3)
0037200 C
0037300 C
0037400 WRITE(99,804) HEADNG
0037500 B04 FORMAT(' ',3X,'HEADING WRT TRUE N',5X,F6.2)
0037600 C
0037700 WRITE(99,805) VARA,EVARA,BALPHA,EBA,EVARX(4),BETAU,EVARB,BBETA,EBB
0037800 B05 FORMAT(' ',5X,'ALPHA (DRIFT ANG)',18X,F7.3,1X,F7.3,5X,F6.2,2X,F6.2,4X,F7.3,
0037900 & /,5X,'BETA (TAS HDG)',20X,F7.3,1X,F7.3,5X,F6.2,2X,F6.2)
0038000 WRITE(99,8010)
0038100 WRITE(99,8010)
0038200 WRITE(99,806)
0038300 B06 FORMAT(' ',3X,'VLF STATION',6X,'VARIANCE',10X,'USEC BIAS',9X,'BIAS',
0038400 & 10X,'SIMULATED BIAS DRIFT',/,6X,'PHASES',6X,'TRUE',4X,'EST.',
0038500 & 7X,'TRUE',5X,'EST.',6X,'VARIANCE',/,25X,'(WEIGHT)',23X,
0038600 & '(WEIGHT)',3X,'START',1X,'END',3X,'DRIFT',3X,'SLOPE')
0038700 WRITE(99,8010)
0038800 C
0038900 C
0039000 DO 50 I=1,NULF
0039100 IF(ISTN(I).EQ.0)WRITE(99,807) I,STNID(I),VARP(I),EVARP(I),BP(I),
0039200 & X(I+4),EVARX(I+4)
0039300 B07 FORMAT(' ',3X,I1,2X,A1,9X,F7.3,2X,F7.3,3X,F7.3,2X,F7.3,4X,F7.3)
0039400 C
0039500 IF(ISTN(I).EQ.1) WRITE(99,8075)I,STNID(I),VARP(I),EVARP(I),BP(I),
0039600 & X(I+4),EVARX(I+4),VLFBSI(I,1),VLFBSI(I,1),DVBIAS(I,1),KBIAS(I,1)
0039700 C
0039800 B075 FORMAT(' ',3X,I1,2X,A1,9X,F7.3,2X,F7.3,3X,F7.3,2X,F7.3,4X,F7.3,4X,
0039900 & I4,2X,I4,2X,F6.2,2X,F6.3)
0040000 50 CONTINUE
0040100 WRITE(99,8022)
0040200 B022 FORMAT(' ')
0040300 C
0040400 WRITE(99,8010)
0040500 C
0040600 C ** CONVERT INPUT LAT/LONG TO DECIMALS **
0040700 C
0040800 LAT=DEGRAD(LATD,LATM,LATS,0)
0040900 LONG=DEGRAD(LONGD,LONGM,LONGS,0)
0041000 C
0041100 C INITIALIZE FIRST TWO TERMS OF X MATRIX
0041200 C
0041300 X(1)=LAT+DBLE(ERRLT)
0041400 X(2)=LONG+DBLE(ERRLG)
0041500 C
0041600 C
0041700 WRITE(99,808) (X(I),I=1,NX)
0041800 B08 FORMAT('0',3X,'INITIAL STATE VECTOR',/,4X,10F9.4)
0041900 C

```

```

0042000      WRITE(99,8081) (EVARX(I),I=1,NX)
0042100 8081 FORMAT('0',3X,'INITIAL P MATRIX DIAGONALS',/,4X,10F9.4)
0042200 C
0042300      WRITE(99,8082) (EVARV,EVARAB,(Q(I,I),I=3,NX)
0042400 8082 FORMAT('0',3X,'INITIAL Q MATRIX DIAGONALS',/,4X,10F9.4)
0042500 C
0042600      WRITE(99,8083) (EVARP(I),I=1,NVLF)
0042700 8083 FORMAT('0',3X,'INITIAL R MATRIX DIAGONALS',/,4X,10F9.4)
0042800 C
0042900 C
0043000 C
0043100 C
0043200 C      ** END OF DATA ECHO SEQUENCE **
0043300 C
0043400 C
0043500 C      *****
0043600 C      * CONVERT VARIABLES TO PROGRAM *
0043700 C      * ORIENTED QUANTITIES(IE NM/SEC *
0043800 C      * AND RADIANS, ETC. *
0043900 C      *****
0044000 C
0044100 C
0044200      TVX=TVX/3600.
0044300      TVY=TVY/3600.
0044400      TBX=TBX/3600.
0044500      TBY=TBY/3600.
0044600      BV=BV/3600.
0044700      ALPHAT=ALPHAT*RAD
0044800      BETAT=BETAT*RAD
0044900      BALPHA=BALPHA*RAD
0045000      BBETA=BBETA*RAD
0045100      GS=GS/3600.
0045200      X(3)=X(3)/3600.
0045300      X(4)=(EBA+EBB)*RAD
0045400      HEADNG=HEADNG*RAD
0045500      EVARX(3)=EVARX(3)/(3600.*3600.)
0045600      EVARX(4)=EVARX(4)*(RAD*RAD)
0045700      STDA=STDA/3600.
0045800      STDB=STDB/3600.
0045900      STDC=STDC/3600.
0046000      STDD=STDD*RAD
0046100      BETAV=BETAV*(RAD*RAD)
0046200      EVARV=EVARV/(3600.*3600.)
0046300      EVARAB=(EVARA+EVARB)*(RAD*RAD)
0046400      Q(3,3)=Q(3,3)/(3600.*3600.)
0046500      Q(4,4)=Q(4,4)*RAD*RAD
0046600 C      ** END OF CONVERSION SEQUENCE **
0046700 C
0046800 C
0046900 C      INITIALIZE P MATRIX. P WAS INITIALLY SET TO 0
0047000 C
0047100      DO 30 I=1,NX
0047200      P(I,I)=EVARX(I)
0047300      CONTINUE
0047400 C
0047500 C      INITIALIZE Q MATRIX. Q WAS INITIALLY SET TO 0
0047600 C
0047700      Q(1,1)=EVARV
0047800      Q(2,2)=EVARAB
0047900 C

```

```

0048000 C
0048100 C ** INITIAL BETA FIX ***
0048200 C
0048300 STBETA=BETAT
0048400 STHDG=HEADNG
0048500 C
0048600 C
0048700 C
0048800 WRITE(99,952)
0048900 952 FORMAT('1')
0049000 C
0049100 C *****
0049200 C * START RANDOM NUMBER GENERATORS TO ENSURE NUMBERS ***
0049300 C * ARE TRULY RANDOM. ***
0049400 C * UPDATE AUG.01.78 THE DO 35 LOOP COULD BE LEFT OUT ***
0049500 C * PROVIDING INITIAL IY>APPROX.38000 ***
0049600 C *****
0049700 C
0049800 DO 34 I=1,NVLF
0049900 VLFSD(I)=SQRT(VARP(I))
0050000 34 CONTINUE
0050100 C
0050200 DO 35 I=1,20
0050300 DO 35 J=1,NVLF
0050400 CALL NDRAN(IY(J),VLFSD(J),0.,RAND(J+4))
0050500 C
0050600 35 CONTINUE
0050700 C
0050800 C ** INITIALIZE TIME COUNTER **
0050900 C
0051000 NN=0
0051100 C
0051200 C*** MODS FOR VARIABLE TIME INCREMENT 23 FEB 81 **
0051300 C
0051400 WRITE(90,930)
0051500 930 FORMAT(' ',TIME INCREMENT? MULTIPLE OF 10 SECS, 12')
0051600 READ(68,630) NINC
0051700 630 FORMAT(12)
0051800 C
0051900 IF(NINC.EQ.NPRINT) GO TO 71
0052000 WRITE(90,931)
0052100 931 FORMAT(' ',CHANGE NPRINT TO MATCH NINC, 12')
0052200 READ(68,630) NPRINT
0052300 C
0052400 C
0052500 C
0052600 71 CONTINUE
0052700 C*** END OF MOD ***
0052800 C
0052900 C ** MOD FOR INITIAL VLF OBS WITH ARB. BIAS **
0053000 C ** THIS MOD USED FOR D/P TO MAG TAPE FOR 7/32 VLF DATA **
0053100 C
0053200 C
0053300 DO 78 I=1,NVLF
0053400 C
0053500 OBS1(I,1)= GI(X(1),X(2),STNID(I),0,CV,WL)
0053600 OBS2(I,1)= OBS1(I,1)+1000.0
0053700 IF(OBS2(I,1).GT.10000.0) OBS2(I,1)=OBS2(I,1)-10000.0
0053800 IF(OBS2(I,1).LT.0.0) OBS2(I,1)=OBS2(I,1)+10000.0
0053900

```



```

0054000 IF(OBS2(I,1),LT,0.0) OBS2(I,1)=OBS2(I,1)+10000.0
0054100 78 CONTINUE
0054200 IF(IDTA.EQ.1) WRITE(40,401) (OBS2(I,1),I=1,NVLF)
0054300 C
0054400 C ** OPTION TO OUTPUT P MATRIX DIAGONAL ELEMENTS **
0054500 C
0054600 PU3=P(3,3)*3600.0*3600.0
0054700 PU4=P(4,4)/RAD/RAD
0054800 IF(IPOUT.EQ.1) WRITE(91,9100) NN,P(1,1),P(2,2),PU3,PU4,(P(J,J),J=5,10)
0054900 C
0055000 C *****
0055100 C ***** RETURN POINT FOR SUCCESSIVE ITERATIONS *****
0055200 C *****
0055300 C
0055400 C
0055500 C
0055600 41 CONTINUE
0055700 C
0055800 C
0055900 C
0056000 C ** DETERMINE IF DATA SHOULD BE WRITTEN. **
0056100 C ** 'NPRINT' WAS PREVIOUSLY USER DEFINED **
0056200 C
0056300 IF(NN/NPRINT*NPRINT.NE.NN) GO TO 455
0056400 C
0056500 WRITE(99,9125) NN,LAT, LONG
0056600 9125 FORMAT('0',I4,1X,I4,1X,'LAT:',F10.5,' LONG:',F10.5)
0056700 C
0056800 C
0056900 C
0057000 C
0057100 C *****
0057200 C ** CALCULATE ERROR QUANTITIES **
0057300 C ** FOR OUTPUT CONSIDERATIONS WE WANT **
0057400 C ** TO LOOK AT USER ORIENTED VALUES **
0057500 C ** SO CONVERT NM/SEC TO KNOTS AND **
0057600 C ** RADIANS TO DEGREES. **
0057700 C ** **
0057800 C ** XU3= X, USER 3 **
0057900 C ** **
0058000 C ** EXPRESS LAT/LONG DEGREE ERRORS IN **
0058100 C ** NM, ERRORS. **
0058200 C *****
0058300 C
0058400 C
0058500 C
0058600 DX(1)=(LAT-X(1))*(60*(6077-31*COS(2*LAT*RAD)))/6077.
0058700 C
0058800 DX(2)=(LONG-X(2))*(60*(6077-31*COS(2*LAT*RAD))
0058900 * *COS(LAT*RAD))/6077.
0059000 C
0059100 IF(IERRPL.EQ.1) WRITE(97,972) NN,DX(1),DX(2)
0059200 972 FORMAT(' ',I4,2X,F12.8,2X,F12.8)
0059300 C
0059400 DX(3)=(BV-X(3))*3600.
0059500 DX(4)=(BALPHA+BBETA-X(4))/RAD
0059600 DO 44 I44=5,NX
0059700 DX(I44)=BP(I44-4)-X(I44)
0059800 44 CONTINUE
0059900 C

```

```

0060000 C
0060100 C
0060200 XU3=X(3)*3600
0060300 XU4=X(4)/RAD
0060400 C
0060500 C
0060600 C *** WRITE UPDATES AND MATRICES ***
0060700 C
0060800 C
0060900 WRITE(99,914) X(1),X(2),XU3,XU4,(X(I),I=5,NX)
0061000 914 FORMAT(' ',4(1X,F10.5),6(2X,F9.5))
0061100 C
0061200 C
0061300 WRITE(99,914)(DX(I),I=1,NX)
0061400 C
0061500 C
0061600 C
0061700 C
0061800 IF(DSRN.EQ.0) GO TO 455
0061900 C
0062000 WRITE(DSRN,901) MTH, DAY, YEAR, T
0062100 C
0062200 C
0062300 WRITE(DSRN,915) NN
0062400 915 FORMAT(' ',5X,'P MATRIX',15,' SECONDS',15,' ')
0062500 CALL MOUTS(P,10,NX,NX,DSRN)
0062600 C
0062700 C
0062800 WRITE(DSRN,916)
0062900 916 FORMAT(' ',5X,'Q MATRIX:')
0063000 CALL MOUTS(Q,2,2,2,DSRN)
0063100 C
0063200 C ** INITIALLY (AT TIME=0) WE DO NOT HAVE **
0063300 C ** PHI,GAMMA,M,OR K MATRICES DEFINED. **
0063400 C ** THEREFORE, DO NOT WRITE. **
0063500 C
0063600 C
0063700 IF(NN.EQ.0) GO TO 455
0063800 C
0063900 WRITE(DSRN,917)
0064000 917 FORMAT(' ',5X,'PHI MATRIX:')
0064100 CALL MOUTS(PHI,10,NX,NX,DSRN)
0064200 C
0064300 WRITE(DSRN,918)
0064400 918 FORMAT(' ',5X,'GAMMA MATRIX:')
0064500 CALL MOUTS(GAMMA,10,NX,2,DSRN)
0064600 C
0064700 WRITE(DSRN,919)
0064800 919 FORMAT(' ',5X,'M MATRIX:')
0064900 CALL MOUTS(M1,10,NX,NX,DSRN)
0065000 C
0065100 C
0065200 WRITE(DSRN,921)
0065300 921 FORMAT(' ',5X,'GAIN MATRIX:')
0065400 CALL MOUTS(K1,10,NX,NVLF,DSRN)
0065500 C
0065600 C
0065700 C
0065800 455 CONTINUE
0065900 C

```

```

0066000 C
0066100 C *** MOD 16 APR 81, FIND EIGENVALUES OF P MATRIX FOR CONDX NUMBER ***
0066200 IF(NN/500*500.EQ.NN .AND. IEIG.EQ.1) CALL EIGVAL(P,NX,NN,IPART)
0066300 C *** END MOD
0066400 C
0066500 C
0066600 C
0066700 C ** IF SIMULATION PERIOD IS OVER, NO MORE ITERATIONS ARE DONE **
0066800 C
0066900 IF(NN.GE.NT) GO TO 70
0067000 C
0067100 C ** OTHERWISE, INCREMENT TIME COUNTER BY TIME INCREMENT **
0067200 C
0067300 NN=NN+NINC
0067400 C
0067500 C *** SET OMEGA FOR OPTIONAL CIRCULAR TRACK ***
0067600 OMEGA= 0.0
0067700 IF(NN.GT.TURN1 .AND. NN.LE.TURN2) OMEGA= GS/RADIUS
0067800 C
0067900 C
0068000 C *****
0068100 C * ADJUST PLANT BIAS IF DRIFTING *
0068200 C * RESET P MATRIX IF DESIRED *
0068300 C *****
0068400 C
0068500 IF(NN.LE.VBST .OR. NN.GT.VBET) GO TO 63
0068600 BV= BV + NINC*KVBIAS
0068700 C2=TAN(BALPHA)
0068800 TBX= ((BV+GS)/(C2*2.0 + 1.0)*0.5) - TVX
0068900 TBY= C2*(TVX+TBX)-TVY
0069000 IF(IPSMRT.EQ.1) P(3,3)= 0.25*KVBIAS
0069100 63 CONTINUE
0069200 C
0069300 IF(NN.LE.HBST .OR. NN.GT.HBET) GO TO 69
0069400 BBETA= BBETA + NINC*KHBIAS
0069500 IF(IPSMRT.EQ.1) P(4,4)= 2.0*KHBIAS
0069600 69 CONTINUE
0069700 C
0069800 C
0069900 C
0070000 C *****
0070100 C * ADJUST VLF BIAS IF WITHIN DRIFT RANGE *
0070200 C *****
0070300 C
0070400 DO 61 I=1,NVLF
0070500 C
0070600 IF(NN.GT.VLFBET(I,4)) GO TO 61
0070700 IF(NN.LE.VLFBST(I,1)) GO TO 61
0070800 JBIAS=1
0070900 IF(NN.GT.VLFBST(I,2).AND.NN.LE.VLFRET(I,2)) JBIAS=2
0071000 IF(NN.GT.VLFBST(I,3).AND.NN.LE.VLFBET(I,3)) JBIAS=3
0071100 IF(NN.GT.VLFBST(I,4).AND.NN.LE.VLFBET(I,4)) JBIAS=4
0071200 BP(I)=BP(I) + NINC*KBIAS(I,JBIAS)
0071300 IFLAG1(I)=1
0071400 61 CONTINUE
0071500 C
0071600 C *****
0071700 C * ADJUST VLF VARIANCE IF NEC. *
0071800 C *****
0071900 C

```

```

0072000 IF(INOISE,EQ,0) GO TO 65
0072100 DO 66 I=1,NVLF
0072200 IF(IVLFN(I),EQ,0) GO TO 66
0072300 IF(NN,NE,VLFVT(I)) GO TO 66
0072400 VARP(I)= VARP(I)
0072500 66 CONTINUE
0072600 65 CONTINUE
0072700 C
0072800 C
0072900 IF(IVAUTO,EQ,1) GO TO 68
0073000 C
0073100 C *****
0073200 C * IF BIAS DRIFTING IS OCCURRING, ADJUST *
0073300 C * THE VARIANCE ON THE BIAS TO REFLECT *
0073400 C * THE UNCERTAINTY, (FOR SMART FILTER) *
0073500 C * *
0073600 C * HERE IFLAG1=1: BIAS IS DRIFTING *
0073700 C * IFLAG2=1: VARIANCE HAS BEEN *
0073800 C * ADJUSTED ALREADY *
0073900 C *****
0074000 C
0074100 DO 62 I=1,NVLF
0074200 IF(IFLAG1(I),EQ,0) GO TO 62
0074300 IF(IFLAG2(I),EQ,1) GO TO 62
0074400 P(I+4,I+4)= 5000 * KBIAS(I,JBIA)
0074500 IF(NN,GT,VLFVET(I,4)) IFLAG2(I)=1
0074600 62 CONTINUE
0074700 C
0074800 C
0074900 C ** ADJUST R MATRIX IF NECESSARY **
0075000 C
0075100 IF(IRSMRT,EQ,0) GO TO 68
0075200 DO 67 I=1,NVLF
0075300 IF(IVLFN(I),EQ,0) GO TO 67
0075400 IF(NN,NE,VLFVT(I)) GO TO 67
0075500 R(I,I) = RCDEF * VARP(I)
0075600 WRITE(90,922) I,I,R(I,I),NN
0075700 922 FORMAT(' 2X, NEW R MATRIX ELEMENT, R(,I1, ,I1,)=,2X,
0075800 8 4B,2,2X, RESET AT, I5, SECS')
0075900 C
0076000 67 CONTINUE
0076100 C
0076200 68 CONTINUE
0076300 C
0076400 C *****
0076500 C * INITIALIZE A COEFFS. READ AND PROCESS *
0076600 C * DOPPLER/HEADING DATA *
0076700 C *****
0076800 C
0076900 C
0077000 A1=0.0
0077100 A2=0.0
0077200 A3=0.0
0077300 A4=0.0
0077400 C
0077500 NDOP= NINC*2 + 1
0077600 C
0077700 CALL DOP2(ROLL,PITCH,VDOP,ALPHAD,BETAD,STDD,BETAT,BBETA
0077800 8 ,IY1,IY2,IY3,IY4,I,OMEGA,NN,STBETA,TURN1,NDOP)
0077900 C

```

0078000 C ** OUTPUT FLIGHT DATA **

0078100 C

0078200 DO 31 I=1,NDOP

0078300 VX(I)=VDOP(I)/SQRT(1.0 + TAN(ALPHAD(I))*TAN(ALPHAD(I)))

0078400 VY(I)=VX(I)*ABS(TAN(ALPHAD(I)))

0078500 C

0078600 VX(I)=VX(I)*3600.0

0078700 VY(I)=VY(I)*3600.0

0078800 IF(BETAD(I).GT.PI) BETAD(I)=BETAD(I)-2*PI

0078900 IF(BETAD(I).LT. -PI) BETAD(I)=BETAD(I)+2.0*PI

0079000 C

0079100 BBB(I)=180.0/3.14159*BETAD(I)

0079200 31 CONTINUE

0079300 IF(IDTA.EQ.1) WRITE(40,400) NN,(VX(I),VY(I),BBB(I),I=1,NDOP)

0079400 400 FORMAT(' ',I4,7,5X,10(4(F8.4,2X,F8.4,2X,F9.4,5X),/,5X),F8.4,2X,F8.4,2X,F9.4)

0079500 C

0079600 C

0079700 CALL ASUB(VDOP,ALPHAD,BETAD,X,A5,A6,A7,AB,I,NDOP)

0079800 C

0079900 C

*** OPTION FOR Q MATRIX FIX FOR CIRCULAR TRACK ***

0080000 C

*** (ADJUST DOPPLER WEIGHT)

0080100 C

0080200 C

0080300

Q(1,1)= EVARU + Q11*OMEGA

0080400

Q(2,2)= EVARAB+ Q22*OMEGA

0080500 C

0080600 C

0080700 C

*** ADJUST P MATRIX ELEMENTS IF OUT OF CIRCULAR TRACK ***

0080800 C

0080900

IF(ICIRC.EQ.0 .OR. ISMRT.EQ.0) GO TO 64

0081000

IF(HEADNG.EQ.STHDG .OR. OMEGA.NE.0.0 .OR. IPFLAG.EQ.1 .OR. NINC.EQ.10) GO TO 64

0081100

P(3,3)=P33 * EVARX(3)

0081200

P(4,4)=P44 * EVARX(4)

0081300 C

0081400

IPFLAG=1

0081500 C

0081600 64 CONTINUE

0081700 C

0081800 C

0081900 C

* * * * *

0082000 C

* START KALMAN FILTERING

0082100 C

0082200 C

0082300

CALL FISUB(A5,A6,A7,AB,PHI,GAMMA,X)

0082400

CALL FSUB11(F)

0082500 C

0082600 C

0082700 C

** CALCULATE UPDATED TRUE POSITION **

0082800 C

0082900

TIME1=FLOAT(NINC)

0083000

CALL TRU(LAT, LONG, HEADNG, GS, TIME1, OMEGA, RADIUS, STHDG, NN, TURN1)

0083100 C

0083200 C

0083300

IF(ICIRC.EQ.1) WRITE(50,501) NN,BETAD,HEADNG

0083400 501

FORMAT(' ',3X,I4,4X,F9.4,4X,F9.4)

0083500 C

0083600 C

0083700

CALL VLFSIM(IY,LAT, LONG, STNID, OBS, BP, VARP, RAND, BIY, VARB, RANDB, NBPER, VBLEV, NN, KAO)

0083800 C

0083900 C

```

0084000 C ** MOD FOR REVERSED, BIASED VLF DATA (FOR MAG TAPE O/P) **
0084100 C
0084200 DO 77 I=1,NVLF
0084300 OBS2(I,1)= 2.0*OBS1(I,1)-OBS(I,1)+1000.0
0084400 IF(OBS2(I,1).GT.10000.0) OBS2(I,1)=OBS2(I,1)-10000.0
0084500 IF(OBS2(I,1).GT.10000.0) OBS2(I,1)=OBS2(I,1)-10000.0
0084600 IF(OBS2(I,1).LT.0.0) OBS2(I,1)=OBS2(I,1)+10000.0
0084700 IF(OBS2(I,1).LT.0.0) OBS2(I,1)=OBS2(I,1)+10000.0
0084800 77 CONTINUE
0084900 IF(IDTA.EQ.1) WRITE(40,401) (OBS2(I,1),I=1,NVLF)
0085000 401 FORMAT(' ',4X,6(F9.3,2X))
0085100 C
0085200 DO 87 I=1,NVLF
0085300 SMALLH(I,1)= -GI( F(1),F(2),STNID(I),0,CV,WL) - X(I+4)
0085400 IF(KAQ(I).EQ.0) OBS(I,1)=SMALLH(I,1)
0085500 87 CONTINUE
0085600 C
0085700 C
0085800 CALL MADD(OBS,SMALLH,NVLF,1,6,1,A)
0085900 C
0086000 C *****
0086100 C * VLF NOISE AND BIAS JUMP *
0086200 C * CHECKING *
0086300 C *****--*****
0086400 C
0086500 IF(IVAUTO.EQ.0) GO TO 80
0086600 C
0086700 IF(NN.LT.300) GO TO 80
0086800 IF(NN.GT.880) CALL VLFIX(P,R,NVLF,KAQ,IVAR,A,VMEAN,VSTD,KVLF,VLFN,NINC,NN,PCOEFF,TOLCF)
0086900 C IF(NN.GT.800) CALL WDIAG(P,10,NX,NX,80,VECTOR,NN)
0087000 C
0087100 DO 79 I79=1,NVLF
0087200 IF(IVAR(I79).EQ.1) GO TO 79
0087300 IF(KAQ(I79).EQ.0) GO TO 79
0087400 C
0087500 KVLF(I79)= KVLF(I79) + 1
0087600 KKVLF= KVLF(I79)
0087700 VLFN(KKVLF,I79) = A(I79,1)
0087800 C
0087900 IF(KVLF(I79).LT. 30) GO TO 79
0088000 CALL STAT(VLFN,30,I79,VMEAN(I79),VSTD(I79))
0088100 IVAR(I79) = 1
0088200 KVLF(I79) = 0
0088300 C
0088400 79 CONTINUE
0088500 C
0088600 80 CONTINUE
0088700 C
0088800 C
0088900 PU3=P(3,3)*3600.0*3600.0
0089000 PU4=P(4,4)/RAD/RAD
0089100 IF(IPOUT.EQ.1) WRITE(91,9100)NN,P(1,1),P(2,2),PU3,PU4,(P(J,J),J=5,10)
0089200 9100 FORMAT(' ',I4,10(1X,E11.4))
0089300 C
0089400 CALL MISUB1(PHI,P,GAMMA,Q,M1,S)
0089500 CALL HSUB1(X(1),X(2),STNID,H,NN)
0089600 CALL KISUB1(M1,H,R,K1,NN,NMAT)
0089700 CALL EMULTM(K1,A,B,NX,1,NVLF,10,6,10)
0089800 C
0089900 C

```

```

0090000      DO 81 I=1,NX
0090100      X(I)= F(I)+B(I,1)
0090200      81  CONTINUE
0090300 C
0090400      CALL PSUB11(M1,K1,H,P)
0090500 C
0090600 C
0090700 C
0090800 C ** VLF NOISE OUTPUT **
0090900      IF(INOISE.EQ.1) WRITE(94,966) NM
0091000 966  FORMAT(' ',2X,I4,2X,'SECONDS')
0091100      DO 88 I=1,NVLF
0091200      AA= OBS(I,1)+SMALLH(I,1)
0091300      BB= RAND(I+4)-OBS(I,1)-SMALLH(I,1)
0091400      IF(ISTN(I).EQ.1 .OR. IULFN(I).EQ.1) WRITE(94,967) STNID(I),AA,RAND(I+4),BB
0091500 967  FORMAT(' ',5X,A1,3(3X,F7.3))
0091600      88  CONTINUE
0091700 C ** END NOISE OUTPUT **
0091800 C      ** AFTER THE A POSTERIORI COVARIANCE MATRIX HAS **
0091900 C      ** BEEN DETERMINED, DO ANOTHER ITERATION      **
0092000 C
0092100      GO TO 41
0092200 C
0092300 C
0092400 C      *****
0092500 C      * PRINT FINAL UPDATE AND TERMINATE      *
0092600 C      *****
0092700 C
0092800 C
0092900 70  CONTINUE
0093000      WRITE(99,923)
0093100 923  FORMAT(' ',5X,'UPDATED X MATRIX')
0093200      DO 75 I=1,NX
0093300      WRITE(99,914) X(I)
0093400 75  CONTINUE
0093500 C
0093600      STOP
0093700      END
EOF

```

```

0000100 SUBROUTINE ASUB(VDOP,ALPHAD,BETAD,X,A5,A6,A7,A8,J,NDOP)
0000200 C
0000300 C ASUB: ROUTINE TO CALCULATE THE 8 A COEFFICIENTS USED IN
0000400 C THE KALMAN FILTER SIMULATION ROUTINE
0000500 C
0000600 REAL VDOP(NDOP),ALPHAD(NDOP),BETAD(NDOP),TERM1(81),TERM2(81),TERM3(81)
0000700 & ,TERM4(81)
0000800 REAL X(10),LATCOR
0000900 C
0001000 C
0001100 COMMON/COM1/A1,A2,A3,A4
0001200 C
0001300 PI=3.14159
0001400 C
0001500 TESTN=0.0
0001600 C
0001700 > K= NDOP-1
0001800 DO 10 I=1,K
0001900 TERM1(I)=0.25*(VDOP(I)*SIN(ALPHAD(I)+BETAD(I))
0002000 & +VDOP(I+1)*SIN(ALPHAD(I+1)+BETAD(I+1)))
0002100 C
0002200 A1=A1+TERM1(I)
0002300 C
0002400 C
0002500 TERM2(I)=0.25*(VDOP(I)*COS(ALPHAD(I)+BETAD(I))
0002600 & +VDOP(I+1)*COS(ALPHAD(I+1)+BETAD(I+1)))
0002700 C
0002800 A2=A2+TERM2(I)
0002900 C
0003000 C
0003100 TERM3(I)=0.25*(SIN(ALPHAD(I)+BETAD(I))+
0003200 & SIN(ALPHAD(I+1)+BETAD(I+1)))
0003300 C
0003400 A3=A3+TERM3(I)
0003500 C
0003600 C
0003700 TERM4(I)=0.25*(COS(ALPHAD(I)+BETAD(I))+
0003800 & COS(ALPHAD(I+1)+BETAD(I+1)))
0003900 C
0004000 A4=A4+TERM4(I)
0004100 C
0004200 10. CONTINUE
0004300 C
0004400 C
0004500 C ** SCALE THE A COEFFS IF ENOUGH DATA HAS BEEN PROCESSED **
0004600 C ** OTHERWISE ,ANOTHER CALL TO ASUB WILL BE MADE **
0004700 C
0004800 C ** CORRECTION FACTOR FOR NM TO DEGREE CONVERSION
0004900 C ** AT PRESENT LATITUDE
0005000 C ** LATCOR= LATITUDE CORRECTION ***
0005100 C
0005200 LATCOR= 6077.0/(60.0 *(6077.0- 31.0*COS(2.0*(1)*3.14159/180.0)) )
0005300 C
0005400 A1=A1 * LATCOR
0005500 A2=A2 * LATCOR
0005600 A3=A3 * LATCOR
0005700 A4=A4 * LATCOR
0005800 20 CONTINUE
0005900 C

```



```
0006000 C
0006100 KK= (NDOP+1)/2
0006200 A5=+(VDOP(KK)-X(3))*SIN(ALPHAD(KK)+BETAD(KK)-X(4)) * LATCOR
0006300 A6=-COS(ALPHAD(KK)+BETAD(KK)-X(4)) * LATCOR
0006400 A7=-A6*(VDOP(KK)-X(3))/COS(X(1)*PI/180.)
0006500 A8= SIN(ALPHAD(KK)+BETAD(KK)-X(4)) *LATCOR /
0006600 8 (COS(X(1)*PI/180.))
0006700 C
0006800 C
0006900 RETURN
0007000 END
EDF
```

```
0000100      FUNCTION DEGRAD(DEGS,MINS,SECS,K)
0000200 C
0000300 C      FUNCTION TO CALCULATE( CONVERT TO) DECIMAL DEGREES
0000400 C      OR RADIANS WHEN GIVEN DEGREES:MINS:SECS FORMAT.
0000500 C
0000600 C      K=0 FOR DEGREES OUTPUT
0000700 C      K=1 FOR RADIANS OUTPUT
0000800 C
0000900      INTEGER DEGS,MINS
0001000      RD=3.14159/180.
0001100 C
0001200      DEGRAD= IABS(DEGS)+IABS(MINS)/60.+ABS(SECS)/3600.
0001300      IF (DEGS.LT.0.OR.MINS.LT.0.OR.SECS.LT.0) DEGRAD=-DEGRAD
0001400      IF (K.EQ.0) RETURN
0001500      DEGRAD=DEGRAD*RD
0001600      RETURN
0001700 C
0001800      END
EDF
```

```

0000100      SUBROUTINE DOPIN(PITCH,ROLL,BETAT,BBETA,EBB,BETAV,EVARV,
0000200      &          EVARA,EVARB,VARA,IY1,IY2,IY3,IY4,EVARX3,EVARX4)
0000220 C
0000240 C          DOPIN: ROUTINE TO READ IN INITIAL DOPPLER/HEADING PARAMTERS
0000260 C
0000300 C
0000400 C
0000500      COMMON /DOPPLER/ TFA,TFB,TFC,BFA,BFB,BFC,EBFA,EBFB,EBFC,
0000600      &          VARFA,VARFB,VARFC,STDA,STDB,STDC,TVX,TVY,TBX,TBY
0000700 C
0000800      READ(64,600) TFA,TFB,TFC
0000900 600  FORMAT(3F11.2)
0001000 C
0001100      READ(64,601) PITCH,ROLL
0001200 601  FORMAT(2F6.3)
0001300 C
0001400      READ(64,603) BETAT,BBETA,BETAV,EBB
0001500 603  FORMAT(4F7.3)
0001600 C
0001700      READ(64,605) IY1,IY2,IY3,IY4
0001800 605  FORMAT(I6,3(2X,I6))
0001900 C
0002000      READ(64,602) BFA,BFB,BFC
0002100      READ(64,602) EBFA,EBFB,EBFC
0002200      READ(64,602) VARFA,VARFB,VARFC
0002300 602  FORMAT(3F8.3)
0002400 C
0002500      READ(64,602) EVARV,EVARA,EVARB
0002600 C
0002610 C
0002620      READ(64,606) EVARX3
0002630      READ(64,606) EVARX4
0002640 606  FORMAT(F9.5)
0004400 C
0004500      PITCH=PITCH*3.141593/180.
0004600      ROLL=ROLL*3.141593/180.
0004700 C
0004800      RETURN
0004900      END
EOF

```

```

REKEYED
0000100 SUBROUTINE DOP1(STD, BETAV, GS, ALPHAT, BETAT, HEADNG, BV, BALPHA, EBA, ROLL, PITCH,
0000200 VARV, VARA)
0000300 C
0000400 C DOPI: ROUTINE FOR PROCESSING THE INITIAL DOPPLE/HEADING
0000500 C PARAMETERS FOR KALMAN FILTER SIMULATION
0000600 C
0000700 C INPUT: PARAMTERS OF THREE BEAM FREQUENCIES
0000800 C OUTPUT: PARAMTERS FOR DOPPLER VELOCITY AND DRIFT ANGLE
0000900 C
0001000 C
0001100 COMMON /DOPLER/ TFA, TFB, TFC, BFA, BFB, BFC, EBFA, EBFB, EBFC, VARFA, VARFB, VARFC,
0001200 & STDA, STDB, STDC, TVX, TVY, TBX, TBY
0001300 C
0001400 C
0001500 FU= (TFB-TFC)/8.925
0001600 FV= (TFA-TFB)/4.4625
0001700 FW= ((TFA+TFC)-50000)/20.576
0001800 C
0001900 TVX= COS(PITCH)*FU + SIN(PITCH)*SIN(ROLL)*FV + SIN(PITCH)*COS(ROLL)*FW
0002000 TVY= COS(ROLL)*FV - SIN(ROLL)*FW
0002100 C
0002200 GS=(TVX**2 + TVY**2) ** 0.50
0002300 C
0002400 C
0002500 C= TVY/TVX
0002600 ALPHAT= ATAN(C)*180./3.141593
0002700 C
0002800 HEADNG= ALPHAT + BETAT
0002900 C
0003000 C *****
0003100 C * CALCULATE TRUE BIASES AND EST. *
0003200 C * BIASES FROM THE BEAM *
0003300 C * FREQUENCIES *
0003400 C *****
0003500 C
0003600 C TRUE BIASES 0003700 C
0003800 TBU= (BFB-BFC)/8.925
0003900 TBV= (BFA-BFB)/4.4625
0004000 TBW= (BFA+BFB)/20.576
0004100 C
0004200 TBX= COS(PITCH)*TBU + SIN(PITCH)*SIN(ROLL)*TBV
0004300 & + SIN(PITCH)*COS(ROLL)*TBW
0004400 C
0004500 TBY= COS(ROLL)*TBV - SIN(ROLL)*TBW
0004600 C
0004700 BV= ((TVX+TBX)**2 + (TVY+TBY)**2) **0.50 - GS
0004800 C
0004900 C=(TVY+TBY)/(TVX+TBX)
0005000 BALPHA= ATAN(C)*180./3.141593 - ALPHAT
0005100 C
0005200 C
0005300 C ESTIMATED BIASES FROM ESTIMATED BEAM FREQUENCY BIASES
0005400 C
0005500 EBU= (EBFB-EBFC)/8.925
0005600 EBV= (EBFA-EBFB)/4.4625
0005700 EBW= (EBFA+EBFB)/20.576

```

```
0005800 C
0005900   EBX= COS(PITCH)*EBU + SIN(PITCH)*SIN(ROLL)*EBV + SIN(PITCH)*COS(ROLL)*EBW
0006000   EBY= COS(ROLL)*EBV - SIN(ROLL)*EBW
0006100 C
0006200   X3= ((TVX+EBX)**2 + (TVY+EBY)**2) **.5 - GS
0006300 C
0006400   C=(TVY+EBY)/(TVX+EBX)
0006500   EBA= ATAN(C)* 180./3.141593 - ALPHAT
0006600 C
0006700   STDA= SQRT(VARFA)
0006800   STDB= SQRT(VARFB)
0006900   STDC= SQRT(VARFC)
0007000   STDD= SQRT(BETAV)
0007100 C
0007200   VARNX= 0.0125*(VARFB+VARFC)
0007300   VARNY= 0.0502*(VARFA+VARFB)
0007400   A= TVX+TBX
0007500   B= TVY+TBY
0007600 C
0007700   VARV= (A**2*VARNX + B**2*VARNY)/(A**2+B**2)
0007800 C
0007900   VARA= (B**2*VARNX + A**2*VARNY)/(A**2 + B**2)**2
0008000   VARA= VARA*(180.0/3.14159)**2
0008100 C
0008200 C
0008300   RETURN
0008400   END
0008500 C
EOF
```

59 RECORDS (VI V 132) SOURCE.CDOP2

REKEYED

0000100 SUBROUTINE DOP2(ROLL,PITCH,VDOP,ALPHAD,BETAD,STDD

0000200 & ,BETAT,BBETA,IY1,IY2,IY3,IY4,JDOP,OMEGA,NN,STBETA,TURN1,NDOP)

0000300 C

0000400 C

DOP2: ROUTINE TO PROCESS DOPPLER/HEADING DATA(SIMULATES DOPPLER/HEADING DATA)

0000500 C

0000600 C

0000700 C

REAL VDOP(NDOP),ALPHAD(NDOP),BETAD(NDOP),NX,NY,NU,NV,NW

0000800 C

0000900 C

COMMON/DOPLER/ TFA,TFB,TFC,BFA,BFB,BFC,EBFA,EBFB,EBFC,VARFA,

0001000 C

& VARFB,VARFC,STDA,STDB,STDC,TVX,TVY,TBX,TBY

0001100 C

0001200 C

VDOP(1)=VDOP(NDOP)

0001300 C

ALPHAD(1)=ALPHAD(NDOP)

0001400 C

BETAD(1)=BETAD(NDOP)

0001500 C

5 CONTINUE

0001600 C

0001700 C

II=2

0001800 C

IF(VDOP(1).EQ.0.00) II=1

0001900 C

DO 10 I=II,NDOP

0002000 C

CALL NDRAN(IY1,STDA,0.,RANDA)

0002100 C

CALL NDRAN(IY2,STDB,0.,RANDB)

0002200 C

CALL NDRAN(IY3,STDC,0.,RANDC)

0002300 C

0002400 C

NU= (RANDB-RANDC)/8.925

0002500 C

NV= (RANDA-RANDB)/4.4625

0002600 C

NW= (RANDA+RANDC)/20.576

0002700 C

0002800 C

NX= COS(PITCH)*NU + SIN(PITCH)*SIN(ROLL)*NV + SIN(PITCH)

0002900 C

& *COS(ROLL)*NW

0003000 C

0003100 C

NY= COS(ROLL)*NV - SIN(ROLL)*NW

0003200 C

0003300 C

VX=TVX+TBX+NX

0003400 C

VY=TVY+TBY+NY

0003500 C

VDOP(I)= (VX**2 + VY**2) **.50

0003600 C

0003700 C

C= VY/VX

0003800 C

ALPHAD(I)= ATAN(C)

0003900 C

0004000 C

0004100 C

CALL NDRAN(IY4,STDD,0.,RANDD)

0004200 C

0004300 C

IF(I.NE.4) BETAT=BETAT +OMEGA*.5

0004400 C

IF(BETAT. GT. 2*3.14159) BETAT=BETAT-2.0*3.14159

0004500 C

0004600 C

BETAD(I)= BETAT + BBETA + RANDD

0004700 C

0004800 C

10 CONTINUE

0004900 C

0005000 C

& ** CORRECT FOR POSSIBLE ROUND-OFF ERROR **

0005100 C

0005200 C

NTURN1=TURN1

0005300 C

IF(OMEGA.NE.0.0) BETAT= STBETA +OMEGA*FLOAT(NN-NTURN1)

0005400 C

IF(BETAT. GT. 2*3.14159) BETAT=BETAT- (IFIX(BETAT/(2.0*3.14159)))*2.0*3.14159

0005500 C

0005600 C

0005700

0005800

0005900 C

EOF

RETURN

END

```

REKEYED
0000100 SUBROUTINE FISUB(A5,A6,A7,A8,PHI,GAMMA,X)
0000200 C
0000300 REAL PHI(10,10),GAMMA(10,10),X(10)
0000400 REAL TERM1(25),TERM2(25),TERM3(25),TERM4(25)
0000500 REAL VDOP(21),ALPHAD(21),BETAD(21),BV,BALPHA,BBETA
0000600 COMMON/COM1/A1,A2,A3,A4
0000700 C
0000800 COMMON/COM4/NX,NVLF
0000900 C
0001000 PI=3.14159
0001100 C INITIALIZE PHI AND GAMMA MATRICES
0001200 C
0001300 DO 5 I=1,NX
0001400 DO 5 J=1,NX
0001500 PHI(I,J)=0.D0
0001600 5 CONTINUE
0001700 C
0001800 DO 6 I=1,NX
0001900 DO 6 J=1,NX
0002000 GAMMA(I,J)=0.D00
0002100 6 CONTINUE
0002200 C
0002300 C DEFINE THE STATE TRANSITION MATRIX
0002400 C
0002500 DO 20 I=1,NX
0002600 PHI(I,I)=1.
0002700 20 CONTINUE
0002800 C
0002900 PHI(1,3)=-A3*SIN(X(4))-A4*COS(X(4))
0003000 PHI(1,4)=(A1-A3*X(3))*COS(X(4))-(A2-A4*X(3))*SIN(X(4))
0003100 PHI(2,1)=-PHI(1,4)*PI/180.*SIN(X(1)*PI/180.)/(COS(X(1)*PI/180.
0003200 & ))*2
0003300 PHI(2,3)=-(-A3*COS(X(4))+A4*SIN(X(4)))/COS(X(1)*PI/180.)
0003400 PHI(2,4)=-(-(A1-A3*X(3))*SIN(X(4))+(A4*X(3)-A2)*COS(X(4)))/
0003500 COS(X(1)*PI/180.)
0003600 C
0003700 C DEFINE GAMMA MATRIX
0003800 C
0003900 GAMMA(1,1)=A6
0004000 GAMMA(1,2)=A5
0004100 GAMMA(2,1)=A8
0004200 GAMMA(2,2)=A7
0004300 C
0004400 DO 30 I=3,10
0004500 GAMMA(I,I)=1.0
0004600 30 CONTINUE
0004700 C
0004800 RETURN
0004900 END
EDF

```



```

0000100      SUBROUTINE FSUB11(F)
0000200 C
0000300 C      FSUB11: ROUTINE TO CALCULATE ELEMENTS OF F VECTOR USED
0000400 C      IN THE PLANT UPDATE EQUATIONS
0000500 C
0000600      REAL F(10),X(10)
0000700      COMMON/COM4/NX,NVLF
0000800      COMMON/COM1/A1,A2,A3,A4
0000900      COMMON/COM3/X
0001000 C
0001100      F(1)=X(1)+(A1-A3*X(3))*SIN(X(4))
0001200      + (A2-A4*X(3))*COS(X(4))
0001300      B=COS(X(1))*3.14159/180.)
0001400      F(2)=X(2)-(A1-A3*X(3))*COS(X(4))/B
0001500      - (A4*X(3)-A2)*SIN(X(4))/B
0001600 C
0001700 C
0001800      DO 10 I=3,NX
0001900      F(I)=X(I)
0002000 10      CONTINUE
0002100 C
0002200 C
0002300 C
0002400      RETURN
0002500      END
0002600 C
EOF

```

```

0000100 C
0000200 FUNCTION GI(LAT, LONG, STNID, IFLAG, CV, WL, NN)
0000300 C
0000400 C      DETERMINES THEORETICAL USEC DIST. (VELOCITY CORRECTED)
0000500 C      FROM LAT/LONG POSITION TO STNID VLF STATION
0000600 C
0000700 C      IFLAG=1 FOR USEC DISTANCE
0000800 C      IFLAG=1 FOR PARTIAL GI WRT LAT
0000900 C      IFLAG=2 FOR PARTIAL GI WRT LONG
0001000 C
0001100 C      IFLAG=1,2 FOR H MATRIX CALCULATIONS
0001200 C
0001300 C
0001400 C      INTEGER STNID, LE
0001500 C      REAL STN(3), C(3), DE(3), E(8), F(10), CV, WL, DIST, LAT, LONG
0001600 C
0001700 C      A5=72.14772853
0001800 C      F(1)=71.90586
0001900 C      A10=(A5*A5-F(1)*F(1))/(A5*A5)
0002000 C      A14=1.0-F(1)/A5
0002100 C      F(2)=(1.0+A14+A14*A14)
0002200 C      F(3)=(F(2)-1.0)
0002300 C      F(4)=(A14*A14)/2.
0002400 C      F(5)=F(3)/2.
0002500 C      F(6)=(A14*A14)/16.
0002600 C      F(7)=(A14*A14)/8.
0002700 C      F(8)=A14*A14
0002800 C      F(9)=F(8)*1.25
0002900 C      F(10)=F(8)/4.
0003000 C
0003100 C      FIND AND READ STATION DATA FOR THE STN TO BE USED
0003200 C
0003300 C      REWIND 60
0003400 C
0003500 C 6 CONTINUE
0003600 C      READ(60,7,END=120) LE
0003700 C 7 FORMAT(A1)
0003800 C      IF(LE.EQ.STNID) GO TO 8
0003900 C      READ(60,10) IDEG, IMIN, SEC, IDEG1, IMIN1, SEC1, CV
0004000 C 10 FORMAT(2(I4,1X,I2,1X,F4.1,1X),F6.4,1X,F7.4)
0004100 C      GO TO 6
0004200 C
0004300 C
0004400 C 8 CONTINUE
0004500 C      READ(60,10) IDEG, IMIN, SEC, IDEG1, IMIN1, SEC1, CV, WL
0004600 C
0004700 C      E(1)=DEGRAD(IDEG, IMIN, SEC, 1)
0004800 C      STN(1)=DEGRAD(IDEG1, IMIN1, SEC1, 1)
0004900 C
0005000 C ***** CALCULATE PARAMETRIC LATITUDE OF STATIONS *****
0005100 C      E(3)=SIN(E(1))
0005200 C      E(4)=COS(E(1))
0005300 C      E(5)=E(3)/E(4)
0005400 C      E(8)=(E(5))*(1.0-A14)
0005500 C      A62=ATAN(E(8))
0005600 C      STN(2)=SIN(A62)
0005700 C      STN(3)=COS(A62)

```

```

0005800 C
0005900 C      START POSITION INFORMATION:
0006000 C
0006100      E(1)= LAT*3.14159/180.
0006200      E(2)= LONG*3.14159/180.
0006300      E(3)=SIN(E(1))
0006400      E(4)=COS(E(1))
0006500      E(5)=E(3)/E(4)
0006600      E(8)=(E(5))*(1.0-A14)
0006700      A62=ATAN(E(8))
0006800      E(6)=SIN(A62)
0006900      E(7)=COS(A62)
0007000 C
0007100 C
0007200      CALL INVER(STN,E,F,C(1),C(2),C(3),A35)
0007300 C
0007400      DIST=C(3)*CV*1.00260844/0.0034
0007500      GI=DIST
0007600      IF(IFLAG.EQ.0) RETURN
0007700 C
0007800 C
0007900      AZS=ATAN2(-C(1),-C(2))
0008000      IF(AZS.LT.0.0) AZS=AZS+6.283185
0008100      IF(IFLAG.EQ.2) GO TO 15
0008200      GI=60*COS(AZS)/(CV*0.16177182)
0008300      RETURN
0008400 C
0008500 15 CONTINUE
0008600      GI=60*SIN(AZS)/(CV*0.16177182)
0008700 C
0008800 C
0008900 C
0009000      RETURN
0009100 C
0009200 C
0009300 120 CONTINUE
0009400      WRITE(98,220) STNID
0009500 220 FORMAT(' 5X, **** VLF STATION ',A3,' DOES NOT EXIST:INPUT CORRECT',
0009600      ' VLF STATION. ENTER AS A ')
0009700 C
0009800      READ(63,600) STNID
0009900 600 FORMAT(A1)
0010000      REWIND 60
0010100      GO TO 6
0010200      END
0010300 C
EOF

```

```

0000100      SUBROUTINE HSUB11(LAT, LONG, STNID, H, NN)
0000200 C
0000300 C      HSUB11: ROUTINE TO CALCULATE H MATRIX ELEMENTS USED IN
0000400 C      KALMAN FILTER ROUTINE
0000500 C
0000600      REAL H(6,10)
0000700      REAL LAT, LONG
0000800      INTEGER STNID(6)
0000900      COMMON/COM4/NX, NVLF
0001000 C
0001100      DO 5 I=1, NVLF
0001200      DO 5 J=1, NX
0001300      H(I, J)=0.
0001400 5    CONTINUE
0001500 C
0001600 C
0001700      DO 15 I=1, NVLF
0001800      H(I, I+4)=1.
0001900 15    CONTINUE
0002000 C
0002100 C
0002200      DO 10 I=1, NVLF
0002300      H(I, 1) = -(G1(LAT, LONG, STNID(I), 1, CV, WL, NN))
0002400      H(I, 2) = (G1(LAT, LONG, STNID(I), 2, CV, WL, NN)) * COS(LAT*3.14159/180.)
0002500 C
0002600 10    CONTINUE
0002700 C
0002800 C
0002900      RETURN
0003000      END
EDF

```

```

0000100          SUBROUTINE K1SUB1(M1,H,R,K1,NN,NMAT)
0000200 C
0000300 C          *K1SUB1: ROUTINE TO DETERMINE KALMAN FILTER GAIN MATRIX
0000400 C
0000500 C
0000600          REAL M1(10,10),H(6,10),R(6,6),K1(10,6),A(10,10),B(10,6),C(6,6)
0000700          REAL DTEMP(6,6),DORIG(6,6)
0000800 C
0000900          COMMON/CON4/NX,NVLF
0001000 C
0001100          CALL MTPRDA(M1,H,NX,NX,NVLF,10,10,6,B)
0001200 C
0001300          CALL EMULTM(H,B,C,NVLF,NVLF,NX,6,10,6)
0001400 C
0001500          CALL MADD(C,R,NVLF,NVLF,6,6,DORIG)
0001600 C
0001700          DO 5 I=1,6
0001800          DO 5 J=1,6
0001900          DTEMP(I,J)=DORIG(I,J)
0002000 5 CONTINUE
0002100 C
0002200 C
0002300          PREC= 5.0E-6
0002400 C
0002500 C
0002600 6 CALL INVERT(DORIG,DTEMP,6,NVLF,NVLF,PREC,DET,TEST)
0002700          IF(TEST.EQ.0.0) GO TO 7
0002800          PREC=2.0*PREC
0002900 C
0003000          DO 8 I=1,6
0003100          DO 8 J=1,6
0003200          D(I,J)=DORIG(I,J)
0003300 8 CONTINUE
0003400          GO TO 6
0003500 C
0003600 7 CONTINUE
0003700 C
0003800          CALL EMULTM(B,DTEMP,K1,NX,NVLF,NVLF,10,6,10)
0003900 C
0004000          IF(NMAT.EQ.0) GO TO 50
0004100 C
0004200 C
0004300 C
0004400 50 CONTINUE
0004500          RETURN
0004600          END
0004700 C
0004800 C
EOF

```

```

0000100      SUBROUTINE MDUTS(A,IDA,MA,NA,DSRN)
0000200 C
0000300 C      ADOPTED FROM UNIVERSITY OF NEW BRUNSWICK FTN LIBRARY
0000400 C      SOURCE: D.G. MACNEIL , UNB COMPUTING CENTRE.
0000500 C
0000600 C
0000700 C      *** SUBROUTINE TO PRINT MATRICES ***
0000800 C
0000900      REAL A(IDA,NA)
0001000      INTEGER DSRN
0001100      LOGICAL FORCE
0001200      REAL*8 FORM(12) / ('1X,I4, ' ',11X' ' /,END/'1X) ' ',
0001300      & E/'1PE12.4,' /,FULL/'E15.7, ' /,F/'0PF12.5,' /,
0001400      & ZERO/'0PF12.0,' /
0001500 C
0001600      INTEGER CMIN,CMAX,ROW,INDEX,0/
0001700      INDEX=INDEX+1
0001800      WRITE(DSRN,11)
0001900 11 FORMAT(' ')
0002000      IF(MA.LT.0) GO TO 70
0002100      M=MA
0002200      NUMC=10
0002300      FORCE=.FALSE.
0002400      GO TO 71
0002500 70 CONTINUE
0002600      NUMC=8
0002700      M=-MA
0002800      FORCE=.TRUE.
0002900 C
0003000 71 CONTINUE
0003100      DO 50 CMIN=1,NA,NUMC
0003200      CMAX=CMIN+(NUMC-1)
0003300      IF(CMAX.GE.NA) CMAX=NA
0003400      IF (FORCE) GO TO 31
0003500      WRITE(DSRN,1) INDEX,(I,I=CMIN,CMAX)
0003600      GO TO 33
0003700 31 CONTINUE
0003800      WRITE(DSRN,3) INDEX,(I,I=CMIN,CMAX)
0003900 C
0004000 33 CONTINUE
0004100      DO 60 ROW=1,M
0004200      IF(((ROW-1)/10)*10.EQ.(ROW-1)) WRITE(DSRN,2)
0004300 32 CONTINUE
0004400      IFORM=1
0004500      DO 5 I=CMIN,CMAX
0004600      IFORM=IFORM+1
0004700      IF(FORCE) GO TO 20
0004800      FORM(IFORM)=E
0004900      TEST=ABS(A(ROW,I))
0005000      IF((TEST.GE.0.1).AND.(TEST.LT.10000.)) FORM(IFORM)=F
0005100      IF (TEST.EQ.0.0) FORM(IFORM)=ZERO
0005200      GO TO 5
0005300 20 CONTINUE
0005400      FORM(IFORM)=FULL
0005500 C
0005600 5 CONTINUE
0005700      FORM(IFORM+1)=END

```

```
0005800 C
0005900 WRITE(DSRN,FORM) ROW,(A(ROW,I),I=CMIN,CMAX)
0006000 60 CONTINUE
0006100 50 CONTINUE
0006200 WRITE(DSRN,2)
0006300 RETURN
0006400 1 FORMAT('0(',I4,',',10(4X,I4,4X))
0006500 2 FORMAT(' ')
0006600 3 FORMAT('0(',I4,',',10(4X,I4,7X))
0006700 END
EDF
```

```
0000100      SUBROUTINE MADD(A,B,N,M,IN,IM,C)
0000200 C
0000300 C          ADD TWO MATRICES: A(N,M) AND B(N,M)
0000400 C
0000500      REAL A(IN,IM),B(IN,IM),C(IN,IM)
0000600 C
0000700      DO 10 I=1,N
0000800      DO 10 J=1,M
0000900          C(I,J)=A(I,J)+B(I,J)
0001000 10      CONTINUE
0001100 C
0001200      RETURN
0001300      END
0001400 C
EOF
```



```

0000100 SUBROUTINE MISUB1(PHI,P,GAMMA,Q,M1,S)
0000200 C
0000300 C MISUB1: ROUTINE TO CALCULATE A PRIORI ERROR COVARIANCE MATRIX (P')
0000400 C
0000500 C
0000600 REAL PHI(10,10),P(10,10),GAMMA(10,10),Q(10,10),M1(10,10)
0000700 REAL A(10,10),B(10,10),C(10,10)
0000800 C
0000900 COMMON/COM4/NX,NULF
0001000 C
0001100 C
0001200 CALL EMULTM(PHI,P,A,NX,NX,NX,10,10,10)
0001300 CALL MTPRDA(A,PHI,NX,NX,NX,10,10,10,C)
0001400 C
0001500 C ** EXP. DATA WEIGHTING **
0001600 C
0001700 DO 10 I=1,10
0001800 DO 10 J=1,10
0001900 C(I,J)=S * C(I,J)
0002000 10 CONTINUE
0002100 C
0002200 CALL EMULTM(GAMMA,Q,B,NX,NX,NX,10,10,10)
0002300 CALL MTPRDA(B,GAMMA,NX,NX,NX,10,10,10,A)
0002400 CALL MADD(C,A,NX,NX,10,10,M1)
0002500 C
0002600 RETURN
0002700 END
0002800 C
EOF

```

0000100 SUBROUTINE PBIAS(VBST,VBET,VBINC,KVBIAS,HBST,HBET,HBINC,KHBIAS,IPSMRT)
0000200 C
0000300 C ** SUBROUTINE TO INPUT PARAMETERS FOR BIAS ON VELOCITY AND HEADING DATA**
0000400 C
0000500 INTEGER VBST,VBET,HBST,HBET
0000600 REAL VBINC,KVBIAS,HBINC,KHBIAS
0000700 C
0000800 WRITE(90,900)
0000900 900 FORMAT(' ', 'ENTER VELOCITY BIAS DRIFT DATA',/,2X,'START END KTS',/,2X,
0001000 & 'IIII IIII FFF,FF')
0001100 READ(68,600) VBST,VBET,VBINC
0001200 600 FORMAT(I4,2X,I4,2X,F6.2)
0001300 KVBIAS= (VBINC/3600.0)/(VBET-VBST)
0001400 C
0001500 WRITE(90,901)
0001600 901 FORMAT(' ', 'ENTER HEADING BIAS DRIFT DATA',/,2X,'IIII IIII FFF,FF')
0001700 READ(68,600) HBST,HBET,HBINC
0001800 KHBIAS= HBINC*0.017453/(HBET-HBST)
0001900 C
0002000 WRITE(90,902)
0002100 902 FORMAT(' ', 'DO YOU WANT P MATRIX RESET FOR PLANT BIAS DRIFT?')
0002200 READ(68,601) IPSMRT
0002300 601 FORMAT(I1)
0002400 C
0002500 RETURN
0002600 END
EDF

44
OF / DE

```
0000100      SUBROUTINE PLOT1(IDNO)
0000200 C
0000300      INTEGER IDNO
0000400      WRITE(90,900)
0000500 900  FORMAT(' ',2X,'RUN ID?',/,2X,'AAAA')
0000600      READ(68,600) IDNO
0000700 600  FORMAT(A4)
0000800      WRITE(90,901) IDNO
0000900 901  FORMAT(' ',2X,'THIS RUN ID IS',2X,A4)
0001000      RETURN
0001100      END
EOF
```

```
0000100 SUBROUTINE PSUB11(M1,K1,H,P)
0000200 C
0000300 C PSUB11: ROUTINE TO DETERMINE A POSTERIORI ERROR COVARIANCE MATRIX **
0000400 C
0000500 C
0000600 REAL M1(10,10),K1(10,6),H(6,10),P(10,10),A(10,10),B(10,10)
0000700 COMMON/COM4/NX,NVLF
0000800 C
0000900 C
0001000 CALL EMULTM(K1,H,A,NX,NX,NVLF,10,6,10)
0001100 C
0001200 CALL EMULTM(A,M1,B,NX,NX,NX,10,10,10)
0001300 C
0001400 C
0001500 DO 10 I=1,NX
0001600 DO 10 J=1,NX
0001700 B(I,J)=-B(I,J)
0001800 10 CONTINUE
0001900 C
0002000 C
0002100 CALL MADD(M1,B,NX,NX,10,10,P)
0002200 C
0002300 RETURN
0002400 END
0002500 C
EDF
```

```
0000100      SUBROUTINE STAT(DTA,ISAMP,ICOL,X,S)
0000200 C
0000300      REAL SUM,DTA(30,6),X,S
0000400 C,
0000500      SUM=0.0
0000600      DO 5 J=1,ISAMP
0000700      SUM=SUM+DTA(J,ICOL)
0000800 5  CONTINUE
0000900 C
0001000      X= SUM/ISAMP
0001100 C
0001200      SUM=0.0
0001300      DO 10 J=1,ISAMP
0001400      SUM=SUM+(DTA(J,ICOL)-X)**2
0001500 10 CONTINUE
0001600 C
0001700      S= SQRT(SUM/(ISAMP-1))
0001800      RETURN
0001900      END
EDF
```

```

0000100 C
0000200 SUBROUTINE TRU(LAT, LONG, HEADNG, GS, TIME, OMEGA, RADIUS, STHDG, NN, TURN1)
0000300 C
0000400 C THIS SUBROUTINE DETERMINES THE FINAL POSITION (LAT, LONG), GIVEN
0000500 C INITIAL POSITION, GROUND SPEED, HEADING, AND THE TIME OF THE FLIGHT
0000600 C
0000700 C VARIABLES USED:
0000800 C LAT: ON INPUT THE INITIAL LATITUDE
0000900 C ON OUTPUT THE NEW (FINAL) LAT.
0001000 C LONG: INITIAL LONGITUDE; NEW LONG.
0001100 C HEADNG: HEADING WRT TRUE NORTH (RADIAN)
0001200 C GS: GROUND SPEED IN NM/SEC
0001300 C TIME: FLYING TIME FROM INITIAL POSITION IN SECS.
0001400 C DLAT; DLONG : DISTANCES IN NM FLOWN IN LAT. AND LONG
0001500 C DIRECTIONS
0001600 C PLAT; PLONG : CHANGES IN LAT. AND LONG. (I.E. CHANGES IN DEGREES)
0001700 C
0001800 C FOR THE INITIAL POSITION THE FOLLOWING CONVENTION IS USED
0001900 C N LAT. : ENTER AS A '+' VALUE
0002000 C S LAT : '-'
0002100 C W 'E' : '-'
0002200 C
0002300 C
0002400 REAL LAT, LONG, HEADNG, GS, TIME
0002500 REAL*8 DLAT, DLONG, PLAT, PLONG
0002600 C
0002700 C CALCULATE THE CHANGES IN DISTANCES
0002800 C
0002900 C *** OPTION FOR CIRCULAR TRACK ***
0003000 C
0003100 IF (OMEGA.EQ.0.0) GO TO 10
0003200 THETA= OMEGA*TIME
0003300 DLAT= 2*RADIUS* SIN(THETA/2.0)* COS(HEADNG+THETA/2.0)
0003400 C
0003500 DLONG= 2.0*RADIUS*SIN(THETA/2.0)* SIN(HEADNG+THETA/2.0)
0003600 NTURN1=TURN1
0003700 IF (OMEGA.NE.0.0) HEADNG= STHDG + OMEGA*FLOAT(NN-NTURN1)
0003800 IF (HEADNG.GT.2.0*3.14159) HEADNG= HEADNG- (IFIX(HEADNG/(2.0*3.14159)))*2.0*3.14159
0003900 GO TO 15
0004000 10 CONTINUE
0004100 C
0004200 DLAT= DBLE(GS)*DCOS(DBLE(HEADNG))*DBLE(TIME)
0004300 DLONG= DBLE(GS)*DSIN(DBLE(HEADNG))*DBLE(TIME)
0004400 C
0004500 15 CONTINUE
0004600 C CALCULATE THE CHANGES IN LAT. AND LONG.
0004700 C
0004800 C
0004900 PLAT=DLAT*6077.D00/(60.D00*(6077.D00-31.D00*DCOS(2.D00*DBLE(LAT))*
0005000 & 3.14159265D00/180.D00)))
0005100 C
0005200 PLONG=DLONG*6077.D00/(60.D00*(6077.D00-31.D00*DCOS(2.D00*DBLE(LAT))*
0005300 & 3.14159265D00/180.D00))*DCOS(DBLE(LAT)*3.14159265D00/180.D00))
0005400 C
0005500 C
0005600 LONG=DBLE(LONG)-PLONG
0005700 LAT=DBLE(LAT)+PLAT

```

0005800 IF(LAT.GT.90) LAT=180-LAT
0005900 IF(LAT.LT.-90) LAT=-180-LAT
0006000 IF(LONG.GT.180) LONG=LONG-360.
0006100 IF(LONG.LT.-180) LONG=LONG+180.
0006200 RETURN
0006300 END.
0006400 C
0006500 C

OF


```

0000100 SUBROUTINE VBIAS(STNID,ISTN,VLFBS,VLFBE,DVBIAS,KBIAS,NVLF,IFLAG2)
0000200 C
0000300 C SUBROUTINE TO CALCULATE BIAS DRIFT CHARACTERISTICS FOR
0000400 C DRIFT ON VLF BIAS
0000500 C
0000600 INTEGER ISTN(6),VLFBS(6,4),VLFBE(6,4),STNID(6),IFLAG2(6)
0000700 REAL DVBIAS(6,4),KBIAS(6,4)
0000800 C
0000900 WRITE(90,900)
0001000 900 FORMAT(' ','WHAT VLF BIASES WILL DRIFT? ENTER A '1')
0001100 WRITE (90,901) (STNID(I),I=1,NVLF)
0001200 901 FORMAT(' ',1X,6A1)
0001300 C
0001400 READ(69,600) (ISTN(I),I=1,NVLF)
0001500 600 FORMAT(6I1)
0001600 WRITE(90,905)
0001700 905 FORMAT(' ',2X,'HOW MANY BIAS CHANGES PER STATION? MAX OF 4')
0001800 READ(68,603) NDRIFT
0001900 C
0002000 C
0002100 C
0002200 WRITE(90,902)
0002300 902 FORMAT(' ','ENTER BIAS DRIFT DATA',/,1X,
0002400 ' & 'START END' USEC CHANGE',/,2X,
0002500 ' & 'IIII IIII FFF.FF (VLFBS(2)=9999, FOR DEFAULT)')
0002600 C
0002700 DO 10 I=1,NVLF
0002800 DO 10 J=1,NDRIFT
0002900 C
0003000 IF(ISTN(I).EQ.0) GO TO 10
0003100 C
0003200 IF(I.GE.2.AND.J.GE.1.AND.VLFBS(2,1).EQ.9999.0) GO TO 11
0003300 C
0003400 WRITE(90,903) STNID(I),J
0003500 903 FORMAT(' ',1X,A1,1X,I1)
0003600 READ(89,601) VLFBS(I,J),VLFBE(I,J),DVBIAS(I,J)
0003700 601 FORMAT(I4,2X,I4,2X,F6.2)
0003800 GO TO 10
0003900 C
0004000 11 CONTINUE
0004100 VLFBS(I,J)=VLFBS(1,J)
0004200 VLFBE(I,J)=VLFBE(1,J)
0004300 DVBIAS(I,J)=DVBIAS(1,J)
0004400 10 CONTINUE
0004500 C
0004600 DO 20 I=1,NVLF
0004700 DO 20 J=1,NDRIFT
0004800 C
0004900 IF(ISTN(I).EQ.0) GO TO 20
0005000 IF(VLFBE(I,J).EQ.VLFBS(I,J)) GO TO 20
0005100 KBIAS(I,J)=DVBIAS(I,J)/(VLFBE(I,J)-VLFBS(I,J))
0005200 20 CONTINUE
0005300 C
0005400 C
0005500 C
0005600 WRITE(90,904)
0005700 904 FORMAT(' ',2X,'DO YOU WANT SMART FILTER?')

```

```
0005800      READ(68,603) ISMRT
0005900 603  FORMAT(I1)
0006000 C
0006100      IF(ISMRT.EQ.1) GO TO 40
0006200      DO 30 I=1,6
0006300      IFLAG2(I)=1
0006400 30  CONTINUE
0006500 C
0006600 40  CONTINUE
0006700      RETURN
0006800      END
EOF
```

```

0000100      SUBROUTINE VLFFIX(P,R,NVLF,KAQ,IVAR,A,VMEAN,VSTD,KVLF,VLFN,
0000200      &      KRES,NINC,NN,PCOEFF,TOLCF)
0000300 C
0000400      REAL P(10,10),R(6,6),A(6,1),VMEAN(6),VSTD(6),VLFN(30,6),RR(6,6)
0000500      REAL STMP(6),MTEMP(6),NCHECK,DMEAN(6),XDATA(12),YDATA(12)
0000600 C
0000700      INTEGER KAO(6),IVAR(6),NVLF,KVLF(6) ,KRES(6)
0000800 C
0000900      IFLAG=0
0001000      DO 50 I=1,NVLF
0001100 C
0001200      IF(KAO(I).EQ.0) GO TO 45
0001300      IF(IVAR(I).EQ.0) GO TO 50
0001400 C
0001500      NVLF(I)=KVLF(I) +1
0001600      KKVLF = KVLF(I)
0001700      VLFN(KKVLF,I)= A(I,1)
0001800 C
0001900 C
0002000      TBAND= VMEAN(I) + 3.0*VSTD(I)
0002100      BBAND= VMEAN(I) - 3.0*VSTD(I)
0002200      IF(VLFN(KKVLF,I).LE.TBAND .AND. VLFN(KKVLF,I).GE.BBAND) GO TO 10
0002300      KRES(I)=KRES(I)+1
0002400 C
0002500 C
0002600 C
0002700      IF(KRES(I).LT.3) GO TO 18
0002800 C
0002900      IF(VLFN(KKVLF,I).GT.TBAND) GO TO 12
0003000 C
0003100 C * RESIDUAL < MEAN-3*SIGMA
0003200 C
0003300      IF(VLFN(KKVLF-1,I).GT.BBAND.OR.VLFN(KKVLF-2,I).GT.BBAND) GO TO 11
0003400      P(I+4,I+4)=PCOEFF*(ABS(VLFN(KKVLF,I))-BBAND)
0003500 C
0003600      WRITE(90,900) I,P(I+4,I+4),NN
0003700      KRES(I)=0
0003800 C
0003900 13  KVLF(I)=0
0004000      GO TO 40
0004100 C
0004200 12  CONTINUE
0004300 C
0004400 C * RESIDUAL > MEAN+3*SIGMA
0004500 C
0004600      IF(VLFN(KKVLF-1,I).LT.TBAND.OR.VLFN(KKVLF-2,I).LT.TBAND) GO TO 11
0004700      P(I+4,I+4)=PCOEFF*(VLFN(KKVLF,I)-TBAND)
0004800 C
0004900      WRITE(90,900) I,P(I+4,I+4),NN
0005000      KRES(I)=0
0005100 C
0005200 14  KVLF(I)=0
0005300      GO TO 40
0005400 C
0005500 11  CONTINUE
0005600 C
0005700      IF(KVLF(I).EQ.11.OR.KVLF(I).EQ.12) GO TO 17
0005800      IF(R(I,I).EQ.10000.0) GO TO 17

```

```

0005900      R(I,I)=10000.0
0006000 C
0006100      WRITE(90,901) I,I,NN
0006200 901  FORMAT(' ',2X,'R(' ',I1,I1,' ') SET TO 10,000.0 AT',2X,I5,' SECS')
0006300 C
0006400      KLVLF=1
0006500      KLVF(I)=1
0006600      VLFN(I,I)=A(I,1)
0006700      GO TO 17
0006800 C
0006900 10  CONTINUE
0007000      KRES(I)=0
0007100 C
0007200 18  IF(R(I,I).NE.10000.0) GO TO 17
0007300      R(I,I)=2.0*VSTD(I)**2.0
0007400      WRITE(90,902) I,I,NN
0007500 17  CONTINUE
0007600 C
0007700 C
0007800      IF(KLVF(I).LT.12) GO TO 50
0007900 C
0008000 C-----
0008100 C
0008200      DO 15 J=1,12
0008300      YDATA(J)=VLFN(J,I)
0008400      XDATA(J)=FLOAT(NN)-FLOAT((12-J)*NINC)
0008500 15  CONTINUE
0008600 C
0008700      XMEAN=(XDATA(12)-XDATA(1))/2.0+ XDATA(1)
0008800      CALL LINREG(XDATA,YDATA,12,XMEAN(I),VSTD(I),SXX,SXY,SYX,B0,B1)
0008900 C
0009000      VTOL=VSTD(I)*TOLCF*SQRT(1.0/12.0 + (XDATA(12)-XMEAN)**2.0
0009100      &      /((SXX/12.0))
0009200      Y12= B0 + B1*(XDATA(12)-XDATA(1))
0009300 C
0009400      IF(ABS(Y12).LE.VTOL) GO TO 40
0009500 C
0009600      KLVF(I)=0
0009700      KRES(I)=0
0009800 C
0009900 30  CONTINUE
0010000      P(I+4,I+4)= PCOEFF*(ABS(Y12)-VTOL)**2.0
0010100 C
0010200      WRITE(90,900) I,P(I+4,I+4),NN
0010300 900  FORMAT(' ',2X,'VLF STN',2X,I2,' P MAT ELEMENT SET TO',
0010400      &      2X,F10.3,' AT',I5,' SECS')
0010500 C
0010600 C-----
0010700 40  CONTINUE
0010800      IF(R(I,I).EQ.10000.0) WRITE(90,902)I,I,NN
0010900 902  FORMAT(' ',2X,'R(' ',I1,I2,' ') RESET AT',2X,I5,' SECS')
0011000      R(I,I)=2.0*VSTD(I)**2.0
0011100      IF(KLVF(I).EQ.12) KLVF(I)=0
0011200      GO TO 50
0011300 C
0011400 45  CONTINUE
0011500      R(I,I)=10000.0
0011600 C
0011700      WRITE(90,904) I,NN,R(I,I),

```

0011800 904 FORMAT(' ,2X,'** STN',I2,' WEIGHTED OUT AT',I5,' SECS, R ELEMENT SET TO',2X,F8.2)
0011900 C
0012000 C
0012100 C
0012200 50 CONTINUE
0012300 C
0012400 RETURN
0012500 END
EDF

```

0000100 C
0000200 SUBROUTINE VLF$IM(IY,BAT, LONG,STNID,OBS,BP,VARP,RAND,BIY,VARB,RANDB,NBPER,VBLEV,NN,KAQ)
0000300 C
0000400 C VLSIM: VLF SIMULATOR
0000500 C
0000600 C
0000700 REAL LAT, LONG, OBS(6,1), BP(6), VARP(10), RAND(10), VLFSD(10), RANDB(6), VBLEV(6)
0000800 INTEGER STNID(6), IY(6), KAQ(6), BIY(6), NBPER(6)
0000900 COMMON/COM4/NX,NVLF
0001000 C
0001100 DO 10 I=1,NVLF
0001200 VLFSD(I)=SQRT(VARP(I))
0001300 CALL NDRAN(IY(I),VLFSD(I),0.,RAND(I+4))
0001400 10 CONTINUE
0001500 C
0001600 BSD=SQRT(VARB)
0001700 C
0001800 DO 15 I=1,NVLF
0001900 IF(NBPER(I).GT.0) GO TO 16
0002000 CALL NDRAN(BIY(I),BSD,0.0,RANDB(I))
0002100 GO TO 15
0002200 16 CONTINUE
0002300 IF(NN/NBPER(I)*NBPER(I).EQ.NN .AND. NN.NE.0) RANDB(I)=-RANDB(I)
0002400 15 CONTINUE
0002500 C
0002600 C
0002700 C
0002800 DO 20 I=1,NVLF
0002900 IF(NBPER(I).EQ.0) GO TO 21
0003000 IF(NN/NBPER(I)*NBPER(I).EQ.NN .AND. NN.NE.0) BP(I)=BP(I)+RANDB(I)
0003100 21 OBS(I,1)=GI(LAT, LONG,STNID(I),0,CU,WL)+BP(I)+RAND(I+4)
0003200 KAQ(I)=1
0003300 20 CONTINUE
0003400 C
0003500 RETURN
0003600 END
0003700 C
EOF

```

```

0000100      SUBROUTINE VNOISE(STNID,NVLF,IVLFN,VLFVT,VARPN,IDNO)
0000200 C
0000300 C      DETERMINES VLF STATION NOISE CHANGES
0000400 C
0000500      REAL VARPN(6)
0000600      INTEGER STNID(6),NVLF,IVLFN(6),VLFVT(6)
0000700 C
0000800      WRITE(90,900)
0000900 900  FORMAT(' ',2X,'WHAT VLF PHASE VARIANCES WILL CHANGE')
0001000      WRITE(90,901) (STNID(I),I=1,NVLF)
0001100 901  FORMAT(' ',1X,6A1)
0001200      READ(68,600) (IVLFN(I),I=1,NVLF)
0001300 600  FORMAT(6I1)
0001400 C
0001500      WRITE(90,902)
0001600 902  FORMAT(' ',2X,'ENTER VARIANCE DATA',/,1X,'TIME NEW VARIANCE',/,2X,
0001700      '      'III FFFFFF.FF')
0001800 C
0001900      DO 10 I=1,NVLF
0002000      IF (IVLFN(I).EQ.0) GO TO 10
0002100      WRITE(90,903) STNID(I)
0002200 903  FORMAT(' ',1X,A1)
0002300      READ(68,601) VLFVT(I),VARPN(I)
0002400 601  FORMAT(I4,3X,F9.2)
0002500 10  CONTINUE
0002600 C
0002700 C
0002800      WRITE(94,904) IDNO
0002900 904  FORMAT(' ',A4)
0003000 C
0003100      RETURN
0003200      END
0003300 C
EDF

```

```

0000100      SUBROUTINE INVER(CC,E,F,A44,A45,A46,A35)
0000200 C
0000300 C      SUBROUTINE INVER:  SODANO INVERSE ALGORITHM TO DETERMINE
0000400 C *      GEODESIC DISTANCES BETWEEN TWO POINTS
0000500 C *      ON THE EARTHS SURFACE **
0000600 C
0000700      DIMENSION CC(3),E(8),F(10)
0000800      PI=3.14159
0000900      A5=72.14772853
0001000      A14=1.0-F(1)/A5
0001100 500      A59=-CC(1)
0001200      A60=-E(2)
0001300      C35=A59-A60
0001400      C36=ABS(C35)
0001500      IF (C36-PI)501,502,502
0001600 502      A16=2.0*PI-C36
0001700      GO TO 505
0001800 501      A16=C36
0001900 505      IF (A16)506,507,506
0002000 507      A16=0.00000005
0002100 506      A17=SIN(A16)
0002200      A18=COS(A16)
0002300      A19=E(6)*CC(2)
0002400      A20=E(7)*CC(3)
0002500      A21=A19+A20*A18
0002600      A22=SQRT((A17*CC(3))**2 + (CC(2)*E(7)-E(6)*CC(3)*A18)**2)
0002700      A23=(A20*A17)/A22
0002800      A24=1.0-A23*A23
0002900      A25=ARCSIN(A22)
0003000      IF (A21)503,504,504
0003100 503      A25=PI-A25
0003200 504      A26=A25*A25
0003300      A27=1.0/A22
0003400      A28=A21/A22
0003500      A29=A24*A24
0003600      A30=(F(2)*A25)+A19*(F(3)*A22-F(4)*A26*A27)
0003700      A31=A24*(F(5)*A25+F(5)*A22*A21+F(4)*A26*A28)
0003800      A32=A19*A19*(-F(4)*A21*A22)
0003900      A33=A29*(F(6)*A25+F(6)*A21*A22-F(4)*A26*A28-F(7)*A22*(A21**3))
0004000      A34=A19*A24*(F(4)*A26*A27+F(4)*A22*A21*A21)
0004100      A46=(A30+A31+A32+A33+A34)*F(1)
0004200      A36=(F(3)*A25)+A19*(-F(4)*A22-A14*A14*A26*A27)
0004300      A37=A24*(-F(9)*A25+F(10)*A22*A21+A14*A14*A26*A28)
0004400      A38=(A36+A37)*A23+A16
0004500      A39=SIN(A38)
0004600      A40=COS(A38)
0004700      A41=(CC(2)*E(7)-A40*E(6)*CC(3))/(A39*CC(3))
0004800      IF (A41)510,509,510
0004900 509      A41=0.00000005
0005000 510      A42=1.0/A41
0005100      A43=ATAN(A42)
0005200      IF (C35) 515,514,514
0005300 514      IF (C35-PI)511,512,512
0005400 511      IF (A41)520,521,521
0005500 520      A43=PI+A43
0005600      GO TO 521
0005700 512      IF (A41)517,518,518

```



```
0005800 515 IF (C35+PI)511,511,516
0005900 516 IF (A41)517,518,518
0006000 517 A43=PI-A43
0006100      GO TO 521
0006200 518 A43=2.0*PI-A43
0006300 521 A43=A43+PI
0006400      A43=A43-2.0*PI
0006500      IF (A43)522,523,523
0006600 522 A43=A43+2.0*PI
0006700 523 A44=SIN(A43)
0006800      A45=COS(A43)
0006900      A35=A46*299.6929037
0007000      RETURN
0007100      END
0007200 C
EOF
```

```

0000100      SUBROUTINE INVERT (XORIG,X,IX,M,N,PREC,DET,TEST)
0000200 C
0000300 C          *** INVERTS A MATRIX ***
0000400 C
0000500      INTEGER*2 I1(255), I2(255)
0000600      DIMENSION X(IX,N),XORIG(M,N)
0000700      EQUIVALENCE (BIG,TEMP,PIV)
0000800 C
0000900 C      INITIALIZATION
0001000      IFLAG=0
0001100      IF (TEST.NE.0.0) IFLAG=1
0001200 C
0001300      DET = 1.
0001400      TEMP = ABS(PREC)
0001500      IF (TEMP .EQ. 0.) TEMP = 0.5E-6
0001600      TEST = 0.
0001700      DO 1 I = 1,M
0001800          I2(I) = 0
0001900      DO 1 J = 1,M
0002000 1 TEST = TEST + X(I,J) * X(I,J)
0002100      TEST = TEMP * SQRT(TEST) / M
0002200 C      IF (IFLAG.EQ.1) PAUSE 1
0002300 C
0002400 C      SEARCH FOR NEXT PIVOT, AVOIDING ROWS AND COLUMNS WHICH ALREADY
0002500 C      CONTAIN A PIVOT.
0002600 C
0002700      DO 9 K = 1,M
0002800          BIG = 0.
0002900          DO 3 I = 1,M
0003000              IF (I2(I) .NE. 0) GO TO 3
0003100          DO 2 J = 1,M
0003200              IF (I2(J) .NE. 0) GO TO 2
0003300              T = ABS(X(I,J))
0003400              IF (T .LE. BIG) GO TO 2
0003500          BIG = T
0003600          IROW = I
0003700          JCOL = J
0003800 2 CONTINUE
0003900 3 CONTINUE
0004000 C      IF (IFLAG.EQ.1) PAUSE 2
0004100      IF (BIG .LE. TEST) RETURN
0004200 C      IF (IFLAG.EQ.1) PAUSE 3
0004300 C
0004400 C      RECORD POSITION OF PIVOT THEN MOVE PIVOTAL ROW SO PIVOT LIES ON
0004500 C      THE DIAGONAL. SCALE PIVOTAL ROW.
0004600 C
0004700      I1(K) = JCOL
0004800      I2(JCOL) = IROW
0004900      IF (JCOL .EQ. IROW) GO TO 6
0005000      DO 5 J = 1,N
0005100          TEMP = X(IROW,J)
0005200          X(IROW,J) = X(JCOL,J)
0005300 5 X(JCOL,J) = TEMP
0005400      DET = -DET
0005500 6 DET = DET * X(JCOL,JCOL)
0005600      PIV = 1. / X(JCOL,JCOL)
0005700      DO 7 J = 1,N

```

```

0005800      IF (J.NE. JCOL) X(JCOL,J) = PIV * X(JCOL,J)
0005900      7 CONTINUE
0006000 C      IF(IFLAG.EQ.1) PAUSE 4
0006100      X(JCOL,JCOL) = PIV
0006200 C
0006300 C      ELIMINATE REMAINING ROWS.
0006400 C
0006500      DO 9 I = 1,M
0006600      IF (I.EQ. JCOL) GO TO 9
0006700      PIV = -X(I,JCOL)
0006800      DO 8 J = 1,N
0006900      IF (J.NE. JCOL) X(I,J) = X(I,J) + PIV * X(JCOL,J)
0007000      8 CONTINUE
0007100      X(I,JCOL) = PIV * X(JCOL,JCOL)
0007200      9 CONTINUE
0007300 C      IF(IFLAG.EQ.1) PAUSE 5
0007400 C
0007500 C      UNSCRAMBLE THE COLUMNS OF THE INVERSE ACCORDING TO THE ORIGINAL
0007600 C      POSITIONS OF THE PIVOTS.
0007700 C
0007800      DO 11 K = 1,M
0007900      I = M+1 - K
0008000      JCOL = I1(I)
0008100      IROW = I2(JCOL)
0008200      IF (IROW.EQ. JCOL) GO TO 11
0008300      DO 10 I = 1,M
0008400      TEMP = X(I,IROW)
0008500      X(I,IROW) = X(I,JCOL)
0008600      10 X(I,JCOL) = TEMP
0008700      11 CONTINUE
0008800 C      IF(IFLAG.EQ.1) PAUSE 6
0008900      TEST = 0.
0009000      RETURN
0009100      END
EOF

```

```

0000100 SUBROUTINE LINREG(X,Y,NSAMP,YMEAN,SE,SXX,SXY,SYY,B0,B1)
0000200 C
0000300 C LINREG: LINEAR REGRESSION ROUTINE
0000400 C
0000500 C INPUT: X DATA,Y DATA, NUMBER OF SAMPLES
0000600 C OUTPUT: REGRESSION VARAINCES, REGRESSION COEFFICIENTS
0000700 C B0: INITIAL POINT OF LINEAR CURVE
0000800 C B1: SLOPE OF LINEAR CURVE
0000900 C
0001000 C
0001100 REAL X(NSAMP),Y(NSAMP)
0001200 C
0001300 CALL VSTAT(X,NSAMP,XMEAN,XSTD)
0001400 CALL VSTAT(Y,NSAMP,YMEAN,YSTD)
0001500 C
0001600 SUM1=0.0
0001700 SUM2=0.0
0001800 SUM3=0.0
0001900 C
0002000 DO 10 I=1,NSAMP
0002100 SUM1= SUM1 + (X(I)-XMEAN)*(Y(I)-YMEAN)
0002200 SUM2= SUM2 + (X(I)-XMEAN)*(X(I)-XMEAN)
0002300 SUM3= SUM3 + (Y(I)-YMEAN)*(Y(I)-YMEAN)
0002400 10 CONTINUE
0002500 C
0002600 C
0002700 SXY= NSAMP*SUM1
0002800 SXX= NSAMP*SUM2
0002900 SYY= NSAMP*SUM3
0003000 C
0003100 SVAR= (SXX*SYY - SXY*SXY)/(NSAMP*(NSAMP-2.0)*SXX)
0003200 C
0003300 SE= SQRT(SVAR)
0003400 C
0003500 B1= SUM1/SUM2
0003600 B0= YMEAN-B1*(XMEAN-X(1))
0003700 C
0003800 RETURN
0003900 END
EDF

```

```
0000100 C
0000200 SUBROUTINE MTPRDA(A,B,L,M,N,IL,IM,IN,D)
0000300 C
0000400 C
0000500 C      MULTIPLY 'A' BY THE TRANSPOSE OF 'B'
0000600 C
0000700 C      REAL A(IL,IM),B(IN,IM),C(10,10),D(IL,IN)
0000800 C
0000900 C      DO 10 I=1,M
0001000 C      DO 10 J=1,N
0001100 C      C(I,J)=B(J,I)
0001200 10 CONTINUE
0001300 C
0001400 C
0001500 C      CALL EMULTM(A,C,D,L,N,M,IL,10,IL)
0001600 C
0001700 C      RETURN
0001800 C      END
0001900 C
EOF
```

```
dl 1 111
0000100 SUBROUTINE VSTAT(DTA,ISAMP,X,S)
0000200 C
0000300 REAL SUM,DTA(ISAMP),X,S
0000400 C
0000500 SUM=0.0
0000600 DO 5 J=1,ISAMP
0000700 SUM=SUM+DTA(J)
0000800 5 CONTINUE
0000900 C
0001000 X= SUM/ISAMP
0001100 C
0001200 SUM=0.0
0001300 DO 10 J=1,ISAMP
0001400 SUM=SUM+(DTA(J)-X)**2
0001500 10 CONTINUE
0001600 C
0001700 S= SQRT(SUM/(FLOAT(ISAMP)-1.0))
0001800 RETURN
0001900 END
EOF
```

SOURCE: PREDOP

```

0000100 C ** ROUTINE TO DETERMINE DOPPLER BEAM FREQUENCY PARAMETERS FOR
0000200 C ** KALMAN FILTER INPUTS, GIVEN DOPPLER GROUND SPEED, HEADING AND
0000300 C ** DRIFT ANGLE BIASES AND VARIANCES, THIS ROUTINE WILL GIVE
0000400 C ** THE THREE BEAM FREQUENCY BIASES AND VARIANCES **
0000500 C
0000600 C
0000700 10 CONTINUE
0000800 WRITE(70,700)
0000900 700 FORMAT(' ',3X,'INPUT GS,BV,BALPHA FFF.FF FF.FF FF.FF GS=999 TO STOP')
0001000 C
0001100 READ(60,600) GS,BV,BALPHA
0001200 600 FORMAT(F6.2,1X,F5.2,1X,F5.2)
0001300 IF(GS.EQ.999) STOP
0001400 C
0001500 TVX=GS
0001600 TVY=0.0
0001700 C2= TAN(BALPHA*3.14159/180.0)
0001800 TBX= ((BV+GS)/(C2**2+1.0)**0.5) - GS
0001900 C
0002000 BFB=8.925*TBX
0002100 TBY= C2*(TVX+TBX)
0002200 BFA= 4.4625*TBY + BFB
0002300 BFC= 0.0
0002400 A=TVX+TBX
0002500 B=TVY+TBY
0002600 C= A**2 + B**2
0002700 15 CONTINUE
0002800 WRITE(70,701)
0002900 701 FORMAT(' ',3X,'INPUT VARIANCE OF VDOP,ALPHA FFF.FFFF FFF.FFFF',
0003000 & ' ',3X,'VARIANCE OF VDOP = 999. TO STOP')
0003100 READ(60,601) VARV,ALPHA
0003200 C
0003300 IF(VARV.EQ.999) STOP
0003400 601 FORMAT(F8.4,1X,F8.4)
0003500 ALPHA=ALPHA*(3.14159/180.0)**2
0003600 C
0003700 C *** CHECK IF VARIANCE IS WITHIN THEORETICAL LIMIT ***
0003800 C
0003900 CHECK1= B**2*C*ALPHA/A**2
0004000 CHECK2= (C*(A**2+4*B**2)/(4*A**2+B**2))*ALPHA
0004100 IF(VARV.GT.CHECK1 .AND. VARV.LT.CHECK2) GO TO 20
0004200 WRITE(70,705) CHECK1,CHECK2
0004300 705 FORMAT(' ',3X,'VELOCITY VARIANCE MUST BE BETWEEN',2X,F7.3,' AND',2X,F7.3)
0004400 GO TO 15
0004500 C
0004600 20 CONTINUE
0004700 C
0004800 C
0004900 VARNY= (ALPHA*C**2 - B**2/A**2*VARV*C) * A**2/(A**4-B**4)
0005000 C
0005100 VARNX= (VARV*C - B**2*VARNY)/A**2
0005200 C
0005300 VARFC = 0.0
0005400 VARFB= VARNX/0.0125
0005500 VARFA = VARNY/0.0502 - VARFB
0005600 C
0005700 WRITE(70,703) BFA,VARFA,BFB,VARFB,BFC,VARFC

```

```
0005800 703 FORMAT(' ',3X,'KNOWN BEAM FREQ QUANTITIES',//,3X,'BEAM A BIAS',
0005900      1,      2X,F9.4,2X,'VARIANCE',2X,F8.2,//,3X,'BEAM B BIAS',
0006000      2,      2X,F9.4,2X,'VARIANCE',2X,F8.2,//,3X,'BEAM C BIAS',
0006100      1,      2X,F9.4,2X,'VARIANCE',2X,F8.2)
0006200 C
0006300      WRITE(70,704)
0006400 704 FORMAT(' ',60(' '))
0006500 C
0006600 C
0006700      GO TO 15
0006800 C
0006900      END
EOF
```



```
dl 1 111
0000100 WRITE(90,900)
0000200 900 FORMAT(' ',2X,'INPUT V,T1,T2,BETA1',/,2X,'FFF.FF FFFF.FF FFFF.FF FFF.FF')
0000300 READ(60,600) V,T1,T2,BETA1
0000400 600 FORMAT(F6.2,1X,F7.2,1X,F7.2,1X,F6.2)
0000500 WRITE(90,907) V,T1,T2,BETA1
0000600 907 FORMAT(' ',2X,'V=',2X,F6.2,2X,'T1=',2X,F7.2,' T2=',2X,F7.2,' BETA1=',F6.2)
0000700 V=V/3600.0
0000800 C
0000900 5 CONTINUE
0001000 WRITE(90,901)
0001100 901 FORMAT(' ',2X,'INPUT RADIUS OF CIRCLE, FFF.FFFF')
0001200 READ(60,601) R1
0001300 601 FORMAT(F8.4)
0001400 C
0001500 R2= V*T1*TAN(V*T2/(2*R1))-1.57079)
0001600 WRITE(90,902) R1,R2
0001700 902 FORMAT(' ',2X,'R1= ',2X,F8.4,3X,'R2=',2X,F8.4)
0001800 IF( ABS(R1-R2) .LE. 0.01) GO TO 10
0001900 GO TO 5
0002000 10 CONTINUE
0002100 OMEGA= V/R1
0002200 ALPHA= OMEGA * T2 * 360/(2*3.14159) - 180.0
0002300 WRITE(90,908) ALPHA
0002400 908 FORMAT(' ',2X,'ALPHA=',2X,F7.3)
0002500 WRITE(90,903) OMEGA
0002600 903 FORMAT(' ',2X,'OMEGA=',2X,F8.5)
0002700 BETA2= BETA1 + OMEGA*T2*360.0/(2*3.14159)
0002800 WRITE(90,904) BETA2
0002900 904 FORMAT(' ',2X,'BETA2=',2X,F6.2)
0003000 STOP
0003100 END
EDF
```

```

0000100 C MAIN ROUTINE FOR KALMAN FILTER NAVIGATION SYSTEM USING
0000200 C FLIGHT DATA CONTAINED IN DATA SET DSRN 40. ORIGINAL
0000300 C ROUTINE SOURCE.KFSIM AND SOURCE.KFSIMCON.
0000400 C MODIFICATIONS FOR FLIGHT DATA ROUTINE MADE OCTOBER 1981.
0000500 C
0000600 C CRAIG MASKELL GRADUATE STUDENT CARLETON UNIVERSITY
0000700 C DEPARTMENT OF MECHANICAL AND AERONAUTICAL
0000800 C ENGINEERING, OTTAWA.
0000900 C
0001000 C PROGRAM DESCRIPTION 0001100 C THIS PROGRAMME USES A SUB-OPTIMAL KALMAN FILTERING
0001200 C TECHNIQUE TO OBTAIN ESTIMATES OF THE LAT AND LONG POSN
0001300 C OF AN AIRCRAFT WHILE ON THE GROUND OR IN THE AIR.
0001400 C
0001500 C FOR FURTHER DESCRIPTION REFER TO TEXT IN --- KFDESC.
0001600 C
0001700 C
0001800 C
0001900 C THE MATRIX PRINTING ROUTINE 'MOUTS' HAS BEEN ADAPTED
0002000 C FROM THE UNIVERSITY OF NEW BRUNSWICK LIBRARY.
0002100 C SOURCE: D.G. MACNEIL, UNB COMPUTING CENTRE
0002200 C
0002300 C
0002400 REAL LAT, LONG, VDOP(81)/81*0.0/, LATS, LONGS
0002500 REAL ALPHAD(81)/81*0.0/, BETAD(81)/81*0.0/
0002600 REAL PHI(10,10), P(10,10)/100*0.000/, GAMMA(10,10)
0002700 REAL K1(10,6), R(6,6), X(10), F(10), OBS(6,1), TDTA(6), ITDTA(6)
0002800 REAL Q(10,10)/100*0.000/, H(6,10)
0002900 REAL ULFSD(6), VECTOR(1,10), SMALLH(6,1), VARFN(6), RR(6,6)
0003000 REAL A(6,1), B(10,1), ULFN(30,6), VMEAN(6), VSTD(6)
0003100 REAL INNOV(6,1), VX(81), VY(81), DTA(30), BI(6), XTIME(30)
0003200
0003300 C
0003400 C
0003500 REAL*8 T, DDOPTIM, ULFTIM
0003600 C
0003700 INTEGER DEGS, MINS, LATD, LATM, LONGD, LONGM, KBR(6)
0003800 INTEGER STNID(6), DSRN, ISTN(6)
0003900 INTEGER IULFN(6), IVAR(6), KAD(6), KULF(6)
0004000 C
0004100 INTEGER IFLAG1(6)/6*0/, IFLAG2(6)/6*0/, IPFLAG
0004200 C
0004300 C
0004400 C
0004500 C
0004600 C*****--*--*--*****
0004700 C*****--*--*--*****
0004800 C
0004900 C
0005000 WRITE(90,900)
0005100 900 FORMAT(' ', 2X, 'INPUT RUN ID, A4')
0005200 READ(68,599) ID
0005300 599 FORMAT(A4)
0005400 WRITE(99,901) ID
0005500 901 FORMAT(' ', 2X, '** KF RUN', 2X, A4)
0005600 C
0005700 C
0005800 PI=3.14159
0005900 RAD=PI/180.

```

```

0006000 C
0006100 C
0006200 C      ****
0006300 C      *
0006400 C      * READ INITIAL DATA,
0006500 C      * USER INPUTS DATA IN USER ORIENTED
0006600 C      * QUANTITIES, I.E. KNOTS, DEGREES, ETC.
0006700 C      *
0006800 C      ****
0006900 C
0007000 C
0007100      READ(66,600) NVLF
0007200 600 FORMAT(I2)
0007300      NX=NVLF+4
0007400      READ(66,601) (STNID(I), I=1, NVLF)
0007500 601 FORMAT(6A1)
0007600      READ(66,600) DSRN
0007700      READ(66,602) NT
0007800 602 FORMAT(I4)
0007900      READ(66,600) NPRINT
0008000 C
0008100 C
0008200      READ(66,603) LATD, LATH, LATS
0008300      READ(66,603) LONGD, LONGH, LONGS
0008400 603 FORMAT(I4, 1X, I2, 1X, F5.3)
0008500 C
0008600      DO 20 I=3, NX
0008700      READ(66,604) X(I)
0008800 604 FORMAT(F6.2)
0008900 20 CONTINUE
0009000 C
0009100      DO 21 I=1, NX
0009200      READ(66,605) P(I, I)
0009300 605 FORMAT(F9.4)
0009400 21 CONTINUE
0009500 C
0009600      READ(66,605) D(1, 1)
0009700      READ(66,605) D(2, 2)
0009800 C
0009900      DO 22 I=1, NVLF
0010000      READ(66,605) R(1, I)
0010100 22 CONTINUE
0010200 C
0010300 C
0010400      WRITE(90, 924)
0010500 924 FORMAT(' ', 2X, 'DO YOU WANT AUTO. VLF NOISE/BIAS CHECK?')
0010600      READ(68, 6072) IVAUTO
0010700 6072 FORMAT(I1)
0010800 C
0010900      WRITE(90, 925)
0011000 925 FORMAT(' ', 2X, 'DO YOU WANT P MATRIX ELEMENTS FIXED?')
0011100      READ(68, 6072) IPFIX
0011200      IF(IPFIX.EQ.1) WRITE(90, 9251)
0011300 9251 FORMAT(' ', 2X, 'INPUT P LEVEL FIX, F8.4')
0011400      READ(68, 6073) PFIX
0011500 6073 FORMAT(F8.3)
0011600 C
0011700      PAUSE 'OPTION TO CHANGE ANY INPUT PARAMETERS'
0011800      WRITE(90, 926)

```

```

0011900 926 FORMAT(' ',2X,'INPUT Q MATRIX ELEMENTS FOR VLF BIAS... FF.FFFFF')
0012000      DO 30 I=1,NVLF
0012100      WRITE(90,927) STNID(I)
0012200 927 FORMAT(' ',1X,A1)
0012300      READ(68,627) Q(I+4,I+4)
0012400 627 FORMAT(F8.5)
0012500 30 CONTINUE
0012600 C
0012700      WRITE(90,934)
0012800 934 FORMAT(' ',2X,'INPUT Q MATRIX ELEMENTS FOR DOPPLER BIAS... FF.FFFFF')
0012900      READ(68,634) Q(3,3)
0013000      READ(68,634) Q(4,4)
0013100 634 FORMAT(F8.5)
0013200 C
0013300 C
0013400      WRITE(90,928)
0013500 928 FORMAT(' ',2X,'DO YOU WANT STN R P MATRIX ELEMENT FIXED?')
0013600      READ(68,6072) IP10FX
0013700      IF(IP10FX.EQ.1) READ(68,629) P10FIX
0013800 629 FORMAT(F6.2)
0013900 C
0014000 C
0014100      WRITE(90,940)
0014200 940 FORMAT(' ',2X,'INPUT KAQ SEQUENCE')
0014300      WRITE(90,941) (STNID(I),I=1,NVLF)
0014400 941 FORMAT(' ',1X,6A1)
0014500      READ(68,942) (KAQ(I),I=1,NVLF)
0014600 942 FORMAT(6I1)
0014700 C
0014800 C
0014900 C
0015000 C
0015100 C      *****
0015200 C      *
0015300 C      * END OF DATA INPUT SEQUENCE*
0015400 C      *
0015500 C      *****
0015600 C
0015700 C
0015800 C
0015900 C
0016000 C      *****
0016100 C      * ECHO INPUT DATA *
0016200 C      *****
0016300      X(1)= DEGRAD(LATD,LATH,LATS,0)
0016400      X(2)= DEGRAD(LONGD,LONGH,LONGS,0)
0016500 C
0016600      WRITE(99,808) (X(I),I=1,NX)
0016700 808 FORMAT('0',3X,'INITIAL STATE VECTOR',/,4X,10F9.4)
0016800 C
0016900      WRITE(99,8081) (P(I,I),I=1,NX)
0017000 8081 FORMAT('0',3X,'INITIAL P MATRIX DIAGONALS',/,4X,10F9.4)
0017100 C
0017200      WRITE(99,8082) (Q(I,I),I=1,NX)
0017300 8082 FORMAT('0',3X,'INITIAL Q MATRIX DIAGONALS',/,4X,10F9.4)
0017400 C
0017500      WRITE(99,8083) (R(I,I),I=1,NVLF)
0017600 8083 FORMAT('0',3X,'INITIAL R MATRIX DIAGONALS',/,4X,10F9.4)
0017700 C

```

```

0017800 C
0017900 C
0018000 C
0018100 C      ** END OF DATA ECHO SEQUENCE **
0018200 C
0018300 C
0018400 C      *****
0018500 C      *
0018600 C      * CONVERT VARIABLES TO PROGRAM: *
0018700 C      * ORIENTED QUANTITIES(IE NM/SEC *
0018800 C      * AND RADIANS, ETC. *
0018900 C      *****
0019000 C
0019100      X(3)= X(3)/3600.0
0019200      X(4)= X(4)*RAD
0019300 C
0019400      P(3,3)= P(3,3)/(3600.0*3600.0)
0019500      P(4,4)= P(4,4)*(RAD*RAD)
0019600 C
0019700      Q(1,1)= Q(1,1)/(3600.0*3600.0)
0019800      Q(2,2)= Q(2,2)*(RAD*RAD)
0019900      Q(3,3)= Q(3,3)/(3600.0*3600.0)
0020000      Q(4,4)= Q(4,4)*(RAD*RAD)
0020100 C
0020200 C
0020300 C
0020400 C
0020500      WRITE(99,952)
0020600 952 FORMAT('1')
0020700 C
0020800 C
0020900      NN=0
0021000 C
0021100 C*** MODS FOR VARIABLE TIME INCREMENT 23 FEB 81 **
0021200 C
0021300      WRITE(90,930)
0021400 930 FORMAT(' ', 'TIME INCREMENT? MULTIPLE OF 10 SECS, I2')
0021500      READ(68,630) NINC
0021600 630 FORMAT(I2)
0021700 C
0021800      IF(NINC.EQ.NPRINT) GO TO 71.
0021900      WRITE(90,931)
0022000 931 FORMAT(' ', 'CHANGE NPRINT TO MATCH NINC, I2')
0022100      READ(68,630) NPRINT
0022200 C
0022300 C
0022400 C
0022500 71 CONTINUE
0022600 C*** END OF MOD ***
0022700 C
0022800      REWIND 40
0022900 C
0023000 C      *****
0023100 C      * GROUND RUN STATISTICS *
0023200 C      *****
0023300 C
0023400      WRITE(90,932)
0023500 932 FORMAT(' ', 2X, 'DO YOU WANT GROUND RUN STATISTICS?')
0023600      READ(68,6072) IGND

```

```

0023700      IF(IGND.EQ.1) CALL GNDRUN(X(1),X(2),NINC,NVLF,STNID,KBR,R)
0023800 C
0023900 C
0024000      WRITE(90,911)
0024100 911  FORMAT(' ',2X,'DO YOU WANT INNOVATIONS SEQUENCE O/P ?')
0024200      READ(68,6072) INOISE
0024300      IF(INOISE.EQ.1) WRITE(94,933) ID
0024400 933  FORMAT(' ',A4)
0024500 C
0024600 C
0024700 C
0024800 C ** MOD FOR ARBITRARY BIAS CALC FOR VLF AT NN=0 **
0024900 C
0025000      CALL VLFBTA(X(1),X(2),OBS,NN,KBR,STNID,DOPTIM,IVLF,KAQ,IDOP,NINC,
0025100      &          VLFTIM,NVLF,TDTA,TTDTA,BI)
0025200 C
0025300 C
0025400 C
0025500 C *****
0025600 C ***** RETURN POINT FOR SUCCESSIVE ITERATIONS *****
0025700 C *****
0025800 C *****
0025900 C
0026000 C
0026100 41      CONTINUE
0026200 C
0026300 C
0026400 C
0026500 C      ** DETERMINE IF DATA SHOULD BE WRITTEN. **
0026600 C      ** 'NPRINT' WAS PREVIOUSLY USER DEFINED **
0026700 C
0026800      IF(NN/NPRINT*NPRINT.NE.NN) GO TO 455
0026900 C
0027000      WRITE(99,9125) NN
0027100 9125  FORMAT('0',1X,I4)
0027200 C
0027300 C
0027400 C      *****
0027500 C      * OUTPUT STATE ESTIMATES *
0027600 C      * *****
0027700 C
0027800 C
0027900 C
0028000      XU3=X(3)*3600
0028100      XU4=X(4)/RAD
0028200 C
0028300 C
0028400 C      *** WRITE UPDATES AND MATRICES ***
0028500 C
0028600 C
0028700      WRITE(99,914) X(1),X(2),XU3,XU4,(X(I),I=5,NX)
0028800 914  FORMAT(' ',4(1X,F10.5),6(2X,F9.5))
0028900 C
0029000 C
0029100      WRITE(99,9140)
0029200 9140  FORMAT(' ')
0029300 C
0029400 C
0029500      IF(DSRN.EQ.0) GO TO 455

```

```

0029600 C
0029700 WRITE(DSRN,901) MTH, DAY, YEAR, T
0029800 C
0029900 C
0030000 WRITE(DSRN,915) NN
0030100 915 FORMAT(' ',5X,'P MATRIX *****',15,' SECONDS *****')
0030200 CALL MOUTS(P,10,NX,NX,DSRN)
0030300 C
0030400 C
0030500 WRITE(DSRN,916)
0030600 916 FORMAT(' ',5X,'Q MATRIX:')
0030700 CALL MOUTS(Q,10,10,10,DSRN)
0030800 C
0030900 C ** INITIALLY (AT TIME=0) WE DO NOT HAVE **
0031000 C ** PHI,GAMMA,M,OR K MATRICES DEFINED. **
0031100 C ** THEREFORE, DO NOT WRITE. **
0031200 C
0031300 C
0031400 IF(NN.EQ.0) GO TO 455
0031500 C
0031600 WRITE(DSRN,917)
0031700 917 FORMAT(' ',5X,'PHI MATRIX:')
0031800 CALL MOUTS(PHI,10,NX,NX,DSRN)
0031900 C
0032000 WRITE(DSRN,918)
0032100 918 FORMAT(' ',5X,'GAMMA MATRIX:')
0032200 CALL MOUTS(GAMMA,10,NX,10,DSRN)
0032300 C
0032400 WRITE(DSRN,919)
0032500 919 FORMAT(' ',5X,'M MATRIX:')
0032600 CALL MOUTS(M,10,NX,NX,DSRN)
0032700 C
0032800 C
0032900 WRITE(DSRN,921)
0033000 921 FORMAT(' ',5X,'GAIN MATRIX:')
0033100 CALL MOUTS(KI,10,NX,NVLF,DSRN)
0033200 C
0033300 C
0033400 C
0033500 455 CONTINUE
0033600 C
0033700 C ** IF SIMULATION PERIOD IS OVER,NO MORE ITERATIONS ARE DONE **
0033800 C
0033900 IF(NN.GE.NT) STOP
0034000 C
0034100 C ** INCREMENT TIME COUNTER BY TIME INCREMENT **
0034200 C
0034300 NN=NN+NINC
0034400 C
0034500 C
0034600 C
0034700 C
0034800 A1=0.0
0034900 A2=0.0
0035000 A3=0.0
0035100 A4=0.0
0035200 C
0035300 NDOP= NINC*2 + 1
0035400 C

```

```

0035500 C
0035600 NREC= (NDOP-1)/4
0035700 READ(40,401) NNFLT
0035800 401 FORMAT(I5)
0035900 C
0036000 DO 23 I=1,NREC
0036100 JST=(I*4)-3
0036200 JEND=JST+3
0036300 READ(40,400) (VX(J),VY(J),BETAD(J),J=JST,JEND)
0036400 400 FORMAT(4(F8.4,2X,F8.4,2X,F9.4,5X))
0036500 23 CONTINUE
0036600 C
0036700 READ(40,402) VX(NDOP),VY(NDOP),BETAD(NDOP)
0036800 402 FORMAT(F8.4,2X,F8.4,2X,F9.4)
0036900 C READ(40,403) (OBS(I),I=1,NVLF)
0037000 403 FORMAT(5X,6(F9.3,2X))
0037100 C
0037200 C
0037300 C ** MOD TO HANDLE DOPPLER GLITCHES IN FLT82-03 DATA **
0037400 C
0037500 KK=NDOP-1
0037600 DO 26 I=1,KK
0037700 DOPCHK=VX(I+1)-VX(I)
0037800 IF (ABS(DOPCHK).LT.20.0) GO TO 26
0037900 IF (DOPCHK.LT.0.0) VX(I+1)=VX(I)
0038000 IF (DOPCHK.GE.0.0) VX(I)=VX(I+1)
0038100 26 CONTINUE
0038200 C
0038300 C ** END MOD **
0038400 DO 24 I=1,NDOP
0038500 VDOP(I)= SQRT(VX(I)*VX(I) + VY(I)*VY(I))
0038600 CC= VY(I)/VX(I)
0038700 ALPHAD(I)=ATAN(CC)
0038800 C
0038900 VDOP(I)=VDOP(I)/3600.0
0039000 IF (BETAD(I).LT.0.0) BETAD(I)=BETAD(I)+360.0
0039100 BETAD(I)=BETAD(I)*RAD
0039200 24 CONTINUE
0039300 C
0039400 CALL ASUBJ(VDOP,ALPHAD,BETAD,X,A1,A2,A3,A4,A5,A6,A7,A8,I,NDOP)
0039500 C
0039600 CALL FISUBJ(A1,A2,A3,A4,A5,A6,A7,A8,PHI,GAMMA,X,NX)
0039700 C
0039800 CALL FSUBJ(F,X,A1,A2,A3,A4,NX)
0039900 C
0040000 CALL VLFDTA(X(1),X(2),OBS,NN,KBR,STNID,DOPTIM,IVLF,KAQ,IDOP,NINC,
0040100 8 VLFTIM,NVLF,TDTA,TTDTA,BI)
0040200 C
0040300 C
0040400 C
0040500 DO 87 I=1,NVLF
0040600 SMALLH(I,1)= +GIJ( F(1),F(2),STNID(I),0,CV,WL,NN) + X(I+4)
0040700 IF (KAQ(I).EQ.0) SMALLH(I,1)=OBS(I,1)
0040800 87 CONTINUE
0040900 C
0041000 C
0041100 CALL SUB(SMALLH,OBS,INNOV,5,1,6,6,6)
0041200 C
0041300 IF (INOISE.EQ.1) WRITE(94,966) NN

```



```

0041400 966 FORMAT(' ',2X,I4,2X,' SECONDS')
0041500 C
0041600 C
0041700 DO 25 I=1,NVLF
0041800 A(I,1)=-INNOV(I,1)
0041900 IF(INOISE.EQ.1) WRITE(94,967) STNID(I),A(I,1)
0042000 967, FORMAT(' ',5X,A1,3X,F9,3)
0042100 C
0042200 C KAQ(I)=1
0042300 25 CONTINUE
0042400 C
0042500 C
0042600 C *****
0042700 C * VLF NOISE AND BIAS JUMP *
0042800 C * CHECKING *
0042900 C *****--*****
0043000 C
0043100 IF(IVAUTO.EQ.0) GO TO 80
0043200 C
0043300 IF(NN.LT.300) GO TO 80
0043400 NNCHK=300+29*NNINC
0043500 IF(NN.GT.NNCHK) CALL VLFFIX(P,R,NVLF,KAQ,IVAR,A,VMEAN,VSTD,
0043600 & KULF,VLFN,NINC,NN,0.50,10.0)
0043700 C
0043800 C
0043900 DO 79 I79=1,NVLF
0044000 IF(IVAR(I79).EQ.1) GO TO 79
0044100 IF(KAQ(I79).EQ.0) GO TO 79
0044200 C
0044300 KULF(I79)= KULF(I79) + 1
0044400 KKULF= KULF(I79)
0044500 VLFN(KKULF,I79) = A(I79,1)
0044600 C
0044700 IF(KULF(I79).LT. 30) GO TO 79
0044800 C
0044900 DO 77 I77=1,30
0045000 DTA(I77)=VLFN(I77,I79)
0045100 XTIME(I77)=300.0+(I77-1)*NNINC
0045200 77 CONTINUE
0045300 C
0045400 CALL LINREG(XTIME,DTA,30,VMEAN(I79),VSTD(I79),SE,SXX,SXY,SY,80,B1)
0045500 C PAUSE 1
0045600 C
0045700 C
0045800 IVAR(I79) = 1
0045900 KULF(I79) = 0
0046000 C
0046100 79 CONTINUE
0046200 80 CONTINUE
0046300 C
0046400 C ** MOD FOR STN R DRIFTING **
0046500 C
0046600 IF(IP10FX.EQ.1) P(10,10)=P10FIX
0046700 C 66 CONTINUE
0046800 C
0046900 IF(IPFIX.EQ.0) GO TO 67
0047000 IF(P(5,5).LT.PFIX) P(5,5)=PFIX
0047100 IF(P(6,6).LT.PFIX) P(6,6)=PFIX
0047200 IF(P(7,7).LT.PFIX) P(7,7)=PFIX

```

```
0047300      IF(P(8,8).LT.PFIX) P(8,8)=PFIX
0047400      IF(P(9,9).LT.PFIX) P(9,9)=PFIX
0047500  67  CONTINUE
0047600 C
0047700 C ** END MOD **
0047800 C
0047900 C
0048000      CALL KFSUB(P,PHI,GAMMA,B,NX,NVLF,STNID,R,INNOV,F,X,NN)
0048100 C
0048200 C
0048300      GO TO 41
0048400 C
0048500      END
EOF
```

END

0|8|0|6|8|3

FIN