NASA CR- 172,199

NASA Contractor Report 172199

NASA-CR-172199
19840003088

Finds: A Fault Inferring Nonlinear Detection System—User's Guide

R.E. Lancraft and A.K. Caglayan

Bolt Beranek and Newman Inc.
Cambridge, MA 02238

Contract NAS1-16579
September 1983

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
2. PROGRAM ORGANIZATION	3
3. DESCRIPTION OF INPUTS	13
3.1 Interactive Nature of FINDS	14
3.2 General Input Parameter File	17
3.3 Sensor Input Parameter File	24
3.4 Filter Input Parameter File	52
3.5 Quick Reference to Input Files	55
3.6 Sample Input File Specification	61
4. DESCRIPTION OF OUTPUTS	65
4.1 Summary File	65
4.2 Time Line File	71
4.3 Output File	77
4.4 Plot File	81
5. CONCLUDING COMMENTS AND RECOMMENDATIONS	91

APPENDIX A. POST-PROCESSING PROGRAMS 93

A.1 Description of Program: PRINTD 93
A.2 Description of Program: PLOTD 95

6. REFERENCES 97

INDEX 99

LIST OF FIGURES

FIG.	1.	FUNCTIONAL FLOW DIAGRAM OF FINDS	5
FIG.	2.	FAULT TOLERANT SYSTEM STRUCTURE	7
FIG.	3.	INPUT/OUTPUT PERSPECTIVE OF FINDS	11
FIG.	4.	FINDS INTERACTIVE SESSION	15
FIG.	5.	FUNCTIONAL BLOCK DIAGRAM OF A GENERIC SENSOR MODEL	26
FIG.	6.	FUNCTIONAL BLOCK DIAGRAM OF A GENERIC FAILURE MODEL	30
FIG.	7.	INPUT FILE EXAMPLE	62
FIG.	8.	SAMPLE SUMMARY FILE	66
FIG.	9.	SAMPLE TIME LINE FILE	76
FIG.	10.	SAMPLE OUTPUT FILE	78

This Page Intentionally Left Blank

LIST OF TABLES

TABLE 1.	ORDERING CONVENTIONS FOR ARRAYS	9
TABLE 2.	SENSOR MODULE MNEMONICS	25
TABLE 3.	SUMMARY OF NAMELIST DESCRIPTIONS	56
TABLE 4.	SUMMARY OF .INP FILES	58
TABLE 5.	SUMMARY OF TIME LINE CODES	73
TABLE 6.	PLOT FILE MNEMONICS	82

This Page Intentionally Left Blank

LIST OF ABBREVIATIONS

TCV	Terminal Configured Vehicle
MLS	Microwave Landing System
RSDIMU	Redundant Strapdown Inertial Measurement Unit
IAS	Indicated Airspeed
IMU	Inertial Measurement Unit
A/C	Aircraft
G&C	Guidance and Control
DME	Distance Measuring Equipment
RA	Radar Altimeter
NF filter	No-Fail Filter
LR	Likelihood Ratio
NL	Non-Linear
TOF	Time of Failure

This Page Intentionally Left Blank

LIST OF SYMBOLS

X_{rw}, Y_{rw}, Z_{rw}	A/C position in the runway frame (m)
$\dot{X}_{rw}, \dot{Y}_{rw}, \dot{Z}_{rw}$	A/C velocity in the runway frame (m/s)
ϕ, θ, ψ	A/C Euler angles which rotate the body axes into coincidence with the runway frame
X_w, Y_w	Horizontal wind components in the runway frame
A_x, A_y, A_z	A/C acceleration in the body frame
P,Q,R	A/C angular rates of change in the body frame
Azm,El,Rng	MLS azimuth, elevation, and range
θ_{xy}	Small misalignment angles
ω_x, ω_y	X and Y axis angular rates from two-degree-of-freedom gyro used in the RSDIMU
NX	The total number of NF filter states (estimates)
NU1	The total number of NF filter input measurements
NYMAX	The maximum possible number of output measurement types
NDISTB	The total number of NF filter process noises
NFMAX	The total number of sensor types
NFT	The total number of sensors used by the NF filter

1. INTRODUCTION

This report describes the digital computer program FINDS (Fault Inferring Nonlinear Detection System) version 3.0 developed by Bolt, Beranek, and Newman Inc. (BBN).[1]

The objective of FINDS is to detect, isolate, and compensate for failures in navigation-aid instruments and on-board flight control and navigation sensors of a Terminal Configured Vehicle (TCV) aircraft in a Microwave Landing System (MLS) environment. FINDS also provides sensor fault tolerant estimates for the aircraft states which are used by an automatic guidance and control system to land the aircraft along a prescribed path. FINDS monitors for failures by evaluating all sensor outputs simultaneously and uses analytic relationships between the various sensor outputs arising from the aircraft point mass equations of motion. Hence, FINDS is an integrated sensor failure detection and isolation system.

Although the specific application in this study is concerned with aircraft sensor failures, the failure detection and isolation algorithm developed is quite general and applicable to input, component and output failure identification problems in discrete-time, nonlinear stochastic systems.

To give somewhat of an historical perspective, it should be mentioned that several programs were merged together to form the TCV environment in which FINDS operates. The NASA-supplied program FILCOMP (formerly called ALERT), was modified to provide the required dynamic environment as well as

---

[1]Special thanks and acknowledgement is given to F.R. Morrell and R. Hueschen for invaluable discussions regarding NASA supplied software used in this program and documented in this report.

the MLS simulation model used. In addition, NASA also furnished a simulation model of a developmental redundant strapdown inertial measurement unit (RSDIMU), obtained from the program BVALT. These two programs were combined with sensor and failure models developed by BBN, to form the simulated TCV environment.

FINDS is written in FORTRAN-77, and is intended for operation on a Digital Equipment Corporation's (DEC) VAX 11-780 or 11-750 super mini computer, using the VMS operating system. The program was written in a modular and flexible fashion to allow for ease of program verification, and future modification.

This volume is intended to serve as a self-contained users guide to FINDS and it's associated post-processing programs. If the reader is interested in a more detailed understanding of the theoretical background, then review of the contract's interim report [1], or final report [2] is recommended.

The organization of this report is as follows: Chapter 2 consists of a brief overview of the nature of FINDS, along with a discussion on program organization. It also contains a section which discusses the installation of the delivered program. Chapter 3 gives a complete description of all program inputs, as well as a typical example which clearly shows how to generate the required input files. One feature of this chapter is a quick reference section. The user can find both consise input specification information, as well as cross references to very detailed descriptions. Chapter 4 reviews the outputs available from FINDS by discussing the outputs of the example from Chapter 3. Concluding remarks are made in Chapter 5. Appendix A consists of descriptions of the post-processing programs which can be used in conjunction with FINDS. Finally, a cross-reference list of all important variables and symbols is included at the end of the report for subsequent quick reference.

2. PROGRAM ORGANIZATION

The intention of this chapter is to give the reader an overall understanding of the operation and organization of the FINDS program. FINDS will be examined from both an input/output point of view as well as from a functional viewpoint. As mentioned in the introduction, FINDS is an amalgamation of several computer programs. Since less attention will be devoted to these NASA-supplied modules in this manual, whenever possible, the reader will be referred to more detailed references. After reading this chapter, the reader should have a sound idea of the nature of FINDS, and how it can be used. Further details can be found in the following chapters of this document and,in the final report [2].

The purpose of FINDS is to detect, isolate, and compensate for failures in navigation-aid instruments and on-board flight control and navigation sensors of a TCV aircraft in an MLS environment. FINDS also provides sensor fault tolerant estimates for the aircraft (A/C) states, which are used by an automatic guidance and control system to land the A/C along a prescribed path. The user can analyze the performance of FINDS over many different conditions. For example, each of the following conditions can be <u>easily</u> simulated by FINDS:

- o different flight paths
- o various sensor replications (single, dual or triple redundancy)[2]
- o different sensor parameters (normal operating noise, bias, scale factor, etc.)

---

[2]Although FINDS can simulate triple redundancy in the sensor module, the filter/detector structure has been tested using at most dual redundancy. Therefore, the filter is constrained to use no more the dual redundancy in FINDS version 3.0

- different sensor configurations (i.e. ability to specify what physical sensors are used, for example to use an IMU or a RSDIMU)
- different disturbance profiles (shear, gusts, and horizontal winds)
- different failure modes, amplitudes and onset times
- effects of multiple failures or simultaneous failures
- different NF filter configurations (set of operational biases to estimate)

Chapter 3 will explain how the user can specify each of these conditions.

Provisions have been made in the program which enable the user to run the TCV simulation with guidance and control (G&C) commands generated either from the no-fail filter state estimates, or the "true", simulated values of these variables. The option of running the NF filter without failure detectors is also provided. FINDS can be run in a Monte Carlo fashion to obtain statistical performance information, or in a comparative fashion (by using the same noise sequence) to obtain relative performance data.

A functional block diagram of the major modules in FINDS, as well as the overall program flow, are detailed in Figure 1. The major functional blocks consist of the following:

- Initialization module to initialize all modules properly
- A module to compute the guidance and control commands required by the A/C simulation
- A facility to save information at the end of a simulation "tick"
- A program stopping criterion (to stop after some maximum elapsed time)
- Integration of the six-degree-of-freedom nonlinear TCV aircraft equations of motion
- A collection of realistic sensor modules (including the NASA-supplied RSDIMU and MLS), which take in the true values of measured variables

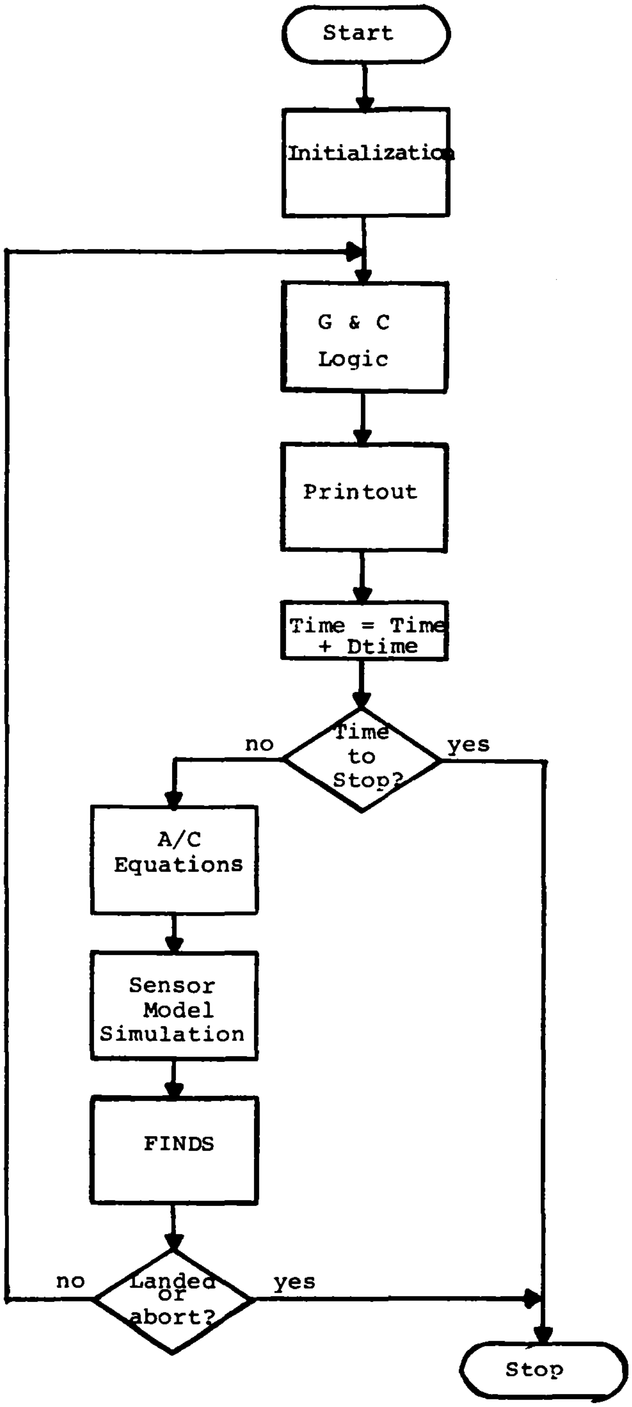


FIG. 1. Functional Flow Diagram of FINDS

from the TCV simulation and put out sensor outputs which account for misalignment, measurement noise, bias drift, normal scale factor errors and limits. This block also contains sensor failure models for simulating increased bias, hardover, null, scale factor, ramp, and increased noise type sensor malfunctions.

- FINDS fault tolerant system algorithms. This module essentially looks like a fault tolerant navigator in this diagram (it supplies reliable state estimates to the G&C module).

- Additional program stopping criterion (if the A/C has landed or if the fault tolerant navigator has issued an "abort")

The fault tolerant system module in Figure 1 represents the "heart" of the program. Figure 2 shows this module broken down into its functional blocks. The figure is made up of the following parts:

- a nonlinear state estimator (also called NF filter, or fault tolerant navigator) which provides sensor fault tolerant estimates for aircraft position, velocity, attitude, and horizontal winds along with normal operating biases for a user-selected sensor subset. This module is separated into "update" and "propagate" cycles in the diagram.

- a bank of detectors which are first-order failure level estimators for estimating bias jump failures in sensor outputs. Each detector operates over a "detection window", which is a fixed integer multiple of decision windows <u>and</u> synched to the start of a decision window.

- a bank of likelihood ratio (LR) computers providing the necessary LR computations for the posed multiple hypothesis testing problem. LR's are computed over a fixed length moving "decision window".

- a Bayesian decision rule which selects the most likely failure mode based on the LR computations. Decisions are made at every simulation "tick", not just at the end of decision windows.

- a sensor healing module which monitors the failed sensors (if any) to determine if they have "healed", or alternatively, if the detection logic had generated a "false alarm" by failing the sensor in the first place.[3] The healers operate over a fixed length moving window

[3] The current healer module is only effective for bias, null, or hardover failures.

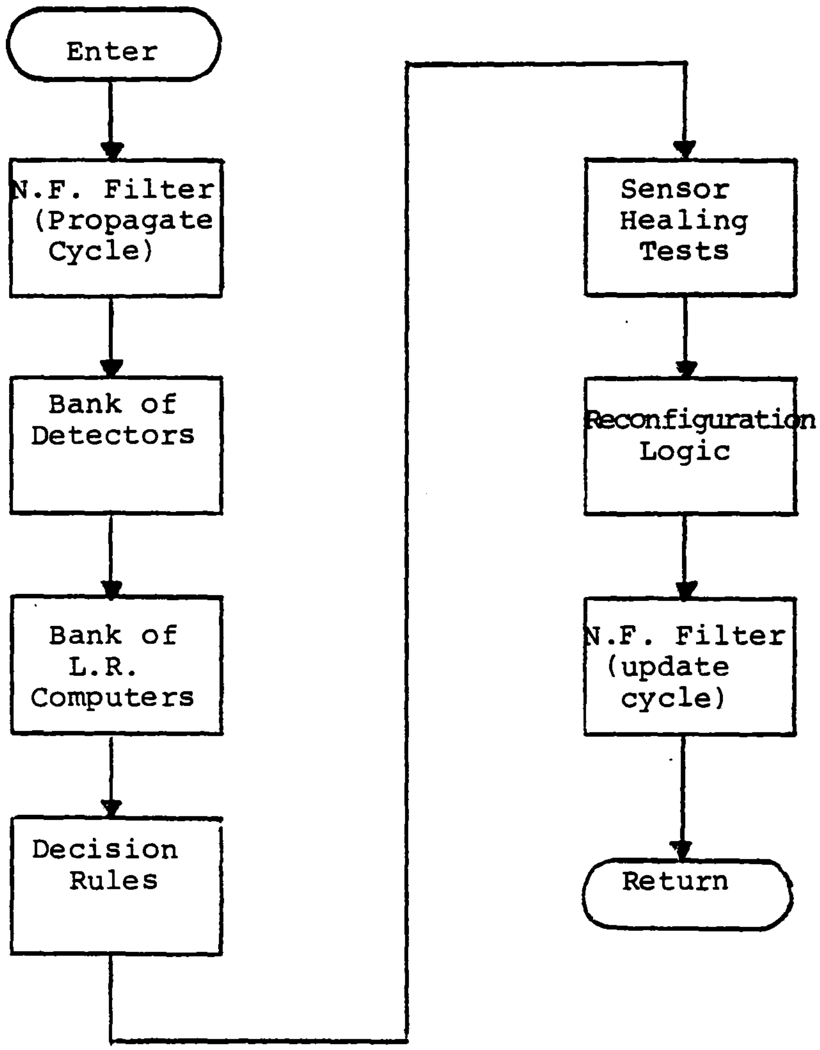


FIG. 2. Fault Tolerant System Structure

which is synched to the decision window. Healing decisions are made only at the end of this window.

- a reconfiguration module which performs the necessary reinitialization in the previous five blocks after the detection and isolation of a sensor failure.

More detail on each of these modules can be found in [1], and [2].

At this point, it will be useful to develop a set of working descriptions which we can use throughout the manual to describe various ideas. For instance, vectors and matrices need to have some sort of symbolic ordering in the program, Table 1 defines several ordering conventions that were adopted (the symbols used in the table are defined in the beginning of the manual or in Table 2 on page 25). The table is organized by "type" and array index. Notice that the vector length is also symbolically defined in the "type" section. For example, if, elsewhere in this manual, reference is made to a vector ordered by "measurement type", then we would know, from Table 1, that the vector is nominally of length NYMAX=8 and ordered as MLS-azimuth, MLS-elevation, MLS-range, Indicated Airspeed, IMU-phi, IMU-theta, IMU-psi, and Radar Altimeter measurements.

Notice from Table 1, the sensor measurements are divided into two groups (when viewed by the NF filter) called "Inputs" and "measurements". The first group, which consists of linear accelerometer and rate gyro measurements (either body mounted or RSDIMU supplied), are referred to as the NF filter's input measurements or simply as input measurements. The NF filter integrates these input measurements during the propagate cycle of the filter to obtain the propagated state estimate. The rest of the sensor measurements are collectively called the NF filter measurements, or simply filter measurements. They are treated as measurements in the NF filter, and are used to compensate the propagated state estimate during the update filter cycle.

The actual input/measurement configuration used by the NF filter is user-definable. The following rules detail this:

TABLE 1.   Ordering Conventions For Arrays

	Array "Type"				
Array Index	NF Filter States (NX)	NF Filter Measurements (NYMAX)	NF Filter Inputs (NU1)	Sensors (NFMAX)	Process Noise (NDISTB)
1	X_{rw}	MLS_{AZ}	A_x	A_x	A_x
2	Y_{rw}	MLS_{EL}	A_y	A_y	A_y
3	Z_{rw}	MLS_{RNG}	A_z	A_z	A_z
4	$\dot{X}_{rw}$	IAS	P	P	P
5	$\dot{Y}_{rw}$	ϕ	Q	Q	Q
6	$\dot{Z}_{rw}$	θ	R	R	R
7	$\dot{\phi}$	ψ		Azm	X_w
8	$\dot{\theta}$	RA		El	Y_w
9	$\dot{\psi}$			Rng	
10	X_w			IAS	
11	Y_w			ϕ	
12				θ	
13				ψ	
14				RA	

- The number of replications of any given sensor used by the NF filter is equal to the number of replications the user has chosen to simulate (none, single, dual or triple redundancy).

- Input measurements can be obtained either from body mounted (flight control quality) sensors, or from an (navigation quality) RSDIMU.

- A single replication of input measurements are used by the NF filter. Additional replications, if simulated, are used as standby equipment.

- Attitude measurements can either be omitted from the filter, obtained from a platform IMU, or from the RSDIMU.

Figure 3 shows an input/output perspective of FINDS from the users viewpoint. Notice that three disk files supply information to the program, with additional input from the controlling terminal (interactive responses) or alternatively from a batch commands file. From an overall point of view, the three disk files suply parameters for general program control and A/C simulation purposes, sensor/failure model simulation, and fault tolerant system parameters, respectively. The files were divided in this manner to facilitate the Monte Carlo running of FINDS. Each of these files will be described in detail in Chapter 3.

On the output side, FINDS generates four disk files of output information, with a fifth file (FINDS.LOG) created by the batch operating system if FINDS was run via the batch processor. FINDS automatically assigns a pre-defined file extension to each of these output files. In as much as the four output files will be discussed in detail in Chapter 4, we will only give a quick summary here.

- FINDS.SUM. The .SUM file basically contains a "summary" of the parameters used to generate the run. It is currently organized into a series of five tables which summarize the sensor and fault tolerant system modules.

- FINDS.TLN. The .TLN file is a "Time line" file of major discrete events which occurred during the course of the run. Whenever an "event" occurs, a coded "snapshot" of that point in the run is saved on the file. This file can later be used to obtain statistical information about FINDS failure detection performance.

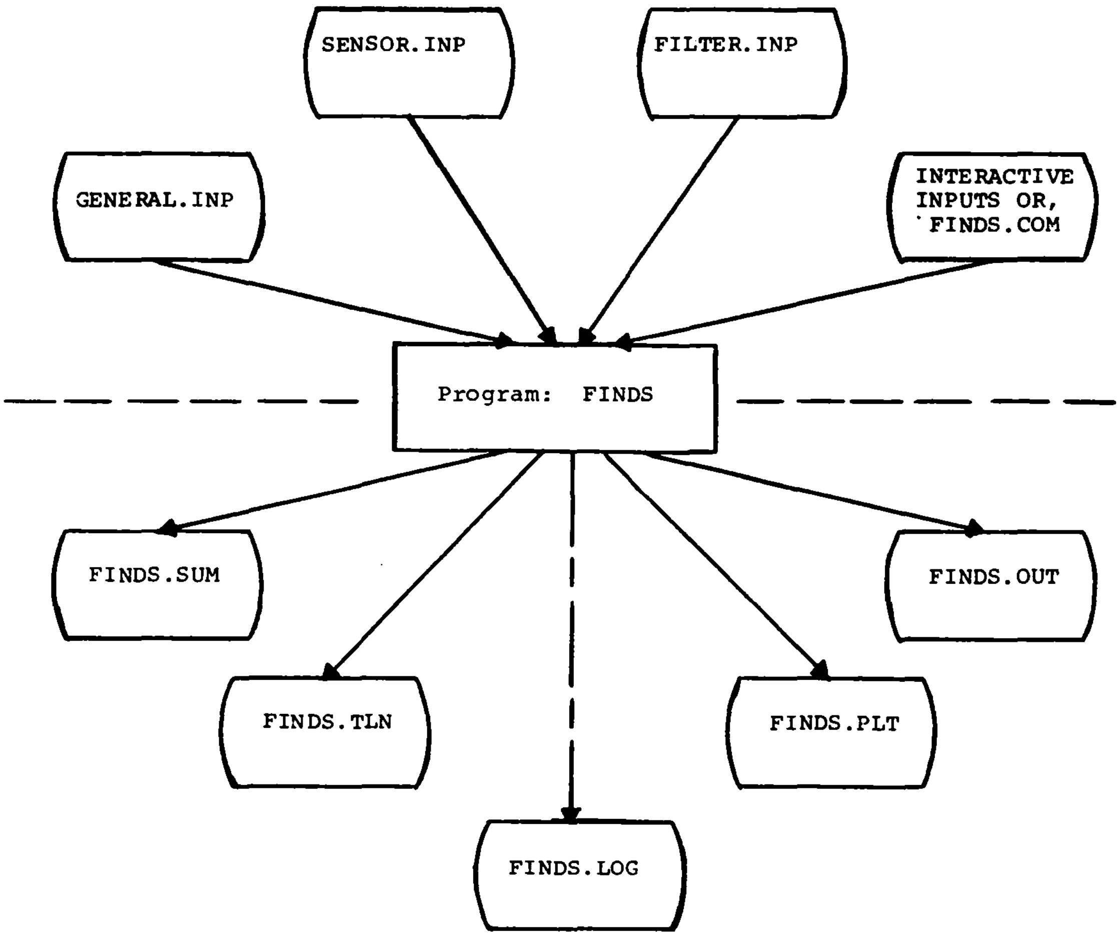


FIG. 3. Input/Output Perspective of FINDS

- FINDS.PLT. The .PLT file is an unformatted (binary) file of time history data. The user can select the frequency and variables to be saved. He can also request that this file not be generated. This file can be post-processed to obtain tables of the data or time-history plots.

- FINDS.OUT. The .OUT file is an output file containing complete echo checks on all input file parameters as well as ASCII time history output. Options are available to either eliminate the time history outputs, or to suppress the generation of this file.

- FINDS.LOG. The .LOG file contains useful run time statistics and error messages (if any).

From the above descriptions, we see that FINDS records detailed (continuous) time history data, discrete event information, and partial or complete summaries of the input data used to produce the run. Furthermore, because of the file formats choosen, uniform naming conventions, and the separation of data by type (i.e.discrete, continuous, summary, etc.) the information contained in each run can be post-processed to obtain ensemble as well as temporal metrics in a convienient fashion. The collection of all runs can almost be viewed as a distributed data base which can be saved on tape and re-examined as new post-procesing programs become available. The only facility missing from FINDS 3.0, which would truly make its outputs part of a data base, is a common file which logs the names of all the runs created. This facility is a simple but important one to add in the future and is mentioned here, only to draw attention to the fact that if such a facility were available , nearly the entire post anlaysis phase of a Monte-Carlo study would be fully automated.

3. DESCRIPTION OF INPUTS

This section describes, in detail, each of the four (4) input files required by FINDS. As you recall, from Figure 3 and the organizational discussions of the last chapter, the four input files consist of the following:

1. "Interactive" responses from the controlling terminal (TT:) or from a batch processor command file. We will refer to the batch command file as the FINDS.COM file.

2. "General" program control and A/C simulation inputs. This file will be referred to as the GENERAL.INP file in the ensuing discussions.

3. "Sensor" module inputs in the SENSOR.INP file.

4. "Filter" and detector module inputs. This file will be called the FILTER.INP file.

In naming each of these files we've adopted arbitrary names which are valid VAX/VMS file specifications. Where the partial form is: "file-name.file-extension". Although the VMS operating system allows very large file specifications (up to 128 characters), FINDS arbitrarily limits the file name to eight (8) characters, and the file ext. to three (3).

The .INP files are "Namelist" files; that is, they contain only Fortran namelist style input data. If the reader is not familiar with namelist-directed read statements, he should review pp. 5-21, 5-22, and 7-18 to 7-22 in [7] before attempting to generate an .INP file. The basic form of a namelist input is:

$namelist name_[entity=value,...]$[END]

where _=space, and [...] represent optional arguments. Entity is a program variable name, defined in the Fortran namelist statement. Each .INP file is made up of several namelist-directed reads. Where all expected namelists <u>must</u> appear within the .INP file, even if no change to the default entity values are desired.

The rest of this chapter is organized as follows: Section 3.1 discusses the interactive environment and use of the FINDS.COM file. Section 3.2-3.4 define the namelists and their associated entities for each of the three .INP files. Section 3.2 also explains the general format of the namelist descriptions used in Section 3.2 - 3.4. The next Section, Section 3.5, is a "quick reference" section which simply reviews, in a very abbreviated format, all the namelist names and entity names. The last Section, Section 3.6, contains a descriptive example to demonstrate how the input files are generated.

3.1 Interactive Nature of FINDS

This section discusses FINDS' interactive environment. The program prompts, as well as expected user responses, will all be enumerated.

The interactive environment of FINDS offers the user various convenient features. For instance, since the program will normally be run in a Monte-Carlo fashion, the user is faced with running the same problem over and over, varying only the random initial conditions. By using an interactive environment, the user can specify those quantities which are specific to a particular run, and eliminate unnecessary duplication of the input files.

To show how the interactive environment operates and to enumerate the user responses, a typical FINDS session is shown in Figure 4. (Note: in the typescript, line numbers are shown for reference purposes, user responses are in bold print, and carriage return is noted as <CR>.)

As seen from Figure 4, the interactive environment is fairly self-explanatory. Each of the user responses are verified by FINDS before the next question is asked. If a faulty answer is given, FINDS will force the user to supply an acceptable one. Note that lines 3-4, 14-32, and 34-35 show how the

LINE No.	INTERACTIONS
1	**$ RUN FINDS <CR>**
2	Specify FILE NAME for Program Control Info: **GENERAL.INP <CR>**
3	Specify FILE NAME for Sensor Model Info: **SENSOR.IMP <CR>**
4	Can't find File:SENSOR.IMP, TRY AGAIN: **SENSOR.INP <CR>**
5	Specify FILE NAME for Filter-Detector Info: **FILTRT.INP <CR>**
6	Specify a GENERIC name for ALL output files
7	File group name: **FINDSR <CR>**
8	TYPE UP TO 15 LINES OF TEXT, END WITH BLANK LINE
9	**STANDARD SENSOR CONFIGURATION<CR>**
10	**SGLTON BIAS FAILURES -- level = 10 sigma n.o.b.<CR>**
11	**TIME= 75 110 150 220 253<CR>**
12	**TYPE= el-2<CR>**
13	<CR>"
14	ANY CHANGES? : **Y <CR>**
15	> **?<CR>**
16	ANSWER :
17	C , D , I , R , T , F ,
18	> **<CR>**
19	EDITOR COMMANDS ARE:
20	C : Change entire title.
21	D : Delete an entire line.
22	I : Insert title above a line.
23	R : Replace one line with another.
24	T : Type current title.
25	F : Finished editing.
26	NOTES :
27	LINE = 0 : Aborts command and returns to prompt level
28	Currently blank lines of text are NOT allowed
29	> **R<CR>**
30	LINE NO.:**2<CR>**
31	**"SINGLTON BIAS FAILURES -- level = 10 sigma n.o.b.<CR>**
32	> **F<CR>**
33	Input SEED for Random Number Generation (odd #) **0 <CR>**
34	Save default variables in .PLT file? **OK <CR>**
35	ANSWER Y, y, N, n:**N<CR>**

FIG. 4. FINDS Interactive Session

LINE No.	INTERACTIONS
36	Indicate (Y,N) which variable groups to save in the PLT file
37	No-Fail filter state est error?: **Y <CR>**
38	No-Fail filter state est Uncertainty?: **Y <CR>**
39	Bias filter state est?: **Y <CR>**
40	No-Fail filter inputs?: **N<CR>**
41	No-Fail filter Outputs?: **N <CR>**
42	No-Fail filter residuals?: **N <CR>**
43	Li, Fi, Ii for detectors?: **N <CR>**
44	Expanded (& filtered) residuals?: **N <CR>**
45	A/C latitude & longitude?: **N <CR>**
46	Ground track info?: **Y <CR>**
47	True attitudes?: **Y <CR>**
48	True body accel?: **N <CR>**
49	Airspeed?: **N <CR>**
50	Performance measures?: **Y <CR>**
51	Body P,Q,R info?: **N <CR>**
52	Control info?: **Y <CR>**
53	RSDIMU info?: **N <CR>**
54	Measurement ERROR histories?: **N<CR>**
55	FORTRAN STOP

FIG. 4. FINDS Interactive Session, concluded

program reacts to improper input and how it allows for user intervention. By responding to line 33 with a zero input, the program chooses a "random" initial seed, where the seed value is based on the current time of day. The last eighteen (18) user responses, lines 36-54, define the set of variables to be saved in the .PLT file. The implications of Y/N responses to these questions are discussed in Table 6 on page 82. If this file is not requested, or if the answer to line 34 is Yes, then these questions will not be asked.

If all the user inputs from our interactive session (bolded text) are saved in a BATCH.COM file, then FINDS can be run in "batch" mode, via the VAX/VMS batch processor.

3.2 General Input Parameter File

This section describes the namelist directed inputs contained in the GENERAL.INP file. This section, as well as the next two sections, contain namelist descriptions. The general format for each namelist description will be as follows:

NAMELIST: namelist-name

Description: A one- or two-sentence description of the namelist.

Comments: Additional information, required before the variables can be specified.

Variables: A list of the variables (namelist entities), along with their definitions, range, engineering units, type, order (scalar, vector, matrix), and default value.

In order to condense the individual variable descriptions, a shorthand convention has been adopted. The following rules define this convention:

- Once a variable has been defined, it may be used to define subsequent variables.
- The symbol "=>" should be read "is associated with" or "implies".
- The symbol"-" or "->" should be read "to", for example A(1) -> A(5).
- A variable's engineering units are always given in parenthesis, at the end of the description.
- Standard abbreviations are used for units, if space is confined.
- "Namelist style" repetition factors are used in specifying both the units and default values for vectors and matrices.
- The following coded information is contained within square brackets at the end of each description.

[A,B[=R:C], D=V] where:

- A describes the variable type, as:
  R=> Real
  I=> Integer
  L=> Logical (.true. or .false.)
  A=> Alpha-numeric (character string)

- B describes the variables structure and order, where:[4]
  S=> Scalar
  V=R=> Vector with R rows
  M=R:C=> matrix with R rows and C columns

- D describes V the default value of the variable (if no namelist entry is made by the user), where:
  D=V=> V is the default value.

For example:

Foo: A dummy variable. $0 \leq$ Foo ≤ 10.0. (2*cubits,3*furlongs), [R,V=5,D=4*3.0,10.0].

describes a real, vector valued variable of length 5, called "Foo". The program expects the value for Foo to be between 0.0 and 10.0. Its default values are 3.0 cubits for the first two elements, 3.0 furlongs for the next two elements, and 10.0 furlongs for the last element.

NAMELIST: PROGCN

Description: PROGCN contains general program control quantities.

Variables:

DTIME: The simulation's integration step size. (seconds), [R,S,D=0.05]

FILTER: Flag to indicate whether or not the fault tolerant navigator is to be run, where .True. => run the navigator. (unitless), [L,S,D=.False.]

---

[4]Note: by order we mean that portion of the variable which FINDS uses, and must therefore be user specified, <u>not</u> necessarily the dimensioned size of a variable.

IEST: Flag to indicate whether the guidance and control logic uses simulation supplied ("True") quantities (IEST=0), or fault tolerant navigator supplied (estimated) quantities (IEST=1). (unitless) [I,S,D=0]

TSTOP: Maximum simulation run time. (seconds), [R,S,D=400.0]

YSTOP: Distance away from the center of the runway at which point the program will stop during rollout. (feet), [R,S,D=300.0]

NAMELIST: PRNTPL

Description: PRNTPL contains quantities used to specify print and plot options.

Variables:

PLOTS: Flag to indicate if FINDS should save an unformatted (binary) file of time history data (file ext.=.PLT), which can be processed offline. Where PLOTS=.true. => save a .PLT. file. (unitless), [L,S,D=.False.]

OUTPS: Flag to indicate if FINDS should save an ASCII file of time history data and Namelist Echo printout (file ext.=.OUT) Where OUTPS =.true.=> save an .OUT file. (Unitless), [L,S,D=.true.]

IHEAD: Number of lines of ASCII time history data printed in the output file (file ext.=.OUT) before a header line is printed. (unitless) [I,S,D=8]

IPRNT: Printout frequency for the output file (file ext.=.OUT), where printout occurs every IPRNT simulation steps. If IPRNT = 0 time history printout is suppressed. (unitless) [I,S,D=20]

KPRINT: Flag indicating printout mode, where: 0=>normal printout and 1=> debug printout. (unitless), [I,S,D=0]

NTICK: Plot save frequency (file ext.= .Plt), where data is recorded every NTICK simulation steps. (unitless), [I,S,D=20]

CPUCHK: Flag to indicate if FINDS should perform a CPU check during execution. Where, if CPUCHK=.true => a CPU check is performed and a summary printed at the end of the summary file (file ext.=.SUM). (unitless), [L,S,D=.False.]

NAMELIST: ACSTAT

Description: ACSTAT contains the quantities necessary to specify the aircraft's initial conditions.

Variables:

PITCH: Pitch angle. (degrees), [R,S,D=0]

ROLL: Roll angle. (degrees), [R,S,D=0]

YAW: Yaw angle. (degrees), [R,S,D=0]

CG: Center of gravity with respect to the mean aerodynamic chord. (feet), [R,S,D=20.0]

RINLO: Longitude to CG of the A/C (degrees) [R,S,D=0]

RINLA: Latitude to CG of the A/C (degrees) [R,S,D=0]

AIRSPE: Airspeed (knots) [R,S,D=0]

HIGHT: Vertical height measured to the CG of the A/C (feet), [R,S,D=0]

WEIGHT: Total initial weight of the A/C (pounds) [R,S,D=90000.0]

RHO: Atmospheric density (slugs/ft^3) [R,S,D=0.002308119]

THRUST: Thrust (pounds) [R,S,D=13900.0]

RE: Radius of the earth (feet), [R,S,D=20,925,705.0]

PDEG: Pitch rate (degrees/second) [R,S,D=0]

QDEG: Roll rate (degrees/second) [R,S,D=0]

RDEG: Yaw rate (degrees/second) [R,S,D=0]

NAMELIST: WINDAT

Description: WINDAT contains quantities necessary to define the wind models simulated.

Comments: See [11] for a discussion of wind models.

Variables:

GUST: Flag to indicate if the gust model is to be simulated. Where, gust=.true. => generate gusts using a Dryden gust model. (unitless), [L,S,D=.FALSE.]

IGUST: Indicates the nature of the simulated gusts (if gust=True). Where igust = 0 => clear air turbulence, and igust = 1 => thunderstorm-like turbulence. (unitless), [I,S,D=0]

SCALEW: Scale factor for the u component of the Dryden gust model. (feet), [R,S,D=1400.0]

SDSUBW: Standard deviation of the gust winds (feet/s), [R,S,D=4.0]

SHEAR: Flag to indicate if shear winds are to be simulated. Where, if shear=.True., shear winds are generated. (unitless), [L,S,D=.False.]

SH1: Shear gradient if the A/C's altitude is between 200 and 100 feet. (feet/sec/foot). [R,S,D=-0.0675]

SH2: Shear gradient if the A/C's altitude is between 100 feet and the ground. (feet/sec/foot). [R,S,D=-0.135]

WINDSS: Magnitude of the steady state horizontal wind. (knots), [R,S,D=0.0]

PSIW: Angular direction of the steady state winds. Measured positive CCW to the runway heading. (degrees), [R,S,D=150.00]

NAMELIST: GSLOC

Description: GSLOC contains the quantities necessary to specify the glideslope and localizer geometries.

Variables:

GPIPRN: Glide path intercept point measured from the runway origin. In other words, this is the distance along the runway center line to the point where the glideslope beam would intercept it. (feet) [R,S,D=1000.0]

LOCRAN: distance from the runway origin (threshold of runway) to the localizer antenna. (feet) [R,S,D=10000.0]

PSIR: Runway heading measured positive CCW to true north. (degrees) [R,S,D=0.0]

THETAG: Glide slope angle measured (positive) from the runway to the glideslope beam. (degrees) [R,S,D=3.0]

NAMELIST: STARTER

Description: STARTER contains the quantities necessary to specify the desired nominal route the aircraft is to follow.

Comments: Nominal route construction is also referred to as waypoint path construction. A waypoint is a point in space which marks a change in the nominal route. In this version of the program, waypoints are only used to mark three conditions:

1. The start of the route. This is required to be the first waypoint. The first leg of the route is further required to be a straight line segment.

2. Changes in A/C heading.

3. The 3-D touchdown point, that is, the desired point on the runway at touchdown. This waypoint is required to be the last waypoint specified.

How can the user change the nominal altitude or velocity profile? These quantities can be specified at waypoints and a path is determined based on them, but they must be associated with a change in heading in this version of the program. That is, one cannot institute a change from level flight to one of constant sink rate, unless it also involves heading change. Therefore, one must plan the route carefully, based on the above constraints.

The following preliminary steps are normally required before use of STARTER

1. Draw the desired ground track in the runway (rw) plane, noting the (x,y,z) coordinates of path intercept points, along with glideslope and velocity changes.

2. Obtain the latitude, longitude and mean sea level (MSL) to the origin of rw frame.

3. Transform all points in the rw frame to appropriate latitude, longitude and altitude

Additional details on waypoint path construction can be found in [3]-[5].

variables:

NW: The total number of waypoints to be specified, where: 2 < NW < 10. (unitless), [I,S,D=none]

WLO: Longitude of each waypoint. (degrees), [R,V=NW,D=none]

WLA: Latitude of each waypoint. (degrees), [R,V=NW,D=none]

SLO: Longitude of Vortac (not used in FINDS 3.0) and MLS frame origins respectively. (degrees), [R,V=2,D=none]

SLA: Latitude of Vortac (not used in FINDS 3.0) and MLS frame origins respectively (degrees), [R,V=2,D=none]

SHZ: Mean sea level altitude of Vortac (not used in FINDS 3.0) and MLS frame origins respectively. (feet), [R,V=2,D=none]

HG: Mean sea level altitude of each waypoint. (feet), [R,V=NW,D=none]

VGD: Desired airspeed at each waypoint. NOTE: not ground speed. (knots), [R,V=NW,D=none]

RT: Radius of turn at each intermediate waypoint. There are no "RT" items for the first and last waypoints. (feet), [R,V=NW-2,D=none]

NEWPTH: Flag to indicate waypoint modification at transition to MLS coverage, where:

0=> continue with original path after MLS transition (unitless), [I,S,D=0]

1=> introduce a new waypoint at MLS transition and construct a new path to next waypoint

KPATH: Flag to indicate the type of waypoint path construction used, where:

0=> zero cross-track error method. This attempts to null out the cross track error at the expense of ground track angle errors. Therefore, the A/C is forced to follow the nominal ground track. This method normally exhibits the most oscillatory behavior.

1=> Tangent path update method. This method attempts to construct a path from the current A/C position to a tangent point on the next waypoint turn circle. This method is smoother than the previous method in general, since it allows for some deviation from nominal. One characteristic of the method is that the total turn angle can be different than the nominal turn angle.

2=> Continued track method. This method extrapolates the current segment and determines its intersection with the next (straight line) segment. It then simply redefines the next waypoint to be the computed intersection point. This method is characteristically the smoothest. (unitless), [I,S,D=none]

IC: Flags which indicate whether the waypoint longitudes and latitudes (WLO & WLA) are to be interpreted as the centers of turns or as the intercept point of the current path with the next path, where: 0 => corner point; 1 => center of turn. (unitless), [I,V=NW-2,D=none]

TANLIM: If KPATH=1, and the distance to the next turn is less than TANLIM at the point of MLS interception, then the program will default to KPATH=0 or KPATH=1. [R,S,D=3000.0]

SNDP: Indicator of the sign of the heading change to each intermediate waypoint path segment. For example, SNDP = -1.0 would indicate that heading will be decreasing (left turn) while turning onto the next path segment. (unitless), [R,V=NW-2,D=none]

3.3 Sensor Input Parameter File

This section describes the namelist directed inputs contained in the SENSOR.INP file. Each namelist defines the parameters necessary to simulate the normal operation or various failure modes of a given sensor.

Since, by design, there is considerable structure and similarity in the sensor modules, the following table of mnemonics should aid the reader in better understanding the individual variables to be defined.

TABLE 2. Sensor Module Mnemonics

Namelist name prefixes:

MP:	Model Parameters
FP:	Failure model Parameters
FDI:	Failure Detection and Isolation

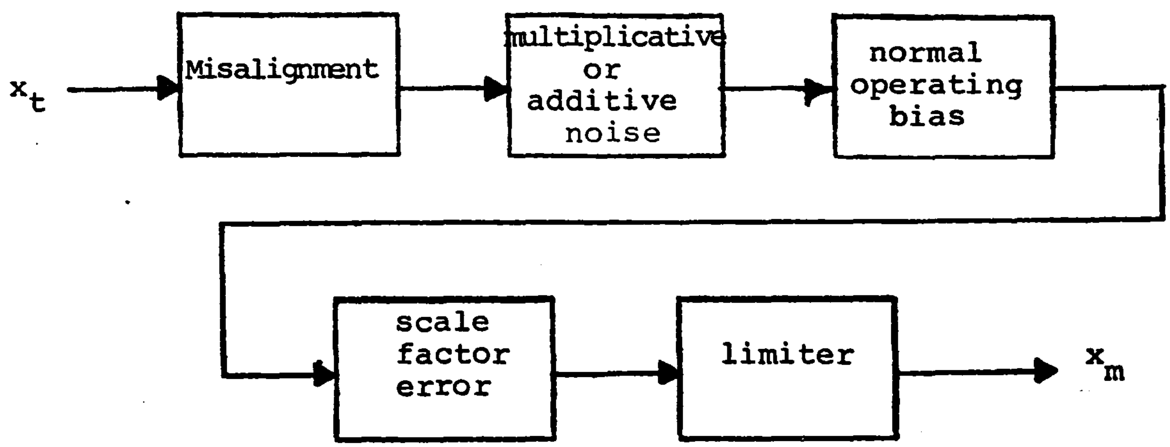
Sensor module identifiers, variable suffixes:

RA:	Radar Altimeter
AS:	Indicated Air Speed
RG:	body mounted Rate Gyro
LA:	body mounted Linear Accelerometers
AG:	inertial measurement unit Attitude Gyro
RA:	Radar Altimeter
ML or MLS:	Microwave Landing System
GR:	rate Gyros contained in the RSDIMU
AR:	linear Accelerometers contained in the RSDIMU

Variable name prefixes:

ID:	Initialization iDentifier
IREP:	Integer REPlication factor
DT:	sample time (dt)
SDN:	Standard Deviation of the white Noise
SDMN:	Standard Deviation of the Multiplicative Noise
SDB:	Standard Deviation of the normal operating Biases
SDSF:	Standard Deviations of the Scale Factor biases
SLIM:	Stop LIMits
SDTM:	Standard Deviations of the misalignment Transformation Matrix
FIN:	Failure level for Increased Noise failures
FIB:	Failure level for Increased Bias failures
FR:	Failure level for Ramp failures
FSF:	Failure levels for Scale Factor failures
TF:	Time of Failure matrix (scheduled failures)

Using the mnemonics defined in Table 2 we will define a generic sensor model, and a generic failure model. The generic sensor model will define the normal operation of all but two of the simulated sensors. The two sensor modules which do not adhere to this model are the MLS and the RSDIMU sensors. These two were both NASA-supplied and will therefore be explained separately and briefly. The failure model to be defined is used by <u>all</u> the sensor modules. Both generic models define features which can be simulated; however, not all the sensors utilize the full set of features. Variables defined in the individual namelist description will make it clear what subset of the generic models are actually being simulated for any given sensor.



The generic sensor model is shown in a functional block diagram in Figure 5 where we define:

Xt = true value of the quantity to be sensed (vector variable in general).

Xm = measured value of Xt.

Misalignments are produced via:

$$Xm = TM * Xt \tag{1}$$

where TM = misalignment transformation matrix of the form (assuming Xt is third order):

$$TM = \begin{bmatrix} 1 & -\theta_{yz} & \theta_{zy} \\ \theta_{xz} & 1 & \theta_{zx} \\ -\theta_{xy} & \theta_{yx} & 1 \end{bmatrix} \tag{2}$$

where each of the small misalignment angles are chosen as:

θ = SDTM if ID=1 (i.e., deterministic initialization)
or;
θ = SDTM*sample, if ID=2 (i.e., random initialization)

"Sample" is a sample from a normal, zero mean unit variance random number generator.

Multiplicative or additive noises are added to Xm by:

$$Xm = (1+MNOISE)*Xm \tag{3}$$

or,

$$Xm = Xm + ANOISE \tag{4}$$

where MNOISE and ANOISE represent multiplicative and additive noise terms chosen as:

```
MNOISE = SDMN, and
ANOISE = SDN              if ID=1
or,
MNOISE = SDMN*sample, and
ANOISE = SDN*sample       if ID=2.
```

Normal operating biases are then added via:

$$Xm = Xm + BIAS \tag{5}$$

where BIAS is chosen as:

```
BIAS = SDB                if ID=1,
or,
BIAS = SDB*sample         if ID=2.
```

Scale factor errors are added as:

$$Xm = (1+SF)Xm \tag{6}$$

where SF is the scale factor error term chosen as:

```
SF = .01*SDSF             if ID=1,
or,
SF = .01*SDSF*sample      if ID=2.
```

The resulting signal is then processed by either a two-sided symmetric limiter:

$$\begin{aligned} Xm &= -SLIM && \text{if } Xm < -SLIM \\ Xm &= Xm && \text{if } -SLIM \leq Xm \leq SLIM \\ Xm &= SLIM && \text{if } XM > SLIM \end{aligned} \tag{7}$$

or a two-sided anti-symmetric limiter:

$$\begin{aligned} Xm &= SLIM(1) && \text{if } Xm < SLIM(1) \\ Xm &= Xm && \text{if } SLIM(1) \leq Xm \leq SLIM(2) \\ Xm &= SLIM(2) && \text{if } Xm > SLIM(2). \end{aligned} \tag{8}$$

Equations (1)-(8) completely define the general form of the sensor modules.

An important feature of the sensor module defined above, is the parameter "ID". This parameter allows the user to set the sensor's model paramters in a random (appropriate for most is instances) or deterministic fashion (useful for debugging or forcing specific settings).

The generic failure model, mentioned earlier is shown functionally in Figure 6. Notice that increased noise, bias, scale factor and ramp failures are processed before the sensor model, while hardover and null failures are added after the measurement is simulated. The reason for this, as we will see, is because the first four failure types actually modify the sensor's model parameters used in the simulation.

In order to determine if it is time to simulate a failure, TF is monitored by the program at every simulation step. TF determines when , how and what specific sensor will fail. It therefore serves as a mapping between sensor type/replication, failure mode and time of failure. This mechanism is easiest to see by way of example. Let us assume we would like to simulate a bias failure in the second replication of the radar altimeter measurement. We would see by reading the description of namelist FPRA found on page 36 that TFRA is a matrix arranged (symbolically) as:

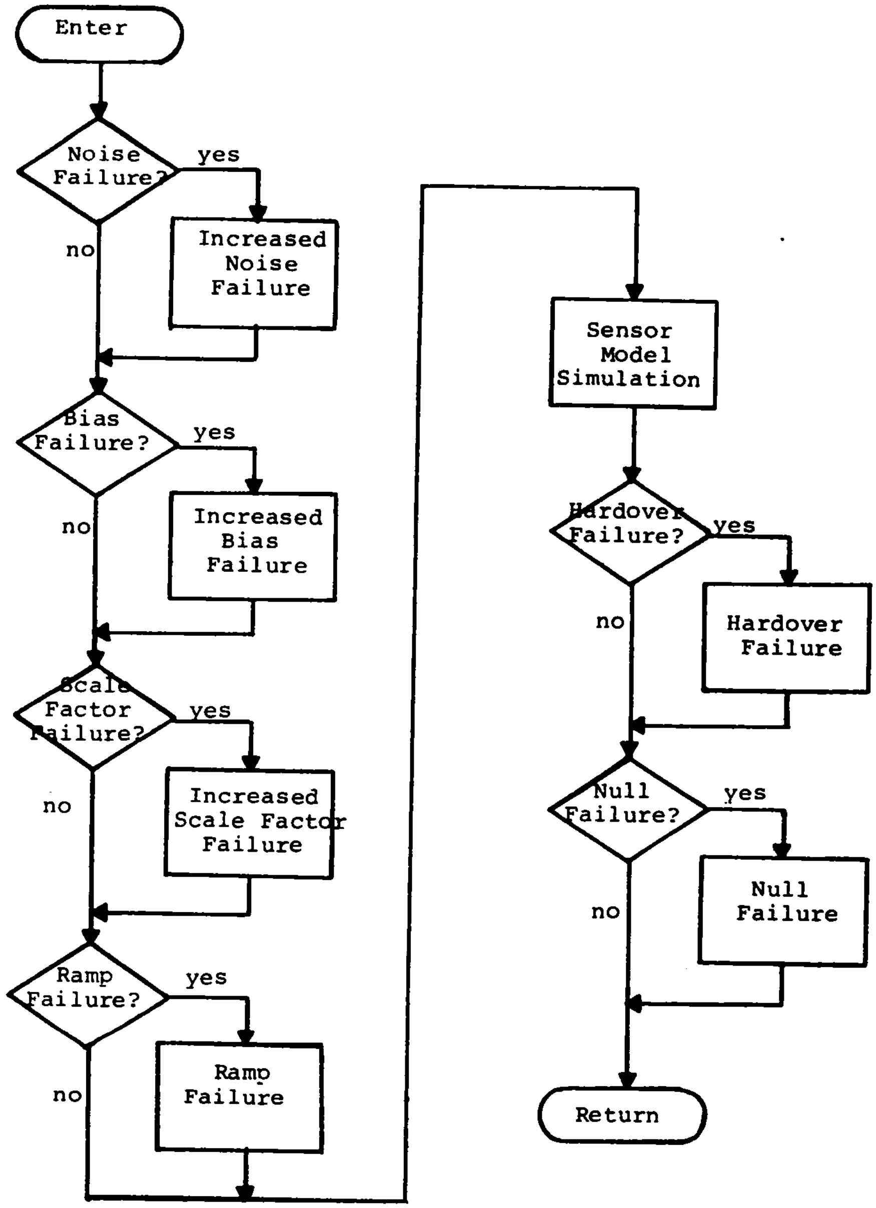


FIG. 6. Functional Block Diagram of a Generic Failure Model

	NOISE	BIAS	S.F.	HARDOVER	NULL	RAMP
RA_1	.	.	.	.	.	.
RA_2	.	Tfail	.	.	.	.
RA_3	.	.	.	.	.	.

Where the elements of TF are desired failure times. The symbol "." signifies the default (program supplied) value, which is greater than the desired simulation stopping time. Therefore, if no specification is made by the user in TF, then no failures will be simulated. To simulate a failure, the desired failure time (Tfail) should be set into the element of TFRA, whose row index corresponds to the desired failure type/replication, and whose column index corresponds to the desired failure mode. In our example, TFRA(2,2) = Tfail would accomplish this.

The actual time of failure, used by the program, is determined by:

$$TF = TF \qquad \text{if } ID=1 \tag{9}$$
or
$$TF = |TF| + INTERVAL*USAMPLE \qquad \text{if } ID = 2, \text{ or } TF < 0.0$$

Where SAMPLE is a random sample from a uniform [0,1] distribution, and INTERVAL is currently 6 seconds. Therefore, if random initialization is choosen (ID=2) the time of failure will be between TF and TF + 6 seconds.

Once the program has determined that it is time for a given failure to occur (time $\geq$ TF(i,j)) one of the following operations are performed:

- o Increased noise failure:

$$MNOISE = SDMN*FIN \tag{10}$$

if multiplicative noise is being simulated or,

$$ANOISE = SDN*FIN \tag{11}$$

if additive noise is simulated in the sensor model.

- Increased Bias failure:

$$BIAS = SDB*FIB \tag{12}$$

- Increased Scale factor failure:

$$SF = SDSF*FSF \tag{13}$$

- Ramp failure:

$$BIAS_{k+1} = BIAS_k + DT*FR \tag{14}$$

where k and k+1 refer to the time scale.

- Hardover failure:

$$Xm = SLIM \qquad \text{if } Xm \geq 0 \tag{15}$$

$$Xm = -SLIM \qquad \text{if } Xm < 0 \tag{16}$$

o Null failure

$$Xm = 0.0 \tag{17}$$

Equations 9-17 completely describe the generic failure model employed.

The MLS simulation module, developed by NASA, is described in detail in [3], and [5]. The MLS model is sufficiently complex that it is beyond the scope of this report; however, in order to give the reader some notion of the simulation model, a brief description is included.

The MLS ground system basically consists of an azimuth antenna and elevation antenna. The azimuth antenna is usually located on the runway centerline beyond the stop end, and the elevation antenna is located on either side of the runway in the vicinity of the glidepath intercept point (GPIP). The azimuth beam sweeps "TO" and "FRO", up to $\pm 60^{\circ}$ about the runway centerline and the elevation beam sweeps up and down similarly between 0 and 20° of coverage. The MLS also consists of a DME antenna to give precision range information which is colocated with the azimuth antenna.

The airborne equipment consists of a receiving antenna, an angle receiver for the elevation and azimuth signal detection, and a DME receiver for the range signal. The receiver measurements are digital in nature ($\sim$ 13 Hz for Azimuth, 40 Hz for elevation and 30 Hz for range).

The MLS simulation model computes the azimuth, elevation and range signals based on the aircraft's position and velocity in runway rectangular coordinates. The model also determines whether the A/C is within coverage of the ground antenna.

The MLS has a time multiplex format in which the antenna scans are not separated at equal time intervals. This time shift is referred to as "jitter". The model computes time increments which represent the signal decoded time relative to current simulation time. Given these time increments, the aircraft's position is determined for the time when the receiver decoded the signal. This is done only for the azimuth and elevation signals. From the position corrected due to time shift, the model computes the azimuth, elevation and range measurement. Time correlated noise, generated by passing Gaussian random numbers with zero mean and standard deviation one through a linear filter, is added to the various signals along with a constant bias. These parameters, as well as many of those mentioned in the previous paragraphs, can be specified by the user at run time.

The model generates frame flags and outlier data (spurious spikes) by use of a random number generator with uniform distribution. These same frame flags are also set when the aircraft is out of a given signal coverage.

The RSDIMU is an experimental redundant strapdown inertial measurement unit developed by NASA Langley Research Center. The unit consists of four two-degree-of-freedom (TDOF) tuned-rotor gyros, and four TDOF pendulous accelerometers in a skewed and separable semi-octahedron array. When coupled to flight computers, the RSDIMU becomes a self-contained, flightworthy inertial navigation system providing medium range accuracy with fail-op/fail/op capability as described in [6], [10] and [8].

The RSDIMU sends to the external flight computers gyro and accelerometer data resolved along the ideal instrument fixed coordinate reference set. Spin axis velocity and earth-rate terms provided to the IMU from the flight control computers are used for calibration and compensation.

The RSDIMU has its own local failure detection and isolation scheme (parity checks, edge vector test, or generalized likelihood ratio test) which can survive two gyro and two accelerometer failures [6], as well as redundancy management logic.

NAMELIST: MPRA

Description: MPRA specifies all model parameters necessary for simulation of the radar altimeter sensors.

Variables:

IDRA: Flag enabling deterministic or random parameter initialization. Where IDRA=2 normal operating biases are randomly selected, and IDRA=1 & TOF deterministic initialization is performed. (unitless), [I,S,D=2].

IREPRA: The desired number of replications to be simulated. Currently, $0 \leq IREPRA \leq 3$. (unitless), [I,S,D=1].

DTRA: The desired sample time for radar altimeter measurements. Note: This feature is currently disabled in FINDS version 3.0. (seconds), [R,S,D=DTIME].

SDNRA: Standard deviations of the additive noises. One value for each replication. (meters), [R,V=IREPRA,D=3*0.3048].

SDBRA: Standard deviations of a zero mean Gaussian distribution from which the normal operating biases are chosen, if IDRA=2. Otherwise, if IDRA=1, SDBRA represents the actual normal operating bias levels. One value for each replication. (meters), [R,V=IREPRA,D=3*0.3048].

SLIMRA: Symmetric stop limits on the radar altimeter signal generated. Note: this feature is currently disabled in FINDS version 3.0. However, SLIMRA is used as the hardover failure level. (See NAMELIST FPRA). (meters), [R,S;D=10000.0].

NAMELIST: FPRA

Description: To specify all the desired failure modes and onset times for the radar altimeter sensors. Currently allowed failure modes include: increased noise, increased bias, hardover, null and ramp.

Variables:

FINRA: Failure level for increased noise type failures. Specified as the number of standard deviations of the normal (1σ) noise level of the first replication (SDNRA(1)). Note if SDNRA(1)=0.0 (i.e. noiseless) then FINRA will be interpreted as the <u>actual</u> failure level. (unitless, or meters), [R,S,D=10.0]

FIBRA: Failure level for increased bias type failures. Specified by number of standard directions of the (1σ) normal operating bias level of the first replication (SDBRA(1)). Note if SDBRA(1) = 0.0 (i.e., no bias) then FIBRA will be interpreted as the <u>actual</u> failure level. (unitless, or meters), [R,S,D=5.0].

FRRA: Failure level for ramp failures. Specified by the slope of the ramp failure. (meters/sec) [R,S,D=0.0].

TFRA: Time of failure matrix used to select the sensor and failure mode to be simulated. Where the rows of TFRA represent the replication number, and the columns represent the failure modes ordered as: increased noise, bias, hardover, null, and ramp failures respectively. (seconds), [R,M=IREPRA:5, D=15*(TSTOP+DTIME)]

NAMELIST: MPAS

Description: MPAS specifies all model parameters necessary for simulation of the airspeed sensors.

Variables:

IDAS: Flag enabling deterministic or random parameter initialization. If IDAS=2 normal operating biases are randomly selected. If IDAS=1 deterministic initialization is performed. (unitless), [I,S,D=2].

IREPAS: The desired number of replications to be simulated. Currently, $0 \leq IREPAS \leq 3$. (unitless), [I,S,D=1].

DTAS: The desired sample time for airspeed measurements. Note: This feature is currently disabled in FINDS version 3.0. (seconds), [R,S,D=DTIME].

SDMNAS: Standard deviations of the multiplicative noises. One value for each replication. (percent), [R,V=IREPAS,D=3*2.0].

SDBAS: Standard deviations of a zero mean Gaussian distribution from which the normal operating biases are chosen, if IDAS=2. Otherwise, if IDAS=1, SDBAS represents the <u>actual</u> normal operating bias levels. One value for each replication. (meters/s), [R,V=IREPAS,D=3*1.0].

SLIMAS: <u>Asymmetric</u> stop limits on the airspeed signal generated. Note: SLIMAS is used as the hardover failure level. (See NAMELIST FPAS). (meters/s), [R,S;D=400.0,100.0].

NAMELIST: FPAS

Description: To specify all the desired failure modes and onset times for the airspeed sensors. Currently allowed failure modes include: increased noise, increased bias, hardover, null and ramp.

Variables:

FINAS: Failure level for increased <u>multiplicative</u> noise type failures. Specified as the number of standard deviations of the normal (1σ) noise level of the first replication (SDNAS(1)). Note if SDNAS(1)=0.0 (i.e. noiseless) then FINAS will be interpreted as the <u>actual</u> failure level. (unitless, or percent), [R,S,D=10.0]

FIBAS: Failure level for increased bias type failures. Specified by number of standard deviations of the (1σ) normal operating bias level of the first replication (SDBAS(1)). Note if SDBAS(1) = 0.0 (i.e., no bias) then FIBAS will be interpreted as the <u>actual</u> failure level. (unitless, or meters/s), [R,S,D=5.0].

FRAS: Failure level for ramp failures. Specified by the slope of the ramp failure. (meters/s/s) [R,S,D=0.0].

TFAS: Time of failure matrix used to select the sensor and failure mode to be simulated. Where the rows of TFAS represent the replication number, and the columns represent the failure modes ordered as: increased noise, bias, hardover, null, and ramp failures respectively. (seconds), [R,M=IREPAS:5, D=15*(TSTOP+DTIME)]

NAMELIST: MPRG

Description: MPRG specifies all model parameters necessary for simulation of the body mounted rate gyros.

Comments: MPRG supplies parameters for three different types of rate gyros (one for each axis), and up to three (3) replications of each type. The types are: pitch rate (P), roll rate (Q), and yaw rate (R) gyros. Vector valued variables referenced in this section are ordered in one of the following ways:

1. $[P_1,Q_1,R_1;\ P_2,Q_2,R_2;\ P_3,Q_3R_3]$. Where the subscript indicates the replication number.

2. [P,Q,R]

3. $[\theta_{xy},\theta_{xz},\ \theta_{yx},\ \theta_{yz},\ \theta_{zx},\ \theta_{zy}]$

Variables:

IDRG: Flag enabling deterministic or random parameter initialization. If IDRG=2 => normal operating biases, scale factor errors, and the misalignment transformation are all randomized. If IDRG=1=> deterministic initialization is performed. (unitless), [I,S,D=2].

IREPRG: The desired number of replications to be simulated. Currently $0 \leq$ IREPRG ≤ 3. (unitless), [I,S,D=1]

DTRG: The desired sample time for rate gyro measurements. Note: this feature is currently disabled in FINDS version 3.0. (seconds), [R,S,D=DTIME].

SDNRG: Standard deviations of the additive noises. One value for each type and replication (See 1. in Comments). (degrees/s), [R,V=IREPRG*3,D=9*0.02].

SDBRG: Standard deviations of a zero mean, Gaussian distribution from which the normal operating biases are chosen, if IDRG=2.

Otherwise, if IDRG=1, SDBRG represents the actual normal operating bias levels. One value for each type and replication (see 3.3. in Comments). (degrees/s), [R,V=IREPRG*3, D=9*0.278E-4].

SDSFRG: Standard deviations of a zero mean, Gaussian distribution from which the scale factor errors are chosen, if IDRG=2. Otherwise, if IDRG=1, SDBFRG represents the actual scale factor errors. One value for each type and replication (See 1. in Comments). (percent), [R,V=IREPRG*3,D=0.01].

SLIMRG: Symmetric stop limits on the rate gyro signals. One value for each gyro type. (See 2. in Comments). SLIMRG is also used as the hardover failure level (See also NAMELIST FPRG). (degrees/s), [R,V=3, D=100.0].

SDTMRG: Standard deviations of a zero mean, Gaussian distribution from which the non-unity elements of the misalignment transformation matrix are chosen, if IDRG=2. Otherwise, if IDRG=1, SDTMRG represents the actual elements in the misalignment transformation matrix (See 3. in Comments). (degrees), [R,V=6, D=0.4].

NAMELIST: FPRG

Description: FPRG specifies all the desired failure modes and onset times for the rate gyro sensors. Currently allowed failure modes include: increased noise, increased bias, increased scale factor, hardover, null, and ramp.

Variables:

FINRG: Failure level for increased noise type failures. Specified as the number of standard deviations of the normal (1σ) noise level of the first replication (SDNRG(1)->SDNRG(3)). Note: if SDNRG(i)=0.0 (i.e. noiseless) then FINRG(i) will be interpreted as the actual failure level. One value for each gyro type. (unitless, or degrees/s), [R,V=3, D=3*10.0].

FIBRG: Failure level for increased bias type failures. Specified as the number of standard deviations of the (1σ) normal operating bias level (SDBRG(1)->SDBRG(3)). Note: if SDBRG(i)=0.0 (i.e. no bias) then FIBRG(i) will be interpreted as the actual failure level. One value for each gyro type. (unitless, or degrees/s), [R,V=3, D=3*100.0].

FRRG: Failure level for ramp failures. Specified by the slope of the ramp failure. One value for each gyro type. (degrees/s/s), [R,V=3, D=0.0].

FSFRG: Failure level for scale factor failures. Specified as the number of standard deviations of the normal (1σ) scale factor error level (SDSFRG(1)->SDSFRG(3)). Note: if SDSFRG(i)=0.0 (i.e. no scale factor errors) then FSFRG(i) will be interpreted as the actual failure level. One value for each gyro type. (unitless, or degrees/s), [R,V=3, D=100.0].

TFRG: Time of failure matrix used to select the sensor and failure mode to be simulated. Where the rows of TFRG correspond to the sensor type/replication number and the columns represent the failure modes. The rows are ordered as per item 1. on page 38, and the columns are ordered: increased noise, bias, scale factor, hardover, null, and ramp failures. (seconds), [R,M=IREPRG*3:5, D=54*(TSTOP+DTIME)].

NAMELIST: MPLA

Description: MPLA specifies all model parameters necessary for simulation of the body mounted linear accelerometers.

Comments: MPLA supplies parameters for three different types of linear accelerometers (one for each axis), and up to three (3) replications of each type. The types are: forward (Ax), lateral (Ay), and vertical (Az) accelerometers. Vector valued variables referenced in this section are ordered in one of the following ways:

1. $[Ax_1, Ay_1, Az_1;\ Ax_2, Ay_2, Az_2;\ Ax_3, Ay_3, Az_3]$. Where the subscript indicates the replication number.

2. [Ax,Ay,Az]

3. $[\theta_{xy}, \theta_{xz}, \theta_{yx}, \theta_{yz}, \theta_{zx}, \theta_{zy}]$

Variables:

IDLA: Flag enabling deterministic or random parameter initialization. If IDLA=2 => normal operating biases, scale factor errors, misalignment transformation angles and TOF are all randomized. If IDLA=1=> deterministic initialization is performed. (unitless), [I,S,D=2].

IREPLA: The desired number of replications to be simulated. Currently $0 \leq$ IREPLA ≤ 3. (unitless), [I,S,D=1]

DTLA: The desired sample time for accelerometer measurements. Note: this feature is currently disabled in FINDS version 3.0. (seconds), [R,S,D=DTIME].

SDNLA: Standard deviations of the additive noises. One value for each type and replication (See 1. in Comments). (g's), [R,V=IREPLA*3,D=9*0.01].

SDBLA: Standard deviations of a zero mean, Gaussian distribution from which the normal operating biases are chosen, if IDLA=2. Otherwise, if IDLA=1, SDBLA represents the actual normal operating bias levels. One value for each type and replication (see 1 in Comments). (g's), [R,V=IREPLA*3, D=9*0.01].

SDSFLA: Standard deviations of a zero mean, Gaussian distribution from which the scale factor errors are chosen, if IDLA=2. Otherwise, if IDLA=1, SDBFLA represents the actual scale factor errors. One value for each type and replication (See 1. in Comments). (percent), [R,V=IREPLA*3,D=0.25].

SLIMLA: Symmetric stop limits on the rate gyro signals. One value for each gyro type. (See 2. in Comments). SLIMLA is also used as the hardover failure level (See also NAMELIST FPLA). (g's), [R,V=3, D=2*0.5,2.0].

SDTMLA: Standard deviations of a zero mean, Gaussian distribution from which the non-unity elements of the misalignment transformation matrix are chosen, if IDLA=2. Otherwise, if IDLA=1, SDTMLA represents the actual elements in the misalignment transformation matrix (See 3. in Comments). (degrees), [R,V=6, D=0.36].

NAMELIST: FPLA

Description: FPLA specifies all the desired failure modes and onset times for the linear accelerometer sensors. Currently allowed failure modes include: increased noise, increased bias, increased scale factor, hardover, null, and ramp.

Variables:

FINLA: Failure level for increased noise type failures. Specified as the number of standard deviations of the normal (1σ) noise level of the first replication (SDNLA(1)->SDNLA(3)). Note: if SDNLA(i)=0.0 (i.e. noiseless) then FINLA(i) will be interpreted as the actual failure level. One value for each accelerometer type. (unitless, or g's), [R,V=3, D=3*10.0].

FIBLA: Failure level for increased bias type failures. Specified as the number of standard deviations of the (1σ) normal operating bias level (SDBLA(1)->SDBLA(3)). Note: if SDBLA(i)=0.0 (i.e. no bias) then FIBLA(i) will be interpreted as the actual failure level. One value for each accelerometer type. (unitless, or g's), [R,V=3, D=3*10.0].

FRLA: Failure level for ramp failures. Specified by the slope of the ramp failure. One value for each accelerometer type. (g's/s), [R,V=3, D=0.0].

FSFLA: Failure level for scale factor failures. Specified as the number of standard deviations of the normal (1σ) scale factor error level (SDSFLA(1)->SDSFLA(3)). Note: if SDSFLA(i)=0.0 (i.e. no scale factor errors) then FSFLA(i) will be interpreted as the actual failure level. One value for each accelerometer type. (unitless, or g's), [R,V=3, D=100.0].

TFLA: Time of failure matrix used to select the sensor and failure mode to be simulated. Where the rows of TFLA represent the sensor type/replication number and the columns represent the failure modes. The rows are ordered as per 1. on page 40, and the columns are ordered: increased noise, bias, scale factor, hardover, null, and ramp failures. (seconds), [R,M=IREPLA*3:5, D=54*(TSTOP+DTIME)].

NAMELIST: MPAG

Description: MPAG specifies all model parameters necessary for simulation of the body mounted attitude gyros.

Comments: MPAG supplies parameters for three different types of attitude gyros (one for each axis), and up to three (3) replications of each type. The types are: pitch (ϕ), roll (θ), and yaw (ψ) attitude gyros. Vector valued variables referenced in this section are ordered in one of the following ways:

1. $[\phi_1,\theta_1,\psi_1;\ \phi_2,\theta_2,\psi_2;\ \phi_3,\theta_3,\psi_3]$. Where the subscript indicates the replication number.

2. [ϕ,θ,ψ]

Variables:

IDAG: Flag enabling deterministic or random parameter initialization. If IDAG=2 => normal operating biases are randomized. If IDAG=1=> deterministic initialization is performed. (unitless), [I,S,D=2].

IREPAG: The desired number of replications to be simulated. Currently $0 \leq$ IREPAG ≤ 3. (unitless), [I,S,D=1]

DTAG: The desired sample time for attitude gyro measurements. Note: this feature is currently disabled in FINDS version 3.0. (seconds), [R,S,D=DTIME].

SDNAG: Standard deviations of the additive noises. One value for each type and replication (See 1. in Comments). (degrees), [R,V=IREPAG*3,D=9*0.23].

SDBAG: Standard deviations of a zero mean, Gaussian distribution from which the normal operating biases are chosen, if IDAG=2. Otherwise, if IDAG=1, SDBAG represents the actual normal operating bias levels. One value for each type and replication (See 1. in Comments). (degrees), [R,V=IREPAG*3, D=9*0.1].

SLIMAG: Symmetric stop limits on the attitude gyro signals. One value for each gyro type. (See 2. in Comments). SLIMAG is also used as the hardover failure level (See also NAMELIST FPAG). (degrees), [R,V=3, D=2*80.0,600.0].

NAMELIST: FPAG

Description: FPAG specifies all the desired failure modes and onset times for the attitude gyro sensors. Currently allowed failure modes include: increased noise, increased bias, hardover, null, and ramp.

Variables:

FINAG: Failure level for increased noise type failures. Specified as the number of standard deviations of the normal (1σ) noise level of the first replication (SDNAG(1)->SDNAG(3)). Note: if SDNAG(i)=0.0 (i.e. noiseless) then FINAG(i) will be interpreted as the actual failure level. One value for each gyro type. (unitless, or degrees), [R,V=3, D=3*10.0].

FIBAG: Failure level for increased bias type failures. Specified as the number of standard deviations of the (1σ) normal operating bias level (SDBAG(1)->SDBAG(3)). Note: if SDBAG(i)=0.0 (i.e. no bias) then FIBAG(i) will be interpreted as the actual failure level. One value for each gyro type. (unitless, or degrees), [R,V=3, D=3*5.0].

FRAG: Failure level for ramp failures. Specified by the slope of the ramp failure. One value for each gyro type. (degrees/s), [R,V=3, D=0.0].

TFAG: Time of failure matrix used to select the sensor and failure mode to be simulated. Where the rows of TFAG represent the sensor type/replication number and the columns represent the failure modes. The rows are ordered as per 1 on page 42 and the columns are ordered: increased noise, bias, scale factor, hardover, null, and ramp failures. (seconds), [R,M=IREPAG*3:5, D=45*(TSTOP+DTIME)].

NAMELIST: MPMLS

Description: MPMLS specifies all model parameters necessary for simulation of the microwave landing system (MLS) receivers.

Comments: MPMLS supplies parameters for three different MLS measurement types, and up to three (3) replications of each type. The types are: MLS azimuth angle (Azm), elevation angle (El), and range (Rng). Vector valued variables referenced in this section are ordered in one of the following ways:

1. [Azm_1,El_1,Rng_1;Azm_2,El_2,Rng_2; Azm_3,El_3,Rng_3]. Where the subscript indicates the replication number.

2. [Azm,El,Rng]

3. [RNG_{min},θ_{az}, θ_{el}] where

RNG_{min}=>minimum distance from the antenna to coverage measured along the runway. $RNG_{min} \geq 0$

θ_{az}=>azimuth coverage of antenna is $\pm\theta_{az}$

θ_{el}=>elevation coverage of antenna is $\pm\theta_{el}$

Variables:

IDML: Flag enabling deterministic or random parameter initialization. If IDML=2 => normal operating biases & TOF are randomized. If IDML=1=> deterministic initialization is performed. (unitless), [I,S,D=2].

IREPML: The desired number of replications to be simulated. Currently $0 \leq$ IREPML ≤ 3. (unitless), [I,S,D=1]

IWHMLS: Flag to indicate if white or colored MLS noise is desired. Where, IWHMLS=0=> colored noise will be simulated, and IWHMLS≠0=> white noise will be used. (unitless), [I,S,D=0].

DTML: The desired sample time for MLS measurements. Note: this feature is currently disabled in FINDS version 3.0. (seconds), [R,S,D=DTIME].

SDNML: Standard deviations of the additive noises. One value for each type and replication (See 1. in Comments). (2*degrees,m), [R,V=IREPML*3,D=3*{2*0.03,3.0}].

SDBML: Standard deviations of a zero mean, Gaussian distribution from which the normal operating biases are chosen, if IDML=2. Otherwise, if IDML=1, SDBML represents the actual normal operating bias levels. One value for each type and replication (see 1 in Comments). (2*degrees,m), [R,V=IREPML*3, D=3*{2*0.03,14.0}].

XAZ: Location of the MLS azimuth antenna in the runway frame. (feet), [R,V=3, D=8546.8, 2*0.0].

XEL: Location of the MLS elevation antenna in the runway frame. (feet), [R,V=3,D=1000.0,254.78,04.7].

BOUNAZ: Limits on the coverage of the azimuth antenna. Ordered as 3 in comments. (feet,2*degrees), [R,V=3,D=100.0,60.0,20.0].

BOUNEL: Limits on the coverage of the elevation antenna. Ordered as 3 in comments. (Feet,2*degrees), [R,V=3,D=50.0,70.0,20.0].

BOUNRN: Limits on the coverage of the range antenna. Ordered as 3 in comments. (feet,2*degrees), [R,V=3,D=100.0,60.0,20.0].

AC: Probability of not "dropping" or missing any measurements. This can also be interpreted as the fractional percentage of time when measurements are correctly decoded. AC is also used as the probability of not taking in a "bad" measurement. Ordered as azimuth, elevation and range measurements. $0.0 \leq AC \leq 1.0$. (unitless), [R,V=3,D=3*1000.0].

CA: Number of standard deviations of the normal operating bias to be added to the measurement if a "bad" measurement is simulated. Ordered the same as AC. (unitless), [R,V=3,D=3*0.0].

NAMELIST: FPMLS

Description: FPML specifies all the desired failure modes and onset times for the MLS sensors. Currently allowed failure modes include: increased noise, increased bias, null, and ramp.

Variables:

FINML: Failure level for increased noise type failures. Specified as the number of standard deviations of the normal (1σ) noise level of the first replication (SDNML(1)->SDNML(3)). Note: if SDNML(i)=0.0 (i.e. noiseless) then FINML(i) will be interpreted as the actual failure level. One value for each MLS type. (unitless, or 2*degrees,m), [R,V=3, D=3*10.0].

FIBML: Failure level for increased bias type failures. Specified as the number of standard deviations of the (1σ) normal operating bias level (SDBML(1)->SDBML(3)). Note: if SDBML(i)=0.0 (i.e. no bias) then FIBML(i) will be interpreted as the actual failure level. One value for each MLS type. (unitless, or 2*degrees,m), [R,V=3, D=3*10.0].

FRML: Failure level for ramp failures. Specified by the slope of the ramp failure. One value for each MLS type. (2*degrees/s,m/s), [R,V=3,D=0.0].

TFML: Time of failure matrix used to select the sensor and failure mode to be simulated. Where the rows of TFML represent the sensor type/replication number and the columns represent the failure modes. The rows are ordered as per 1. on page 44, and the columns are ordered: increased noise, bias, scale factor, hardover, null, and ramp failures.(Even though hardover failures are not implemented). (seconds), [R,M=IREPML*3:5, D=45*(TSTOP+DTIME)].

NAMELIST: MPGR

Description: MPGR specifies all model parameters necessary for simulation of the redundant strapdown inertial measurement unit's (RSDIMU) two degree of freedom gyros.

Comments: MPGR supplies parameters for three different types of rate gyros (one for each axis), and four (4) replications of each type (one for each face of the RSDIMU). The types are: x axis rate (ω_x), and y axis rate (ω_y), where the x and y axis refer to the x and y sensor measurement axis on each face of the RSDIMU. (See [6], [8] or [10] for more detailed descriptions of the RSDIMU). Vector valued variables referenced in this section are ordered in one of the following ways:

1. $[\omega_{x1},\omega_{y1};\omega_{x2},\omega_{y2};\omega_{x3},\omega_{y3};\omega_{x4},\omega_{y4}]$. Where the subscript indicates the RSDIMU face or replication number.

2. $[\theta_{x1},\theta_{y1}, \theta_{x2}, \theta_{y2}, \theta_{x3}, \theta_{y3}\theta_{x4},\theta_{y4}]$

Variables:

IDGR: Flag enabling deterministic or random parameter initialization. If IDGR=2 => normal operating biases, scale factor errors, g-sensitive drift bias, misalignment transformation angles and TOF are all randomized. If IDGR=1=> deterministic initialization is performed. (unitless), [I,S,D=2].

DTGR: The desired sample time for rate gyro measurements. Note: this feature is currently disabled in FINDS version 3.0. (seconds), [R,S,D=DTIME].

IRSOIM: Flag to indicate if the RSDIMU is to be simulated. If IRSOIM=0=> simulate the RSDIMU, otherwise if IRSDIM≠0 => do not run the RSDIMU module. (unitless), [I,S,D=1].

SDNGR: Standard deviations of the additive noises. One value for each type and replication (See 1. in Comments). (degrees/hour), [R,V=8, D=8*0.125].

SDBGR: Standard deviations of a zero mean, Gaussian distribution from which the normal operating biases are chosen, if IDGR=2. Otherwise, if IDGR=1, SDBGR represents the <u>actual</u> normal operating bias levels. One value for each type and replication (See 1. in Comments). (degrees/hour), [R,V=8, D=8*0.015].

SDSFGR: Standard deviations of a zero mean, Gaussian distribution from which the scale factor errors are chosen, if IDGR=2. Otherwise, if IDGR=1, SDBFGR represents the actual scale factor errors. One value for each type and replication (See 1. in Comments). (percent), [R,V=8,D=8*0.0075].

SLIMGR: Symmetric stop limits on the rate gyro signals. One value for each gyro type. (See 2. in Comments). SLIMGR is also used as the hardover failure level (See also NAMELIST FPGR). (degrees/s), [R,V=8, D=8*30.0].

SDTMGR: Standard deviations of a zero mean, Gaussian distribution from which the mounting misalignment angles are chosen, if IDGR=2. Otherwise, if IDGR=1, SDTMGR represents the actual misalignment angles (See 2. in Comments). (degrees), [R,V=16, D=16*0.003333].

SDGDGR: Standard deviation of a zero mean, Gaussian distribution from which the g-sensitive drift biases are chosen, if IDGR=2. Otherwise, if IDGR=1, SDGDGR represents the actual g-sensitive drift bias. (degrees/hour), [R,S,D=0.015].

GRQSF: Scale factor for quantization of the raw gyro measurements. This can be interpreted as the number of quantization levels to represent one radian. (levels/radian), [R,S,D=1.6241323E+5].

NAMELIST: FPGR

Description: FPGR specifies all the desired failure modes and onset times for the RSDIMU rate gyro sensors. Currently allowed failure modes include: increased noise, increased bias, increased scale factor, hardover, null, and ramp.

Comments: Except where noted, the variables defined below are all ordered in accordance with item 1 in the previous comments section on page 47

Variables:

FINGR: Failure level for increased noise type failures. Specified as the number of standard deviations of the normal (1σ) noise level (SDNGR). Note: if SDNGR(i)=0.0 (i.e. noiseless) then FINGR(i) will be interpreted as the actual failure level. (unitless, or degrees/hour), [R,V=8, D=8*5.0].

FIBGR: Failure level for increased bias type failures. Specified as the number of standard deviations of the (1σ) normal operating bias level (SDBGR). Note: if SDBGR(i)=0.0 (i.e. no bias) then FIBGR(i) will be interpreted as the actual failure level. (unitless, or degrees/hour), [R,V=8, D=8*5.0].

FRGR: Failure level for ramp failures. Specified by the slope of the ramp failure. (degrees/s/s), [R,V=8, D=0.0].

FSFGR: Failure level for scale factor failures. Specified as the number of standard deviations of the normal (1σ) scale factor error level(SDSFGR). Note: if SDSFGR(i)=0.0 (i.e. no scale factor errors) then FSFGR(i) will be interpreted as the actual failure level. (unitless, or percent), [R,V=8, D=8*100.0].

TFGR: Time of failure matrix used to select the sensor and failure mode to be simulated. Where the rows of TFGR represent the sensor type/replication number and the columns represent the failure modes. The rows are ordered as per item 1, on page 47 and the columns are ordered: increased noise, bias, scale factor, hardover, null, and ramp failures. (seconds), [R,M=8:5, D=40*(TSTOP+DTIME)].

NAMELIST: FDIGR

Description: FDIGR specifies the failure detection and isolation algorithms to be used by the rate gyros in the RSDIMU module.

Variables:

IPARGR: Flag to indicate what type of parity test to perform. Where: 0=> None, 1=> V parity, 2=> least squares, 3=> edge vector test. (unitless), [I,S,D=3].

IPFOGR: Flag to indicate the desired order of the parity filter. Where 0=> No filter, 1=> first order filter, 2=> second order filter. For example, if IPFOGR=2, then the filter will be: Ha(s)=PFKGR/(s+PFTCGR)**2. (unitless), [I,S,D=1].

PFTCGR: Parity filter time constant selected by IPFOGR. (Sec), [R,S,D=1.0].

PFKGR: Gain of the parity filter selected by IPFOGR. (unitless), [R,S,D=1.0].

PFTHGR: Parity test threshold. (degrees/hour), [R,S,D=20.0].

NAMELIST: MPAR

Description: MPAR specifies all model parameters necessary for simulation of the RSDIMU two-degree-of-freedom linear accelerometers.

Comments: MPAR supplies parameters for two (2) different types of linear accelerometers (one for each axis), and four (4) replications of each type. The types are: x axis (Ax), and y axis (Ay), Where x and y axis refers to the x and y accelerometer measurement axis on each face of the RSDIMU. (See [6], [8] or [10] for more detailed descriptions of the RSDIMU).

Vector valued variables referenced in this section are ordered in the following way:

$[Ax_1, Ay_1, Ax_2, Ay_2, Ax_3, Ay_3, Ax_4, Ay_4]$

Where the subscript indicates the replication number.

Variables:

IDAR: Flag enabling deterministic or random parameter initialization. If IDAR=2 => normal operating biases scale factor errors, are all randomized. If IDAR=1=> deterministic initialization is performed. (unitless), [I,S,D=2].

DTAR: The desired sample time for accelerometer measurements. Note: this feature is currently disabled in FINDS version 3.0. (seconds), [R,S,D=DTIME].

SDNAR: Standard deviations of the additive noises. (g's), [R,V=8, D=8*1.25E-5].

SDBAR: Standard deviations of a zero mean, Gaussian distribution from which the normal operating biases are chosen, if IDAR=2. Otherwise, if IDAR=1, SDBAR represents the <u>actual</u> normal operating bias levels. (g's), [R,V=8, D=8*1.0E-4].

SDSFAR: Standard deviations of a zero mean, Gaussian distribution from which the scale factor errors are chosen, if IDAR=2. Otherwise, if IDAR=1, SDBFAR represents the <u>actual</u> scale factor errors. One value for each type and replication (percent), [R,V=8, D=8*0.0075].

SLIMAR: Symmetric stop limits on the rate gyro signals. One value for each gyro type. SLIMAR is also used as the hardover failure level (See also NAMELIST FPAR). (g's), [R,V=8, D=8*10.0].

NAMELIST: FPAR

Description: FPAR specifies all the desired failure modes and onset times for the RSDIMU linear accelerometer sensors. Currently allowed failure modes include: increased noise, increased bias, increased scale factor, hardover, null, and ramp.

Comments: Except where noted, the variables defined below are all ordered in accordance with the previous comments section, on page 50.

Variables:

FINAR: Failure level for increased noise type failures. Specified as the number of standard deviations of the normal (1σ) noise level of the first replication (SDNAR). Note: if SDNAR(i)=0.0 (i.e. noiseless) then FINAR(i) will be interpreted as the <u>actual</u> failure level. (unitless, or g's), [R,V=8, D=8*5.0].

FIBAR: Failure level for increased bias type failures. Specified as the number of standard deviations of the (1σ) normal operating bias level (SDBAR). Note: if SDBAR(i)=0.0 (i.e. no bias) then FIBAR(i) will be interpreted as the <u>actual</u> failure level. (unitless, or g's), [R,V=8, D=8*5.0].

FRAR: Failure level for ramp failures. Specified by the slope of the ramp failure. (g's/s), [R,V=8, D=0.0].

FSFAR: Failure level for scale factor failures. Specified as the number of standard deviations of the normal (1σ) scale factor error level (SDSFAR). Note: if SDSFAR(i)=0.0 (i.e. no scale factor errors) then FSFAR(i) will be interpreted as the <u>actual</u> failure level. (unitless, or g's), [R,V=8, D=8*100.0].

TFAR: Time of failure matrix used to select the sensor and failure mode to be simulated. Where the rows of TFAR represent the sensor type/replication number and the columns represent the failure modes. The rows are ordered as per the previous comments section, and the columns are ordered: increased noise, bias, scale factor, hardover, null, and ramp failures. (seconds), [R,M=8:5, D=40*(TSTOP+DTIME)].

NAMELIST: FDIAR

Description: FDIAR specifies the failure detection and isolation algorithms to be used by the linear accelerometers in the RSDIMU module.

Variables: Flag to indicate what type of parity test to perform. Where 0=>none, 1=> V parity, 2=> least squares, 3=> edge vector test. (unitless), [I,S,D=0].

IPFOAR: Flag to indicate the desired order of the parity filter. Where 0=> No filter, 1=> first order filter, 2=> second order filter. For example, if IPFOGR=2, then the filter will be: Ha(s)=PFKAR/(s+PFTCAR)**2. (unitless), [I,S,D=1].

PFTCAR: Parity filter time constant selected by IPFOAR. (Sec), [R,S,D=1.0].

PFKAR: Gain of the parity filter selected by IPFOAR. (unit less), [R,S,D=1.0].

PFTHAR: Parity test threshold. (g's), [R,S,D=0.005].

3.4 Filter Input Parameter File

This section describes the namelist directed inputs contained in the FILTER.INP file. The reader may need to refer back to Chapter 2 for definitions of some of the variables used in the descriptions below, as well as to review the overall filter structure. Note that often in the descriptions, filter parameters are requested by sensor type. This is done so that the filter can be initialized as a function of the desired sensor replication specified in the SENSOR.INP file. Note: Although triple redundancy can be simulated, FINDS 3.0 uses at most <u>dual</u> redundancy in the NF filter and detectors.

NAMELIST: FILTIN

Description: FILTIN contains the quantities needed to initialize the fault tolerant navigator.

Comments: To determine the ordering of vectors contained in FILTIN, the reader should refer to Table 1 on page 9.

Variables:

ICNTRL: Flag to indicate mode of operation. Where 0=> normal (full) operation, and 1=> run NF filter without failure detectors (unitless), [I,S,D=0]

IIMUF: Flag to indicate if the NF filter is to utilize measurements from the IMU. Where 0=> don't use IMU, and ≠0 => use IMU. (dimensionless), [I,S,D=0]

IRSDF: Flag to indicate if the NF filter is to use body mounted rate gyros and accelerometers (IRSDF=0), or RSDIMU derived quantities (IRSDF≠0). (unitless), [I,S,D=0]

IRSDFY: Flag, indicating that computed attitudes from the RSDIMU are to be used in the NF filter. If IRSDFY=0=>don't use it, and if IRSDF≠1 <u>and</u> IIMUF=0 <u>and</u> IRSDFY≠1 then use it. (unitless) [I,S,D=0]

INOBPS: List of all normal operating biases to be identified by the NF filter, where each entry is a coded integer representing the sensor type. In other words, INOBPS(j)=> the j-th bias estimate in the bias filter is of type INOBPS(j). If all elements are 0 (zero), then the bias filter will not be invoked. Note: NBMXI is implicitly defined as the number of non-zero elements in INOBPS. (unitless), [I,V=NBMXI, D=0]

WINDOW: Decision window length. Used by the Bayesian decision logic. Window > 0.0. (seconds), [R,S,D=1.0]

KMXEWH: Detector estimation window lengths, specified in integer multiples of WINDOW seconds. KMXEWH determines the length of time before which each detector will be reset. KMXEWH > 0 (unitless), [I,V=NFMAX, D=NU1*3,NYMAX*1].

KMXHLR: Healing test window length, specified in integer multiples of WINDOW seconds. Healing decisions are made only at the end of the healing window. KMXHLR > 0 (unitless), [I,S,D=3]

SIG: NF filter process noise standard deviations, associated with a single sensor replication. Ordered as process noises, measurement noises, and RSDIMU rate gyro and accelerometer noise, respectively. SIG $\geq$ 0.0. (3*m/s/s, 3*deg/s, 2*m/s, 2*deg, m, m/s, 3*deg, m, deg/s, m/s/s), [R,V=NDISTB+NYMAX+2, D=SDNLA,SDNRG,2*0.0,SDNML*2, SDNAS(1)*AIRSPE, SDNAG,SDNGR(1),SDNAR(1)].

PBFOI: Initial bias filter covariance, specified as the standard deviation of the diagonal elements. Ordering is by sensor type. PBFOI $\geq$ 0.0. (3*m/s/s,3*deg/s,2*deg,m,m/s,3*deg,m) [R,V=NFMAX, D=3*.3048, 3*.256,2*.09,90.0,1.5,2*.8,2.4,1.0]

PBFIC: Standard deviation of <u>information</u> for the failure level estimators. Ordering is by sensor type. PBFIC $\geq$ 0.0. (3*s*s/m,3*s/deg,2*1/deg,1/m, s/m,3*s/deg,1/m), [R,V=NFMAX, D=3*.1,(NFMAX-3)*0]

SDXIC: Statistics for the initial NF filter state estimation error, where each initial estimation error is a random sample from a normal distribution with mean zero, and variance=SDXIC(i)**2. SDXIC is ordered by state index. (3*m,3*m/s,3*deg, 2*m/s), [R,V=NX,D=2*15.0,5.0,3*1.5,2*.1,.2,2*.5]

SDPIC: Standard deviation of the diagonal elements of the initial NF filter estimation error covariance. Ordered by state index. (3*m,3*m/s,3*deg,2*m/s), [R,V=NX,D=2*40.0, 30.0,2*4.0,1.25,2*.5,1.5,2*.75]

PRIORI: A priori probability of a single sensor failure. Used by the LR computers. Ordered by sensor index 0 $\leq$ PRIORI $\leq$ 1.0. (unitless), [R,V=NFMAX,NFMAX*0.00001/22]

PRIORJ: A priori probability of a dual sensor failure. Used by the LR computers. 0.0 $\leq$ PRIORJ $\leq$ 1.0. Ordered as MLS azimuth, elevation and range. (unitless), [R,V=3, D=3*{(0.00001/22)**2}].

POSBND: Position error bound for the NF filter's divergence test. The NF filter is declared divergent if the absolute sum of the position errors exceeds POSBND. POSBND $\geq$ 0.0. (meters), [R,S,D=200.0]

VELBND: Velocity error bound for NF filter's divergence test. The NF filter is declared divergent if the absolute sum of the velocity errors exceeds VELBND. VELBND $\geq$ 0.0. (meters/sec), [R,S,D=30.0]

ANGBND: Angular error bound for the NF filter's divergence test. The NF filter is declared divergent if the absolute sum of the angular errors exceeds ANGBND. ANGBND $\geq$ 0.0. (degrees), [R,S,D=10.0]

BTHRSH: Maximum expected normal operating bias levels for each sensor. Used by the sensor healing test. Ordered by sensor type. BTHRSH $\geq$ 0.0. (3*m/s/s, 3*deg/s, 2*deg,m, m/s,3*deg,m), [R,V=NFMAX, D=3*.3, 3*8.34E-5, 2*.09,9.0,1.7,3*.3,.6]

FTHRSH: Minimum expected failure levels for each sensor. Used by the sensor healing test. Ordered by sensor type. FTHRSH $\geq$ 0.0. (3*m/s/s,3*deg/s,2*deg,m,m/s,3*deg,m), [R,V=NFMAX,D=3*.5,5*.1,30.0,3.0, 3*.5,1.0]

HRADAR: Altitude, below which the radar altimeter measurements can be reliably used by the NF filter. (meters), [R,S,D=30.48].

3.5 Quick Reference to Input Files

This section contains two quick reference tables which provide a convienient cross reference to the previous namelist descriptions, as well as a concise description of each. The first table lists all required namelist names, along with the page containing their complete description. The last table also provides a list of the variable names contained in each namelist and their associated lengths in a FORTRAN-like style.

TABLE 3. Summary of Namelist Descriptions

Namelist	Page	Description
PROGCN	18	Contains general program control quantities.
PRNTPL	19	Contains quantities used to specify print and plot options.
ACSTAT	20	Contains the quantities necessary to specify the aircraft's initial conditions.
WINDAT	20	Contains quantities necessary to define the wind models simulated.
GSLOC	21	Contains the quantities necessary to specify the glideslope and localizer geometries.
STARTER	22	Contains the quantities necessary to specify the desired nominal route the aircraft is to follow.
MPRA	35	Specifies all model parameters necessary for simulation of the radar altimeter sensors.
FPRA	36	Specifies all the desired failure modes and onset times for the radar altimeter sensors.
MPAS	36	Specifies all model parameters necessary for simulation of the airspeed sensors.
FPAS	37	Specifies all the desired failure modes and onset times for the airspeed sensors.
MPRG	38	Specifies all model parameters necessary for simulation of the body mounted rate gyros.
FPRG	39	Specifies all the desired failure modes and onset times for the rate gyro sensors.
MPLA	40	Specifies all model parameters necessary for simulation of the body mounted linear accelerometers.

Table 3. Summary of Namelist Descriptions, concluded

Namelist	Page	Description
FPLA	41	Specifies all the desired failure modes and onset times for the linear accelerometer sensors.
MPAG	42	Specifies all model parameters necessary for simulation of the body mounted attitude gyros.
FPAG	43	Specifies all the desired failure modes and onset times for the attitude gyro sensors.
MPMLS	44	Specifies all model parameters necessary for simulation of the microwave landing system (MLS) receivers.
FPML	46	Specifies all the desired failure modes and onset times for the MLS sensors.
MPGR	47	Specifies all model parameters necessary for simulation of the redundant strapdown inertial measurement unit's
FPGR	48	Specifies all the desired failure modes and onset times for the RSDIMU rate gyro sensors.
FDIGR	49	Specifies the failure detection and isolation algorithms to be used by the rate gyros in the RSDIMU module.
MPAR	50	Specifies all model parameters necessary for simulation of the RSDIMU two-degree-of-freedom linear accelerometers.
FPAR	51	Specifies all the desired failure modes and onset times for the RSDIMU linear accelerometer sensors.
FDIAR	52	Specifies the failure detection and isolation algorithms to be used by the linear accelerometers
FILTIN	53	Contains the quantities needed to initialize the fault tolerant navigator.

TABLE 4. Summary of .INP Files

FILE-NAME	VARIABLE NAMES [(rows,columns)]
NAMELIST NAME:	
GENERAL.INP	
ACSTAT:	PITCH, ROLL, YAW, RINLO, RINLA, AIRSPE, HIGHT, WEIGHT, CG, RHO, THRUST, RE, PDEG, QDEG, RDEG
WINDAT:	GUST, IGUST, SCALEW, SDSUBW, SHEAR, SH1, SH2, WINDSS, PSIW
PROGCN:	TIME, FILTER, IEST, TSTOP, YSTOP
PRNTPL:	PLOTS, OUTPS, IHEAD, IPRNT, KPRINT, NTICK, CPUCHK
GSLOC:	GPIPRN, LOCRAN, PSIR, THETAG
STARTER:	NW, WLO(NW), WLA(NW), SLO(2), SLA(2), SHZ(2), HG(NW), VGD(NW), RT(NW-2), NEWPTH, KPATH, IC, TANLIM, SNDP(NW-2)
SENSOR.INP	
MPRA:	IDRA, IREPRA, DTRA, SDNRA(IREPRA), SDBRA(IREPRA), SLIMRA
FPRA:	FINRA, FIBRA, FRRA, TFRA(IREPRA,5)
MPAS:	IDAS, IREPAS, DTAS, SDMNAS(IREPAS), SDBAS(IREPAS), SLIMAS
FPAS:	FINAS, FIBAS, FRAS, TFAS(IREPAS,5)
MPRG:	IDRG, IREPRG, DTRG, SDNRG(IREPRG*3), SDBRG(IREPRG*3), SDSFRG(IREPRG*3), SLIMRG(3), SDTMRG(6)

TABLE 4. Summary of .INP Files, continued

FILE-NAME	VARIABLE NAMES [(rows,columns)]
NAMELIST NAME:	
FPRG:	FINRG(3), FIBRG(3), FRRG(3), FSFRG(3), TFRG(IREPRG*3,6)
MPLA:	IDLA, IREPLA, DTLA, SDNLA(IREPLA*3), SDBLA(IREPLA*3), SDSFLA(IREPLA*3), SLIMLA(3), SDTMLA(6)
FPLA:	FINLA(3), FIBLA(3), FRLA(3), FSFLA(3), TFLA(IREPLA*3,6)
MPAG:	IDAG, IREPAG, DTAG, SDNAG(IREPAG*3), SDBAG(IREPAG*3), SLIMAG(3)
FPAG:	FINAG(3), FIBAG(3), FRAG(3), TFAG(IREPAG*3,5)
MPMLS:	IDML, IREPML, IWHMLS, DTML, SDNML(IREPML*3), SDBML(IREPML*3), XAZ(3), XEL(3), BOUNAZ(3), BOUNEL(3), BOUNRN(3), AC(3), CA(3)
FPMLS:	FINML(3), FIBML(3), FRML(3), TFML(IREPML*3,5)
MPGR:	IDGR, DTRI, IRSOIM, SDNGR(8), SDBGR(8), SDSFGR(8), SLIMGR(8), SDTMGR(16), GRQSF, SDGDGR
FPGR:	FINGR(8), FIBGR(8), FRGR(8), FSFGR(8), TFGR(8,6)
FDIGR:	IPARGR, IPFOGR, PFTCGR, PFKGR, PFTHGR
MPAR:	IDAR, DTRI, SDNAR(8), SDBAR(8), SDSFAR(8), SLIMAR(8), ARQSF

TABLE 4. Summary of .INP Files, concluded

FILE-NAME	VARIABLE NAMES [(rows,columns)]
NAMELIST NAME:	
FPAR:	FINAR(8), FIBAR(8), FRAR(8), FSFAR(8), TFAR(8,6)
FDIAR:	IPARAR, IPFOAR, PFTCAR, PFKAR, PFTHAR
FILTRT.INP	
FILTIN:	ICNTRL, IIMUF, IRSDF, IRSDFY, INOBPS(NBMXI), WINDOW, KMXEWH(NFMAX), KMXHLR, SIG(NDISTB+NYMAX+2), PBFOI(NFMAX), PBFIC(NFMAX), SDXIC(NX), SDPIC(NX), PRIORI(NFMAX), PRIORJ(3), POSBND, VELBND, ANGBND, BTHRSH(NFMAX), FTHRSH(NFMAX), HRADAR

3.6 Sample Input File Specification

In this section, the input files for a typical FINDS run are given. The files are discussed by way of interpretive comments attached to various lines in each of the .INP files. Comments will be separated by "!" and will be in bolded text in the rest of the section. Incidentally, comments are _not_ allowed in the actual namelist files.

GENERAL.INP

```
$ACSTAT                              ! Initial conditions for the A/C
 PITCH=-2.4383, ROLL=0., YAW=170.
 RINLA=40.29510787,                  ! Initial latitude & longitude
 RINLO=-77.14320754,
 PDEG=0., QDEG=0., RDEG=0.,
 AIRSPE=145.,HIGHT=3106.4,
 WEIGHT=80000., CG=20., THRUST=13900.,
 RE=20914003.03,
$
$WINDAT
 SHEAR=.FALSE.,                      ! No shear winds
 IGUST=1,SDSUBW=2.,
  GUST=.F.,                          ! No gusts
  PSIW=165., WINDSS=30.0,            ! S.S. horizontal winds =
$                                    ! 30 knots @165 degrees
$PROGCN
 FILTER=.T.,                         ! Run the fault tolerant navigator
 IEST=1,                             ! Use filter estimates for G&C
 TSTOP=300.,                         ! Stop the program after 300
$                                       ! elapsed simulation seconds
$PRNTPL
 CPUCHK=.T.,                         ! Perform a CPU check
 PLOTS=.T.,                          ! Save .PLT file
 OUTPS=.T.,                          ! Save .OUT file
 IHEAD=10,       ! Write a new header every 50 sec in the .OUT file
 IPRNT=0,               ! Suppress time history printout in .OUT file
 NTICK=20,              ! Save data every 1 sec in .PLT file
$
$GSLOC
 PSIR=30.,       ! Use default GS & LOC geometries, rw heading = 30 deg
$
```

FIG. 7. Input File Example

```
$STARTER                          ! Define the flight path
NEWPTH=1, NW=4,                   ! Two heading changes in grd trk
WLO(1)=-77.14538380, WLO(2)=-77.13148869,      ! Longitudes of waypoints
WLO(3)=-77.05845224, WLO(4)=-77.02320521,
WLA(1)=40.29759451, WLA(2)=40.23788553,        ! Latitudes of waypoints
WLA(3)=40.20574384, WLA(4)=40.25237254,
HG(1)=3263.3,HG(2)=2102.4,                ! Altitudes of waypoints
HG(3)=953.9,HG(4)=.0,
VGD(1)=145.,VGD(2)=135.,                  ! Desired airspeed @ waypoints
VGD(3)=125.,VGD(4)=125.,
RT(1)=7500.,RT(2)=7500.,                  ! Radius of turns
IC(1)=0, IC(2)=0,                 ! WLO & WLA are corner points
SNDP=-1.,-1.,                     ! Both turns are left handed
SLO(2)=-77.025,                   ! Long, Lat, & altitude of MLS
SLA(2)=40.25, SHZ(2)=0.,
KPATH=2,                  ! Use continued track path construction
TANLIM=0.,                ! Use KPATH=2 at all times
$
```

SENSOR.INP

```
$MPRG
 IREPRG=2,                        ! Two replications of rate
$                         ! gyros, all other parameters are defaulted.
$FPRG $                   ! No rate gyro failures to be simulated
$MPLA
 IREPLA=2, $              ! Two replications of linear accelerometers
$FPLA $
$MPAG
 IREPAG=2, $              ! Two replications of attitude gyros
$FPAG $
```

FIG. 7. Input File Example, continued

```
$MPRA
IREPRA=2, $                 ! Two replications of radar altimeters
$FPRA $
$MPAS
IREPAS=2, $                 ! Two replications of airspeed indicators
$FPAS $
$MPGR
IRSDIM=1, $                 ! Do not simulate the RSDIMU
$FPGR $
$FDIGR $
$MPAR $
$FPAR $
$FDIAR $
$MPMLS
IREPML=2,                           ! Two replications of MLS
SDBML(3)=4.0, SDBML(6)=4.0, $       ! Use 4 M nob on MLS range
$FPMLS
FIBML(2)=8.0,TFML(5,2)=75.0,        ! introduce a 5 sigma bias failure on the
$                           ! second replication of elevation at 75 sec.
```

FILTRT.INP

```
$FILTIN
IRSDF=0,                    !Filter uses body mounted inputs and
IIMUF=1,                    ! includes the IMU outputs as measurements.
INOBPS=1,2,3,4,5,6,19*0,            !Request that x,y and z accel.
                            ! as well as P,Q,R, gyro biases be identified
SIG(4)=0.0894, SIG(5)=0.0894, SIG(6)=0.0894,     ! set P,Q,R noise levels
SIG(13)=0.251, SIG(14)=0.251, SIG(15)=0.251,     ! & IMU noises
PRIORI=5*4.6E-9, 4.6E-10, 3*3.5E-9, 3.15E-8,     ! set prior prob
2*2.25E-8, 2.25E-9, 3.5E-10, 17*0.0,
PRIORJ=3*1.225E-19, 7*0.0, $
```

FIG. 7. Input File Example, concluded

4. DESCRIPTION OF OUTPUTS

This chapter examines the outputs available from FINDS. Each of the four major output files, summarized in Chapter 2, will be discussed in the succeeding sections. The sections are organized into a discussion section and a notated sample listing of the file. The comments will be separated by "!" and will appear bolded in this chapter.

4.1 Summary File

The summary file (file ext=.SUM) is an ASCII text file. It contains a summary of the key parameters used to generate the run (from the fault tolerant system's point of view). The file is organized into five tables which summarize the sensor/failure models and fault tolerant system modules. If a failure is detected, additional information is printed to clarify the filter/detector operation at that instant of time. At the end of the listing, if requested, a CPU summary is provided. This summary simply shows the total CPU time used by FINDS, along with a breakdown of the fractional contributions to it by module.

The following notated sample listing, generated by running FINDS using the input data from section 3.6, further describes this file.

```
FINDS  Version 3.0 (d.p.)

14-SEP-83   08:54:29
 STANDARD SENSOR CONFIGURATION
 SINGLETON BIAS FAILURES -- level = 10 sigma n.o.b.
 TIME= 75         110       150       220       253
 TYPE= el-2

Random number seed =        · 32069
Integration time interval =  0.050 seconds
Guidance & Control logic driven by the FILTER


INPUT FILES USED:
General Program Control:DRA1:[RLANCRAFT.RUNS]GENERAL.INP;2
Sensor Model Data:       DRA1:[RLANCRAFT.RUNS]SENSOR.INP;1
Filter-Detector Data:    DRA1:[RLANCRAFT.RUNS]FILTRT.INP;2

OUTPUT FILES GENERATED:
General Output:          DRA1:[RLANCRAFT.RUNS]FINDSR.OUT;1
Summary Output:          DRA1:[RLANCRAFT.RUNS]FINDSR.SUM;1
Plot output:             DRA1:[RLANCRAFT.RUNS]FINDSR.PLT;1
Time Line Output:        DRA1:[RLANCRAFT.RUNS]FINDSR.TLN;1


Steady state winds =     30.000 knots at 165.00 degrees (from North)
There are  1 scheduled failures in this run
```

FIG. 8. Sample Summary File

TABLE I : SENSOR MODEL PARAMETERS

Sensor Type / Replication	Units	Noise Level (Std. Dev.)	Bias Level	Scale Factor (%)	Stop Limits	Failure Type	Failure Magnitude	Onset Time
Body Mounted * Accelerometers								
X / 1	m/s/s	9.8066E-02	2.0141E-01	3.5479E-01	4.9033E+00			
Y / 1	m/s/s	9.8066E-02	6.6280E-03	-6.6226E-01	4.9033E+00			
Z / 1	m/s/s	9.8066E-02	-5.7531E-03	-1.9329E-01	1.9613E+01			
X / 2	m/s/s	9.8066E-02	-1.4061E-01	1.5305E-02	4.9033E+00			
Y / 2	m/s/s	9.8066E-02	2.3216E-02	1.2125E-02	4.9033E+00			
Z / 2	m/s/s	9.8066E-02	-1.4968E-01	-2.8674E-01	1.9613E+01			
Body Mounted ** Rate Gyros								
P / 1	deg/s	2.0000E-02	-2.3726E-05	5.1733E-04	1.0000E+02			
Q / 1	deg/s	2.0000E-02	4.8488E-05	-1.4305E-02	1.0000E+02			
R / 1	deg/s	2.0000E-02	-7.7594E-06	1.0958E-02	1.0000E+02			
P / 2	deg/s	2.0000E-02	-4.2221E-05	9.3981E-03	1.0000E+02			
Q / 2	deg/s	2.0000E-02	-2.6633E-05	1.4372E-02	1.0000E+02			
R / 2	deg/s	2.0000E-02	-1.5924E-05	4.1667E-03	1.0000E+02			
M. L. S. ***								
Azm/ 1	deg	3.0000E-02	3.8437E-02	N/A	N/A			
El / 1	deg	3.0000E-02	-4.4473E-02	N/A	N/A			
Rng/ 1	m	3.0000E+00	7.4070E+00	N/A	N/A			
Azm/ 2	deg	3.0000E-02	-9.8190E-03	N/A	N/A			
El / 2	deg	3.0000E-02	5.6108E-02	N/A	N/A	INC. BIAS	2.4000E-01	80.571
Rng/ 2	m	3.0000E+00	6.9795E+00	N/A	N/A			
Indicated **** Airspeed					(upper/lower)			
IAS/ 1	m/s	2.0000E+00	3.9299E-01	N/A	2.0578E+02			
IAS/ 2	m/s	2.0000E+00	2.5764E-01	N/A	5.1444E+01			
I. M. U.								
phi/ 1	deg	2.3000E-01	1.8383E-02	N/A	8.0000E+01			
the/ 1	deg	2.3000E-01	6.0136E-02	N/A	8.0000E+01			
psi/ 1	deg	2.3000E-01	-5.4620E-02	N/A	6.0000E+02			
phi/ 2	deg	2.3000E-01	7.2554E-02	N/A	8.0000E+01			
the/ 2	deg	2.3000E-01	-8.6147E-02	N/A	8.0000E+01			
psi/ 2	deg	2.3000E-01	-1.3620E-01	N/A	6.0000E+02			
Radar Altimeter								
RA / 1	m	3.0480E-01	-2.0271E-01	N/A	N/A			
RA / 2	m	3.0480E-01	-3.3161E-01	N/A	N/A			

FOOTNOTES TO TABLE I :

* : Body mounted accelerometers are also misaligned by the following transformation:

$$\mathrm{TMLA} = \begin{matrix} 1.0000E+00 & -3.6228E-03 & 3.3961E-03 \\ 1.5940E-02 & 1.0000E+00 & 1.4211E-02 \\ 2.5820E-03 & 1.5736E-03 & 1.0000E+00 \end{matrix}$$

** : Body mounted rate gyros are also misaligned by the following transformation:

$$\mathrm{TMRG} = \begin{matrix} 1.0000E+00 & -7.4602E-03 & -7.1255E-03 \\ -4.5203E-03 & 1.0000E+00 & -3.4715E-03 \\ -1.5571E-03 & -9.6759E-03 & 1.0000E+00 \end{matrix}$$

*** : M.L.S. noise is TIME CORRELATED

**** : Since indicated airspeed noise is multiplicative, the standard deviation of the noise is expressed in percent (%).

FIG. 8. Sample Summary File, continued

TABLE II : NO-FAIL FILTER STATE INITIAL CONDITIONS

Variable	Est. Error	Uncertainty (S.D.)	Units
States:			
x-rw	8.7185E+00	4.0000E+01	m
y-rw	6.4227E+00	4.0000E+01	m
z-rw	-2.2559E+00	3.0000E+01	m
x-dot-rw	-5.9524E-01	4.0000E+00	m/s
y-dot-rw	-1.8651E+00	4.0000E+00	m/s
z-dot-rw	1.8234E+00	1.2500E+00	m/s
phi	-2.7424E-02	5.0000E-01	deg
theta	-1.0536E-01	5.0000E-01	deg
psi	-7.5294E-03	1.5000E+00	deg
x-wind-rw	4.5721E-02	7.5000E-01	m/s
y-wind-rw	3.8239E-01	7.5000E-01	m/s
Ave. Biases:			
x-accel	1.6571E-01	3.0480E-01	m/s/s
y-accel	-1.4457E-01	3.0480E-01	m/s/s
z-accel	1.1009E-02	3.0480E-01	m/s/s
P-gyro	1.7914E-04	2.5600E-01	deg/s
Q-gyro	-5.5519E-05	2.5600E-01	deg/s
R-gyro	-1.3474E-04	2.5600E-01	deg/s

TABLE III : NO-FAIL FILTER NOISE AND INITIAL BIAS LEVELS

Variable	Noise S.D. Per Repl.	Ave. Bias	Replications Used	Units
Process Noises:				
x-accel	9.8066E-02	1.6571E-01	1	m/s/s
y-accel	9.8066E-02	-1.4457E-01	1	m/s/s
z-accel	9.8066E-02	1.1009E-02	1	m/s/s
P-gyro	8.9400E-02	1.7914E-04	1	deg/s
Q-gyro	8.9400E-02	-5.5519E-05	1	deg/s
R-gyro	8.9400E-02	-1.3474E-04	1	deg/s
x-wind-rw	0.0000E+00	N/A	N/A	m/s
y-wind-rw	0.0000E+00	N/A	N/A	m/s
Measurement Noises:				
MLS azim	6.0000E-02	1.4309E-02	2	deg
MLS el	6.0000E-02	5.8176E-03	2	deg
MLS rng	6.0000E+00	7.1933E+00	2	m
IAS	1.4919E+00	3.2531E-01	2	m/s
IMU phi	2.5100E-01	4.5469E-02	2	deg
IMU theta	2.5100E-01	-1.3005E-02	2	deg
IMU psi	2.5100E-01	-9.5411E-02	2	deg
Radar alt	3.0480E-01	0.0000E+00	0	m

FIG. 8. Sample Summary File, continued

TABLE IV : DETECTOR RESET PARAMETERS

Detectors assume M.L.S. noise is COLORED
Detector Window Length = 1.00000 Seconds

Sensor Type	Apriori Prob	Est. Window (sec)	Estimation Information (S.D.)	Units
Singular Failures:				
x-accel	4.6000E-09	3.00000	9.2903E-03	m/s/s
y-accel	4.6000E-09	3.00000	9.2903E-03	m/s/s
z-accel	4.6000E-09	3.00000	9.2903E-03	m/s/s
P-gyro	4.6000E-09	3.00000	0.0000E+00	deg/s
Q-gyro	4.6000E-09	3.00000	0.0000E+00	deg/s
R-gyro	4.6000E-10	3.00000	0.0000E+00	deg/s
MLS azim	3.5000E-09	1.00000	0.0000E+00	deg
MLS el	3.5000E-09	1.00000	0.0000E+00	deg
MLS rng	3.5000E-09	1.00000	0.0000E+00	m
IAS	3.1500E-08	1.00000	0.0000E+00	m/s
IMU phi	2.2500E-08	1.00000	0.0000E+00	deg
IMU theta	2.2500E-08	1.00000	0.0000E+00	deg
IMU psi	2.2500E-09	1.00000	0.0000E+00	deg
Radar alt	3.5000E-10	1.00000	0.0000E+00	m
Simultaneous Multiple Failures:				
MLS azim	1.2250E-14			
MLS el	1.2250E-14			
MLS rng	1.2250E-14			

TABLE V : SENSOR HEALER PARAMETERS

Healer Window Length = 3.00000 Seconds

Sensor Type	Bias Est Thresold	Failure Est Threshold	Decision Threshold (Square Units)	Units
x-accel	2.0000E-01	5.0000E-01	-1.1327E-01	m/s/s
y-accel	2.0000E-01	5.0000E-01	-1.1327E-01	m/s/s
z-accel	2.0000E-01	5.0000E-01	-1.1327E-01	m/s/s
P-gyro	1.2000E-03	1.2000E-02	-4.7111E-03	deg/s
Q-gyro	1.2000E-03	1.2000E-02	-4.7111E-03	deg/s
R-gyro	1.2000E-03	1.2000E-02	-4.7111E-03	deg/s
MLS azim	6.0000E-02	1.5000E-01	-2.1200E-02	deg
MLS el	6.0000E-02	1.5000E-01	-2.1200E-02	deg
MLS rng	1.2000E+01	2.0000E+01	-2.1200E+02	m
IAS	1.2000E+00	2.5000E+00	-1.3107E+01	m/s
IMU phi	2.0000E-01	5.0000E-01	-3.1152E-01	deg
IMU theta	2.0000E-01	5.0000E-01	-3.1152E-01	deg
IMU psi	2.0000E-01	5.0000E-01	-3.1152E-01	deg
Radar alt	6.1000E-01	1.5000E+00	-5.4709E-01	m

FIG. 8. Sample Summary File, continued

```
DECIDE: Sensor type MLS el      , Replication No. 2 Failed at time=  81.000
        Estimated Failure Level =   3.23494907E-03  deg
ALAMDA       : 1.296D+02 1.276D+02 1.249D+02 1.400D+02 1.344D+02
               1.341D+02 1.286D+02 1.263D+02 1.286D+02 1.268D+02
               1.261D+02 1.278D+02 1.296D+02 9.990D+30 1.283D+02
               1.039D+02 1.285D+02 1.266D+02 1.311D+02 1.276D+02
               1.306D+02 9.990D+30 1.090D+02
ALAMDJ       : 1.410D+02 1.334D+02 1.411D+02
P0 BEFORE    : 6.840D+00 7.243D+00 8.686D+00 5.051D-01 3.690D-01
               5.721D-01 1.047D-08 1.043D-08 1.051D-08 4.675D-01
               2.936D-01
V mi         :-2.738D+02-6.853D+01 4.260D+03-4.389D+01-2.109D+01
               7.766D+02-1.338D-03-7.316D-05-3.090D-05 3.039D+01
               2.731D+01-6.716D-01 5.330D+00-1.852D+01 5.509D-05
              -2.043D-05-2.745D-06
PB BEFORE    : 2.670D-03 2.837D-03 2.972D-03 6.154D-11 6.227D-11
               6.070D-11
P0 AFTER     : 7.638D+00 7.293D+00 2.017D+02 5.256D-01 3.737D-01
               6.988D+00 1.049D-08 1.043D-08 1.051D-08 4.774D-01
               3.015D-01
PB AFTER     : 2.675D-03 3.140D-03 6.620D-03 6.157D-11 6.228D-11
               6.070D-11

         GLIDESLOPE CAPTURE AT TIME = 157.40


         GLIDESLOPE CAPTURE PLUS 10 AT TIME = 167.35


         LOCALIZER ONCOURSE AT TIME = 192.25

         *        BEGIN DECRAB AT TIME = 245.05         *
Radar Altimeter turn-on at t= 251.70
P0 BEFORE    : 2.311D+00 1.587D+00 2.632D-01 2.336D-01 3.233D-01
               1.783D-01 2.207D-09 2.206D-09 2.218D-09 4.967D-03
               4.615D-03
V mi         : 0.000D+00 0.000D+00 0.000D+00 0.000D+00 0.000D+00
               0.000D+00 0.000D+00 0.000D+00 0.000D+00 0.000D+00
               0.000D+00 0.000D+00 0.000D+00 0.000D+00 0.000D+00
               0.000D+00 0.000D+00
PB BEFORE    : 4.912D-04 5.036D-04 5.530D-04 8.661D-13 8.665D-13
               5.827D-13
P0 AFTER     : 2.311D+00 1.587D+00 2.632D-01 2.336D-01 3.233D-01
               1.783D-01 2.207D-09 2.206D-09 2.218D-09 4.967D-03
               4.615D-03
PB AFTER     : 4.912D-04 5.036D-04 5.530D-04 8.661D-13 8.665D-13
               5.827D-13
CAT II DECISION HEIGHT         AIRCRAFT HEIGHT =      99.94    AIRCRAFT LATERAL DEVIATION =     -2.15
AIRSPEED ERROR =       0.22     GLIDESLOPE OFFSET =   199.56     TIME =  251.750 SEC
          *         BEGIN FLARE AT TIME = 257.15         *
    ENDPOINT DISPERSION DATA      HIGHTR =        0.16     XPOS     =   1324.81     YPOS =     -0.54
    HDOT =        -3.04      AIRSPEED=,      195.49     TIME =  264.950 SEC
FINAL POSITION =  1.3248E+03 -5.3835E-01 -9.8967E+00
END EXECUTION



          FINDS CPU USAGE SUMMARY

Total CPU time spanned =       5622.00 seconds
Initialization =                   0.06 %
A/C simulation + Plots =           4.71 %
Sensor model simulation =          1.78 %
Fault tolerant filter =           93.11 %

CPU Breakdown for Fault Tolerant Filter :
Initialization + misc. =           0.63 %
No-Fail Filter =            .     30.41 %
Bank of Detectors + LR comp. =    68.52 %
Decision, healing, & reconf =      0.43 %

CPU Breakdown for Bank of Detectors & LR computers :
Initialization and overhead =     40.78 %
Inner loop set-up =               29.58 %
Detectors =                       12.76 %
LR computers =                    14.31 %
Multiple detectors + H0 =          2.57 %
```

FIG. 8. Sample Summary File, concluded

4.2 Time Line File

The discrete time line file (file ext=.TLN) is a variable length formatted ASCII file.[5] It attempts to capture the discrete nature of the automated approach and landing environment. The file consists of coded information stored in a fixed format. Whenever an "event" occurs, (such as glideslope capture or a simulated failure) a coded "snapshot" of important variables are recorded. This file can be used to obtain statistical information, (when combined with other .TLN files from a Monte Carlo simulation), about FINDS failure detection performance (false alarm rate, average time to detect, etc.) as well as estimation and system performance measures at particular points in the flight path.

The time line file is organized into a header record, which identifies the file type; a number of records of comment lines (the number is determined from the header record); and a sequence of three (3) record discrete snapshots. Each "snapshot" consists of the following items:

- o Time of occurrence
- o A coded integer representing the discrete event
- o Four (4) integer descriptors (I.D.) which further clarify the event
- o Eighteen (18) floating point descriptors (F.P.D.)

Table 5 defines the last three items for each discrete event. However, in order to interpret the table properly, a few comments are needed:

- o integer descriptor #4 specifies the following: 0=>F.P.D.s are

---

[5]In order to reduce disk storage requirements (a significant consideration in Monte Carlo studies), future releases of FINDS should store this file in an unformatted (binary) fashion.

absolute values, 1=>F.P.D.s are errors (true-estimated). I.D. #4 is therefore not included in Table 5.

- o The following "basic" floating point description outputs are available, to choose from (coded A -> F):

A: $X_{rw}, Y_{rw}, Z_{rw}, Z_{rw}$, Airspeed

B: Azm,El,Rng

C: X_{rw}, Y_{rw}, Z_{rw}

D: $X_{rw}, Y_{rw}, Z_{rw}, \dot{X}_{rw}, \dot{Y}_{rw}, \dot{Z}_{rw}, \phi, \theta, \psi$,
X_w, Y_w, Airsped error from desired, crosstrack error from desired, altitude error from desired

E: D+F.P.D(17) = estimated singular failure level
+ F.P.V. (18) = uncertainty in estimate

F: D+F.P.D (15) = first multiple failure level,
+F.P.D (16) = uncertainty in FPD (15)
+F.P.D (17) = second multiple failure level,
+F.P.D (18) = uncertainty in F.P.D (17).

- o A blank entry in the table signifies the descriptor is not used.

- o Failure type is coded as 1=>increased noise, 2=>bias, 3=>scale factor, 4=>hardover, 5=>null, 6=>ramp failures.

- o Table 1 on page 9 defines the sensor type index

TABLE 5. Summary of Time Line Codes

Event Name	Event #	I.D.#1	I.D.#2	I.D.#3	F.P.D. Format
Begin Run	1	IEST			A
End Run	2				A
MLS Turn-on	3				B
Filter-detector Turn-on	4	ICNTRL	IIMUF	IRSDF	C
RA Turn-on	5				D
G.S. Capture	6				A
G.S. Capture plus 10	7				D
LOE on course	8				D
Begin Decrab	9				D
Begin Flare	10				D
CAT II Decision height	11				D
Not Used	12				
Not Used	13				
Failure Onset	14	failure type	sensor type	sensor rep.	D

TABLE 5. Summary of Time Line Codes, continued

Event Name	Event #	I.D.#1	I.D.#2	I.D.#3	F.P.D. Format
Singular fail. detection	15		sensor type	sensor rep.	E
Multiple fail. detection	16	1=>Azm 2=>El 3=>Rng			F
Sensor "healed"	17		sensor type	sensor rep.	D
N.F. filter collapsed	18	1=>input 2=>output	sensor type		D
N.F. filter expanded	19	1=>input 2=>output	sensor type		D
N.F. filter divergent	20	1=>position 2=>rates 3=>attitudes 4=>height<0			D
Mission abort	21				C
Localizer capture	22				A

The following sample time line file, Figure 9, was generated using the input files defined in Figure 7 on page 62.

To help the reader understand this important file, lets discuss three snapshots, marked A,B,C in Figure 9.

A This snapshot marks the filter/detector algorithm start-up. Notice that the event occurs at Time= 35.25 seconds, and that the A/C's position is: X_{rw}=-3116 M, Y_{rw}=-9311 M, Z_{rw}=-806 M.

B This event represents the simulated increased bias failure of the second replication of MLS elevation. The event happens at Time=80.6 seconds.

C Here we see that at 81 seconds the decision logic has declared the second replication of MLS elevation to be faulty, and removed it from the problem. Notice that the estimated failure level is 0.185 degrees. By compairing C to B we see that it took .4 sec. to detect this failure.

Based on the information contained in this file several useful measures can be defined. For example, in a single run the following items could be compaired or computed:

- Time to correctly detect a failure, i.e. time of detection minus time of failure.
- Time to correctly heal a repaired sensor, i.e. time of healing minus time of incorrect failure or repair.
- Total number of false alarms, i.e. detection of a failure that wasn't simulated.
- Total number of missed detections, i.e. failure to correctly detect a failure.
- Total number of incorrect healings.
- State estimation error at various points in the run.
- Performance errors at critical points in the run.

By computing the above quantities for each run, and averaging or summing

```
TLN1      4
 STANDARD SENSOR CONFIGURATION
 SINGLETON BIAS FAILURES -- level = 10 sigma n.o.b.
 TIME= 75          110       150       220       253
 TYPE= e1-2
    0.0000     1     1     0     0     0-6.6355E+02-1.1201E+04-9.3696E+02 3.2496E+00
   7.4594E+01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
   0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
   35.2500     3     0     0     0     0 5.8184E+01 4.6151E+00 1.0959E+04 0.0000E+00
   0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
   0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
   35.2500     4     0     1     0     0-3.1163E+03-9.3117E+03-8.0674E+02 0.0000E+00
   0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00  A
   0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
   80.6000    14     2     8     2     1 1.5203E+00 1.2454E+01 2.4628E+00-1.7391E-01
   3.8993E-01-3.0700E-01-6.8313E-02-7.4681E-02 5.5642E-02 1.0018E+01-1.0417E+01  B
  -1.9757E+00-1.9175E+01 2.1727E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
   81.0000    15     0     8     2     1 2.7786E+00 8.3696E+00 4.0544E+00-2.0024E-01
   4.0973E-01 2.9930E-02-8.2717E-02-5.9572E-02 1.8718E-01 9.9836E+00-1.0438E+01  C
  -1.9697E+00-1.9678E+01 4.0353E+00 0.0000E+00 0.0000E+00 1.8535E-01 2.3777E-02
  157.4000     6     0     0     0     0-5.1428E+03-8.0305E+02-3.0122E+02 3.5258E+00
   6.6105E+01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
   0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
  167.3500     7     0     0     0     0 4.9751E+00-2.0130E-01-2.3954E+00-6.5165E-01
  -3.2478E-01 9.2921E-02-1.3047E-01-3.2320E-02 9.6692E-02 9.4701E+00-9.6710E+00
   3.2928E-02-5.9013E-01-1.0699E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
  192.2500     8     0     0     0     0 2.9448E+00 3.5465E+00-1.6928E+00-7.2610E-01
   5.2285E-01 3.8571E-02-1.4277E-01-5.6151E-02 8.7711E-02 9.9642E+00-9.4142E+00
   4.4256E-01 5.7629E+00 7.2765E-02 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
  245.0500     9     0     0     0     0 2.2182E+00-2.6317E-01-7.1819E-01-7.7573E-01
  -2.1049E-01-9.5215E-02 2.4582E-02 6.5698E-02 1.1209E-02 1.0285E+01-9.0163E+00
   8.7472E-02-2.0651E+00-2.9306E-01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
  251.7000     5     0     0     0     1 6.1919E+00 1.3219E+00-1.8943E-01-7.0267E-01
   1.3991E-01-6.3539E-02 4.3436E-02 1.3055E-01 2.5067E-02 1.0291E+01-8.9651E+00
   3.0494E-01 1.9892E+00-4.0475E-01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
  251.7000    18     2     8     1     1 6.1919E+00 1.3219E+00-1.8943E-01-7.0267E-01
   1.3991E-01-6.3539E-02 4.3436E-02 1.3055E-01 2.5067E-02 1.0291E+01-8.9651E+00
   3.0494E-01 1.9892E+00-4.0475E-01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
  251.7000    19     2    14     1     1 6.1919E+00 1.3219E+00-1.8943E-01-7.0267E-01
   1.3991E-01-6.3539E-02 4.3436E-02 1.3055E-01 2.5067E-02 1.0291E+01-8.9651E+00
   3.0494E-01 1.9892E+00-4.0475E-01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
  251.7500    11     0     0     0     0 3.4610E+00 1.3012E+00-1.9396E-01-7.4006E-01
   1.2394E-01-1.5049E-01 8.4812E-02 9.7071E-02-2.0560E-03 1.0291E+01-8.9642E+00
   2.6950E-01 1.9560E+00-2.7817E-01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
  257.1500    10     0     0     0     0 3.1236E+00 4.5899E-01-2.1601E-01-8.0380E-01
  -1.6903E-02 2.9301E-02 8.9153E-02 1.0078E-01-5.2047E-03 1.0288E+01-8.9214E+00
   2.2618E-01-1.0014E+00-6.6519E-01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
  264.9500     2     0     0     0     0 4.0380E+02-1.6409E-01-3.0165E+00 9.4027E-01
   5.9585E+01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
   0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
```

FIG. 9. Sample Time Line File

across an ensemble of like runs, statistical performace measures can be obtained.

4.3 Output File

The output file (file ext=.OUT) is an ASCII text file. It contains a complete echo check of all namelist and interactive inputs and therefore completely defines the run. It also contains as an option, a table of time history outputs. If OUTPS=.False., this file will not be generated. (The default for Monte-Carlo simulation study). This file is basically used for verification purposes.

The following sample output details this file.

```
FINDS  Version 3.0 (d.p.)

14-SEP-83   08:54:29
 STANDARD SENSOR CONFIGURATION
 SINGLETON BIAS FAILURES -- level = 10 sigma n.o.b.
 TIME= 75        110      150      220      253
 TYPE= el-2

Random number seed =          32069
Integration time interval =  0.050 seconds
Guidance & Control logic driven by the FILTER


INPUT FILES USED:
General Program Control:DRA1:[RLANCRAFT.RUNS]GENERAL.INP;2
Sensor Model Data:        DRA1:[RLANCRAFT.RUNS]SENSOR.INP;1
Filter-Detector Data:     DRA1:[RLANCRAFT.RUNS]FILTRT.INP;2

OUTPUT FILES GENERATED:
General Output:           DRA1:[RLANCRAFT.RUNS]FINDSR.OUT;1
Summary Output:           DRA1:[RLANCRAFT.RUNS]FINDSR.SUM;1
Plot output:              DRA1:[RLANCRAFT.RUNS]FINDSR.PLT;1
Time Line Output:         DRA1:[RLANCRAFT.RUNS]FINDSR.TLN;1


$ACSTAT
PITCH    =  -2.43830000000000       ,
ROLL     =  0.000000000000000E+00,
YAW      =   170.000000000000       ,
RINLO    =  -77.1432075400000       ,
RINLA    =   40.2951078700000       ,
AIRSPE   =   145.000000000000       ,
HIGHT    =   3106.40000000000       ,
WEIGHT   =   80000.0000000000       ,
CG       =   20.0000000000000       ,
RHO      =  2.308119000000000E-03,
THRUST   =   13900.0000000000       ,
RE       =   20914003.0300000       ,
PDEG     =  0.000000000000000E+00,
QDEG     =  0.000000000000000E+00,
RDEG     =  0.000000000000000E+00
$END
$WINDAT
GUST     = F,
IGUST    =              1,
SCALEW   =   1400.00000000000       ,
SDSUBW   =   2.00000000000000       ,
SHEAR    = F,
SH1      = -6.749999999999999E-02,
SH2      = -0.135000000000000       ,
WINDSS   =   30.0000000000000       ,
PSIW     =   165.000000000000
$END
$PROGCN
DTIME    =  5.000000000000000E-02,
FILTER   = T,
IEST     =              1,
TSTOP    =   300.000000000000       ,


YSTOP    =   300.000000000000
$END
$PRNTPL
PLOTS    = T,
OUTPS    = T,
IHEAD    =             10,
IPRNT    =              0,
KPRINT   =              0,
NTICK    =              2,
CPUCHK   = T
$END
$GSLOC
GPIPRN   =   1000.00000000000       ,
LOCRAN   =   10000.0000000000       ,
PSIR     =   30.0000000000000       ,
THETAG   =   3.00000000000000
$END
```

FIG. 10. Sample Output File

```
BODY MOUNTED RATE GYRO MODEL PARAMETER INPUTS
$MPRG
IDRG    =             2,
IREPRG  =             2,
DTRG    =  5.0000000000000000E-02,
SDNRG   = 9*2.0000000000000000E-02,
SDBRG   = 9*2.7800000000000000E-05,
SDSFRG  = 9*1.0000000000000000E-02,
SLIMRG  = 3*100.000000000000         ,
SDTMRG  = 6*0.400000000000000
$END
$FPRG1
FINRG   = 3*10.0000000000000         ,
FIBRG   = 3*3600.000000000000        ,
FRRG    = 3*5.0000000000000000E-02,
FSFRG   = 3*10.0000000000000         ,
IFRG    = 6*0,
TFRG    = 54*300.050000000000
$END
BODY MOUNTED ACCELEROMETER MODEL PARAMETER INPUTS
$MPLA
IDLA    =             2,
IREPLA  =             2,
DTLA    =  5.0000000000000000E-02,
SDNLA   = 9*1.0000000000000000E-02,
SDBLA   = 9*1.0000000000000000E-02,
SDSFLA  = 9*0.250000000000000        ,
SLIMLA  = 2*0.500000000000000        ,    2.00000000000000     ,
SDTMLA  = 6*0.360000000000000
$END
$FPLA1
FINLA   = 3*10.0000000000000         ,
FIBLA   = 3*10.0000000000000         ,
FRLA    = 3*360.000000000000         ,
FSFLA   = 3*10.0000000000000         ,
IFLA    = 6*0,
TFLA    = 54*300.050000000000
$END
ATTITUDE GYRO MODEL PARAMETER INPUTS
$MPAG
IDAG    =             2,
IREPAG  =             2,

DTAG    =  5.0000000000000000E-02,
SDNAG   = 9*0.230000000000000        ,
SDBAG   = 9*8.0000000000000000E-02,
SLIMAG  = 2*80.0000000000000         ,    600.000000000000
$END
$FPAG1
FINAG   = 3*10.0000000000000         ,
FIBAG   = 3*10.0000000000000         ,
FRAG    = 3*0.300000000000000        ,
IFAG    = 5*0,
TFAG    = 45*300.050000000000
$END
RADAR ALTIMETER MODEL PARAMETER INPUTS
$MPRA
IDRA    =             2,
IREPRA  =             2,
DTRA    =  5.0000000000000000E-02,
SDNRA   = 3*0.304800000000000        ,
SDBRA   = 3*0.304800000000000        ,
SLIMRA  =   1000.000000000000
$END
$FPRA1
FINRA   =   5.00000000000000         ,
FIBRA   =   10.0000000000000         ,
FRRA    =   1.00000000000000         ,
IFRA    = 5*0,
TFRA    = 15*300.050000000000
$END
```

•

•

•

FIG. 10. Sample Output File, continued

```
$STARTER
NW      =             4,
WLO     =  -77.14538380000000     ,  -77.13148869000000     ,
 -77.05845224000000     ,  -77.02320521000000     , 6*0.0000000000000000E+00,
WLA     =   40.29759451000000     ,   40.23788553000000     ,
  40.20574384000000     ,   40.25237254000000     , 6*0.0000000000000000E+00,
SLO     =  -77.16489400000000     ,  -77.02500000000000     ,
SLA     =   40.40301600000000     ,   40.25000000000000     ,
SHZ     =   150.0000000000000     ,  0.0000000000000000E+00,
HG      =   3263.300000000000     ,   2102.400000000000     ,
  953.9000000000000     , 7*0.0000000000000000E+00,
VGD     =   145.0000000000000     ,   135.0000000000000     ,
2*125.0000000000000     , 6*0.0000000000000000E+00,
RT      = 4*7500.000000000000     , 4*0.0000000000000000E+00,
NEWPTH  =             1,
KPATH   =             2,
IC      = 4*0, 6*1,
TANLIM  =  0.0000000000000000E+00,
SNDP    = 2*-1.000000000000000     , 6*0.0000000000000000E+00
$END
$FILTIN
ICNTRL  =             0,
IIMUF   =             1,
IRSDF   =             0,
IRSDFY  =             0,
INOBPS  =             1,          2,          3,          4,          5,
        6, 19*0,
WINDOW  =   1.000000000000000     ,
KMXEWH  = 6*3, 8*1, 11*0,
KMXHLR  =             3,
SIG     = 3*9.8066352000000002E-02, 3*8.9400000000000000E-02,
2*0.0000000000000000E+00, 2*6.0000000000000000E-02,   6.000000000000000     ,
  1.491885891300010     , 3*0.2510000000000000     ,  0.3048000000000000     ,
 1.5528249843748540E-04,  5.4820757344355615E-04, 12*0.0000000000000000E+00,
PBFOI   = 3*0.3048000000000000     , 3*0.2560000000000000     ,
2*9.0000000000000000E-02,   90.00000000000000     ,   1.500000000000000     ,
2*0.8000000000000000     ,   2.400000000000000     ,   1.000000000000000     ,
11*0.0000000000000000E+00,
PBFIC   = 3*0.1000000000000000     , 27*0.0000000000000000E+00,
SDXIC   = 2*15.00000000000000     ,   5.000000000000000     ,
3*1.500000000000000     , 2*0.1000000000000000     ,  0.2000000000000000     ,
2*0.5000000000000000     , 14*0.0000000000000000E+00,
SDPIC   = 2*40.00000000000000     ,   30.00000000000000     ,
2*4.000000000000000     ,   1.250000000000000     , 2*0.5000000000000000     ,
  1.500000000000000     , 2*0.7500000000000000     , 14*0.0000000000000000E+00,
PRIORI  = 5*4.6000000000000000E-09,  4.6000000000000000E-10,
3*3.5000000000000000E-09,  3.1500000000000000E-08, 2*2.2500000000000000E-08,
 2.2500000000000000E-09,  3.5000000000000000E-10, 17*0.0000000000000000E+00,
PRIORJ  = 3*1.2250000000000000E-14, 7*0.0000000000000000E+00,
POSBND  =   200.0000000000000     ,
VELBND  =   30.00000000000000     ,
ANGBND  =   10.00000000000000     ,
BTHRSH  = 3*0.2000000000000000     , 3*1.2000000000000000E-03,
2*6.0000000000000000E-02,   12.00000000000000     ,   1.200000000000000     ,
3*0.2000000000000000     ,  0.6100000000000000     , 11*0.0000000000000000E+00,
FTHRSH  = 3*0.5000000000000000     , 3*1.2000000000000000E-02,
2*0.1500000000000000     ,   20.00000000000000     ,   2.500000000000000     ,
3*0.5000000000000000     ,   1.500000000000000     , 11*0.0000000000000000E+00,
HRADAR  =   30.48000000000000
$END
DECIDE: Sensor type MLS el      , Replication No. 2 Failed at time=  81.000
        Estimated Failure Level =  3.23494907E-03  deg

         GLIDESLOPE CAPTURE AT TIME = 157.40

         GLIDESLOPE CAPTURE PLUS 10 AT TIME = 167.35

         LOCALIZER ONCOURSE AT TIME = 192.25

         *       BEGIN DECRAB AT TIME = 245.05          *
Radar Altimeter turn-on at t= 251.70
CAT II DECISION HEIGHT          AIRCRAFT HEIGHT =       99.94    AIRCRAFT LATERAL DEVIATION =     -2.15
AIRSPEED ERROR =       0.22     GLIDESLOPE OFFSET =     199.56    TIME =  251.750 SEC
         *        BEGIN FLARE AT TIME = 257.15          *
    ENDPOINT DISPERSION DATA      HIGHTR =       0.16     XPOS     =    1324.81      YPOS =      -0.54
    HDOT =      -3.04      AIRSPEED=,      195.49      TIME =  264.950 SEC
```

FIG. 10. Sample Output File, concluded

4.4 Plot File

The plot file (file ext=.PLT) is an unformatted (binary) file of user selectable time history data. The purpose of this file is to capture the continuous element of the environment. Coded into the file are a set of unique 5-character variable names, along with the engineering units of each variable saved. The user selects the variables he wishes to save by answering "Y" to a series of interactive prompts issued by FINDS. These prompts were displayed in Section 3.1 on page 14.

Table 6 provides a list of the unique variable names, and their units, along with their definitions. It separates these definitions by the interactive questions asked by FINDS. Note that "error" always refers to "true - estimated" values, and error from desired refers to "true - desired" quantities. These definitions are needed in order for the user to use the post processing programs described in Appendix A.

The following replicated sensor mnemonics, refered to as "q mnemonics" in the table, are employed; where the "i" stands for the sensor replication number:

AZ = MLS azimuth
EL = MLS Elevation
RN = MLS range
IA = Indicated Airspeed
PH = Phi
TH = Theta
PS = Psi
RA = Radar Altimeter

other mnemonics are defined locally in the table in order to condense the descriptions and improve readability.

TABLE 6. Plot File Mnemonics

Prompting Question

Variable name	Definition
1. No-Fail filter state est error?:	
EX	X_{rw} error
EY	Y_{rw} error
EZ	Z_{rw} error
EXD	$\dot{X}_{rw}$ error
EYD	$\dot{Y}_{rw}$ error
EZD	$\dot{Z}_{rw}$ error
EPHI	phi error
ETHE	theta error
EPSI	psi error
EXW	X_w error
EYW	Y_w error
2. No-Fail filter state est Uncertainty?:	
SDXi	Standard deviation of the total, updated, state estimation error, i.e., $P_x(k+1/k+1)$. Total of eleven (11) items. Ordered by state index. For example, SDX11 => Phi estimation error standard deviation, and the units are degrees.

Since the .PLT file is binary, it must be post-processed. Two programs are provided to perform this function. Program PRINTD can be used to print

TABLE 6. Plot File Mnemonics, continued

Prompting Question

Variable name	Definition
3. Bias filter state estimate?	
B-AX	Ax bias error
B-AY	Ay bias error
B-AZ	Az bias error
B-P	P bias error
B-Q	Q bias error
B-R	R bias error
B-Azm	Azm bias error
B-El	El bias error
B-RNG	Rng bias error
B-IAS	IAS bias error
B-PHI	phi bias error
B-THE	theta bias error
B-PSI	psi bias error
B-RA	radar altimeter bias error
SDBi	Standard deviation of the updated bias filter estimation error covariance. Ordered by sensor index. For example, SDB7 => estimation error standard deviation from MLS azimuth bias, and the units are degrees.

tables of data, and program PLOTD can be used to make plots of the data. Both of these post-processing programs are described in Appendix A.

TABLE 6. Plot File Mnemonics, continued

Prompting Question

Variable name	Definition
4. No-Fail filter inputs?	
U-AX	x accelerometer input to the filter
U-AY	y accelerometer input to the filter
U-AZ	z accelerometer input to the filter
U-P	P gyro input to the filter
U-Q	Q gyro input to the filter
U-R	R gyro input to the filter
5. No-Fail filter outputs?:	
Y-AZ	MLS Azimuth output estimate
Y-El	MLS Elevation output estimate
Y-RNG	MLS Range output estimate
Y-IAS	IAS output estimate
Y-PHI	Phi output estimate
Y-THE	Theta output estimate
Y-PSI	Psi output estimate
Y-RA	Radar Altimeter output estimate

TABLE 6. Plot File Mnemonics, continued

Prompting Question

Variable name Definition

6. No-Fail filter Residuals?:

RBp	No fail filter residuals (using averaged measurements) where p => use same mnemonics as question 5. i.e. RBPSI => residuals for psi measurements.

7. Li, Fi, Ii for detectors?:

L-q	L-q = a posteriori probability of failure for sensor q (NFT items in all)
F-q	F-q = failure level estimate for sensor q (NFT items in all)
I-q	I-q = Information in F-q for sensor q (NFT items in all)
L-NF	L-NF = a posteriori probability no failures have occurred (NFT items in all)
HYP-I	HYP-I = the selected failure hypothesis
	q => mnemonics defined on page 81 (i.e. L-P, F-IA2, etc.)

8. Expanded (& filtered) residuals?:

RF-m	RF-m = non-averaged residuals for sensor m (3 or 6 items in all)
RN-m	RN-m = RF-m filtered through a colored noise filter, which attempts to remove the time correlation present in all MLS measurements. m = replicated MLS sensor where: AZi => azimuth, ELi => elevation, RNi => range, and i => replication number.

TABLE 6. Plot File Mnemonics, continued

Prompting Question

Variable name	Definition
9. A/C latitude & longitude	
ALAT	A/C latitude estimate
ALONG	A/C longitude estimate
ALATD	Latitude rate estimate
ALOND	Longitude rate estimate
10. Ground track info?	
XPOS	true X_{rw}
YPOS	true Y_{rw}
ZPOS	true Z_{rw}
XVEL	true $\dot{X}_{rw}$
YVEL	true $\dot{Y}_{rw}$
ZVEL	true $\dot{Z}_{rw}$
11. True attitudes?	
PHI-T	true phi
THE-T	true theta
PSI-T	true psi

TABLE 6. Plot File Mnemonics, continued

Prompting Question

Variable name	Definition
12. True body accel?:	
Ax-T	true x acceleration
Ay-T	true y acceleration
Az-T	true z acceleration
13. Airspeed?:	
AIRST	true airspeed error from desired
AIRSE	estimated airspeed error from desired
14. Performance measures?:	
CTE-T	true cross track error from desired (in meters)
CTE-E	estimated cross track error from desired (in meters)
DH-T	true vertical track error from desired (in meters)
DH-E	estimated vertical track error from desired (in meters)
CTEDT	true cross track error rate from desired (in m/s)
CTEDE	estimated cross track error rate from desired (in m/s)
15. Body P,Q,R info?:	
P-T	True pitch rate
Q-T	True roll rate
R-T	True yaw rate

TABLE 6. Plot File Mnemonics, continued

Prompting Question

Variable name	Definition
16. Control info?:	
DELE	elevator input (degrees)
DELA	aileron input (degrees)
DELR	rudder input (degrees)
THRST	thrust (pounds)
17. RSDIMU info?:	
Ax-mr	RSDIMU estimated Ax measurement
Ay-mr	RSDIMU estimated Ay measurement
Az-mr	RSDIMU estimated Az measurement
P-mr	RSDIMU estimated P measurement
Q-mr	RSDIMU estimated Q measurement
R-mr	RSDIMU estimated R measurement
PH-mr	RSDIMU computed phi from the first navigator
TH-mr	RSDIMU computed theta from the first navigator
PS-mr	RSDIMU computed psi from the first navigator
PHI-mr	RSDIMU computed phi from the first navigator
TH2-mr	RSDIMU computed theta from the first navigator
PS2-mr	RSDIMU computed psi from the first navigator
PijRI	Value of the parity equation for rate gyro combination ij

TABLE 6. Plot File Mnemonics, concluded

Prompting Question

Variable name	Definition
PijLA	Value of the parity equation for linear accelerometer combination ij (6 variables in all)
Li-F	Failure flag for linear accelerometers, 0=> ok, 1=> failed. (4 variables in all)
Gi-F	Failure flag fore rate gyros, 0=> ok, 1=> failed. (4 variables in all)

18. Measurement ERROR histories?:

AXE1	simulated X accelerometer error, (measured (i.e. noisy) - true value, first replication only)
AYE1	simulated Y accelerometer error
AZE1	simulated z accelerometer error
PE1	simulated P gyro error
QE1	simulated Q gyro error
RE1	simulated R gyro error
AME1	simulated MLS azimuth error
ELE1	simulated MLS elevation error
RNE1	simulated MLS range error
ASE1	simulated indicated airspeed error
PHE1	simulated phi error
THE1	simulated theta error
PSE1	simulated psi error
RAE1	simulated radar altimeter error

This Page Intentionally Left Blank

5. CONCLUDING COMMENTS AND RECOMMENDATIONS

This manual has provided detailed operational information regarding the program FINDS, version 3.0. In preparing this manual special consideration was given to be sure that all symbols and user specified variables were completely defined and crossreferenced in several ways so that this manual could be used conveniently as a reference vehicle. It is our hope that by supplying this level of detailed documentation the program can be used effectively and extended easily. The reader who is interested in an initial performance evaluation of the program should read the contract's final report [2].

As is always the case with computer software, there is always room for improvement. After using the program for an extended period of time, the following recommendations relating to the "mechanics" of the program are suggested for future program modification. (Performance considerations are deferred to the Final report [2].)

- o Save the time line file as a fixed length random access (binary) file to save disk space and increase the file's utility.
- o Increase the amount of information stored in each "snapshot" contained in the time line file.
- o Have each program run update a common file with information about its generated output files, so that all the output files will be part of a distributed data base. In this way, the post analysis process can be more fully automated. (See page 12.)
- o Generate a program to post-process the time line file and generate the statistics mentioned on page 75.
- o Buffer the interactive responses so that they can be typed out onto the log files when running in batch mode. In this way the log file could provide a convenient "audit trail" for each run.
- o Move the failure specification, currently in the sensor input file, into the interactive portion of the program. This would reduce the

total number of files needed in most instances, and would insure that the failures were set properly at all times.

- Allow the filter/detector portion of FINDS to use up to triple redundancy.

APPENDIX A
POST-PROCESSING PROGRAMS

As mentioned in Chapter 4, several support programs are available for processing the unformatted (PLT) file created by FINDS. These programs are all interactive computer programs and are fairly self-explanatory to use (user prompts, data validity checks, and error recovery are all performed). Two programs currently exist. They are: PRINTD, for printing the time history data, and PLOTD, for plotting the data.

To introduce the user to the capabilities of these programs, brief descriptions of each will be presented. The rest of the appendix is organized into two sections which document PRINTD and PLOTD, respectively.

A.1 Description of Program: PRINTD

Purpose: To examine the unformatted (binary) time history data stored in the .PLT file, generated by FINDS. PRINTD can be used to either display selected data in tabular form or compute temporal means and autocorrelations. The results are presented to the user's terminal, the system line printer, or a user-specified data file.

OPERATION: To invoke PRINTD, the user simply types (at the monitor level):

$ RUN PRINTD

PRINTD will then prompt the user for various directive commands. Time history data is identified by a unique name saved in the .PLT file. (See Table 6 on page 82 for a list of these names).

Outputs: Either;

(a) A table of selected data, where each column of data is headed by the data name and engineering units. Data can be "windowed" by selecting upper and lower limits on time. Within a window, one can further segment the data by specification of a constant skip factor.

or,

(b) All of the following:

1. The sample mean computed as:

$$SM = \frac{1}{N} \sum_{i=1}^{N} X(i)$$

where N is the total number of points in the data, and X is the selected data name.

2. The sample variance computed as:

$$SV = \frac{1}{N} \sum_{i=1}^{N} [X(i)-SM]^2$$

3. The sample autocorrelation function, normalized by the sample variance, computed as:

$$SACF(k) = \frac{1}{SV*N} \sum_{i=1}^{N} [X(i)-SM]\ [X(i-k)-SM]$$

4. The decision of a whiteness test performed on the selected data. It assumes that each SACF(k) are samples from the same normal distribution, and should lie in the band $\pm$ 1.96 N more than 95% of the time for the signal to be considered white. (See [9] for a more detailed description of this test).

A.2 Description of Program: PLOTD

Purpose: To plot the unformatted (binary) time history data stored in the .PLT file generated by FINDS.

Operation: To invoke PLOTD, the user simply types (at the monitor level):

$ RUN PLOTD

PLOTD will then prompt the user for various directive commands. Time history data is identified by a unique name saved in the .PLT file. (See Table 6 on page 82 for a list of these names).

Outputs: PLOTD can be used to generate plots of one, or several variables versus time, or to create cross plots of one variable vs. another. Currently, PLOTD creates a single plot/page. The plots can be drawn on any of the following devices or saved in a device independent disk file: TEKTRONIX 4010 or 4014, and RETROGRAPHICS.

This Page Intentionally Left Blank

6. REFERENCES

[1] "An Aircraft Sensor Fault Tolerant System", A.K. Caglayan and R.E. Lancraft, NASA CR-165876, April 1982.

[2] "A Fault Tolerant System for an Integrated Avionics Sensor Configuration", A.K. Caglayan and R.E. Lancraft, BBN Report No. 5352, Bolt Beranek and Newman, Inc., Cambridge, MA, October 1983.

[3] "Terminal Area Automatic Navigation, Guidance and Control Research Using the Microwave Landing System (MLS), Part 2 - RNAV/MLS Transition Problems for Aircraft", S. Pines, NASA CR-3511, January 1982.

[4] "Terminal Area Automatic Navigation, Guidance and Control Research Using the Microwave Landing System (MLS, Part 3 - A Comparison of Waypoint Guidance Algorithms for RNAV/MLS Transitions", S. Pines, NASA CR-3512, January 1982.

[5] "Automated Landing, Rollout, and Turnoff Using MLS and Magnetic Cable Sensors", S. Pines, S.F. Schmidt, and F. Mann, NASA CR-2907, October 1977.

[6] "Design of a Developmental Dual Fail Operational Redundant Strapped Down Inertial Measurement Unit", F.R. Morrell and J.G. Russell, NAECON, May 1980.

[7] "VAX-11 FORTRAN Language Reference Manual. (VMS V3.0)", Digital Equipment Corporation, 1982

[8] Preliminary Design of a Redundant Strapped Down Inertial Navigation Unit Using Two-Degree-of-Freedom Tuned-Gimbal Gyroscopes", Litton Guidance and Control Systems, NASA CR-145035, October 1976.

[9] "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems", R.K. Mehra and J. Peschon, Automatica, Vol. 7, 1971.

[10] "Failure Detection and Isolation Analysis of a Redundant Strapdown Inertial Measurement Unit", P. Motyka, M. Landey, and R. McKern, NASA CR-165658, February 1981.

[11] "Wind Models for Flight Simulator Certification of Landing and Approach Guidance and Control Systems", N.M. Barr, D. Gangaas and D.R. Schaeffer, FAA-RD-74-206, December 1974.

This Page Intentionally Left Blank

INDEX

AC 45
ACSTAT 56, 58
AG 25
AIRSPE 20
ANGBND 55
AR 25
AS 25

BOUNAZ 45
BOUNEL 45
BOUNRN 45
BTHRSH 55

CA 46
CG 20
CPUCHK 19

DT 25
DTAG 43
DTAR 50
DTAS 37
DTGR 47
DTIME 18
DTLA 41
DTML 45
DTRA 35
DTRG 38

FDI 25
FDIAR 57, 60
FDIGR 57, 59
FIB 25
FIBAG 44
FIBAR 51
FIBAS 37
FIBGR 49
FIBLA 42
FIBML 46
FIBRA 36
FIBRG 39
FILTER 18
FILTIN 57, 60
FIN 25
FINAG 43
FINAR 51
FINAS 37
FINGR 48
FINLA 41
FINML 46
FINRA 36
FINRG 39
FP 25
FPAG 57, 59
FPAR 57, 60
FPAS 56, 58
FPGR 57, 59
FPLA 57, 59
FPML 57
FPMLS 59
FPRA 56, 58
FPRG 56, 59
FR 25
FRAG 44
FRAR 51
FRAS 37
FRGR 49
FRLA 42
FRML 46
FRRA 36
FRRG 40
FSF 25
FSFAR 51
FSFGR 49
FSFLA 42
FSFRG 40
FTHRSH 55

GPIPRN 21
GR 25
GRQSF 48
GSLOC 56, 58
GUST 21

HG 23
HIGHT 20
HRADAR 55

IC 24
ICNTRL 53, 73
ID 25
IDAG 43
IDAR 50
IDAS 36
IDGR 47
IDLA 40
IDML 45
IDRA 35
IDRG 38
IEST 19, 73
IGUST 21
IHEAD 19
IIMUF 53, 73
INOBPS 53
IPARGR 49
IPFOAR 52
IPFOGR 49
IPRNT 19
IREP 25
IREPAG 43
IREPAS 37
IREPLA 41
IREPML 45
IREPRA 35
IREPRG 38
IRSDF 53, 73
IRSDFY 53
IRSOIM 47
IWHMLS 45

KMXEWH 53
KMXHLR 53
KPATH 23

KPRINT 19

LA 25
LOCRAN 21

ML 25
MP 25
MPAG 57, 59
MPAR 57, 59
MPAS 56, 58
MPGR 57, 59
MPLA 56, 59
MPMLS 57, 59
MPRA 56, 58
MPRG 56, 58

NDISTB 0
NEWPTH 23
NFMAX 0
NFT 0
NTICK 19
NU1 0
NW 23
NX 0
NYMAX 0

OUTPS 19

PBFOI 54
PBFIC 54
PDEG 20
PFKAR 52
PFKGR 49
PFTCAR 52
PFTCGR 49
PFTHAR 52
PFTHGR 49
PITCH 20
PLOTS 19
POSBND 54
PRIORI 54
PRIORJ 54
PRNTPL 56, 58
PROGCN 18, 56, 58
PSIR 22
PSIW 21

QDEG 20

RA 25
RDEG 20
RE 20
RG 25
RHO 20
RINLA 20
RINLO 20
ROLL 20
RT 23

SCALEW 21
SDB 25
SDBAG 43
SDBAR 50
SDBAS 37
SDBGR 47
SDBLA 41
SDBML 45
SDBRA 35
SDBRG 38
SDGDGR 48
SDMN 25
SDMNAS 37
SDN 25
SDNAG 43
SDNAR 50
SDNGR 47
SDNLA 41
SDNML 45
SDNRA 35
SDNRG 38
SDPIC 54
SDSF 25
SDSFAR 50
SDSFGR 48
SDSFLA 41
SDSFRG 39
SDSUBW 21
SDTM 25
SDTMGR 48
SDTMLA 41
SDTMRG 39
SDXIC 54
SH1 21
SH2 21
SHEAR 21
SHZ 23
SIG 54
SLA 23
SLIM 25
SLIMAG 43
SLIMAR 50
SLIMAS 37
SLIMGR 48
SLIMLA 41
SLIMRA 35
SLIMRG 39
SLO 23
SNDP 24
STARTER 56, 58

TANLIM 24
TF 25
TFAG 44
TFAR 51
TFAS 38
TFGR 49
TFLA 42
TFML 46
TFRA 36
TFRG 40
THETAG 22
THRUST 20
TSTOP 19

VELBND 54
VGD 23

WEIGHT 20
WINDAT 56, 58
WINDOW 53
WINDSS 21
WLA 23
WLO 23

XAZ 45
XEL 45

YAW 20
YSTOP 19

1. Report No. NASA CR-172199	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle FINDS: A Fault Inferring Nonlinear Detection System - User's Guide		5. Report Date September 1983 6. Performing Organization Code
7. Author(s) R. E. Lancraft and A. K. Caglayan		8. Performing Organization Report No. 5358
9. Performing Organization Name and Address Bolt Beranek and Newman Inc. 10 Moulton Street Cambridge, MA 02238		10. Work Unit No. 11. Contract or Grant No. NAS1-16579
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Contractor Report 14. Sponsoring Agency Code
15. Supplementary Notes Langley technical monitor: Frederick R. Morrell Computer Program documentation		

16. Abstract

The computer program FINDS is written in FORTRAN-77, and is intended for operation on a Digital Equipment Corporation's (DEC) VAX 11-780 or 11-750 super mini computer, using the VMS operating system.

The objective of FINDS is to detect, isolate, and compensate for failures in navigation-aid instruments and on-board flight control and navigation sensors of a Terminal Configured Vehicle (TCV) aircraft in a Microwave Landing System (MLS) environment. In addition, FINDS provides sensor fault tolerant estimates for the aircraft states which are then used by an automatic guidance and control system to land the aircraft along a prescribed path. FINDS monitors for failures by evaluating all sensor outputs simultaneously using the nonlinear analytic relationships between the various sensor outputs arising from the aircraft point mass equations of motion. Hence, FINDS is an integrated sensor failure detection and isolation system.

17. Key Words (Suggested by Author(s)) sensor failure detection, nonlinear estimation, fault tolerant navigation, analytic redundancy, MLS, No Fail filter		18. Distribution Statement Unclassified - Unlimited Subject Category 04	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 111	22. Price A06

N-305

End of Document