

NAVSIM II A Computer Program for Simulating Aided-Inertial Navigation
for Aircraft.

William S. Bjorkman

(NASA-CR-177438) NAVSIM 2: A COMPUTER
PROGRAM FOR SIMULATING AIDED-INERTIAL
NAVIGATION FOR AIRCRAFT (Analytical
Mechanics Associates) 94 P CSCL 17G

N88-12478

Unclas
G3/04 0110667

CONTRACT NAS2-10850

March 1987

NASA

NASA CONTRACTOR REPORT 177438

NAVSIM II A Computer Program for Simulating Aided-Inertial Navigation
for Aircraft.

William S. Bjorkman
Analytical Mechanics Associates, Inc.
Mountain View, CA 94043

Prepared for
Ames Research Center
Under Contract NAS2-10850
March 1987



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

Section 1 INTRODUCTION

This report describes NAVSIM II, a computer program for analytical simulation of an aided-inertial navigation system for aircraft -- and, in particular, for helicopters. This program was developed for NASA-Ames Research Center by Analytical Mechanics Associates, Inc. (AMA) as one of the tasks under contract NAS2-10850. This report comprises the final report for that contract.

As its name implies, NAVSIM II is a second-generation navigation simulation, superceding an earlier program, NAVSIM, which has been used by NASA/ARC's Guidance and Navigation Branch in support of STOLAND and V/STOLAND flight tests. NAVSIM II was developed independently of NAVSIM, however, using structured design principles and techniques to produce a "clean" program which can be easily understood, modified and extended. The resulting software is functionally modular, reasonably uncoupled, well-commented and written in fairly standard Fortran 77 for ready transportability. Care was taken in the design to minimize the user's workload and confusion. A significant goal of the design and development effort was to produce not just a program for solving a particular problem, but rather the core software for application to solution of many future problems through extension and modification.

NAVSIM II simulates a strapped-down inertial navigation system which may be aided by measurements of:

- * GPS pseudo-range
- * barometric and/or radio altitude
- * Doppler radar (velocity)
- * airspeed
- * position (fixes).

Aiding is implemented by processing these measurements through an extended Kalman filter. Bierman's U-D procedure [1] is used in the filter implementation. Concern is taken for some of the considerations of real-system implementation (processing order, data screening), but the Fortran simulation is not constrained by such common real-system considerations as fixed-point arithmetic, allowable cycle time and word length.

Figure 1 presents an overview of the simulation process implemented in NAVSIM II. The "actual trajectory" generator (TAF COS) provides:

- * initial position, velocity and attitude for initiating the integration of the estimated navigation equations
- * specific forces and body rates to simulate the measured accelerometer and gyro outputs of the strapped-down IMU
- * position and velocity vectors to simulate the "true" measurements from the navaids.

The IMU model adds simulated sensor errors to the "actual" specific forces and body rates to form the simulated inertial-sensor outputs. These outputs are the inputs to the navigation equations (integration of the differential equations for position, velocity and attitude). When measurements are simulated, the estimated position and velocity and the "actual" position and velocity are used by the various measurement simulators to form the "actual" and estimated measurements. The pre-processor logic in each of these modules produces:

- * the measurement residual
- * the sensitivity (partial) of the measurement to filter-state variations
- * the filter weight (user-supplied).

These residuals, sensitivities and weights are processed by the Kalman filter to produce new values of the estimation errors (i.e., the filter-states). These values are added to the previous estimates to form new estimates of position, velocity, tilts and inertial-sensor bias compensations (all used in the navigation equations) and new estimates of measurement bias compensation (used in measurement pre-processing).

The next sections describe the hardware/software configuration simulated by NAVSIM II, the program's application to aircraft navigation investigations and procedures for utilization of the program. Later sections are devoted to documentation of the software, discussion of future extensions and recommendations for program enhancement. Five appendices are included. The first of these presents the mathematical formulation (without derivation). The second contains procedures and examples for program modification. The third discusses the utility software for plotting and for specification of the reference flight path. The last two appendices contain collections of subroutine summaries and common descriptions.

ORIGINAL PAGE IS
OF POOR QUALITY

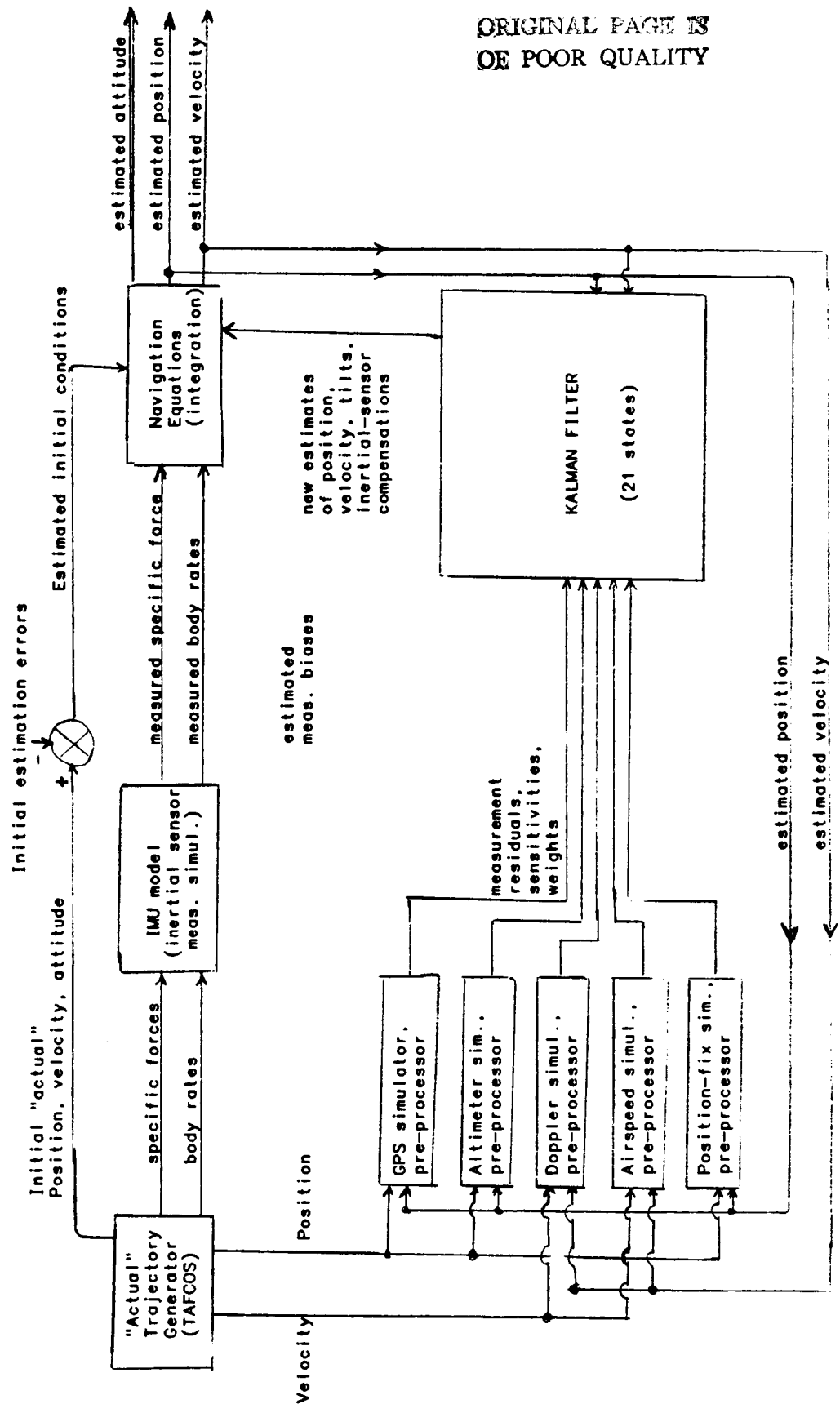


Figure 1: NAVSIM II Simulation Process

Section II THE SIMULATED SITUATION

NAVSIM II simulates an aided-inertial navigation system flown aboard a helicopter. Its equations and logical design represent a certain range of hardware, software and vehicle configurations. It may be used to simulate navigation during several different mission scenarios.

Configuration

The vehicle simulated in NAVSIM II is considered to be a helicopter or other rotorcraft. Nothing in the navigational logic prevents the vehicle from being considered as a CTOL aircraft, however. It is only the flight path, attitude and rate information which are generated for the simulated "true" aircraft which restrict the interpretation. These variables are generated within NAVSIM II according to a realistic model of the UH-1H helicopter flying under control of the TAF COS [2] guidance and automatic flight control system.

The simulation assumes that the airborne hardware includes:

- (1) a strapped-down inertial measurement unit (IMU) which includes a triad of inertial-grade laser gyros and a triad of integrating accelerometers,
- (2) a navigation computer interfaced with the IMU, the various external-measurement devices and a panel or other display device,
- (3) a GPS receiver interfaced with the navigation computer,
- (4) a Doppler radar receiver interfaced with the navigation computer,
- (5) barometric and radio altimeters interfaced with the navigation computer,
- (6) an airspeed indicator interfaced to the navigation computer.
- (7) a device capable of displaying the navigational results and accepting mode controls or other data from the navigator.

The airborne software represented in NAVSIM II is that which is necessary to implement an aided-inertial strapped-down navigation system: namely, software to perform the free-inertial strapped-down navigation function, Kalman filtering software to perform the estimation (aiding) function, and interfacing

software for communicating between the navigation computer and the various external devices. It should be noted that the airborne navigation software would probably differ considerably from that found in NAVSIM II. The simulation is not subject to the time and storage constraints which are found in an airborne implementation, nor is it subject to the variety of problems which inevitably arise in the use of real data and real sensors.

Mission Scenarios

The missions which have been considered in the design of NAVSIM II are those civilian missions which might be undertaken by a rotorcraft equipped with the navigational hardware and software described above. Such missions include search-and-rescue, police surveillance, oil/gas exploration, land management and remote-site transportation, as well as airport-to-airport operation.

The user of NAVSIM II may choose to simulate navigation on any or all portions of a helicopter mission, although the terminal-area portion is generally of the most interest in navigation and guidance analyses. In its simplest form, a mission consists of a departure phase, an en-route phase and an approach-and-landing phase. When the navigation system is inertial, the departure phase must be preceded by an alignment phase during which initial conditions are initialized on position, velocity and attitude estimates in the navigation computations. This alignment phase takes place while the craft is stationary. Transition to the "navigate" mode can be initiated either manually (via the control panel) or automatically, using some criterion programmed into the navigation computer. During flight to the destination, the navigation estimate is continually updated (i.e., computed) through integration of the inertial navigation equations and aided through processing of the various external measurements by the Kalman filter logic. The particular mixture of external measurements to be included would usually be determined by a measurement schedule specified by the navigator before and/or during the flight.

Applications

NAVSIM II is written as a tool which may be used by a knowledgeable engineer to study the effects of the various aided-navigation system parameters. With this tool, he can:

- * design/configure airborne systems
- * plan research flight projects
- * analyze reported results of real or simulated navigation systems
- * perform analytical studies of hypothetical configurations.

This tool may also be used as an educational device for exhibiting the effects of the various parameters on system performance. Among the parameters the user may vary are:

- * inertial system errors such as gyro drifts and accelerometer null biases
- * measurement errors
- * initial condition errors
- * filter weights
- * process noise
- * measurement schedules
- * flightpath.

Section III UTILIZATION

NAVSIM II resides on the MARS Vax 11/780 computer at NASA/ARC. It requires two input files and it produces seven output files in addition to the system output file (log file). These are shown in Figure 2. Both input and output files are self-documenting. That is, the input numbers are all titled within the files and the input files are identified by a header. The output files are tagged with the input header and dated and paged automatically. Output formats are columnar with explicit headings.

The input files are prepared by the user by means of a system editing utility (e.g., EDT) before running NAVSIM II. Formats and contents of these files will be described later in this section.

The program is invoked (executed) interactively by typing

RUN NAVSIM

in response to the operating system's "\$" prompt. The user will then be prompted by

enter navigation file name :

He must enter the name of the navigation-parameter input file name, complete with extension. The program will then prompt

enter trajectory file name:

and this name must be provided. The program will then proceed into execution, not pausing or interacting until completion of its scheduled calculations. System output will be delivered to the terminal as it is produced. This information is not of much general interest. The normal output of interest will be found in the other files. The user may TYPE, PRINT or edit these files to view their contents. Each file produced has the principal name of the input navigation-parameter file as supplied by the user in response to the first prompt from NAVSIM II. The extensions (.OIN, .ER1, .ER2, .SG1, .SG2, .RES, .TRJ) are appended by NAVSIM II to identify the contents.

NAVSIM II may also be run in the batch mode by entering

@NAVSM

to invoke a command procedure. This procedure will prompt the user for the navigation-parameter and trajectory input-file names. Finally, the user is prompted for which batch queue he wishes (SUBMIT, MBAT, LBAT). System output is written to a log-file of which the primary filename is that of the navigation-parameter file (e.g., NAVXXX.LOG).

ORIGINAL PAGE IS
OF POOR QUALITY

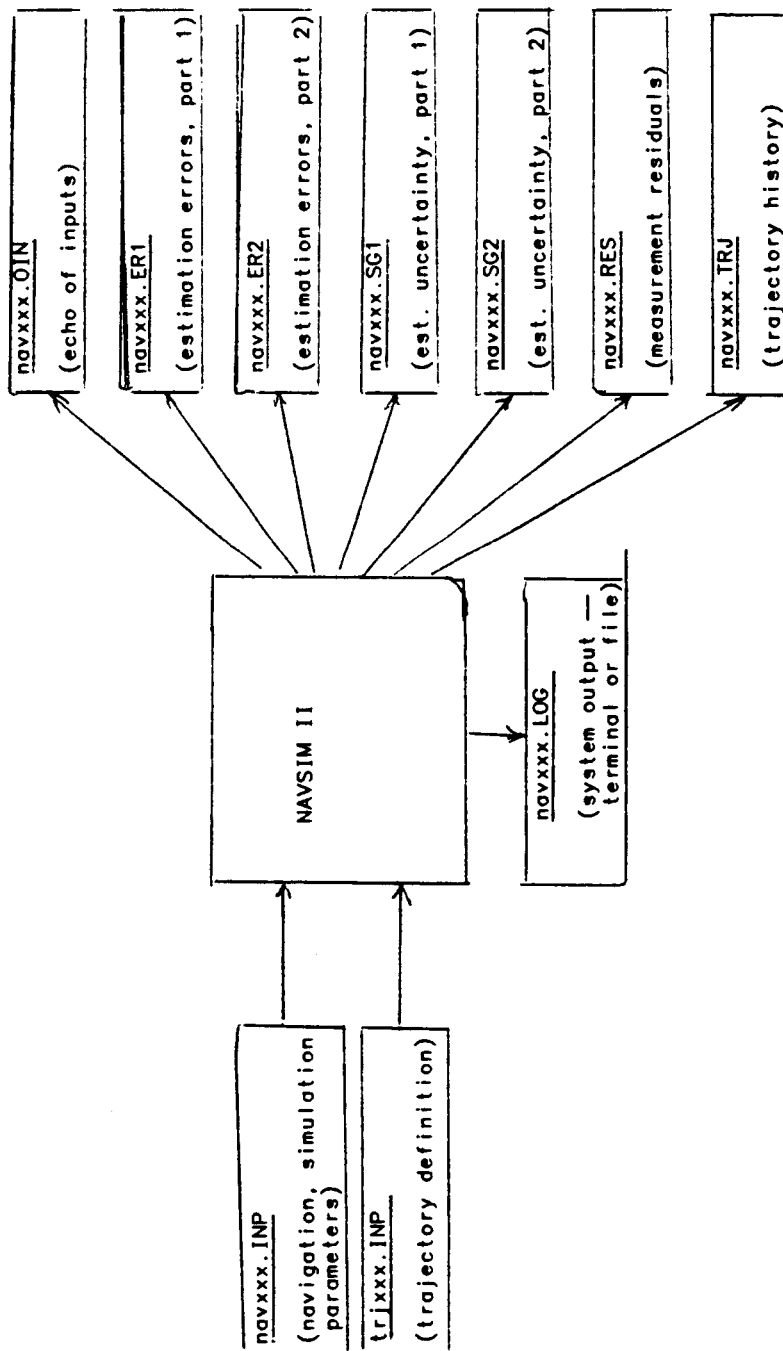


Figure 2: Files Used and Produced by NAVSIM II

Input Files

Navigation Parameters

The navigation-parameter file (e.g., NAVXXX.INP) has a structured format which can be changed only with care and recompilation. This format is shown in Figure 3. Observe first that the first line-entry is a header line. This line becomes the user's identification in the output files which will be produced by running NAVSIM II with this input file. The remaining parameters to be specified are functionally grouped:

Group	Function
*****	*****
1	Run scheduling and Monte Carlo treatment
2	Measurement schedules
3	Navigation data (i.e., filter parameters)
4	Simulated errors
5	GPS satellite information
6	Runway reference.

Within each of the functional groups several numbers (parameter values) are to be supplied. The numbers are preceded by "=" and a description in English. Any line without an "=" is treated strictly as a comment line and may be omitted or added without effect. The numbers are separated by commas and may or may not include decimal points. Scientific notation is not recognized. The input file is read and interpreted in subroutine READINP. READINP interprets each line according to its outline index (i.e., the left-hand line numbers in Figure 3). For example, a line with index 2.5 is expected to contain three GPS measurement-scheduling parameters (see Figure 3), while line 3.3.5 is expected to contain an airspeed measurement weight. Thus, while the parameter definitions may be changed to suit the user, the indices (line numbers) must remain as shown -- unless the user wants to modify READINP.

Figure 3. Navigation Parameter File, navxxx.INP

GPS ONCE/SEC WITH EPHEM ERRORS

```

1      RUN SCHEDULING
1.1    ALIGNMENT START           = 0.
1.2    ALIGNMENT STOP           = 0.
1.3    FILTER START             = 0.
1.4    STOP TIME                = 100.
1.5    NO. OF MONTE CARLO CASES = 5
1.6    START RANDOM NO (500<I<1000) =512
1.7    IC AND IMU ERRORS       = 1
1.8          (0-BIAS; 1-RANDOM)
1.9
2      MEASUREMENT SCHEDULE - SECS
2.1
2.2          MSMNT INT-SEC      START SEC      STOP SEC
2.3    BARO ALTIMETER =      5.,           7200.,      7400.
2.4    RADIO ALTIMETER =      1.,           400.,       7400.
2.5    GPS              =      1.,           0.,         7400.,
2.6    DOPPLER          =     10.,           7200.,      3000.
2.7    AIRSPEED         =     10.,           7200.,      7400.
2.8    POSITION FIX      =     10.,           100.,       7400.
2.9
3      FILTER PARAMETERS
3.1    a priori STATE UNCERTAINTIES (STD. DEVS.)
3.1.1
3.2          PMATRIX (NEGATIVE TO IGNORE STATE)
3.2.1    3 - POSITIONS - FT.           = 300,300,100
3.2.2    3 - VELOCITIES -FT/SEC       = 3,3,3
3.2.3    3 - TILTS - ARC SEC          = 2,2,2
3.2.4    3 - GYRO DRIFT - DEG/HR      = .01,.01,.01
3.2.5    1 - ACCEL NULL BIAS - MICROG = 10.
3.2.6    1 - CLOCK BIAS - NANOSEC     = 20
3.2.7    1 - CLOCK BIAS RATE NSEC/HR  = 10.
3.2.8    3 - DOPPLER ANT. TILT - DEG  = 1,1,1
3.2.9    1 - AIRSPEED SCALE FACTOR -%  = 1
3.2.10   2 - WIND ESTIMATE - FT/SEC    = 5,5
3.2.11
3.3    ESTIMATED MEASUREMENT ACCURACY
3.3.1    BARO ALTIMETER - FT          = 50.
3.3.2    RADIO ALTIMETER - FT         = 5.
3.3.3    GPS RANGE - FT               = 100.
3.3.4    DOPPLER - FT/SEC             = 2.
3.3.5    AIRSPEED - FT/SEC            = 3.
3.3.6    POSITION FIX - FT             = 2.
3.3.7
3.4    FORCING FUNCTIONS
3.4.1    3 - GYRO F.F.  DEG/HR        = 2.0,2.0,2.0
3.4.2          ACCEL F.F.  MICROG     = 50.
3.4.3
3.5    FORCING FUNCTION DECAY RATE
3.5.1    3 - GYRO FF DECAY - SEC      = 600.,600.,600.
3.5.2          ACC FF DECAY - SEC     = 600.
3.5.3

```

Figure 3 (cont'd)

```

4      SIMULATED ERRORS  STANDARD DEVIATIONS
4.1    INITIAL ESTIMATION ERRORS
4.1.1    3 - POSITIONS - FT                = 0,0,0
4.1.2    3 - VELOCITIES - FT/SEC          = 0,0,0
4.1.3    3 - TILTS - ARC SEC              = 0,0,0
4.1.4    3 - GYRO DRIFT - DEG/HR          = 0,0,0
4.1.5    1 - ACCEL NULL BIAS - MICROG     = 0.
4.1.6    1 - CLOCK BIAS - NANOSEC         = 0.
4.1.7    1 - CLOCK BIAS RATE - NSEC/HR   = 0.
4.1.8    3 - DOPPLER ANTENNA TILT - DEG   = 0.,0.,0.
4.1.9    1 - AIRSPEED SCALE FACTOR - %    = 0.
4.1.10   2 - WIND ESTIMATE - FT/SEC       = 0.,0.
4.1.11
4.2      IMU MODEL ERRORS
4.2.1    GYROS          FRAC,DEG/HR = .0000,  .0,  .0
4.2.2    ACCELEROMETERS FRAC,FPS2 = .0000,  .0,  0.
4.2.3
4.3      MEASUREMENT ERRORS
4.3.1    BARO ALTIMETER - FT= 10., 50.,  1.,  1.,
4.3.2    RADIO ALTIMETER FT=  1.,  5.,  2.,  1.5,  500.
4.3.3    GPS RANGE - FT =  5.,  5.,  0.,  2.,
4.3.4    DOPPLER VEL. FT/SEC=  1.,  2.,  4.,  2.5
4.3.5    AIRSPEED - FT/SEC =  1.,  3.,  5.,  3.,  30.
4.3.6    POSITION FIX - FT =  10., 1.,  0.,  3.
4.3.7
5      SATELLITE CONFIGURATION
5.1    NOMINAL VALUES
5.1.1    NO OF ORBITS (MAX 6)              = 3
5.1.2    ORBIT INCLINATION DEG.            =60.
5.1.3    NO OF SATELLITES/ORBIT (MAX 6)= 6,6,6,0,0,0
5.1.4    ASC NODE EACH ORBIT DEG          = 0.,120.,240.,0.,0.,0.
5.1.5    ANOMALY OF FIRST SAT DEG        =0., 20., 40.,0.,0.,0.,
5.1.6
5.2    SATELLITE SELECTION FOR GPS
5.2.1    DESIRED GDOP                      = 0.0
5.2.2    INTERVAL OF FIXED SAT SELECTION SEC =600.
5.2.3    MIN ELEV VIEW ANGLE DEG          =5.0
5.2.4
5.3    EPHEMERIS ERRORS  STANDARD DEVIATION IN FEET
5.3.1    RADIAL          CROSS-TRACK  IN-TRACK
5.3.2    - 5.,          5.,          10.
5.3.3
6      GHA AND RUNWAY REFERENCE
6.1    INITIAL GHA                      =0.0
6.2    RUNWAY LAT,LON,AZ                =37.5,-121.0,0.0
7      END OF INPUT DATA

```

Run scheduling and error treatment.- The four times which are to be specified (all in seconds) are:

1. alignment start time
2. alignment stop time
3. filter start time
4. run stop time.

Alignment should coincide with a stationary trajectory period and use precise position fixes as measurements. Filter start time is a parameter for delaying the start of filtering relative to navigation (i.e., TAF COS) time, which always starts at $t=0$.

The error treatment inputs relate to the Monte Carlo operation of NAVSIM II. The program will repeat the same case n times if n is specified greater than one in line 1.5. The program's "random" number sequence is initiated at the m -th sample, where m is supplied at compilation and where m is the number supplied at input line 1.6. The error flag at line 1.7 permits the use of specific errors (flag=0) in place of random errors (flag=1). Thus, when the flag is zero, the numbers found in Group 4 (see Figure 3) are used directly as the simulated errors. When the flag is non-zero, the Group 4 numbers are used as standard deviations (scales) for the unit-variance "Gaussian" pseudo-random errors to be initiated for each Monte Carlo pass.

Measurement schedule.- The numbers in this schedule represent the interval, start-time and stop-time for each measurement type. The resolution of the interval is one second, since this is the filter's cycle rate. Measurements are omitted or suppressed by delaying the "starting" of observations until after the run's stop-time.

Filter parameters.- The first set of numbers in this group contains the square roots of the a priori covariances of the initial diagonal P-matrix (i.e., D-matrix in the U-D formulation). That is, these numbers represent the initial uncertainty in the estimation of state. The initial diagonal covariance matrix assumes that the initial error states are statistically uncorrelated.

The second set of numbers in the group contains the measurement weights (uncertainties) for use in filter-processing of the various measurements. In the case of Doppler radar and position fixes, the single number specified here is used to weight each axis component of measurement.

The third set of numbers contains the scales for gyro and accelerometer process noise. The incorporation of this process noise keeps the error covariance matrix (P-matrix, U-D matrix) from becoming too optimistic and thereby excluding measurement information from the estimate. The fourth set of numbers represents the time-constants in the process-noise integration (filter) model.

Simulated errors.- Simulated errors include errors in initial estimation of the state elements, IMU model errors and measurement errors. These represent the "actual" system errors of the simulated aided-inertial navigation system. If the error treatment flag (line 1.7) is set to 1, these numbers are used as the standard deviations (scales) for pseudo-random, unit-variance errors. If the flag is set to zero, the numbers represent the specific and particular errors of the system.

The first set of numbers describes the errors in the initial estimate of state. The initial estimate is derived in NAVSIM II (in particular, in subroutine INESTM) by subtracting these errors from the "actual" initial state.

The second set of numbers describes the error models for gyros and accelerometers of the strapped-down IMU. The scale factor, bias and random (Gaussian) errors apply to all three axes of these sensors. Scale factor and bias errors are randomly sampled as constants for each Monte Carlo pass. The "random" value is the scale for inertial sensor noise at every navigation integration step (20 times/second).

The third set of numbers describes the "actual" measurement errors for the various measurement types plus screening limits for bad measurement data and minimal data values for processing. That is, four or five numbers are supplied for each measurement type. The first number is a bias scale for random selection of measurement bias at each Monte Carlo pass. The second number is the scale for Gaussian noise in the measurement. The third number is the percentage of "bad" measurements to be simulated. These "bad" measurements are simulated by use of a uniform random number sequence on the interval (0,1). When the random sample is less than the specified percentage, the measurement residual sent to the filter is replaced by a large number which should be detected by the screening logic. The fourth number is the multiple of the weighted residual to be used as a threshold in screening bad points from filter processing. The weighted residual is the ratio of the measurement residual to the filter weight, $\sqrt{HPH^T + R}$. The fifth number for radar altimeter measurements is the limiting altitude for processing -- that is, measurements are processed only below 500 feet. The fifth number for airspeed measurement represents the lowest airspeed at which the data will be processed.

GPS satellite information.- This group has three sections pertaining to the GPS satellites. The first section pertains to the orbital configuration. The first parameter is the number of orbital planes and the second parameter is the common inclination of these planes to the equator. The next set of numbers specifies the number of satellites in each of the orbits. The next set of numbers specifies the longitude of the ascending node of each orbit, measured in the equator from the vernal equinox. The next set of numbers specifies the anomaly (angle) of the first satellite in each orbit measured along the orbit from the equator. The satellites in any orbit are uniformly distributed in angular measure from this first anomaly and remain so throughout this simulation under the assumption of circular orbits. The orbital period is modifiable only through re-compilation.

The second section of this group pertains to the selection and observation of satellites in GPS measurement. Desired GDOP is used in selecting four satellites from the 4-8 which may be in view. GPS observations cycle through the selected four for a specified interval of time (the next entry). A desired value of zero causes selection of the optimal set from the point-of-view of minimal GDOP. A large value (e.g., 1000) causes selection of the worst set from GDOP considerations. Intermediate values (e.g., 6) cause selection of a set whose GDOP is nearest the desired value. The final number in this section specifies the lowest elevation at which a satellite may be viewed.

The third section of this group relates to satellite ephemeris errors in GPS measurements. The three numbers define the scales for random selection of biases in radial, cross-track and in-track orbital components.

Runway reference.- The first number in this group is the Greenwich hour angle (GHA) at time=0. This affects only the orientation of the satellite orbits with respect to the Earth-fixed reference frame. It implies time-of-day and date and is equivalent to a bias in the longitudes of the ascending nodes of the orbits. The other three numbers are latitude(N), longitude(E) and azimuth (CW from N) of the runway reference coordinate system.

Trajectory specification. - An example of the trajectory specification file (e.g., TRJXXX.INP) is shown in Figure 4. This file is read by TAF COS to describe the reference flight path and initial conditions. The file contains comment lines and numerical-data lines. The latter are distinguished from the former by the occurrence of "=" in the latter. Some care must be taken to exclude "=" from any comment line. The first numerical-data line contains runway-x and runway-y components of wind. The second contains the initial waypoint index in the waypoint table which comes later. The third and fourth numerical-data lines contain initial position and velocity offsets relative to the initial waypoint. That is, initial position and velocity are taken to be those of the initial waypoint plus the offsets. The three numbers represent the x, y and z (downward) components of the offset. It must be noted that these are guidance offsets and not navigation errors.

The remaining numerical-data lines specify trajectory waypoint parameters for TAF COS. The number of waypoints (lines in the table) is limited to 50. The 11 numbers in each line represent the values shown in the following table.

Number in line -----	Value -----
1	waypoint index
2	x-coordinate of start of straight-line path or turn-center
3	y-coordinate of start of straight-line path or turn-center
4	turning radius (zero for straight line)
5	initial heading
6	approximate path length to be traveled to the next waypoint
7	speed (airspeed when mode=1, ground speed otherwise)
8	mode (1 for normal flight, 2 for approach and 3 for hover)
9	approx. time to traverse the path length to the next waypoint
10	initial altitude
11	flight path angle

Appendix C contains a discussion of some utility software which can aid in preparing a reference trajectory file. The file can also be formed directly, by use of the system's editing utilities.

Figure 4. Trajectory Specification File, TRJXXX.INP

TRAJECTORY DATA FOR LOCAL PATH V IS 101FPS
WIND VELOCITY FPS = 0.0,0.0
INITIAL WAYPOINT = 13

AIRCRAFT INITIAL CONDITION
POSITION ERROR-X,Y,Z- FT =0.,0.,0.
VELOCITY ERROR-X,Y,Z- FPS =0.,0.,0.

WAYPOINT TABLE

WP NO	X FT	Y FT	RAD TURN FT	INIT HDG DEG	PATH LNGTH FT	SPEED * FPS	MODE *	TIME IN SEG SEC	ALT FT	FLT PA ANGI DEG
=1,	-9000.,	0.,	0.,	0.,	0.,	-5.,	3.,	20.,	500.,	0.
=2,	-9000.,	-3000.,	-3000.,	0.,	9425.,	101.,	1.,	140.,	500.,	6.0
=3,	-9000.,	-6000.,	0.,	180.,	7000.,	135.,	1.,	44.,	1500.,	0.
=4,	-16000.,	-3000.,	-3000.,	180.,	9425.,	135.,	1.,	70.,	1500.,	0.
=5,	-16000.,	0.,	0.,	0.,	7000.,	135.,	1.,	44.,	1500.,	0.
=6,	-9000.,	-3000.,	-3000.,	0.,	9425.,	135.,	1.,	70.,	1500.,	0.
=7,	-9000.,	-6000.,	0.,	180.,	7000.,	135.,	1.,	44.,	1500.,	0.
=8,	-16000.,	-3000.,	-3000.,	180.,	9425.,	135.,	1.,	70.,	1500.,	0.
=9,	-16000.,	0.,	0.,	0.,	7000.,	101.,	1.,	50.,	1500.,	0.
=10,	-9000.,	-3000.,	-3000.,	0.,	9425.,	101.,	1.,	93.,	1500.,	0.
=11,	-9000.,	-6000.,	0.,	180.,	7000.,	101.,	1.,	60.,	1500.,	0.
=12,	-16000.,	-3000.,	-3000.,	180.,	9425.,	101.,	1.,	93.,	1500.,	0.
=13,	-16000.,	0.,	0.,	0.,	1000.,	101.,	1.,	10.,	1500.,	0.
=14,	-15000.,	2000.,	2000.,	0.,	12566.,	101.,	1.,	150.,	1500.,	-3.92
=15,	-15000.,	0.,	0.,	0.,	6000.,	5.,	2.,	83.,	637.,	-1.31
=16,	-9000.,	0.,	0.,	0.,	0.,	-5.,	3.,	20.,	500.,	0.

* IF MODE IS 1 NORMAL FLIGHT SPEED IS AIRSPEED
IF MODE IS 2 APPROACH SPEED IS GNDSPEED
IF MODE IS 3 HOVER SPEED IS GNDSPEED

Output Description

Printed Output Files

NAVSIM II produces seven output files in addition to the system output file. These files may be displayed by invoking the system PRINT or TYPE utilities. The output files are:

1. navxxx.OIN echo of the input files
2. navxxx.ER1 estimation errors, part 1
3. navxxx.ER2 estimation errors, part 2
4. navxxx.SG1 uncertainty in estimation, part 1
5. navxxx.SG2 uncertainty in estimation, part 2
6. navxxx.RES measurement residuals
7. navxxx.TRJ trajectory (actual).

The -.OIN file echoes the -.INP files of navigation and trajectory data and also lists the randomized bias-type errors which were actually used for each Monte Carlo pass.

The -.ER1 and -.ER2 files tabulate the estimation errors. The sense of these errors is (actual value - estimated value).

The -.SG1 and -.SG2 files tabulate the uncertainties in the estimates as indicated by the square roots of the diagonal elements of the state covariance (P) matrix. One line of each of these files is produced after each time-update and after each measurement event. The lines' sources are identified by an entry in a column labeled "DATA TYPE."

The -.RES file tabulates the measurement residuals which were sent to the filter for processing. In the case of GPS measurements, the satellite number and current GDOP measure are also tabulated.

The -.TRJ file tabulates the true or actual trajectory components of position, velocity and specific force. This file is written only during the first Monte Carlo pass, since the actual trajectory is invariant from pass to pass.

Plots

Utility software for NAVSIM II includes a plotting capability for graphical display of estimation errors, the trajectory and measurement residuals. This capability is invoked by typing

NAVPLT

at a graphics terminal such as a Tektronix 4014. This causes the following menu to be displayed.

Current plots available

1. Error plots
2. Trajectory plots
3. Measurement residual plots
8. DIPTEK (display previously created DIP file)
9. Exit (return to system '\$' level)

Enter number of plot desired:

Input file name:

When the user responds to the first prompt with a 1, 2, 3 or 8 he is asked for a file name. The NAVPLT plotting capability reads the output files produced by NAVSIM II for data to be plotted. Error plots use the -.ER1 and -.SG1 files, trajectory plots use the -.TRJ files and measurement residual plots display data from the -.RES files. DIP files are device-independent-plotting files which are always created for hard-copy display whenever the user responds with a 1, 2 or 3. The DIP files are filed as new versions of:

ERRPLT.DIP	for error plots
TRJYPLT.DIP	for trajectory plots
RESPLT.DIP	for residual plots.

These may be re-displayed later at the terminal using option 8.

When the user has provided a filename following a 1, 2 or 3 response to the initial prompt, he will be asked to furnish scaling information. The user is given the opportunity to override the automatically scaled values for abscissa and ordinate scales. He is prompted with the default values. A simple carriage return will retain those values. An entered value will override the default.

Error plots (Figures 5 and 6) display two plot-pages of estimation errors and their uncertainty envelopes. The first page depicts position errors versus time and the second depicts velocity errors versus time. The major plot title is taken from the first 40 characters of the run title. The uncertainties come from the -.SG1 file while the errors are taken from the -.ER1 file.

Trajectory plots (Figure 7) display the horizontal and vertical projections of the actual (true) trajectory. The vertical plot's abscissa is distance along the horizontal path (rather than x or y). The scales for these plots are fixed, but may be changed by a procedure described in Appendix C.

Measurement residuals (Figure 8) are displayed versus time as discrete points, rather than being connected by lines. As with the position and velocity error plots, the user is given the opportunity to define scales to override the data-based automatic scales. In the case of GPS range measurements, the range residuals for the different satellites are each displayed with different plotting symbols.

GPS ONCE/SEC WITH EPHEM ERRORS POSITION ERRORS

TIME IN SECONDS

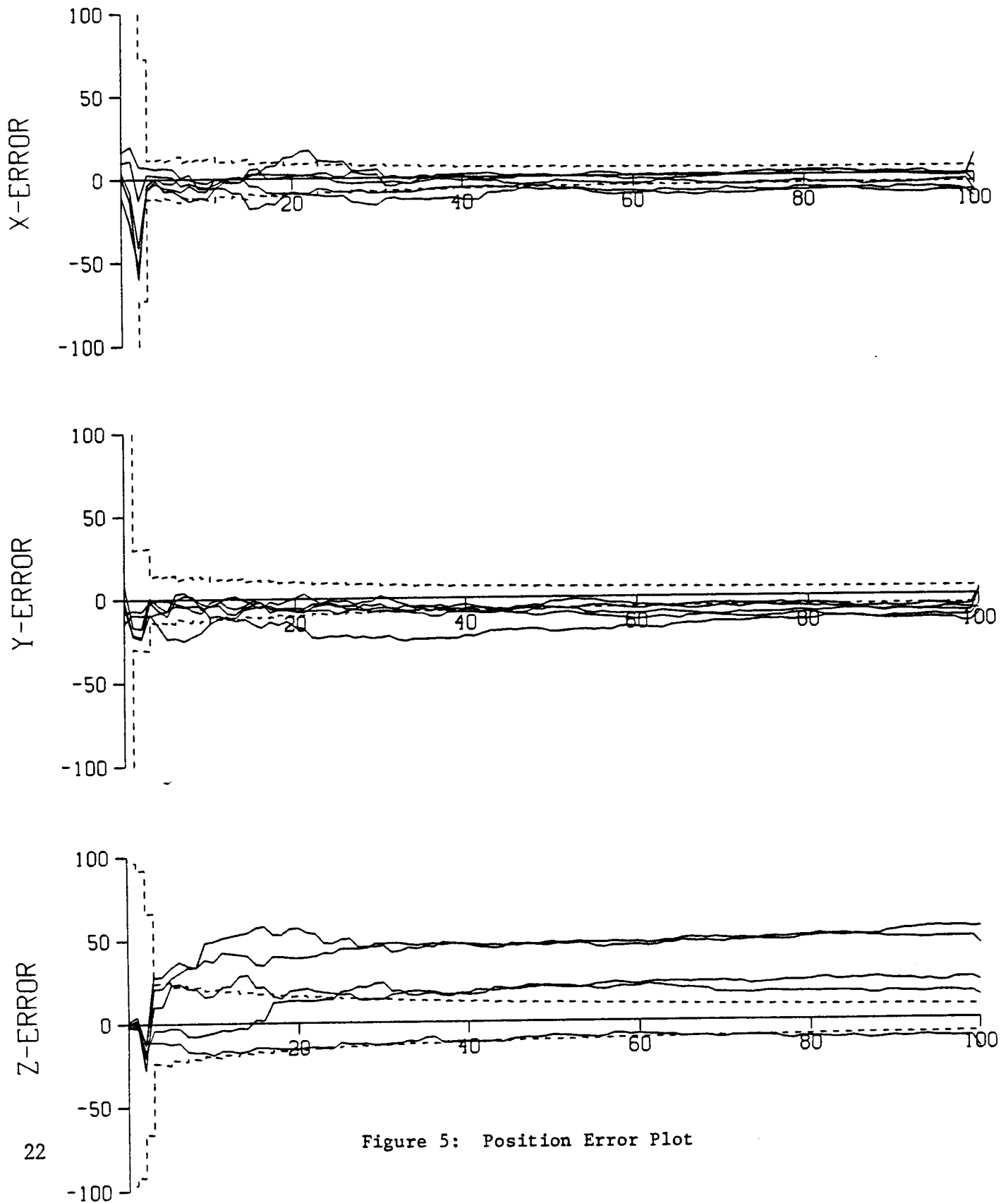


Figure 5: Position Error Plot

GPS ONCE/SEC WITH EPHEM ERRORS VELOCITY ERRORS

TIME IN SECONDS

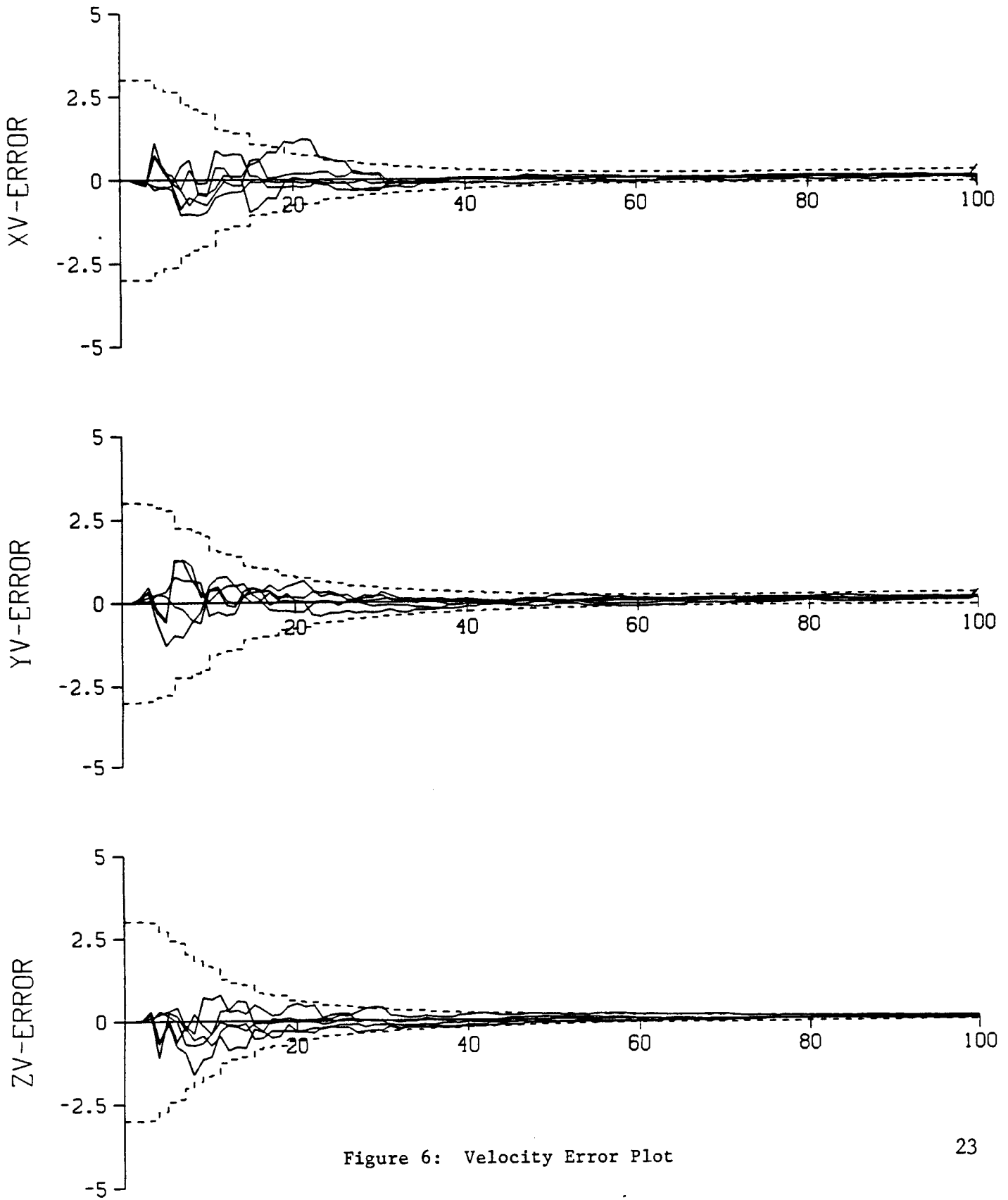
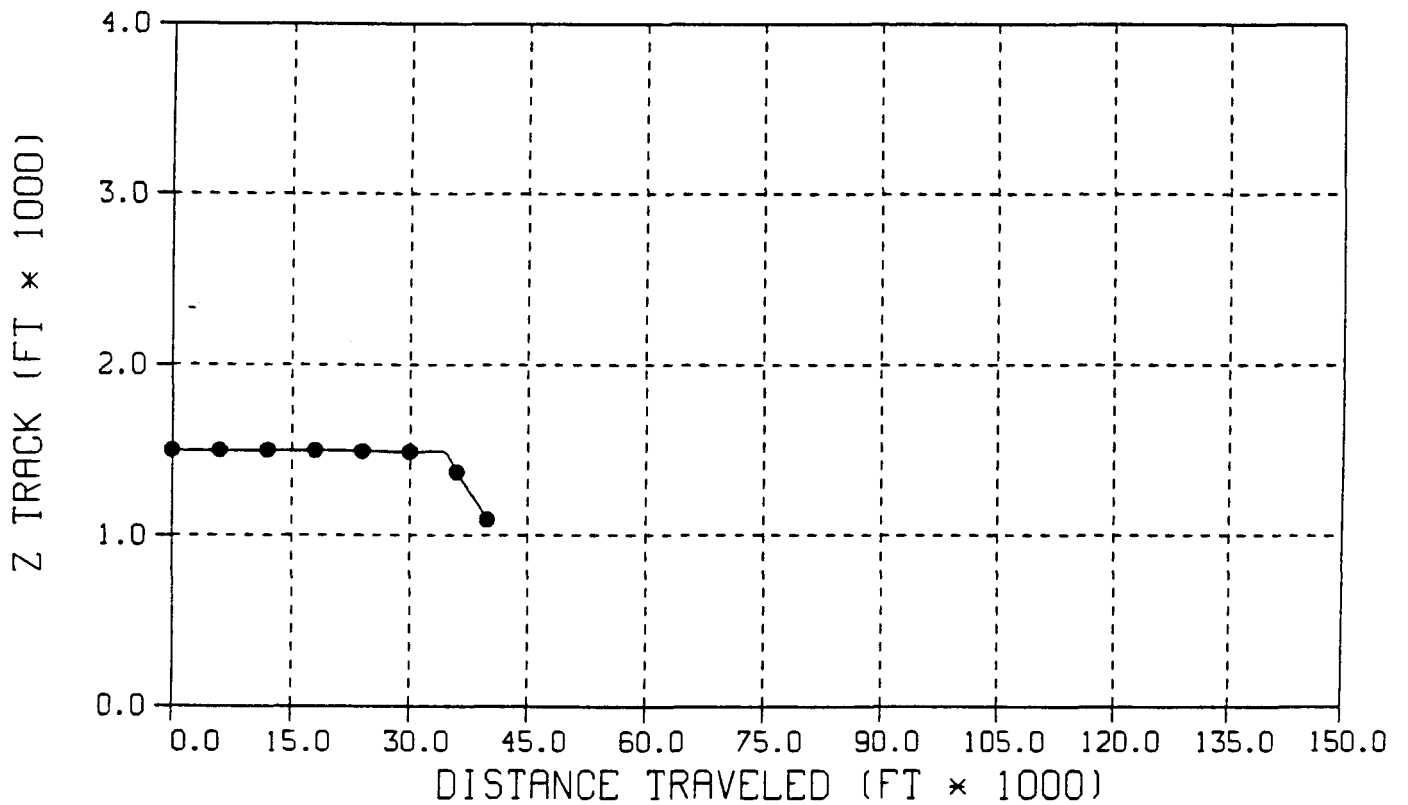
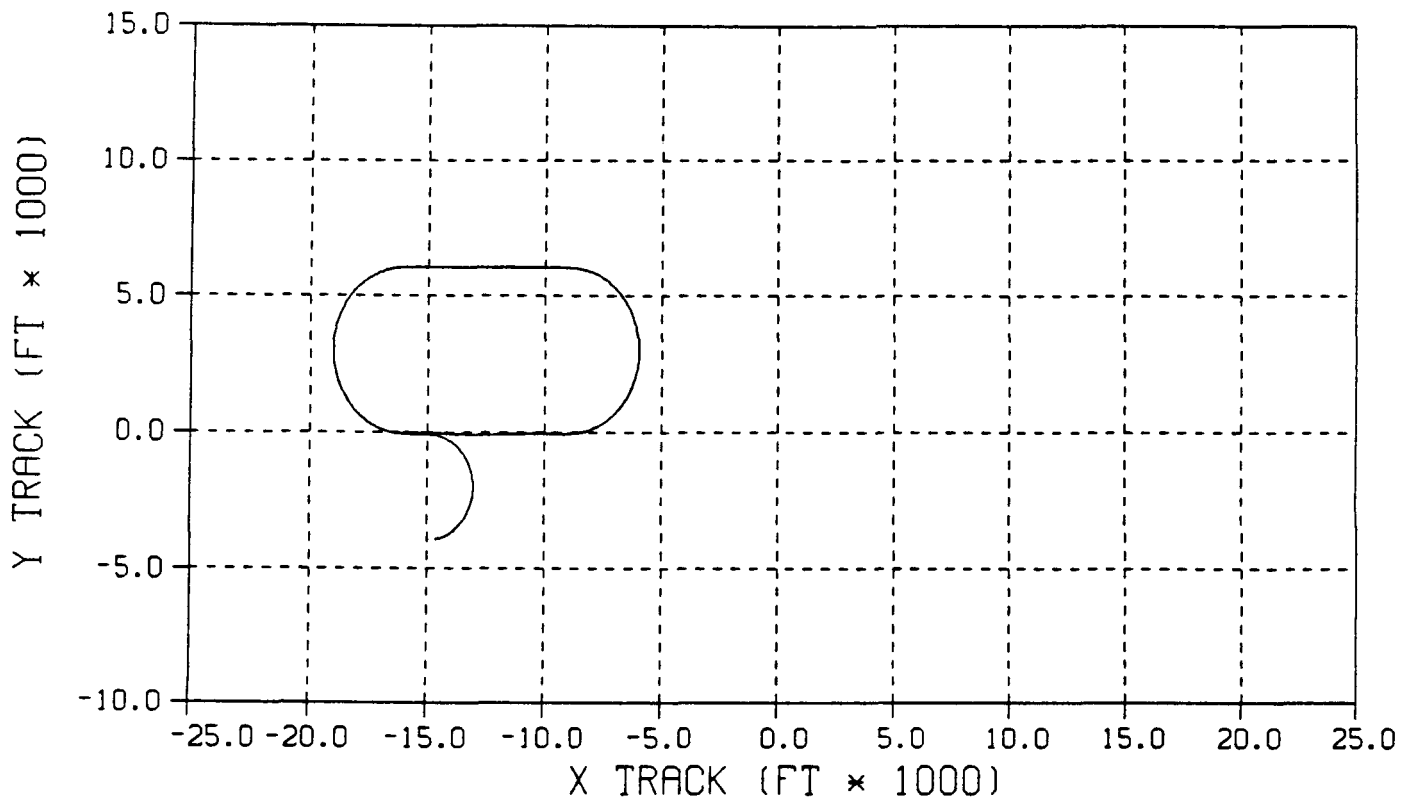


Figure 6: Velocity Error Plot

GPS ONCE/SEC WITH EPHEM ERRORS

TRAJECTORY PLOTS



GPS ONCE/SEC WITH EPHEM ERRORS

MEASUREMENT RESIDUALS - GPS

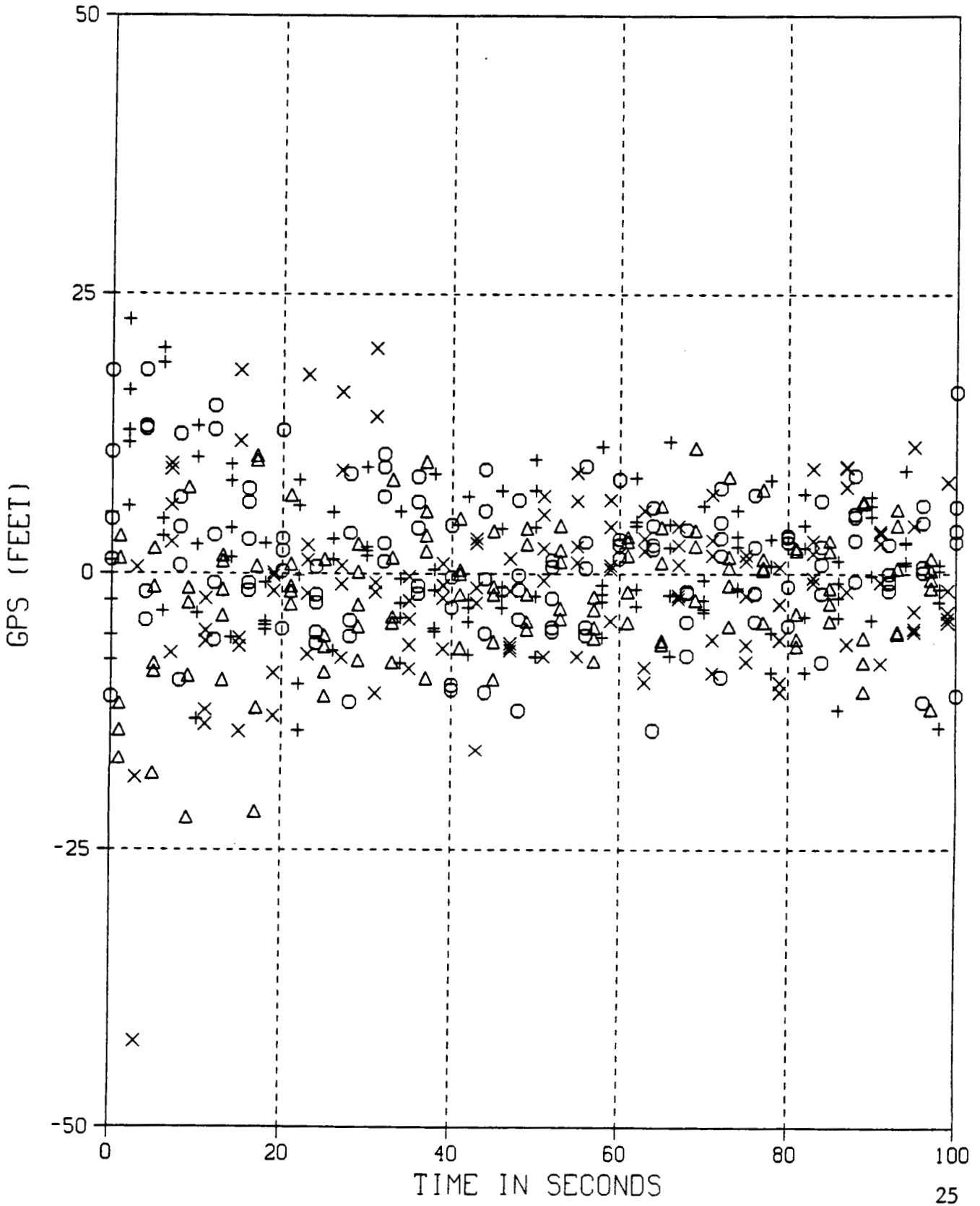


Figure 8: Measurement Residual Plot

Hard-copy plots

Hard-copy plots may be obtained from the DIP files (ERRPLT.DIP, TRJYPLT.DIP, RESPLT.DIP) and displayed with the help of system utilities:

DIPQMS for laser-printer plots
DIPVER for Versatec plots.

To use DIPQMS, the user responds to the system (\$) prompt with

DIPQMS filename

where filename=ERRPLT.DIP, for example. In this case, a file named ERRPLT.QMS will be produced by DIPQMS. This file may be printed by entering:

PRINT/QUEUE=LASER/SETUP=PORTRAIT ERRPLT.QMS

If the user wishes Versatec plots, he first types:

DIPVER filename

to produce the default files PARM.PLV and VECTR1.PLV. He then types

PLOT

and is prompted for:

Prompt	Response
-----	-----
Node:	SAT
Type:	VERSA
Parameter file:	return (defaults to PARM.PLV)
Vector file:	return (defaults to VECTR1.PLV)
Type:	return (ends this dialogue)

The Versatec plots will then be printed at the specified Decnet node, Saturn Vax in this example.

Section IV PROGRAM DESCRIPTION

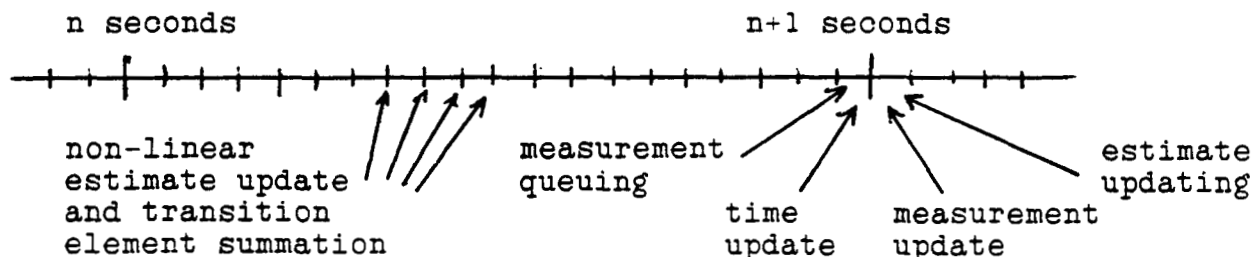
Philosophy of design

The design philosophy of NAVSIM II was to try to build a core navigation simulation program which could be easily used, modified, transported and understood. Toward this end, great care was taken in program design to decouple the various program functions, to modularize logically and to minimize the user's workload. Care was also taken in implementation to write source code which was easy to read so that modification and debug time would be minimized. Program efficiency was considered to be of only secondary importance.

Structure

Figure 9 is a combination roadmap and high-level structure chart of NAVSIM II. It reads left-to-right and top-to-bottom and shows both "who calls whom" and in which order. It shows that the executive routine calls (in order), READINP, INEREF and SEESAT, then calls MONTEC and TAF COS for each Monte Carlo pass and finally calls OUTCLSE to close the output files. The group of subroutines emanating from MONTEC is executed only once per Monte Carlo pass. FILSIM is called by TAF COS every integration step (every 1/20 second) and drives the aided-navigation simulation logic. The navigation logic (integration of the equations of motion) is carried out 20 times per second, while the filter's cycle time is once per second. Thus NAVEQ and AUPT are called 20 times per second while the remaining subordinate routines of FILSIM are called only once per second.

Subroutine MEASIM implements the measurement schedule, calling each requested measurement simulation module at the scheduled times. The measurement inputs to the filter -- residuals, sensitivities and weights -- are queued up for sequential processing by the filter logic (MEASUP, UDMES). The filter's three significant operations are time-updating of the U-D matrix (in TIMEUP), measurement updating (in MEASUP) and estimate-updating (in CHNGX). The remaining functions are supportive of these three operations. The following time-line sketch shows the operational sequence.



ORIGINAL PAGE IS
OF POOR QUALITY

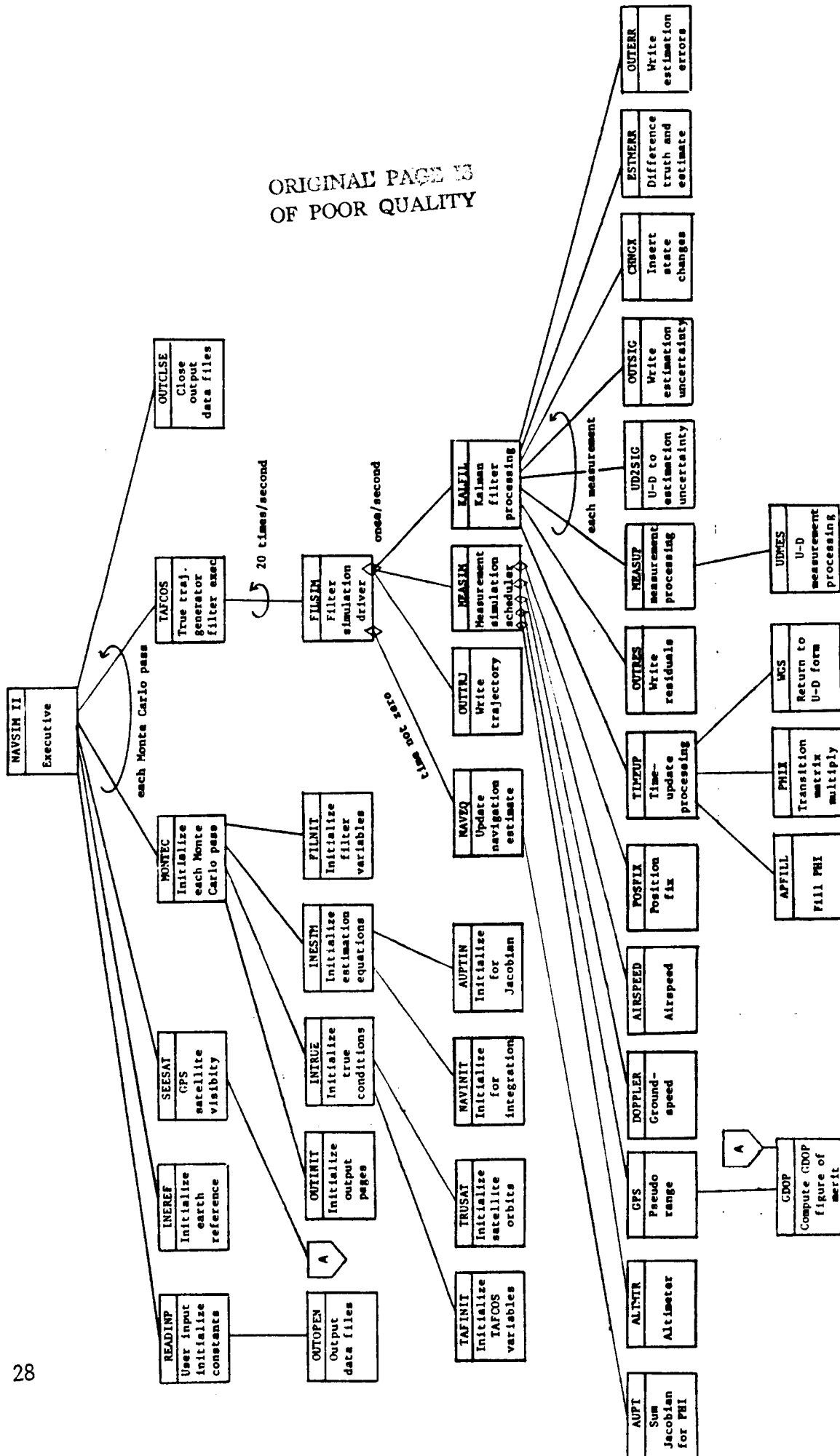


Figure 9: High-level Structure of NAVSIM II

Data Flow

Figure 10 is a high-level data flow diagram of NAVSIM II. A more detailed data flow picture requires an item-by-item and subroutine-by-subroutine exposition. This detail can be found by examination of the commons descriptions found later in this section. Figure 10 shows the major program groups and the flow of data among them. The once-per-run initialization routines read the input files and initialize earth-reference variables and satellite orbits. The once-per-pass routines re-establish initial conditions for the actual trajectory and for the estimated trajectory. These are passed to the integration routines, TAF COS and NAVEQ, along with the actual and estimated IMU errors. The error state covariance is also reset to its initial (input) value and passed to the Kalman filter routines. Initial measurement biases are also established by random sampling for use by the measurement simulation routines. The integration routines supply the actual and estimated position and velocity to the measurement routines and compute the state transition matrix elements for use by the Kalman filter's time-update routines. In return, updates to the estimated state are provided to NAVEQ by the Kalman filter routines (esp. CHNGX). The measurement simulation routines supply measurement residuals, sensitivities and filter weights to the Kalman filter routines where they are processed to produce changes in the estimates of navigation state (returned to the integration routines) and measurement biases (returned to the measurement simulation routines). These are the major data flows of NAVSIM II. Data flows which are strictly for output purposes are not shown.

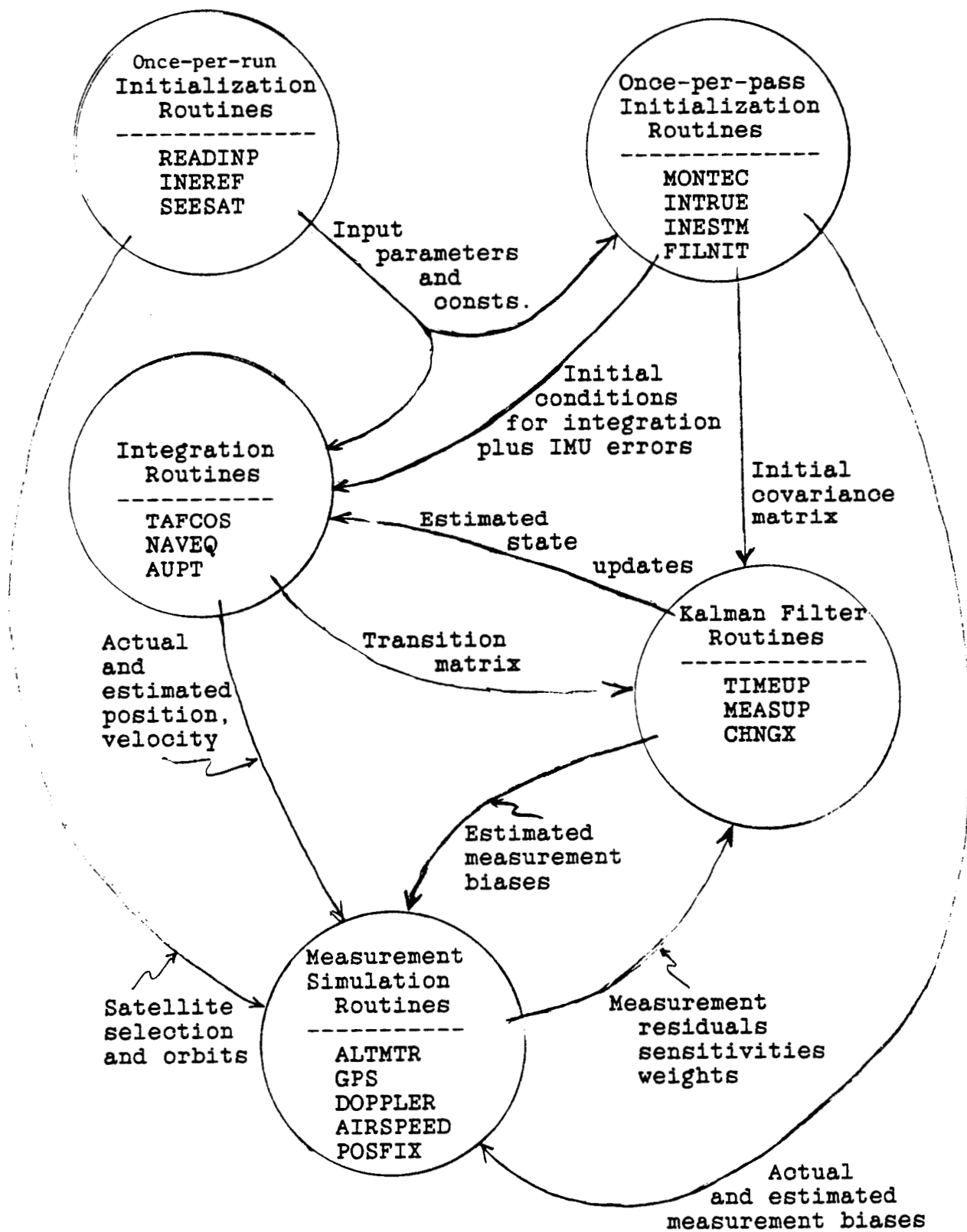


Figure 10: Data Flow of NAVSIM II

Subroutines

The subroutines which comprise NAVSIM II are listed in Table 1 with a very brief description of each. A more-detailed summary of each of the subroutines is given in Appendix D. The summary lists parameters set and used by the routine as well as names of subroutines the routine-of-interest references and the other NAVSIM II routines which reference it. No mathematical formulations or algorithms are described; these can be found in Appendix A, in the references and in the Fortran source code.

Commons

Most inter-module communication in NAVSIM II is by means of labeled commons (as opposed to call-lists). These commons have been structured from logical consideration of their contents. Table 2 lists the commons and gives a very brief description of each. A description of each common will be found in Appendix E. In addition to a definition of each parameter in the common, the description contains a set/used table by parameter and by accessing module.

Files

NAVSIM II reads two input files and writes seven output files in addition to the system output file. The input files are read in by modules READINP and TAF COS. Output files are written primarily in module OUTPUT. All input and output files are in ASCII format. Table 3 lists these files.

Table 1
Subroutines/Modules

Module -----	Function -----
NAVSIM (main)	Driver or executive for NAVSIM II
AIRSPEED	Formulates and queues up residuals, sensitivities and weight for airspeed measurements
ALTMTR	Formulates and queues up residuals, sensitivities and weights for barometric and radio altimeter measurements
APFILL	Fills transition matrix elements for time-update
AUPT	Computes and sums state-rate sensitivities to state variations for transition matrix computation
BARN1	Generates pseudo-random numbers
CHNGX	Incorporates state-change estimates into the estimated state for the extended filter
CNVLIN	Converts ASCII-string numbers into internal floating-point numbers
DOPPLER	Formulates and queues up residuals, sensitivities and weights for Doppler radar measurements
ESTMERR	Computes the state estimation error
FILNIT	Initializes the Kalman filter parameters
FILSIM	Drives the simulation of the airborne Kalman filter
GDOP	Computes GDOP for four satellites
GPS	Formulates and queues up residuals, sensitivities and weights for GPS measurements
INEREF	Initializes equator-to-runway transformation, earth-rate vector and radius of curvature
INESTM	Initializes the estimated state for each Monte Carlo pass
INTRUE	Initializes the simulated actual state for each Monte Carlo pass
KALFIL	Controls high-level Kalman filter sequencing logic

MEASIM	Controls measurement simulation according to input schedule
MEASUP	Controls processing of one measurement by the Kalman filter
MONTEC	Controls initialization process at the start of each Monte Carlo pass
NAVEQ	Integrates navigational equations for one time-step
OUTPUT	Performs most of the file-output tasks for NAVSIM II
PHIX	Performs linearized time-update (multiplies vector by the state transition matrix)
POSFIX	Formulates and queues up residuals, sensitivities and weights for position-fix measurements
READINP	Reads, interprets and distributes data from the navigation-parameter input file
SEESAT	Determines GPS coverage (satellite visibility) and optimal GDOP
TAFCOS	Generates helicopter's "actual" trajectory, forces, body rates and attitude
TIMEUP	Controls time-updating of the covariance matrix and incremental state for the Kalman filter
TRIMAT	Displays the U-D covariance matrix in upper-triangular format
TRUSAT	Initializes the "actual" GPS satellite ephemerides and errors
UDMES	Updates the U-D matrix and incremental-state estimate for one measurement
UD2SIG	Computes standard deviations (state uncertainties) from a U-D matrix
WGS	Reduces an input rectangular matrix to U-D (upper-triangular) form

Table 2. List of Commons

ACTUAL	- Simulated truths in sensor models
AUPCOM	- Time-constants for inertial-sensor process noise
CONST	- Global constants (radians-to-degrees, etc.)
DELCOM	- Incremental state estimates
EROTIC	- Earth's rotational rate and radius vectors
ERRCOM	- Error state initial condition and sensor errors
GRVCOM	- Estimated gravity vector
KALM	- Kalman filter covariance parameters
KALMIN	- Kalman filter input parameters
MESCOM	- Measurement processing parameters
MESKED	- Measurement scheduling parameters
MESWTS	- Measurement weights and errors
ORBITS	- Orbital data for GPS satellites
OUTCOM	- Output file descriptors
PRTCOM	- Extra-output print option flags
REFCOM	- Earth-referenced location/frame variables
STATCM	- Current estimates of state
TIMCOM	- Time indicators
TIMEIN	- Case-control times
TRANS	- Transition-matrix parameters
TRUORB	- True GPS orbital parameters
TRUTHC	- simulated true aircraft state
VISCOM	- GPS visibility parameters

Table 3. Input/output Files of NAVSIM II

	Logical Unit	File	Contents/purpose
Input	1	TRJXXX.INP	reference trajectory specification
	4	navxxx.INP	navigation parameter inputs
	5	System inp.	Input filenames from user
Output	6	System out.	miscellaneous & debug print
	7	navxxx.TRJ	tabulated trajectory coords.
	8	navxxx.RES	measurement residuals
	9	navxxx.ER1	estimation errors, part 1
	10	navxxx.ER2	estimation errors, part 2
	11	navxxx.SG1	estimation uncertainty, part 1
	12	navxxx.SG2	estimation uncertainty, part 2
13	navxxx.OIN	echo of the inputs (nav + trj)	

Section V RECOMMENDATIONS

NAVSIM II is a tool which can aid researchers in understanding and predicting the effects of various system parameters and errors on performance of an aided-inertial navigation system for helicopter flight. As any tool, it must be used to be of any value. Its use will undoubtedly lead to its modification as enhancements and extensions to its original capabilities are seen to be needed. It was with the likelihood of modification in mind that NAVSIM II was designed with attention to structure.

Enhancements

A number of useful enhancements to NAVSIM II can be suggested, even at the program's debut. But for time limitations, many of these would already be included among NAVSIM II's capabilities and features. The order of the following suggested enhancements does not imply importance or priority.

1. Save and re-start from a final covariance matrix. This would enable shorter runs which concentrate on the terminal or approach phase while representing correlated uncertainties from take-off and cruise phases. The saved covariance matrix could also be used as the source of Monte Carlo initial condition errors for the terminal phase.
2. Enable variation of the filter's cycle time and estimation-update time. This would allow longer Kalman cycles, thereby reducing the computational load and enabling investigation into corresponding dependency on assumptions of linearity. Delay in updating the estimate with estimated state increments is common in airborne implementations. Such delay may tax linearity assumptions and usually generates special requirements in calculation of partials and residuals. Increasing the filter's cycle time will generally require an increase in the size of the measurement-processing queue.
3. Permit a skip-rate for print-files. This would reduce the size of the output files produced by NAVSIM II at the expense of detail and some additional complexity in the plotting software.
4. Provide flexible plotting to select data from arbitrary columns of output from several specified runs. This capability would permit display of the

effects of parametric differences -- not just Monte Carlo variation. This enhancement could be extended to include specified mathematical operations on the output columns of numbers before plotting -- thus enabling such displays as translated and rotated coordinates, mean-square errors, metric units.

5. Provide the square-root and conventional P-matrix implementations as options to the U-D filter formulation. This should make no difference in the results, but should help to satisfy any questions of the method.
6. Provide more flexibility in specifying filter weights and process noise. This flexibility would permit examination of ad hoc methods for tuning the filter, such as scheduled variations in the level of process noise (random forcing functions). This capability, along with more flexibility in scheduling of the various measurements, would enhance the simulations of INS alignment and pseudo-differential GPS.
7. Provide an option for simulation of a stable-platform type of INS.
8. Enable Runge-Kutta integration of the navigation equations. This would enable the use of larger integration steps and might reduce run-time.
9. Provide an interactive program for producing and/or editing the navigation-parameter input file through prompts.
10. Improve the means of specifying the reference trajectory. The utility provided with NAVSIM II is helpful in this regard, but is still hard to use.
11. Extend the measurement set to include GPS delta-range (range rate).

Extensions

Several logical extensions to NAVSIM II are envisioned. Extensions differ from enhancements in that they make a significant change in the purpose or use of the program rather than simply enhancing its current function.

1. Extension to guidance. This extension would feed navigation errors back into the guidance calculations (TAF COS), thereby potentially simulating guidance errors which include navigation errors. The resulting program would be primarily a guidance research tool.
2. Flight test analysis. This extension would use flight-recorded data in a filter-smoother implementation to examine the effect of specifiable parameters on the resultant navigational estimate. This extension would require a major modification of the program.

Section VI

REFERENCES

- [1] Bierman, G. J., Factorization Methods for Discrete Sequential Estimation, Academic Press, New York (1977)
- [2] Meyer, G., Hunt, R. L., Su, R., "Design of a Helicopter Autopilot by Means of Linearizing Transformations," Agard Conference Proceedings No. 321, Advances in Guidance and Control Systems (October 1982)
- [3] Schmidt, S. F., Bjorkman, W. S. and Conrad, B., "New Mechanization Equations for Aided Inertial Navigation Systems," NASA CR-2352 (December 1973)
- [4] Bierman, G. J. and Nead, M. W., "A Parameter Estimation Subroutine Package," JPL Publication 77-26 (July 1977)
- [5] Denaro, R. P. and Cabak, A. R., "Simulation and Analysis of Differential Global Positioning System for Civil Helicopter Operations," NASA CR-166534 (December 1983)
- [6] Erzberger, H., "Automation of On-Board Flightpath Management," NASA Technical Memorandum 84212 (December 1981)

PRECEDING PAGE BLANK NOT FILMED

Appendix A
MATHEMATICAL BASIS

Equations of Motion and Their Errors

The equations of motion (i.e., navigational equations) of the aircraft are (from reference [3]):

$$\dot{\mathbf{v}} = \mathbf{f} + \mathbf{g} - (2\boldsymbol{\omega}_e + \boldsymbol{\omega}) \times \mathbf{v} \quad (\text{A.1})$$

$$\dot{\mathbf{x}} = \mathbf{v} \quad (\text{A.2})$$

Where

\mathbf{f} is the specific force vector.

\mathbf{g} is the gravity (reaction) vector

$\boldsymbol{\omega}_e$ is the earth-rate vector

$\boldsymbol{\omega}$ is the rotation rate of the computational reference frame with respect to the earth

\mathbf{v} is the velocity vector relative to the reference frame's origin

\mathbf{x} is the position vector relative to the reference frame's origin

The computational reference frame is runway - referenced, centered at some point on the runway with x-axis positive in the normal takeoff and landing direction, y-axis to the right and z positive downward.

The vectors must be transformed to common reference frames before the equations of motion can be formed and integrated. Let a lower-left subscript denote the reference frame in which a vector is coordinatized. For example,

${}_b\mathbf{f}$ is the specific force vector expressed in body-frame components.

${}_r\mathbf{f}$ is the specific force expressed in runway-frame components.

The transformation from body to runway coordinates is denoted ${}_r\mathbf{T}^b$.

Thus ${}_r\mathbf{f} = {}_r\mathbf{T}^b {}_b\mathbf{f}$

Further, let the "true" values of a variable be denoted with a bar over the variable's symbol and let the "estimated" value be denoted with a caret.

\bar{f} true specific force

\hat{g} estimated gravity

The implementation of equations A.1 and A.2 (in subroutine NAVEQ) is as follows.

1. Compute measured specific force

$$b_{\text{meas}}^f = \bar{b}^f \cdot \bar{s}_f + \bar{b}^f \text{bias} + \bar{b}^f \text{noise} \quad (\text{A.3})$$

2. Compute corrected specific force in runway coordinates

$$r_{\text{corr}}^f = r^{\hat{b}} b_{\text{meas}}^f - \begin{pmatrix} 0 \\ 0 \\ \hat{f}_b \end{pmatrix} \quad (\text{A.4})$$

3. Compute transformation from navigation-frame (tilted) to runway frame, using estimated tilts, $\hat{\beta}$

$$r^{\text{T}^n} = \mathcal{R}(\hat{\beta}) \cong I - (\hat{\beta} \times) \quad (\text{A.5})$$

4. Compute acceleration in the navigation frame by adding gravitational and coriolis contributions to the corrected specific force

$$n^a = n^{\text{T}^r} r_{\text{corr}}^f + n^{\hat{g}} + n^{\hat{c}} \quad (\text{A.6})$$

5. Integrate acceleration to render velocity

$$n^v(t) = n^v(t-\delta) + n^a \cdot \delta \quad (\delta = \text{integration step}) \quad (\text{A.7})$$

6. Compute average velocity at mid-integration-step

$$n^v_{\text{avg}} = (n^v(t) + n^v(t-\delta))/2 \quad (\text{A.8})$$

7. Integrate velocity to render position

$${}_{\mathbf{r}}\mathbf{x}(t) = {}_{\mathbf{r}}\mathbf{x}(t-\delta) + {}_{\mathbf{r}}\mathbf{T}^{\mathbf{n}} \mathbf{v}_{\text{avg}} \cdot \delta \quad (\text{A.9})$$

8. Compute measured body rate vector

$${}_{\mathbf{b}}\omega_{\text{meas}} = {}_{\mathbf{b}}\omega \cdot \bar{s}_{\omega} + {}_{\mathbf{b}}\bar{\omega}_{\text{bias}} + {}_{\mathbf{b}}\bar{\omega}_{\text{noise}} \quad (\text{A.10})$$

9. Compute corrected body rate vector

$${}_{\mathbf{b}}\hat{\omega}(t) = {}_{\mathbf{b}}\omega_{\text{meas}} - {}_{\mathbf{b}}\hat{\omega}_{\text{bias}} \quad (\text{A.11})$$

10. Compute average rotation vector

$$\theta = .5 \cdot \left({}_{\mathbf{b}}\hat{\omega}(t) + {}_{\mathbf{b}}\hat{\omega}(t-\delta) \right) \cdot \delta \quad (\text{A.12})$$

11. Update the estimated body-to-runway transformation

$${}_{\mathbf{b}}\mathbf{T}^{\mathbf{r}}(t+\delta) = \mathcal{R}(\theta) {}_{\mathbf{b}}\hat{\mathbf{T}}^{\mathbf{r}}(t) \quad (\text{A.13})$$

12. Update gravitational acceleration

$${}_{\mathbf{n}}\hat{\mathbf{g}}(t+\delta) = \mathbf{g}_o (1+2x^3/r_e) {}_{\mathbf{n}}\mathbf{T}^{\mathbf{r}} \begin{bmatrix} -x^1/r_e \\ -x^2/r_e \\ \sqrt{1 - (x^1/r_e)^2 - (x^2/r_e)^2} \end{bmatrix} \quad (\text{A.14})$$

13. Update coriolis acceleration

$${}_{\mathbf{n}}\hat{\mathbf{c}}(t+\delta) = {}_{\mathbf{n}}\hat{\mathbf{v}} \times \left\{ 2 {}_{\mathbf{n}}\mathbf{T}^{\mathbf{r}} \mathbf{r} \omega_e + \hat{\beta} \times \mathbf{r} \omega_e + {}_{\mathbf{r}}\hat{\mathbf{T}}^{\mathbf{b}} ({}_{\mathbf{b}}\bar{\omega}_{\text{bias}} - {}_{\mathbf{b}}\hat{\omega}_{\text{bias}}) \right\} \quad (\text{A.15})$$

14. Update the drift-rate estimate

$$\hat{\omega}_d = -r_e^\omega \times r^\beta + r^{T^b} (\bar{b}_{bias} - \hat{b}_{bias}) \quad (A.16)$$

15. Update the tilt estimate

$$\hat{\beta}(t+\delta) = \hat{\beta}(t) + (r_d^\omega \times \hat{\beta}/2 + \hat{\omega}_d) \cdot \delta \quad (A.17)$$

Estimation errors propagate through the equations of motion because these non-linear differential equations are functionally dependent on the errors in the estimate. Additional errors develop in the estimate because some IMU errors are included in the error model ("true" errors) but are not included in the estimated state.

Variational Equations

The error covariance matrix is time-updated linearly, using the state transition matrix and forcing function sensitivity matrix in the update. These propagation matrices are derived from solutions to appropriate variational equations.

The variational equations for NAVSIM II are developed in Reference [3]. They are conveniently summarized using vector-matrix symbology. If we let z represent the error vector, its time-variation is approximately by

$$\dot{z} = F_z z + F_n n \quad (\text{A.18})$$

where n is the process-noise vector and where F_z and F_n are matrices.

$$F_z = \frac{\partial \dot{z}}{\partial z}, \quad F_n = \frac{\partial \dot{z}}{\partial n}$$

Equation (A.2.1) has the solution

$$z(t) = \phi(t) z(0) + \phi_n(t) n(0) \quad (\text{A.19})$$

where ϕ is the state transition matrix and ϕ_n is the forcing-function sensitivity matrix. These obey the following differential equations.

$$\dot{\phi} = F_z \phi \quad \text{with} \quad \phi(0) = I \quad (\text{A.20})$$

$$\dot{\phi}_n = F_z \phi_n + F_n \quad \text{with} \quad \phi_n(0) = 0 \quad (\text{A.21})$$

These differential equations are not solved explicitly in NAVSIM II. Alternatively, the assumption is made that

$$\Phi \approx I + \sum_{i=1}^{20} F_z(\hat{x}(i\delta)) \cdot \delta$$

$$\Phi_n \approx F_n(\Delta) \cdot \Delta$$

where δ is the integration step and Δ is the filter-update interval. The product Φ_z of equation A.2.2 is formed in subroutine PHIX, taking advantage of the known sparseness of Φ in formation of the product.

The error state vector, z , for NAVSIM II has 21 elements:

- | | | |
|--------|---------------------|--|
| 1-3. | \tilde{x} | position error vector, feet, runway coordinates |
| 4-6. | \tilde{v}_n | velocity error vector, ft/sec, navigation-frame coordinates |
| 7-9. | $\tilde{\beta}$ | tilt error vector, radians, runway coordinates |
| 10-12. | $\tilde{\omega}_d$ | drift rate error vector, rad/sec, runway coordinates |
| 13. | \tilde{f}_b | vertical acceleration bias, ft/sec/sec, runway coordinates |
| 14. | \tilde{c}_b | GPS clock bias error, nanosec |
| 15. | $\tilde{\dot{c}}_b$ | GPS clock bias rate error, nanosec/second |
| 16-18. | $\tilde{\alpha}_d$ | Doppler radar antenna pointing error vector, degrees |
| 19. | $\tilde{s}_{a/s}$ | Airspeed scale factor error, dimensionless |
| 20-21. | \tilde{w}_v | wind speed error vector, ft/sec, runway coordinates (level only) |

The partial derivative, F_z , of \dot{z} with respect to z is developed in reference [3]. It can be written as a matrix.

$$F_z = \begin{array}{|c|} \hline 0 & F_{xv} & F_{x\beta} & 0 & 0 & 0 \\ \hline F_{vx} & F_{vv} & F_{v\beta} & F_{v\omega} & F_{vf} & 0 \\ \hline 0 & 0 & F_{\beta\beta} & F_{\beta\omega} & 0 & 0 \\ \hline 0 & 0 & 0 & F_{\omega\omega} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & F_{ff} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & D \\ \hline \end{array}$$

The non-zero elements of F_z are all 3 x 3 matrices except for F_{ff} (a scalar), F_{vf} (3 x 1) and D (8 x 8). They are implemented in subroutine AUPT.

$$F_{xv} = I \quad (3 \times 3 \text{ identity matrix})$$

$$F_{x\beta} = \hat{n} \times \hat{v} \quad (\text{cross-product matrix})$$

$$F_{vx} = \frac{\partial \dot{v}}{\partial x} = \text{function of } \hat{g} \text{ and } \hat{x} \quad (\text{see ref. [3]})$$

$$F_{vv} = -2 \hat{r} \omega \times \quad (\text{cross-product matrix})$$

$$F_{v\beta} = -\hat{r}_g x - (\hat{n}_v x) (\hat{r}_e^\omega x)$$

$$F_{v\omega} = \hat{n}_v x \hat{n}_T^b$$

$$F_{vf} = (0 \ 0 \ 1)$$

$$F_{\beta\beta} = -\hat{c}_e^\omega x$$

$$F_{\beta\omega} = \hat{n}_T^b$$

$$F_{\omega\omega} = -\frac{1}{\tau_\omega} I$$

$$F_{ff} = -\frac{1}{\tau_f}$$

$D = I_g +$ off-diagonal term for clock bias wrt clock bias rate

Filter Equations

The conventional Kalman filter equations are:

$$\text{Time-update: } P_+ = \Phi P_- \Phi^T + \Phi_n W \Phi_n^T \quad (\text{A.22})$$

$$\text{Measurement update: } \hat{z}_+ = \hat{z}_- + K (y - H\hat{z}_-) \quad (\text{A.23})$$

$$\text{with } K = P_+ H^T (H P_+ H^T + R)^{-1}$$

$$\text{and } P_{++} = (I - KH) P_+ \quad (\text{A.24})$$

In these equations,

Φ is the state transition matrix

Φ_n is the process-noise sensitivity matrix

P is the error state covariance matrix

W is the process noise covariance matrix

\hat{z} is the estimated error state

y is the measurement residual value

H is the sensitivity (vector) of measurement to state variation

R is the covariance of measurement uncertainty.

At time-updates, P is updated (from P_- to P_+) by equation A.22. At each measurement update, the error state estimate is updated (from \hat{z}_- to \hat{z}_+) by equation A.23 and P is modified (from P_+ to P_{++}) by equation A.24. Then P_{++} is used in place of P_+ for the next measurement and P_+ becomes P_- for the next time update.

In the U-D formulation, U is an upper-triangular matrix and D is a diagonal matrix such that at any time

$$P = UDU^t$$

The details of the time and measurement update processes are to be found in reference [1].

Measurement Models and Errors

The measurements modeled in NAVSIM II are

1. baro-altitude (ALTMTR)
2. radio altitude (ALTMTR)
3. GPS pseudo-range (GPS)
4. Doppler radar velocity (DOPPLER)
5. airspeed (AIRSPEED)
6. position fix (POSFIX)

The filter sees each of these measurements generically as:

1. a measurement residual, y
2. a measurement sensitivity vector, H , and
3. a measurement uncertainty (weight), r

In the following measurement-model summaries, x is a three-component position vector, v is a three-component velocity-vector. Barred variables (\bar{x}) represent "true" values, while caret variables (\hat{x}) represent estimated values. The letter b will denote "bias" and n will denote "noise" (gaussian).

Altimeter Measurement

This measurement is height above mean sea level for the barometric altimeter and height above the local terrain (defaulted to be sea level) for the radio altimeter.

$h = -x(3)$ for baro-altimeter

$h = x(3) - h_{\text{terrain}}$ for radio-altimeter

residual, $y = (\bar{h} + \bar{b} + \bar{n}) - \hat{h}$

sensitivity, $H = (0, 0, -1, 0, \dots, 0)$

weight, $r = \text{input value}$

GPS Measurement

This measurement is slant range distance from the aircraft's position to the observed satellite's concurrent position. The satellite-to-be-observed varies at each observation, with the choice cycling among a set of four satellites which optimize the GDOP criterion. Let R_s represent the observed satellite's position in the runway coordinate frame. Then

$$\text{slant range, } s = |R_s - x| \equiv |S|$$

$$\text{residual, } y = (\bar{s} + \bar{e} + \bar{c} + \bar{b} + \bar{n}) - (\hat{s} + \hat{c})$$

$$\text{sensitivity, } H = (-\hat{S}(1)/s, -\hat{S}(2)/s, -\hat{S}(3)/s, 0, 0, \dots, c_0, 0, \dots, 0)$$

weight, $r = \text{input value.}$

In the residual equation, \bar{e} is the observed satellite's ephemeris error vector (treated as a bias in the orbital frame and transformed to runway coordinates. The letter c denotes the clock bias of the GPS receiver, treated as a range bias. In the sensitivity equation, the first three terms are runway-coordinate components of the normalized estimated slant-range vector and c_0 is the speed of light (sensitivity of slant-range error to clock bias).

The residual is not implemented as shown above, but rather in a form which does not involve differencing large numbers to render a small result. The implementation shown above is written for better understanding of the essential computation.

Doppler Measurement

These measurements are modeled as the three components of ground velocity along each of the three antenna axes. These axes are assumed to differ from the aircraft's body axes by only the small rotation which represents antenna misalignment. The body-axis-component vector of ground velocity is written in terms of the runway-referenced velocity:

$${}_b \mathbf{v} = {}_b \mathbf{T}^r \mathbf{v}$$

where ${}_b \mathbf{T}^r$ is the transformation from runway to body. The "true" antenna misalignment is assumed to be zero, while the estimated misalignment is in error by the estimate. It follows that the true and estimated antenna-axis components of ground speed are:

$${}_a \bar{\mathbf{v}} = {}_b \bar{\mathbf{T}}^r \bar{\mathbf{v}}$$

$$\hat{\mathbf{v}}_a \approx (\mathbf{I} + \hat{\alpha} \mathbf{x}) {}_b \hat{\mathbf{T}}^r \hat{\mathbf{v}}$$

$$\text{residual for axis } i, y = ({}_a \bar{v}(i) + \bar{b}(i) + \bar{n}(i)) - \hat{v}_a(i)$$

sensitivity for axis i ,

$$\mathbf{H} = (0, 0, 0, t_{i1}, t_{i2}, t_{i3}, 0, \dots, vx_{i1}, vx_{i2}, vx_{i3}, 0, \dots, 0)$$

weight, r = input value

In the preceding sensitivity equation, t_{i1} , t_{i2} and t_{i3} are elements of the i -th row of ${}_b \hat{\mathbf{T}}^r$, the runway-to-body transformation. The vs_{ij} are the i -th row elements of the skew-symmetric "v-cross" matrix:

$$\hat{\mathbf{v}} \mathbf{x} = \begin{bmatrix} 0 & -v(3) & v(2) \\ v(3) & 0 & -v(1) \\ -v(2) & v(1) & 0 \end{bmatrix}$$

This matrix is the sensitivity of the three-measurement set with respect to antenna misalignments.

Airspeed Measurement

This measurement is the speed of the aircraft with respect to the local air mass. Winds are assumed to be horizontal, so the air velocity (vector) is

$$V_{\text{air}} = \begin{bmatrix} v(1) - v_{\text{xwind}} \\ v(2) - v_{\text{ywind}} \\ v(3) \end{bmatrix}$$

and the airspeed is its magnitude,

$$v_{\text{air}} = |V_{\text{air}}|.$$

The measurement is modeled with a scale factor offset, s , as well as simulated bias and noise errors.

$$\text{residual, } y = ((1 + \bar{s}) * \bar{v}_{\text{air}} + \bar{b} + \bar{n}) - (1 + \hat{s}) * \hat{v}_{\text{air}}$$

$$\text{sensitivity, } H = (0, 0, 0, h_4, h_5, h_6, 0, \dots, h_{19}, h_{20}, h_{21})$$

weight, r = input value.

In the sensitivity equation, the velocity sensitivities are

$$h_4 = (1 + \hat{s}) * (v(1) - v_{\text{xwind}}) / \hat{v}_{\text{air}}$$

$$h_5 = (1 + \hat{s}) * (v(2) - v_{\text{ywind}}) / \hat{v}_{\text{air}}$$

$$h_6 = (1 + \hat{s}) * v(3) / \hat{v}_{\text{air}}$$

The sensitivity to airspeed scale factor variation is

$$h_{19} = \hat{v}_{\text{air}}$$

and the sensitivities to wind velocity estimates are

$$h_{20} = -h_4$$

$$h_{21} = -h_5$$

Position-fix Measurement

These measurements are modeled as observations of the current "true" cartesian position components in the runway coordinate system.

$$\text{x-fix residual, } y = (\bar{x}(1) + \bar{b}(1) + \bar{n}(1)) - \hat{x}(1)$$

$$\text{y-fix residual, } y = (\bar{x}(2) + \bar{b}(2) + \bar{n}(2)) - \hat{x}(2)$$

$$\text{x-fix sensitivity, } H = (1, 0, \dots, 0)$$

$$\text{y-fix sensitivity, } H = (0, 1, 0, \dots, 0)$$

$$\text{weight, } r = \text{input value.}$$

Position-fix errors in x and y are treated as uncorrelated in the current NAVSIM II implementation.

Bad Measurement Treatment

NAVSIM II includes a screening feature for excluding "bad" measurements (outliers) from incorporation by the Kalman estimator. This feature is not necessary in a simulation where the "perfect" measurements are corrupted only by biases and white noise, but has been included to simulate a "real" system. The criterion for excluding a measurement is that the absolute value of the residual (i.e., innovation) exceed a user-specified multiple, n, of the current measurement weight:

$$|y| > n * \sqrt{HPH^T + r}$$

In order to exercise this feature, NAVSIM II simulates a specified percentage of "bad" measurements for each measurement type. The percentage is used as a threshold for random sampling from a uniform distribution covering 100%. That is,

if random sample < input percentage, then exclude.

Exclusion is implemented by setting the simulated measurement residual to

$$y = \sqrt{2} * n * r.$$

Appendix B

PROGRAM MODIFICATION

NAVSIM II has been designed, structured and documented under the assumption that it will serve as the basis for extended simulations, and that in that role it will inevitably be modified. This appendix addresses the elements of such modification and points out the parts of this report which can simplify the effort. Two likely cases are considered:

1. a new measurement type is to be added, or
2. a new measurement-bias-type state is to be added.

Any modification to the program involves editing and compiling the appropriate modules, then linking the new or modified object module(s) with the remaining modules to form the new executable program. Each of the subroutines of NAVSIM II is contained in a separate module with the same name as the subroutine (+'.FOR'). Only modules AUPT.FOR, NAVEQ.FOR and OUTPUT.FOR contain multiple entry points. The link-module is a command procedure called LNAVSIM.COM. The current program is linked by

@LNAVSIM

The link-module attaches an object library, UTILITY.OLB, to satisfy several generic references:

DOT	vector dot-product
CROSS	vector cross-product
VNORM	vector normalization
MVTRN	matrix-vector multiplication
GTRN	generate a series of canonical rotations
TWIST	generate a rotation matrix from vector rotation.

If the modification requires a new module, then LNAVSIM.COM must be edited to include mention of the new module. It is suggested that a "working" version of the program be saved (ie., copied or renamed) until the new version is well checked out.

The following discussion references figures, file-conventions and terminology described elsewhere in this report. In addition, it is assumed that the reader has access to Fortran listings of the modules mentioned herein.

Case 1: New Measurement Type

Suppose one wished to add a new measurement (e.g., range to a fixed transponder) to the list of currently available measurements. He would then:

1. formulate the measurement residual and sensitivities in a module called by MEASIM,
2. provide for input of related parameters and measurement schedules in the input (*.INP) file, commons and in subroutine READINP,
3. provide for output of the measurement residual in OUTRES (in module OUTPUT.FOR).

Let us call the new measurement module TRANGE (for "transponder range") for example. The "true" measurement might be modeled as

$$\text{true meas} = (1 + \bar{s}) * \sqrt{(\bar{x} - x_T)^2 + (\bar{y} - y_T)^2 + (\bar{z} - z_T)^2} + \bar{\text{bias}} + \overline{\text{random}}$$

The estimated range might be

$$\text{est. range} = \sqrt{(\hat{x} - x_T)^2 + (\hat{y} - y_T)^2 + (\hat{z} - z_T)^2} + \hat{\text{bias}}$$

The measurement residual would be modeled in TRANGE as

$$\text{residual} = \text{true meas.} - \text{est. range}$$

and queued up for filter-processing in YRES(MINDEX), where MINDEX is the measurement-queue index, incremented on entry into TRANGE. The sensitivity of the estimated range to the filter state variables is

$$\partial r / \partial (x, y, z) = (\hat{x} \quad \hat{y} \quad \hat{z}) / r$$

where

$$r = \sqrt{(\hat{x} - x_T)^2 + (\hat{y} - y_T)^2 + (\hat{z} - z_T)^2}$$

and

$$\partial r / \partial \text{bias} = 1 \quad (\text{assuming range bias is a filter state})$$

The generally-sparse sensitivity array, H, is upper-loaded with the potentially non-zero elements of measurement sensitivity with respect to state. These elements are then related to the appropriate state-vector element by the indexing array, M.

H(1,MINDEX) = \hat{x} / r	M(1,MINDEX) = 1	(sens. to state 1, x)
H(2,MINDEX) = \hat{y} / r	M(2,MINDEX) = 2	(sens. to state 2, y)
H(3,MINDEX) = \hat{z} / r	M(3,MINDEX) = 3	(sens. to state 3, z)
H(4,MINDEX) = 1	M(4,MINDEX) = 22	(sens. to state 22, bias)
	M(5,MINDEX) = 0	(no more non-zero sens.)

The measurement type, weight and time must also be set.

```
MTYPE(MINDEX) = 10
SIGMES(MINDEX) = SIGM(10)
MSTIM(MINDEX) = MTIME
```

The following commons are used for input and output communication to TRANGE.

```
/ACTUAL/    modified to carry true transponder location
             coordinates and true scale factor

/MESCOM/    carries queued measurement processing data from
             TRANGE to the filter -- modified to change dimensions
             when measurement types grow to exceed 10

/MESWTS/    provides actual bias and scale for gaussian errors
             plus measurement weights and bad-measurement
             simulation parameters

/STATCM/    provides estimated position, estimated transponder
             location, estimated range bias for use in computing
             the residual -- modified to add range bias estimate
             and estimated transponder location

/TIMCOM/    provides time of measurement for storing in measurement
             queue for display purposes

/TRUTHC/    provides true position for formulating the true
             measurement.
```

Each of the commons is provided as a separate module ("name".CMN) which is incorporated into the various subroutines via an "INCLUDE" statement. Each commons module contains a comment line listing the accessing subroutines. This line tells which subroutines must be recompiled when the common is changed. The modifier must remember to modify this line to include TRANGE, for example, for each accessed commons module. The commons descriptions files (commons.DOC), contained in Appendix E of this report, must also be updated to include new accessing routines (e.g., TRANGE), set/used information and descriptions of any new elements added to the common.

Having the measurement subroutine and commons formulated and modified, one would next modify MEASIM to reference the measurement subroutine (i.e., TRANGE), following the procedures used in calling the other measurement subroutines.

Module OUTPUT must be modified at entries OUTRES, OUTSIG and OUTSIGHT to take the new measurement type into consideration. The measurement-type labels array, CSIGTYP, must be re-dimensioned to 11 and "TRNG" inserted as the 10th measurement type. TIME then becomes the 11th type by changing IDTYP in entry OUTSIGHT. The format in OUTRES will have to be changed to tabulate the new range residual -- at least as far as the column headings are concerned. Transponder range can simply replace the vertical component of position-fix measurement at this time. Changes in output formats may affect the ability of the plotting software to read the files as plotting input.

The input file, navxxx.INP, and subroutine READINP will have to be modified to accept and treat the new measurement parameters. Group 2 inputs (see Figure 3) would be extended to input a measurement schedule for transponder range. Group 3 would be extended to include a measurement accuracy (uncertainty) for transponder range. Group 4 would be extended to include specifications for the bias and random error scales and bad-measurement parameters for the new measurement. Specifications for the scale-factor error and the transponder location(s) and location errors would require a new category in the navxxx.INP list -- perhaps logically inserted as group 4.4. Subroutine READINP would be modified to (1) read in the new parameters and (2) to transfer them to the appropriate commons.

The measurement errors and transponder locations are actuated in subroutines INTRUE (where the "actual" errors and transponder locations are actuated for each Monte Carlo pass) and INESTM (where the estimated parameters are established at each Monte Carlo pass).

In each case that a subroutine or common is modified, the programmer should be sure to document the modification and to extend the module's prologue to include the modification.

Case 2: Addition of a new state

Suppose one wished to extend the list of states to include a transponder range bias for the new measurement just discussed. This task is simpler than if the new state were a "dynamic" state affecting the equations of motion. The dimensions of some arrays must be increased as must the range of some DO-loops. The subroutines affected are:

- APFILL - the number of potentially non-zero first-order transition matrix elements increases from 73 to 74,
- AUPT - the NRAP (number of elements in a row of AP) array in /TRANS/ is extended to 22 elements, the 22nd being 1 (one element in the new row, on the diagonal)
- the NBEG (index of the element which begins each row) array in /TRANS/ is extended to 22 elements, the 22nd being 74 for the 74th packed-array element
- CHNGX - the range bias estimate is added to /STATCM/ and incremented by extended DELX(22) of /DELCOM/
- ESTMERR - the range bias error in /ESTERR/ is computed as the difference between the actual bias in /ACTUAL/ and the estimated bias in /STATCM/
- FILNIT - the dimensions of the covariance arrays in /KALM/ and /KALMIN/ are changed, as well as dimension-constants.
- UD(253) ==> UD(276)
- GAMMA(21,6) ==> GAMMA(22,6)
- PDIAGO(21) ==> PDIAGO(22)
- NST = 21 ==> NST = 22
- IMAXWM = 21 ==> IMAXWM = 22
- LENUD = 231 ==> LENUD = 253
- add UD(253) = PDIAGO(22)**2
- INESTM - ERRSTATEIC(21) ==> ERRSTATEIC(22) in /ERRCOM/
- estimated range bias is defined as the difference between the actual bias from /ACTUAL/ and the error from /ERRCOM/
- INTRUE - actual range bias is initiated at each Monte Carlo pass and communicated via /ACTUAL/
- MEASUP - internal array-dimensions are increased from 22 to 23 as are /MESCOM/ arrays H, MH and M
- OUTPUT - unit 12 output in OUTSIG entry is changed and SIG array is re-dimensioned to 22

PHIX - internal array dimensions are increased to 22
TIMEUP - internal array UDI is re-dimensioned to 22.

The new state requires two input values in navxxx.INP -- an a priori uncertainty (outline index 3.2.11) and initial estimation error (outline index 4.1.11). These must be accommodated in READINP, the former being stored into PDIAGO(22) and the latter into ERRSTATEIC(22).

Each of the affected commons (*.CMN) must be changed and their descriptive files (*.DOC) changed as well. Then all subroutines which call any of the modified commons must be re-compiled before linking. Of course, the affected subroutine prologues must be modified to reflect the change.

Appendix C

UTILITY PROGRAMS

This appendix contains information on the utility programs used in conjunction with NAVSIM II: the plotting utility, NAVPLT, and the trajectory reference utility, QNEW.

Plotting Utility

NAVPLT is a Fortran-based data display program which runs from the command procedure shown in Figure C1. It uses tabular data files produced by NAVSIM II as its source of data. NAVPLT's program options and capabilities include:

1. displaying
 - a. position and velocity errors versus time
 - b. ground track and altitude versus position
 - c. measurement residuals versus time,
2. re-display of previously generated Versatec DIP files
3. automatic or prescribed scaling.

NAVPLT is written to be run interactively, giving the user the opportunity to specify data source files and to control the plot scales. It always creates DIP files, however, so that the user can obtain hard-copy plots as well. Procedures for running the program are found in Section III.

Figure C1: RNAVPLT Command Procedure

```
$! ***** MAIN DRIVER FOR NAVSIM PLOTS *****
$!
$ DIRECT = F$DIRECTORY()
$ SET DEFAULT 'DIRECT'
$ TYPE SYSS$INPUT
```

```
*****
* Welcome to NAVSIM data plotting program *
*****
```

```
$ TOP:
$ TYPE SYSS$INPUT
```

Current plots available :

- 1 = ERROR PLOTS
- 2 = TRAJECTORY PLOTS
- 3 = MEASUREMENT RESIDUAL PLOTS
- 8 = DIPTEK (DISPLAY PREVIOUSLY CREATED DIP FILE)
- 9 = EXIT (RETURN TO SYSTEM '\$' LEVEL)

```
$ INQUIRE IDO "ENTER NUMBER OF PLOT DESIRED"
$ IF IDO .EQ. 1 THEN GOTO ERRPLTS
$ IF IDO .EQ. 2 THEN GOTO TRJYPLTS
$ IF IDO .EQ. 3 THEN GOTO RESPLTS
$ IF IDO .EQ. 8 THEN GOTO DIPTEK
$ IF IDO .EQ. 9 THEN GOTO EXIT
$ WRITE SYSS$OUTPUT ""
$ WRITE SYSS$OUTPUT "*** ILLEGAL INPUT *** ",IDO
$ GOTO TOP
$!
```


Figure C1: RNAVPLT (continued)

```

$ ERRPLTS:
$ FILE := 'DIRECT'"IDO.INP"
$ OPEN/WRITE INPT 'FILE'
$ WRITE INPT IDO
$ CLOSE INPT
$ ASSIGN 'FILE' FOR007
$ ASSIGN SYS$COMMAND FOR005
$ R FSNO:[FSNAMA.QPLOT]RERRPLT
$ DEASSIGN FOR005
$ DEASSIGN FOR007
$ DEL IDO.INP;*
$ WRITE SYS$OUTPUT "^X"
$ GOTO TOP
$!
$ TRJYPLTS:
$ FILE := 'DIRECT'"IDO.INP"
$ OPEN/WRITE INPT 'FILE'
$ WRITE INPT IDO
$ CLOSE INPT
$ ASSIGN 'FILE' FOR007
$ ASSIGN SYS$COMMAND FOR005
$ R FSNO:[FSNAMA.QPLOT]RTRJYPLT
$ DEASSIGN FOR005
$ DEASSIGN FOR007
$ DEL IDO.INP;*
$ WRITE SYS$OUTPUT "^X"
$ GOTO TOP
$!
$ RESPLTS:
$ FILE := 'DIRECT'"IDO.INP"
$ OPEN/WRITE INPT 'FILE'
$ WRITE INPT IDO
$ CLOSE INPT
$ ASSIGN 'FILE' FOR007
$ ASSIGN SYS$COMMAND FOR005
$ R FSNO:[FSNAMA.QPLOT]RRESPLT
$ DEASSIGN FOR005
$ DEASSIGN FOR007
$ DEL IDO.INP;*
$ WRITE SYS$OUTPUT "^X"
$ GOTO TOP
$!
$ DIPTEK:
$ ASSIGN SYS$COMMAND FOR005
$ R FSNO:[FSNAMA.QPLOT]DIPTEK
$ DEASSIGN FOR005
$ WRITE SYS$OUTPUT "^X"
$ GOTO TOP
$!
$ EXIT:
$ EXIT

```

The default values of NAVPLT are defined in a value-setting file, SETVAL.INP. An example of this file is shown as Figure C2. As seen in the figure, the file is divided into sections, one for each main plot type (error, trajectory, residuals). Each section is divided into lines which contain the following fields:

- Field 1. Line number: three numbers, separated by periods, are used by the program to indicate which plot parameters to access (do NOT change these numbers)
- Field 2. Text: labels and parameter names
- Field 3. Parameter value: numeric value to be used as a limiting value for the plot.

Each section contains the following elements:

- Element 1. Title line: lines numbered 0.0.0 title the section
- Element 2. Use line: line number <n>.0.1 indicates to the program whether the user wishes to use the values in SETVAL.INP or to obtain the values interactively. A value of (1) indicates that the defined SETVAL values are to be used, while (0) will obtain the values interactively.
- Element 3. Information labels: if the third number is (0), (for example, 1.3.0), the line-entry is a label
- Element 4. Plot parameters: these lines contain an (=) and a following number which defines the parameter's value. Min/max values must contain a decimal point, while indicators must not contain a decimal point.

Numeric values which follow an (=) must be preceded by a space. The min/max values for trajectory plots have been scaled down by 1000 to facilitate plot axis labeling.

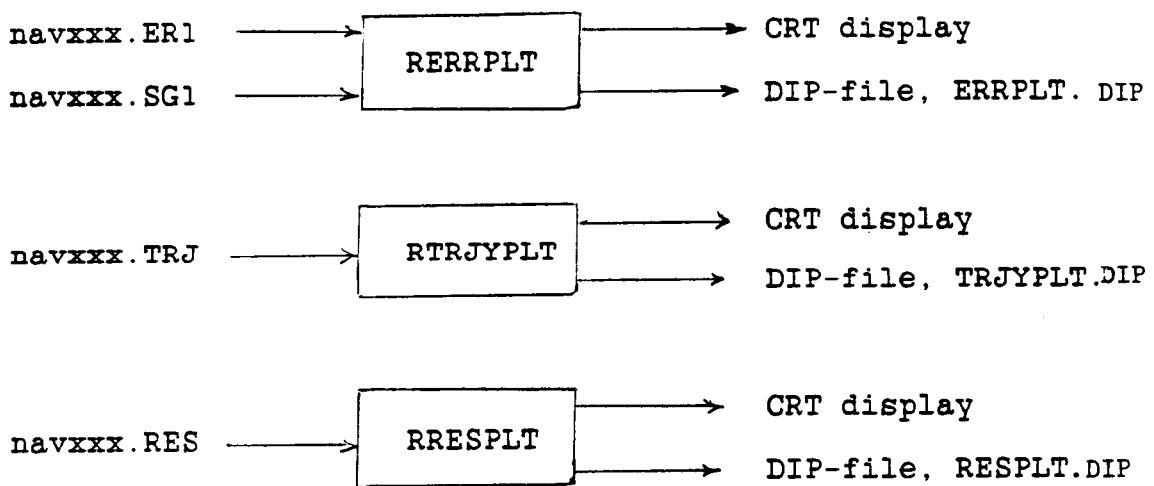
Figure C2: SETVAL.INP Plot-parameter Default File

```
0.0.0 ERROR PLOTS
1.0.1 USE = 1 ! = 1 USE VALUES, = 0 ASK FOR VALUES
1.1.0 Maximum Y-axis values for Error plots
1.1.0 Position Errors
1.1.2 X-ERROR = 3002.
1.1.3 Y-ERROR = 3002.
1.1.4 Z-ERROR = 502.
1.2.0 Velocity Errors
1.2.1 XV-ERROR = 10.
1.2.2 YV-ERROR = 10.
1.2.3 ZV-ERROR = 10.
0.0.0 TRAJECTORY PLOTS
2.0.1 USE = 1 ! = 1 USE VALUES, = 0 USE DEFAULTS
2.1.0 Max, Min, Delta for X-axis: PLOT #1
2.1.1 MAX VALUE = 25.
2.1.2 MIN VALUE = -25.
2.1.3 DELTA VALUE = 5.
2.2.0 Max, Min, Delta for X-axis: PLOT #2
2.2.1 MAX VALUE = 150.
2.2.2 MIN VALUE = 0.
2.2.3 DELTA VALUE = 15.
2.3.0 Max, Min, Delta for Y-axis: PLOT #1
2.3.1 MAX VALUE = 15.
2.3.2 MIN VALUE = -10.
2.3.3 DELTA VALUE = 5.
2.4.0 Max, Min, Delta for Y-axis: PLOT #2
2.4.1 MAX VALUE = 4.
2.4.2 MIN VALUE = 0.
2.4.3 DELTA VALUE = 1.
2.5.0 Marker every Nth point
2.5.1 N = 60
2.6.0 Marker symbol (DISSPLA marker values)
2.6.1 SYMBOL = 15
0.0.0 RESIDUAL PLOTS
3.0.1 USE = 1
3.1.0 Maximum values for residual plots
3.1.1 ALTITUDE = 220.
3.1.2 GPS = 2800.
3.1.3 DOPPLER = 9.
3.1.4 AIRSPEED = 1.
3.1.5 POSITION FIX = -50.
3.2.0 Minimum values for residual plots
3.2.1 ALTITUDE = -310.
3.2.2 GPS = -4300.
3.2.3 DOPPLER = -12.
3.2.4 AIRSPEED = -1.
3.2.5 POSITION FIX = 50.
7.7.7 END
```

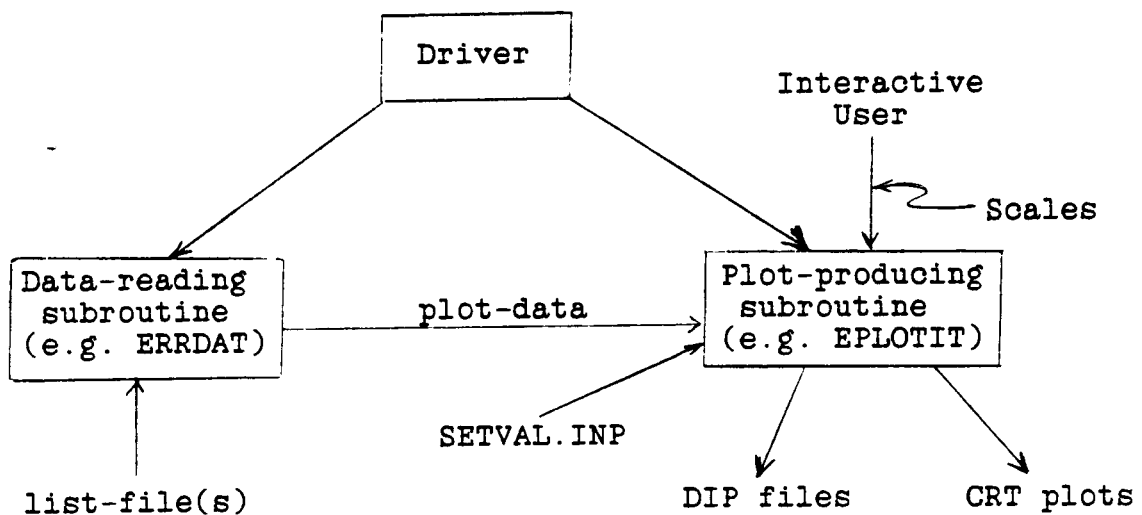
Program structure: -- Examination of Figure C1 will show that NAVPLT is composed of four individual executable programs:

1. RERRPLT for position and velocity error plots
2. RTRJYPLT for trajectory plots
3. RRESPLT for measurement residual plots
4. DIPTEK for re-display of previously generated DIP files.

These programs are called into execution through interactive control of the NAVPLT command procedure. Each of the first three programs reads its input from the list-files produced by NAVSIM II.



Each of these three programs has the structure shown in the following sketch.



Reference Trajectory Utility

An interactive program is included with NAVSIM II as a utility for helping the user to specify the reference trajectory. The trajectory generation routine, TAF COS, requires trajectory-specification parameters in the particular format described in Section III and shown in Figure 4. The utility program, QNEW, was developed in order to calculate the required trajectory-specification parameters from responses to a set of appropriate prompts. QNEW enables en-route trajectories as well as terminal-area trajectories and provides for off-axis flight paths in the terminal area. It will find appropriate circular arcs and straight-line segments to connect different terminals. The output from QNEW is a file written to FOR010.DAT in the format required by TAF COS. This file is not quite complete, however, to be used as the TRJXXX.INP file needed by NAVSIM II and must be edited somewhat. The editing requirements will be obvious when the user compares the FOR010.DAT file with the example trajectory-input file shown in Figure 4.

The source-files for the various modules which comprise program QNEW are all found in a single module, QNEW.FOR. This module may thus be edited and then re-compiled and re-linked with the simple commands:

```
$ FOR QNEW
$ LINK QNEW
```

The interactive dialogue with QNEW begins with:

```
ENTER LAT-LNG AND X-AXIS HEADING OF YOUR COORDINATE SYSTEM DEG
```

The purpose of this information is to enable translation and rotation of coordinate frames for en-route and terminal-to-terminal trajectories. If the entire trajectory is to be flown in the same coordinate system, the user's response to this prompt is irrelevant. Otherwise, he should respond with latitude (+North, degrees), longitude (+East, degrees) and x-axis heading (+CCW from North, degrees), separated with commas. Subsequent prompts ask for the various parameters which must be supplied in order to define each "leg" of the trajectory. The prompting sequence will vary according to the responses provided by the user. The program treats three types of flight segment:

1. local flight
2. long-distance flight
3. hover and ascent/descent.

In any case, the user must supply the initial position (x,y,z) and ground speed.

Local flight

The program treats both straight-line segments and circular arcs. The inputs are:

for straight lines

either heading and distance to the end-point
or x,y coordinates of the end-point

altitude and speed at the end point

for circular arcs

either final heading at the end of the arc
or fraction of a full circle to traverse

radius and sense of the turn

altitude and speed at the end point

Long-distance flight

The program determines circular arcs and straight-line segments to tangentially connect turning-circles at each end of the "leg." The two end-points may be specified in different coordinate systems (i.e., at different latitudes and longitudes and at different x-axis headings). This program feature implements some of the formulations found in [6] "Automation of Onboard Flight Management," NASA TM-84212 by H. Erzberger. The user is required to supply turn-radii at the initial and final points of the "leg." He may specify turn senses or let the program determine these so that the transfer path length is minimized. The final heading (i.e., at the end of the turn-straight-turn combination) may either be specified or else may be determined by the program to be that which will direct the aircraft toward the next (as yet unspecified) destination point.

Hover and ascent/descent

The only specification for hover is the time duration. Ascent and descent require specification of final altitude as well.

The path-generation feature is not as user-friendly as it could be made to be with additional effort, but it can be useful in assisting in construction of complex reference paths for input to TAF COS.

Appendix D

SUBROUTINE SUMMARIES

This appendix contains a brief summary of each of the subroutines of NAVSIM II. The summarized subroutines appear in alphabetical order. Included in each summary are:

1. a brief statement of the subroutine's purpose,
2. a list of every communicated parameter which is set or used by the subroutine,
3. a list of the routines which reference the subroutine being summarized,
4. a list of the subroutines referenced by the subroutine being summarized.

SUBROUTINE AIRSPEED

```

C* *****
C*
C*
C*           AIRSPEED - Models airspeed measurements (MTYPE=7)
C*
C* Intermodule Data :
C*   /ACTUAL/  ACTUAL_ASSF  U   True airspeed scale factor
C*   /MESCOM/  MINDEX      S/U  Measurement queue index
C*           MTYPE(10)    S   Measurement-type code array
C*           H(22,10)     S/U  Measurement sensitivity vector
C*           M(22,10)     S   Identifying array for H
C*           MSTIM(10)    S   Time measurement taken
C*           SIGMES(10)   S   Measurement weight
C*           YRES(10)     S   Measurement residual array
C*   /MESWTS/  PBIAS(10)   U   Bias of measurement
C*           PGAUSS(10)  U   Scale for random measurement noise
C*           PBADPCT(10) U   Percentage of bad measurements
C*           PSCRNLMT(10) U   Threshold for screening bad points
C*           PMIN(10)    U   Minimum acceptable measurement
C*           SIGM(10)    U   Measurement weights
C*   /STATCM/  ASSFE       U   Airspeed scale factor estimate
C*           VN(3)       U   Velocity vector
C*           WINDE(2)    U   Wind estimate
C*   /TIMCOM/  MTIME       U   Time counter at filter rate
C*   /TRUTHC/  TVW(3)      U   True wind velocity
C*           V(3)        U   True velocity in Earth-fixed coords
C*
C* Called By :  MEASIM
C*
C* Modules Referenced : BARN1
C*
C* *****

```


SUBROUTINE ALTMTR (KTYPE)

```

C*****
C*
C*
C* ALTMTR(KTYPE) - Models baro- or radio-altimeter
C*
C* Intermodule Data :
C*           KTYPE           U   If KTYPE=1 - Baro-altimeter
C*                               If KTYPE=2 - Radio-altimeter
C*   /MESCOM/  MTYPE(10)     S   Measurement type code
C*             H(22,10)      S   Measurement sensitivity vector
C*             M(22,10)      S   Identifying array for H
C*             YRES(10)      S   Measurement residual
C*             SIGMES(10)    S   Measurement weight
C*             MSTIM(10)     S   Time measurement taken
C*             MINDEX        S/U  Measurement queue index
C*   /MESWTS/  SIGM(10)      U   Measurement weights
C*             PBIAS(10)     U   Bias of measurement
C*             PGAUSS(10)    U   Scale for random measurement noise
C*             PBADPCT(10)  U   Percentage of bad measurements
C*             PSCRNLMT(10) U   Threshold for bad points
C*             PMIN(10)      U   Minimum measurement for processing
C*   /STATCM/  X(3)          U   Estimated position vector
C*   /TIMCOM/  MTIME         U   Time counter at filter rate
C*   /TRUTHC/  R(3)          U   True position vector
C*
C* Called By : MEASIM
C*
C* Modules Referenced : BARN1
C*
C*****

```

SUBROUTINE APFILL

```
C*****
C*
C*
C* APFILL - To fill the transition matrix elements
C*
C* Intermodule Data :
C*   /TRANS/      AP(100)      S   Integral of AK
C*                AK(100)      S/U  Sum of the non-zero elements
C*                                of the transition matrix
C*
C* Called By : TIMEUP
C*
C* Modules Referenced : NONE
C*
C*****
```

SUBROUTINE AUPT

```
C*****
C*
C*
C* AUPT - Calculates non-zero elements of the transition matrix
C* AUPTIN - (Entry) Initializes constant values
C*
C* Intermodule Data :
C*   /AUPCOM/   TAUGYR(3)  U   Time constants for gyro noise, seconds
C*             TAUACB      U   Time constants for accelerometer noise,
C*   /CONST/   GO          U   Gravitational acceleration at sea level
C*   /EROTIC/  WEV(3)     U   Earth's rotational rate vector
C*                                 in Earth-fixed coordinates, rad/sec
C*             RE          U   Earth's radius, ft
C*   /GRVCOM/  GVEC(3)    U   Gravity vector, ft/sec**2
C*   /STATCM/  X(3)       U   Position vector, ft
C*             VN(3)       U   Velocity vector, ft/sec
C*             TEB(3,3)    U   Estimated Earth to body transformation
C*   /TIMCOM/  DT          U   Integration step size, sec
C*   /TRANS/   AP(100)    S   Integral of AK
C*             AK(100)    S/U  Sum of the non-zero elements of the
C*                                 Jacobian matrix
C*             AM(100)    S/U  Non-zero elements of the Jacobian matrix
C*             NAP(100)   S   State index for the Jth non-zero element
C*                                 of the transition matrix
C*             NRAP(21)   S   Number of non-zero elements in the Ith
C*                                 row of the transition matrix
C*             NBEG(21)   S   Index of the first non-zero element in
C*                                 the Ith row of the transition matrix
C*             ORDER     S   Highest order for transition-matrix
C*                                 updating in subroutine PHIX
C*
C* Called By :   INESTM
C*              NAVEQ
C*
C* Modules Referenced : NONE
C*
C*****
```

REAL FUNCTION BARN1(I, IKEY, IFRN)

```
C*****
C*
C*
C* BARN1 - random number generator
C*
C* Intermodule Data :
C*      I          U    If I < 0  Gaussian distribution
C*                  U    If I > 0  Uniform distribution
C*      IKEY       U    If IKEY < 0 Use compiled value for IFRN
C*                  U    If IKEY > 0 U value transmitted in call
C*      IFRN      U    First random number
C*
C* Local Data :
C*      SD          The desired standard deviation----= 1.
C*      AMEAN       The desired mean-----= 0.
C*      H           The population size-----= 12.
C*
C* Called By :  AIRSPEED
C*              ALTMTR
C*              DOPPLER
C*              FILNIT
C*              GPS
C*              INESTM
C*              INTRUE
C*              NAVEQ
C*              READINP
C*
C* Modules Referenced : NONE
C*
C*****
```

SUBROUTINE CHNGX

```
C
C* *****
C*
C*
C* CHNGX - Puts estimated state changes into estimated state
C*
C* Intermodule Data :
C*   /DELCOM/   DELX(21)   S/U   Incremental state estimates
C*   /STATCM/   X(3)       S/U   Position vector
C*             VN(3)       S/U   Velocity vector
C*             TILTS(3)    S/U   Tilt vector
C*             GDRE(3)     S/U   Gyro drift rate estimates
C*             ACNBE       S/U   Vertical acceleration bias estimate
C*             CLKBE       S/U   Clock bias estimate
C*             CKBRE       S/U   GPS clock bias rate estimate
C*             ASSFE       S/U   Airspeed scale factor estimate
C*             WINDE(2)    S/U   Wind estimate
C*
C* Called By :   KALFIL
C*
C* Modules Referenced : MVTRN
C*                       TWIST
C* *****
```

SUBROUTINE CNVLIN(NB,B,NMAX,XOUT,NBAD)

```
C*****
C*
C*
C* CNVLIN(NB,B,NMAX,XOUT,NBAD) - Accepts an ascii string and
C*                               converts up to 6 numbers starting
C*                               with =, and separated with commas
C*
C* Intermodule Data :
C*      NB      U      Number of characters in input line
C*      B      U      Input line
C*      NMAX    U      Maximum number of values to be returned
C*      XOUT    S      Returned result
C*      NBAD   S/U    Error flag
C*
C* Called By :  READINP
C*
C* Modules Referenced : NONE
C*****
```

SUBROUTINE DOPPLER

```

C* *****
C*
C*
C* DOPPLER - Simulate Doppler measurements
C*
C* Intermodule Data :
C*   /CONST/      D2R          U Radian to degree conversion
C*   /MESCOM/     MTYPE(10)   S Measurement type code
C*                                     If MTYPE = 1 Baro-altimeter
C*                                     If MTYPE = 2 Radio-altimeter
C*                                     If MTYPE = 3 GPS pseudo range
C*                                     If MTYPE = 4,5,6 Doppler velocity
C*                                     If MTYPE = 7 Airspeed measurement
C*                                     If MTYPE = 8,9 Position fix
C*           H(22,10) S/U Measurement sensitivity vector
C*           M(22,10) S Identifying array for H
C*           YRES(10) S/U Measurement residual array
C*           SIGMES(10) S/U Measurement weight
C*           MINDEX S/U Measurement queue index
C*           MSTIM(10) S Time of measurement
C*   /MESWTS/     SIGM(10)    U Measurement weights
C*           PBIAS(10) U Bias of measurement
C*           PGAUSS(10) U Scale for random measurement noise
C*   /STATCM/     VN(3)      U Velocity vector
C*           DNTE(3)    U Doppler antenna mounting estimate
C*           TEB(3,3)  U Estimated Earth to body transformation
C*   /TIMCOM/     MTIME      U Time counter at integration rate
C*   /TRUTHC/     V(3)       U True velocity in Earth-fixed coords.
C*           ETB(3)    U Earth to body transformation
C*
C* Called By : MEASIM
C*
C* Modules Required : BARN1
C*                   CROSS
C*                   MVTRN
C* *****

```

SUBROUTINE ESTMERR

```

C*****
C*
C*
C* ESTMERR - This module computes the difference between true
C*           and estimated states
C*
C* Intermodule Data :
C*   /ACTUAL/   ACTUAL_GDR(3)  U   True gyro drift rate, rad/sec
C*              ACTUAL_ACNB(3) U   True accelerometer null bias, ft/sec**2
C*              ACTUAL_GSF(3)  U   True gyro scale factor errors
C*              ACTUAL_ASF(3)  U   True accelerometer scale factor errors
C*              ACTUAL_CLKB    U   True GPS receiver clock bias, nanosec
C*              ACTUAL_CKBR    U   True clock bias rate, nano-sec
C*              ACTUAL_DNT(3)  U   True doppler antenna mounting, radians
C*              ACTUAL_ASSF    U   True airspeed scale factor
C*   /ESTERR/   DX(3)          S   Position estimation error in feet
C*              DXD(3)         S   Velocity estimation error in radians
C*              DOM(3)         S   Tilt estimation error in radians
C*              CLKBER         S   Error in clock bias estimate
C*              GDRER(3)       S   Error in gyro drift rate estimates
C*              ACNBER         S   Error in accelerometer null bias est.
C*              CKBRER         S   Error in clock bias rate estimates
C*              DNTER(3)       S   Error in Doppler antenna mounting est.
C*              ASSFER         S   Error in airspeed scale factor
C*              WINDER(2)     S   Error in wind estimate
C*   /STATCM/   X(3)          U   Position vector
C*              VN(3)         U   Velocity vector
C*              TILTS(3)      U   Tilt vector
C*              CLKBE         U   Clock bias estimate
C*              GDRE(3)       U   Gyro drift rate estimate
C*              ACNBE         U   Accelerometer null bias estimate
C*              CKBRE         U   Clock bias rate estimate
C*              DNTE(3)       U   Doppler antenna mounting estimate
C*              ASSFE         U   Airspeed scale factor estimate
C*              WINDE(2)     U   Wind estimate
C*   /TRUTHC/   R(3)          U   True position vector in Earth-fixed
C*              V(3)          U   True velocity vector in Earth-fixed
C*              coordinates, ft/sec
C*
C* Called By :   KALFIL
C*
C* Modules Referenced : MVRN
C*                      TWIST
C*****

```


SUBROUTINE FILNIT

```

C*****
C*
C*
C* FILNIT - Initialize arrays and variables
C*
C* Intermodule Data :
C*   /AUPCOM/  TAUGYR(3)  U   Time constants for gyro noise, sec
C*             TAUACB    U   Time constant for accelerometer noise,
C*   /CONST/   GO        U   Gravitational acceleration
C*                                 ft/sec**2
C*             D2R       U   Degree to radian conversion factor
C*             SPS2SPH   S   Seconds per second to seconds/hour
C*             FPS22UG   S   Feet per second squared to micro-g
C*             RPS2DPH   S   Radians per second to degrees/hour
C*             R2ASEC    S   Radians to arc-seconds
C*             RAT2PCT   S   Ratio to percent
C*   /DELCOM/  DELX(21)  S   Incremental state estimates
C*   /KALM/    NST       S/U  Number of states total
C*             NFF       S   Number of forcing functions
C*             IMAXWM    S   Dimension of WM
C*             UD(253)   S/U  U-D covariance matrix
C*             GAMMA(21,6) S   Forcing function sensitivities
C*             QD(6)     S   Process noise covariance
C*   /KALMIN/  USIGM(10) U   Measurement weights
C*             PDIAGO(21) U   Initial state uncertainties
C*             QDIAGO(6)  U   Process noise uncertainties
C*             BIASM(10)  U   Measurement bias scale
C*             GAUSSM(10) U   Scale of random measurement noise
C*             BADPCT(10) U   Percentage of bad measurements
C*             SCREEN_LIMIT(10) U   Threshold for bad measurements
C*             AMIN(10)  U   Minimum value for taking meas.
C*   /MESCOM/  MINDEX    S   Measurement queue index
C*   /MESWTS/  SIGM(10)  S   Measurement weights
C*             PBIAS(10) S   Bias of measurement
C*             PGAUSS(10) S   Scale for random measurement noise
C*             PBADPCT(10) S   Percentage of bad points
C*             PSCRNLMT(10) S   Threshold for screening bad data
C*             PMIN(10)  S   Minimum of acceptable measurements
C*   /TIMCOM/  DT        U   Integration step size, seconds
C*
C* Called By : MONTEC
C*
C* Modules Referenced : BARN1
C*
C*****

```

SUBROUTINE FILSIM

```
C*****
C*
C*
C* FILSIM - Filter simulation
C*      Called by TAF COS 20 times per second
C*      If NTIME>0 it calls NAVEQ
C*      It calls MEASIM at NTIME=0,20,40,... then KALFIL
C*
C* Intermodule Data :
C*   /TIMCOM/  NTIME      U   Time counter at integration rate
C*
C* Called By : TAF COS
C*
C* Modules Referenced : KALFIL
C*                      MEASIM
C*                      NAVEQ
C*                      OUTTRJ
C*
C*****
```

REAL FUNCTION GDOP*4(GU, I,J,K,L)

```
C*****
C*
C*
C* GDOP(GU,I,J,K,L) - Computes geometric dilution of precision for 4
C*                   satellites in a real*4 FUNCTION
C*
C* Intermodule Data :
C*       GU(3,10)  U   Matrix of unit slant-range vectors
C*       I,J,K,L  U   Unit slant-range vectore are
C*                   located in columns I,J,K,L of GU
C*
C* Called By : GPS
C*             SEESAT
C*
C* Modules Referenced : INVERT
C*
C*****
```

SUBROUTINE GPS

```

C* ****
C*
C* GPS - Models pseudo-range measurements for GPS (MTYPE=3)
C*
C* Intermodule Data :
C*   /ACTUAL/   ACTUAL_CLKB   U   True clock bias
C*   /CONST /   R2D           U   Radian to degree conversion
C*             FTPKM          U   Feet per kilometer
C*             C              U   Speed of light, ft/sec
C*   /EROTIC/   WE           U   Earth's rotation rate, rad/sec
C*             RE           U   Earth's radius, ft
C*             TI2CO(3,3)    U   Inertial-to-runway transformation
C*   /MESCOM/   MTYPE(10)    S   Measurement type code
C*             H(22,10)     S/U  Measurement sensitivity vector
C*             M(22,10)     S   Identifying array for H
C*             YRES(10)     S/U  Measurement residual array
C*             SIGMES(10)   S   Measurement weight
C*             MSTIM(10)    S   Time of measurement
C*             MINDEX       S/U  Measurement queue index
C*   /MESWTS/   SIGM(10)     U   Measurement weights
C*             PBIAS(10)    U   Bias of measurement
C*             PGAUSS(10)   U   Scale for random measurement noise
C*   /ORBITS/   OINC         U   Orbital inclination
C*             ASCNOD(6)    U   Ascending nodes
C*             ANOMLY(6,6)  U   Mean anomalies
C*             RORB         U   Orbital radius for GPS orbits
C*             ORBITAL_RATE U   Ang. rate for sats., rad/sec
C*             UM           S/U  Grav. const., km**3/sec**2
C*             EFEMER(3)    U   Ephem. errors feet
C*             RCA(3,6,6)   U   Radial, cross, in-track ephemeris
C*                               errors, feet
C*   /STATCM/   X(3)         U   Position vector
C*             CLKBE        U   Clock bias estimate
C*   /TIMCOM/   MTIME        U   Time counter at integration rate
C*   /TRUORB/   RSO(3,24)    U   Initial satellite position vector
C*             VSO(3,24)    U   Initial satellite velocity vector
C*             CB(20,24)    S/U  Orbital parameters
C*   /TRUTHC/   R(3)         U   True position, feet
C*   /VISCOM/   ISAT         S/U  Current satellite number
C*             ELMIN        U   Minimum elevation angle
C*             GDOP         S   Calculated GDOP
C*             ILSAT        S/U  Index for ISATS to select ISAT
C*             NSEC_CHNG_SATS U   Time interval with fixed sat. set
C*             LSATS(4,20)  U   Array of visible satellites
C*
C* Called By : MEASIM
C*
C* Modules Referenced : BARN1
C*                     DOT
C*                     GDOP
C*                     MVTRN
C*                     STEPD
C*                     VNORM
C*

```

SUBROUTINE INEREF

```
C*****
C*
C*
C* INEREF - Initializes Earth rate vector and radius of curvature
C*
C* Intermodule Data :
C*   /CONST/   R2D           U   Radius to degree conversion
C*   /EROTIC/  WEV(3)        U   Earth's rotation rate vector
C*            WE            U   Earth's rotation rate
C*            REV(3)        U   Earth's radius vector
C*            RE            S/U  Earth's radius
C*            TI2CO(3,3)    U   Inertial-to-runway transformation
C*   /REFCOM/  VLAT          U   Runway reference latitude
C*            VLON          U   Runway reference longitude
C*            RAZ           U   Runway azimuth
C*            GHAO          U   Initial Greenwich hour angle
C*
C* Called By :  NAVSIM
C*
C* Modules Referenced : GTRN
C*
C*****
```

SUBROUTINE INESTM

```

C*****
C*
C*
C* INESTM - Initialize estimation equations
C*
C* Intermodule Data :
C*   /ACTUAL/   ACTUAL_GDR(3)  U   True gyro drift rate, rad/sec
C*              ACTUAL_ACNB(3) U   True accelerometer null bias,
C*                               ft/sec/sec
C*   /CONST/   R2ASEC          U   Radians to arc-seconds
C*   /ERRCOM/  ERRSTATEIC(21) U   Initial input error conditions
C*              ERGGYR(3)      U   Accel. errors, sf, bias, random
C*              ERRACC(3)      U   Accel. errors, sf, bias, random
C*              ICERR          U   Error treatment flag
C*              IGRAND         U   Random gyro flag
C*              IARAND         U   Random accelerometer flag
C*   /STATCM/  ACNBE           S   Accelerometer null bias estimate
C*              ASSFE          S   Airspeed scale factor estimate
C*              CLKBE          S   Clock bias estimate
C*              CKBRE          S   Clock bias rate estimate
C*              DNTE(3)        S   Doppler antenna mounting estimate
C*              GDRE(3)        S   Gyro drift rate estimate
C*              TILTS(3)       S   Tilt vector
C*              VN(3)          S   Velocity vector
C*              WINDE(2)       S   Wind estimate
C*              X(3)           S   Position vector
C*   /TRUTHC/  R(3)           U   True position vector in
C*                               Earth-fixed co-ordinates, feet
C*              V(3)          U   True velocity vector in
C*                               Earth-fixed co-ordinates, ft/sec
C*
C* Called By : MONTEC
C*
C* Modules Referenced : AUPTIN
C*                      BARN1
C*                      NAVINIT
C*                      TWIST
C*****

```

SUBROUTINE INTRUE

```

C*****
C*
C*
C* INTRUE - Initialize all true conditions
C*
C* Intermodule Data :
C*   /ACTUAL/   ACTUAL_GDR(3)   S   True gyro drift rate, rad/sec
C*              ACTUAL_ACNB(3)  S   True accel. null bias, ft/sec
C*              ACTUAL_GSF(3)   S   True gyro scale factors
C*              ACTUAL_ASF(3)   S   True accelerometer scale factors
C*              ACTUAL_CLKB     S   True GPS clock bias, nanosec
C*              ACTUAL_CKBR     S   True clock bias rate, nano-sec
C*              ACTUAL_DNT(3)   S   True Doppler antenna errors, deg
C*              ACTUAL_ASSF     S   True airspeed scale factor
C*   /CONST/   FPS22UG         U   Feet per second to micro-g's
C*              RAT2PCT         U   Ratio to per cent
C*              RPS2DPH         U   Radians per second to deg./hr
C*              SPS2SPH         U   Seconds/hr
C*   /ERRCOM/   ERRSTATEIC(21) U   Initial input error conditions
C*              ERRGYR(3)       U   Gyro errors 1 sf, 2 bias, 3 rand
C*              ERRACC(3)       U   Accel. input errors
C*              ICERR           U   Input errors in measurement
C*
C* Called By :  MONTEC
C*
C* Modules Referenced : BARN1
C*                      TAFINIT
C*                      TRUSAT
C*****

```

SUBROUTINE KALFIL

```

C*****
C*
C*
C* KALFIL - Master driver for the KALMAN filter
C*       Calls TIMEUP, then calls MEASUP until MINDEX=0
C*       Called by FILSIM once per second
C*
C* Intermodule Data :
C*   /KALM/      NST          U   Number of states total
C*              UD(253)      U   U-D covariance matrix - upper loaded
C*   /MESCOM/    MTYPE(10)   U   If MTYPE=1 : Baro-altimeter
C*                                     If MTYPE=2 : Radio-altimeter
C*                                     If MTYPE=3 : GPS pseudo-range
C*                                     If MTYPE=4,5,6 : Doppler velocity meas.
C*                                     If MTYPE=7 : Airspeed measurement
C*                                     If MTYPE=8,9 : Position fix
C*              MINDEX      S/U  Measurement queue index
C*   /TIMCOM/    MTIME       U   Time counter at integration rate
C*
C* Called By :   FILSIM
C*
C* Modules Referenced :  CHNGX
C*                       ESTMERR
C*                       MEASUP
C*                       OUTERR
C*                       OUTRES
C*                       TIMEUP
C*                       UD2SIG
C*****

```


SUBROUTINE MEASIM

```
C*****
C*
C*
C* MEASIM - Simulates measurements for NAVSIM II.
C*       Called once per second by FILSIM
C*
C* Intermodule Data :
C*   /TIMCOM/  MTIME           U   Time counter at filter rate
C*   /MESKED/  MESINT(10)     U   Measurement interval
C*             MESSTOP(10)    U   Measurement stop time
C*             MESTIME(10) S/U  Current time for each meas.
C*
C* Called By :  FILSIM
C*
C* Modules Referenced : AIRSPEED
C*                     ALTMTR
C*                     DOPPLER
C*                     GPS
C*                     POSFIX
C*****
```

SUBROUTINE MEASUP(MM)

```
C*****
C*
C* MEASUP(MM) - Processes measurement data and updates the
C*               UD matrix and the state estimates
C*
C* Intermodule Data :
C*      MM           U   Measurement queue index
C*  /DELCOM/ DELX(21) S/U Incremental state estimates
C*  /KALM/   NST      U   Number of states total
C*           UD(253)  S/U U-D covariance of uncertainty (R*8)
C*  /MESCOM/ H(22,10) U   Measurement sensitivity vector
C*           M(22,10) U   Identifying array for H above
C*           YRES(10) U   Measurement residual array
C*           SIGMES(10) U Measurement weights for the filter
C*  /PRTCOM/ PRTM     U   Extra-output print flag for meas.
C*  /TIMCOM/ MTIME    U   Time counter at filter rate
C*
C* Called By : KALFIL
C*
C* Modules Referenced : TRIMAT
C*                     UDMES
C*
C*****
```

SUBROUTINE MONTEC

```
C*****
C*
C* MONTEC - Initialize variables at start of each Monte Carlo pass.
C*       Scales initial errors
C*
C* Intermodule Data :
C*   /MESKED/  MESSTART(10)  U   Measurement start time (sec)
C*           MESTIME(10)   S   Current next-measurement time
C*   /TIMCOM/  MTIME        S   Time counter at filter rate
C*           NTIME        S   Time counter at integration rate
C*   /VISCOM/  ILSAT        S   Index for selecting 4 satellites
C*
C* Called By :  NAVSIM
C*
C* Modules Referenced : FILNIT
C*                   INESTM
C*                   INTRUE
C*                   OUTINIT
C*****
```

SUBROUTINE NAVEQ

```

C*****
C*
C*
C* NAVEQ   - Integrate with spheroidal, rotating Earth
C* NAVINIT - (Entry) Initialize internal variables for NAVEQ
C*
C* Intermodule Data :
C*   /ACTUAL/  ACT_GDR(3) U   True gyro drift rate, rad/sec
C*             ACT_ACNB(3)U   True accel. null bias, ft/sec**2
C*             ACT_GSF(3) U   True gyro scale factor, unitless
C*             ACT_ASF(3) U   True accel. scale factor, unitless
C*   /CONST /  GO           U   Gravitational acceleration ft/sec**2
C*   /EROTIC/  RE           U   Radius of the earth, feet
C*             WEV(3)       U   Earth's rotation rate vector, rad/sec
C*   /GRVCOM/  GVEC(3)     S/U  Estimated gravity vector, ft/sec**2
C*   /STATCM/  GDRE(3)     U   Estimated gyro drift rate, rad/sec
C*             TEB(3,3)    S/U  Estimated earth-to-body transformation
C*             TILTS(3)   S/U  Estimated tilt vector, radians
C*             VN(3)       S/U  Estimated velocity, feet/second
C*             X(3)        S/U  Estimated position, feet
C*   /TIMCOM/  DT           U   Integration time-step, seconds
C*             MTIME       S   Time counter at filter rate (1/sec)
C*             NTIME       U   Time counter at integration rate (20/sec)
C*   /TRUTHC/  A(3)        U   True specific force vector, g's
C*             ETB(3,3)    U   True earth-to-body transformation
C*             W           U   Angular velocity (body rate) vector
C*             WD          U   Angular acceleration vector
C*
C* Called By : FILSIM
C*             INESTM
C*
C* Modules Required : AUPT
C*                   CROSS
C*                   MVTRN
C*                   TWIST
C*****

```

PROGRAM NAVSIM

```
C* ****
C*
C*
C*      NAVSIM II EXECUTIVE : FILED AS NAVSIM.FOR
C*
C* Intermodule Data :
C*   /TIMEIN/   MNTCRL  U      Number of Monte Carlo passes
C*   /OUTCOM/   MCPASS  S      Current Monte Carlo pass
C*
C* Modules Referenced : INEREF
C*                      MONTEC
C*                      OUTCLSE
C*                      READINP
C*                      SEESAT
C*                      TAFCOS
C* ****
```

SUBROUTINE OUTPUT

```

C*****
C* OUTPUT - Control output
C* ENTRY OUTOPEN - Open output files
C* ENTRY OUTINIT - Initialize at beginning of each Monte Carlo pass
C* ENTRY OUTTRJ - Write trajectory data to output file
C* ENTRY OUTRES - Write residual data to output file
C* ENTRY OUTERR - Write measurement errors to output files
C* ENTRY OUTSIG - Write standard deviation of estimate for measurements
C* ENTRY OUTSIGT - Write standard deviations of error for time update
C* ENTRY OUTCLSE - Close output files
*
C* Intermodule Data :
C* /CONST/ FPS22UG U Feet per second squared to micro-g's
C* R2ASEC U Radians to arcseconds
C* RAT2PCT U Ratio to percent
C* RPS2DPH U Radians per second to degrees per hour
C* SPS2SPH U Seconds/second to seconds/hour
C* /ESTERR/ ACNBER U Accelerometer null bias estimation error
C* ASSPER U Airspeed scale factor estimation error
C* CKBRER U GPS clock bias estimation error, nanosec
C* CLKBER U GPS clock bias rate estimation error
C* DOM(3) U Tilt estimation error
C* DX(3) U Position estimation error
C* DXD(3) U Velocity estimation error
C* DNTER(3) U Doppler antenna misalignment est. error
C* GDRER(3) U Gyro drift rate estimation error
C* WINDER(2) U Wind velocity estimation error
C* /KALM/ UD(253) U Upper-loaded covariance matrix
C* NST U Total number of states
C* /MESCOM/ MINDEX U Measurement queue index
C* MTYPE(10) U Measurement type code
C* YRES(10) U Measurement residuals
C* /OUTCOM/ FILNM U Input filename. Character*40
C* NB U Number of bytes in FILNM
C* TITLE U First line of input file
C* DATBF U Date of run
C* TIMBF U Time of run
C* MCPASS U Monte Carlo pass number
C* /TIMCOM/ MTIME U Time counter at filter rate
C* /TRUTHC/ A(3) U True specific force vector, g's
C* R(3) U True position vector, feet
C* V(3) U True velocity vector, feet/sec
C* /VISCOM/ ISAT U GPS satellite index
C* GDOPC U Computed geometric dilution of precision
C*
C* Called By : NAVSIM
C* FILSIM
C* KALFIL
C* MONTEC
C* READINP
C*
C* Modules Referenced : UD2SIG

```

SUBROUTINE PHIX (X,M,H,MH,IFLAG)

```

C*****
C*
C* PHIX(X,M,H,MH,IFLAG) - Linear time-update routine multiplies
C*                        vectors by the state transition matrix
C*
C* Intermodule Data :
C*      X                S/U  Vector to be multiplied
C*      M                U    Identifying array for upper-loaded H
C*      H                S    Upper-loaded sensitivity vector
C*      MH               S    Output-identifying array for H*Phi
C*      IFLAG           U    If IFLAG = 0  X(t) = PHI*X(o)
C*                        If IFLAG = 1  H(o) = H(t)*PHI
C*      /STATEC/       NSC(21) U    If NSC(I) = 1  Include the Ith state
C*                        If NSC(I) = 0  Exclude the Ith state
C*      NS(21)         U    Array of indices of included states
C*      NT              U    Total number of states included
C*      /TRANS/       AP(100)  U    Transition matrix elements (non-diagonal)
C*      NAP(100)      S    State index for the Jth non-zero element
C*                        of the transition matrix
C*      NRAP(21)      S    Number of non-zero elements in the Ith
C*                        row of the transition matrix
C*      NBEG(21)      S    Index of the first non-zero element in
C*                        the Ith row of the transition matrix
C*      ORDER          S    Highest order for transition-matrix
C*                        updating in subroutine PHIX
C*
C* Called By : TIMEUP
C*
C* Modules Referenced : NONE
C*
C*****

```

SUBROUTINE POSFIX

```
C*****
C*
C*
C* POSFIX - Measurement for position fix (MTYPE=8,9)
C*
C* Intermodule Data :
C*
C* /MESCOM/  MINDEX      S/U  Measurement queue index
C*          H(22,10)    S    Measurement sensitivity vector
C*          M(22,10)    S    Identifying array for upper-loaded H
C*          MSTIM(10)   S    Time (counter) of the measurement
C*          MTYPE(10)  S    Measurement type (8,9 for position fix)
C*          SIGMES(10) S    Measurement weights
C*          YRES(10)   S    Measurement residual vector
C* /MESWTS/  SIGM(10)   U    Measurement weights for each meas. type
C* /STATCM/  X(3)       U    True position vector, feet
C* /TRUTHC/  R(3)       U    Estimated position vector, feet
C*
C* Called By : MEASIM
C*
C* Modules Referenced : BARN1
C*
C*****
```


SUBROUTINE READINP

```

C*****
C*
C* READINP - Reads input data file and copies data to
C*           appropriate common arrays
C*
C* Intermodule Data :
C*   /AUPCOM/  TAUGYR(3)  S/U  Time constants for gyro noise
C*            TAUACB    S/U  Time constants for accel. noise
C*   /CONST/  GO        S    Gravity at sea level, ft/sec/sec
C*            D2R       S    Degrees to radians conversion
C*            R2D       S    Radians to degrees conversion
C*            FTPKM     S    Feet per kilometer
C*            C         S    Speed of light, feet/nanosec
C*            FPS22UG   S/U  Feet/sec/sec to micro g's
C*            R2ASEC    S/U  Radians to arcseconds
C*            RAT2PCT   S/U  Ratio to percent
C*            RPS2DPH   S/U  Radians/sec to degrees/hour
C*            SPS2SPH   S/U  Seconds/second to seconds/hour
C*   /ERRCOM/  ERRSTATEIC(21) S/U Initial input error conditions
C*            ERGGYR(3)  S    Gyro scale, bias, random errors
C*            ERRACC(3)  S    Accel. scale, bias, random errors
C*            ICERR     S/U  Error treatment flag (const,random)
C*            IGRAND    S    Gyro noise flag, 0 for no noise
C*            IARAND    S    Accelerometer noise flag
C*   /KALMIN/  BADPCT(10) S/U Input scale percentage of bad meas.
C*            BIASM(10) S/U Input measurement biases
C*            GAUSSM(10) S/U Input random measurement noise scale
C*            PDIAGO(21) S/U Initial state uncertainties
C*            QDIAGO(6)  S/U Process noise uncertainties
C*            SCREEN_LIMIT(10) S/U Bad measurement threshold
C*            USIGM(10) S/U Measurement weights
C*            AMIN(10)  S/U Minimum measurement for processing
C*   /MESKED/  MESSTART(10) S/U Measurement start times
C*            MESINT(10) S/U Measurement interval
C*            MESSTOP(10) S/U Measurement stop time
C*   /ORBITS/  NORBS     S/U Number of satellite orbital planes
C*            OINC      S/U Orbital inclination, degrees
C*            ANOMLY(6,6) S/U Mean anomalies of each satellite
C*            ASCNOD(6)  S/U Ascending node of each orbit
C*            NSPORB(6)  S/U Number of satellites per orbit
C*            EFEMER(3)  S    GPS ephemeris errors
C*                                     radial, cross-track, along-track
C*            RORB      S    Orbital radius for GPS, km
C*            ORBITAL_RATE S Angular rate, rad/sec
C*            UM        S    Gravitational const, km**3/sec**2

```

READINP (continued)

```

C*   /OUTCOM/  DATEBF      S/U  Date of this run. Character*9
C*           TIMBF       S/U  Time of this run. Character*8
C*           FILNM      S/U  Input filename. Character*40
C*           NB         S/U  Number of bytes in FILNM
C*           TITLE     S/U  First line of input file
C*   /PRTCOM/  PRTT       S      Extra-output flag for time update
C*           PRMT       S      Extra-output flag for meas. update
C*   /REFCOM/  GHAO      S/U  Initial Greenwich Hour Angle
C*           RAZ        S/U  Runway azimuth (deg CW from North)
C*           VLAT       S/U  Runway latitude, deg
C*           VLON       S/U  Runway longitude, deg
C*   /STATEC/  NSC(21)   S/U  States-considered flags
C*           NS(21)    S/U  Upper-loaded s-c flags
C*   /TIMCOM/  NTMAX     S      Stop-time counter
C*   /TIMEIN/  TAL1      S/U  Time to start alignment
C*           TAL2      S/U  Time to stop alignment
C*           TFL1      S/U  Time to start filter
C*           TSSTOP    S/U  Stop time for the run
C*           MNTCRL    S/U  Number of Monte Carlo passes
C*   /VISCOM/  DESIRED_GDOP S/U  Desired value of GDOP
C*           ELMIN     S/U  Minimum elevation for GPS obs.
C*           NSEC_CHNG_SATS S/U  Time interval for fixed
C*                                     4-satellite configuration

```

C* Called By : NAVSIM

C* Subroutines Referenced : BARN1
CNVLIN
OUTOPEN
TIMEP

C*****

SUBROUTINE SEESAT

```

C* *****
C*
C* SEESAT - Shows coverage for a stationary aircraft
C*
C* Intermodule Data :
C*   /CONST/   FTPKM      U      Feet per kilometer
C*             R2D        U      Radians to degrees conversion
C*   /EROTIC/  WE         U      Earth's rate, rad/sec
C*             REV(3)     U      Earth's radius vector, feet
C*   /ORBITS/  NORBS      U      Number of orbital planes
C*             OINC       U      Orbital inclination, degrees
C*             ANOMLY(6,6) U      Mean anomalies, degrees
C*             ASCNOD(6)  U      Ascending nodes of each plane
C*             NSPORB(6)  U      Number of satellites per orbit
C*             RORB       S/U    Orbital radius, km
C*             ORBITAL_RATE S/U  Angular rate, rad/sec
C*             UM         S/U    Grav. const., km**3/sec**2
C*   /TIMEIN/  TSSTOP     U      Stop time for the run
C*   /VISCOM/  DESIRED_GDOP U      Desired value for GDOP
C*             ELMIN      U      Minimum elevation for observing
C*             ISAT       S      Current satellite index
C*             NSEC_CHNG_SATS U    Time interval with same 4 sats.
C*             LSATS(4,20) S/U    Array of visible satellites
C*
C* Called By : NAVSIM
C*
C* Subroutines Referenced : CROSS
C*                          DOT
C*                          GDOP
C*                          VNORM
C* *****

```

SUBROUTINE TAFCOS

```

C*****
C*
C* TAFCOS - Generates helicopter's "actual" trajectory,
C* forces, body rates and attitude
C*
C* Entry TAFINIT - Initialization of initial conditions
C*
C* Intermodule data:
C* /EROTIC/ WEV(3) U Earth's rotation rate vector,
C* radians/second
C* RE U Earth's radius in feet
C* /OUTCOM/ FILTRJ U Trajectory-file name (characters)
C* NTRJ U Number of characters in FILTRJ
C* /TIMCOM/ NTIME S/U Time counter, integration rate
C* NTMAX U Stop time in counts
C* DT S/U Integration step, seconds (DTP)
C* /TRUTHC/ R(3) S/U True position vector, feet (ER)
C* V(3) S/U True velocity, ft/sec (EV)
C* A(3) S/U True specific force vector,
C* in g's (FAB)
C* ETB S/U (3,3) Earth-to-body transformation
C* matrix (direction cosines)
C* W(3) S/U Angular velocity vector, rad/sec
C* (OMAA)
C* WD(3) S/U Angular acceleration vector,
C* rad/sec/sec (DOMAA)
C* TVW(3) S/U True wind velocity, ft/sec (WIND8)
C*
C* Called by: NAVSIM
C* INTRUE (initialization, TAFINIT entry)
C*
C* Modules referenced: CNVLIN
C* FILSIM
C* MVTRN
C* TWIST
C*****

```

SUBROUTINE TIMEUP

```
C*****
C*
C*
C* TIMEUP - Does a linear time update for the estimated state and UD
C*           matrix. The result is stored in the W matrix, then WGS is
C*           called to retriangularize the W matrix into the UD matrix
C*
C* Intermodule Data :
C*   /DELCOM/  DELX(21)  S/U  Incremental state estimates
C*   /KALM/    NST       U    Number of states total
C*           NFF       U    Number of forcing functions
C*           IMAXWM    U    Maximum dimension of WM matrix
C*           UD(253)   S/U   UD covariance matrix (real*8)
C*           WM(21,27) S/U   Rectangular UD extended (real*8)
C*           GAMMA(21,6) U    Forcing-function sensitivity matrix
C*           QD(6)     U    Forcing-function variances
C*   /MESCOM/  H(22,10)  S/U   Measurement sensitivities
C*           M(22,10)  U    Identifying array for upper-loaded H
C*           MH(22,10) S/U   Post-multiplication M array
C*   /PRTCOM/  PRTT     U    Extra-output flag for time-update
C*   /STATEC/  NT       S/U   Number of states
C*   /TIMCOM/  MTIME    U    Time counter at filter rate
C*
C* Called By : KALFIL
C*
C* Modules Referenced : APFILL
C*                   PHIX
C*                   TRIMAT
C*                   WGS
C*****
```

C-2

SUBROUTINE TRIMAT(A,N,CAR,TEXT,NCHAR,NAMES)

```
C*****
C*
C* TRIMAT(A,N,CAR,TEXT, NCHAR,NAMES) - To display a vector-stored
C*   upper-triangular matrix in a two-dimensional triangular
C*   format
C*
C* Intermodular Data :
C*   A(N*(N+1)/2)  U   Vector containing upper triangular
C*                   matrix
C*   N              U   Dimension of matrix
C*   CAR(N)        U   Parameter names
C*   TEXT( )      U   An array of field data characters to be
C*                   printed as a title preceding the matrix
C*   NCHAR        U   Number of chars. to be printed in TEXT( ).
C*                   ABS(NCHAR).LE.126. -NCHAR is used
C*                   to avoid skipping to a new page to print
C*   NAMES        U   .TRUE. to print parameter names
C*
C* Called By : MEASUP
C*            TIMEUP
C*
C* Modules Referenced : NONE
C*
C*****
```

SUBROUTINE TRUSAT

```
C*****
C*
C*
C* TRUSAT - Initializes "true" GPS satellite orbits as deviations
C*       from input circular orbits
C*
C* Intermodule Data :
C*   /ORBITS/   NORBS      U   Number of orbital planes
C*              OINC      U   Orbital inclination, degrees
C*              ANOMLY(6,6) U   Initial mean anomalies, degrees
C*              ASCNOD(6)  U   Ascending nodes, degrees
C*              NSPORB(6)  U   Number of satellites per orbit
C*              EFEMER(3)  U   Ephemeris errors, radial, cross,
C*                               along (ft)
C*              RCA(3,6,6) S   Randomized ephemeris errors by
C*                               satellite
C*              UM        U   Universal gravitational const.,
C*                               km**3/sec**2
C*   /TRUORB/   CB(20,24)  S/U
C*              RSO(3,24)  S/U
C*              VSO(3,24)  S/U
C*
C* Called By :   INTRUE
C*
C* Modules Referenced : ORB2X
C*                       STEPD
C*                       X2ORB
C*****
```

SUBROUTINE UD2SIG(U,N,SIG,TEXT,NCT)

```
C*****
C*
C* UD2SIG(U,N,SIG,TEXT,NCT) - Compute standard deviations
C*                          (sigmas) from U-D covariance
C*
C* Intermodule Data :
C*   U(N*(N+1)/2)  U   Input vector stored array containing
C*                  the U-D factors.  The D (diagonal)
C*                  elements are stored on the diagonal
C*   N              U   Dimension of U and SIG
C*   SIG(N)         S/U  Vector of output standard deviations
C*   TEXT( )        U   Array of characters to be printed
C*                  above the array of sigmas
C*   NCT            U   Number of characters in text.
C*                  0 .LE. NCT .LE. 126
C*                  If NCT = 0, no sigmas are printed
C*
C* Called By : KALFIL
C*
C* Modules Referenced : NONE
C*
C*****
```


SUBROUTINE UDMES (U,N,R,A,G,ALPHA,SCRLIM)

COMPUTES ESTIMATE AND U-D MEASUREMENT UPDATED
COVARIANCE, $P = UDU^{**T}$

*** INPUTS ***

U UPPER TRIANGULAR MATRIX, WITH D ELEMENTS STORED
AS THE DIAGONAL. U IS VECTOR STORED AND
CORRESPONDS TO THE PRIORI COVARIANCE. IF
STATE ESTIMATES ARE COMPUTED, THE LAST COLUMN
OF U CONTAINS X.

N DIMENSION OF THE STATE ESTIMATES.

R MEASUREMENT VARIANCE

A VECTOR OF MEASUREMENT COEFFICIENTS, IF DATA
THEN $A(N+1) = Z$

ALPHA IF ALPHA LESS THAN ZERO NO ESTIMATES ARE
COMPUTED (AND X AND Z NEED NOT BE INCLUDED)

*** OUTPUTS ***

U UPDATED, VECTOR STORED FACTORS AND ESTIMATE
AND $U((N+1)(N+2)/2)$ CONTAINS $(Z-A^{**T}X)/ALPHA$

ALPHA INNOVATIONS VARIANCE OF THE MEASUREMENT
RESIDUAL

G VECTOR OF UNWEIGHTED KALMAN GAINS, $K=G/ALPHA$

A CONTAINS $U^{**T}A$ AND $(Z-A^{**T}X)/ALPHA$

Called by : MEASUP

Modules Referenced : NONE

```

SUBROUTINE WGS (W,IMAXW,IW,JW,D,U,V)
C*****
C*
C* WGS(W,IMAXW,IW,JW,D,U,V) - Modified Gramm-Schmidt algorithm for
C*   reducing WDW(**T) to UDU(**T) form where U is a vector
C*   stored triangular matrix with the resulting D elements
C*   stored on the diagonal
C*
C* Intermodule Data :
C*   W(IW,JW)      S/U  Input matrix to be reduced to triangular
C*                   format
C*   IMAXW         U    Maximum dimension of W
C*   IW,JW         U    Actual dimension of W
C*   D(IW)         U    Vector of non-negative weights for the
C*                   orthogonalization process. The D's are
C*                   unchanged by the calculation
C*   U(IW*(IW+1)/2) S/U  Output upper-triangular vector-stored
C*                   matrix
C*   V(JW)         S/U  Work vector
C*
C* Called By : TIMEUP
C*
C* Modules Referenced : NONE
C*
C*****

```

Appendix E

COMMONS DESCRIPTIONS

This appendix contains a description of each of the commons used by NAVSIM II. The descriptions appear in alphabetical order. Included in each description are:

1. a brief statement of the common's primary purpose,
2. a list of the parameters which comprise the common, with mnemonic, dimension, definition and units,
3. a set/used table which shows which subroutines access which parameters from the common.

ACTUAL Labeled Common

ACTUAL - Simulated truths in sensor models

ACTUAL_GDR(3)	True gyro drift rate, rad/sec
ACTUAL_ACNB(3)	True accelerometer null bias, ft/sec/sec
ACTUAL_GSF(3)	True gyro scale factor error, unitless
ACTUAL_ASF(3)	True accelerometer scale factor error, unitless
ACTUAL_CLKB	True GPS receiver clock bias, nano-sec
ACTUAL_CKBR	True clock bias rate, nano-sec
ACTUAL_DNT(3)	True Doppler antenna mounting, radians
ACTUAL_ASSF	True airspeed scale factor

Accessing

Subroutines :	AIRSPD	ESTMERR	GPS	INESTM	INTRUE	NAVEQ
ACTUAL_GDR(3)	-	U	-	U	S	U
ACTUAL_ACNB(3)	-	U	-	U	S	U
ACTUAL_GSF(3)	-	-	-	-	S	U
ACTUAL_ASF(3)	-	-	-	-	S	U
ACTUAL_CLKB	-	U	U	-	S	-
ACTUAL_CKBR	-	U	-	-	S	-
ACTUAL_DNT(3)	-	U	-	-	S	-
ACTUAL_ASSF	U	U	-	-	S	-

Mathematical Symbols

$\dot{\alpha}_d$	ACTUAL_GDR
f_b	ACTUAL_ACNB
s_c	ACTUAL_GSF
s_f	ACTUAL_ASF
\bar{c}_b	ACTUAL_CLKB
$\dot{\bar{c}}_b$	ACTUAL_CKBR
$\bar{\alpha}_d$	ACTUAL_DNT
$s_{a/s}$	ACTUAL_ASSF

AUPCOM Labeled Common

AUPCOM - Time constants for inertial-sensor process noise

TAUGYR(3) Array of time constants for gyro noise, sec
TAUACB Time constant for accelerometer noise, sec

Accessing

Subroutines : AUPT FILNIT READINP

TAUGYR	U	U	S/U
TAUACB	U	U	S/U

Mathematical Symbols

τ_{ω}	TAUGYR
τ_{ϕ}	TAUACB

CONST Labeled Common

CONST - Global constants

GO Gravitational acceleration at sea level, ft/sec**2
D2R Degree to radian conversion factor
R2D Radian to degree conversion factor
FTPKM Feet per kilometer
C Speed of light, ft/nanosec
SPS2SPH Seconds per second to seconds per hour
FPS22UG Feet per second squared to micro G's
RPS2DPH Radians per second to degrees per hour
R2ASEC Radian to arc seconds
RAT2PCT Ratio to percent

Accessing
Subroutines : AUPT DOPPLER FILNIT GPS INEREF INESTM NAVEQ

GO	U	-	U	-	-	-	U
D2R	-	U	U	-	-	-	-
R2D	-	-	-	U	U	-	-
FTPKM	-	-	-	U	-	-	-
C	-	-	-	U	-	-	-
SPS2SPH	-	-	S	-	-	-	-
FPS22UG	-	-	S	-	-	-	-
RPS2DPH	-	-	S	-	-	-	-
R2ASEC	-	-	S	-	-	U	-
RAT2PCT	-	-	S	-	-	-	-

OUTPUT READINP SEESAT

GO	-	S	-
D2R	-	S	-
R2D	-	S	U
FTPKM	-	S	U
C	-	S	-
SPS2SPH	U	S/U	-
FPS22UG	U	S/U	-
RPS2DPH	U	S/U	-
R2ASEC	U	S/U	-
RAT2PCT	U	S/U	-

Mathematical Symbols

g_o GO
c₁ C

DELCOM Labeled Common

DELCOM - Incremental state estimates

DELX(21) Incremental state estimates

Accessing

Subroutines : CHNGX FILNIT MEASUP TIMEUP

DELX S/U S S/U S/U

Mathematical Symbols

\hat{z} , \hat{x} DELX

EROTIC Labeled Common

EROTIC - Earth's rotational rate, radius and inertial transformation

WEV(3) Earth's rotation rate vector in Earth-fixed coordinates, radians/second
 WE Earth's rotation rate in radians/second
 REV(3) Earth's radius vector in Earth-fixed co-ordinates, feet
 RE Earth's radius in feet
 TI2CO(3,3) Inertial transformation from Earth's equator-of-date to initial Earth-fixed runway coordinates

Accessing

Subroutines	ALTMTR	AUPT	GPS	INEREF	NAVEQ	SEESAT	TAF COS
WEV	-	U	-	S	U	-	U
WE	-	-	U	S/U	-	U	-
REV	-	-	-	S	-	U	-
RE	U	U	U	S/U	U	-	U
TI2CO(3,3)	-	-	U	S	-	-	-

Mathematical Symbols

a_e WEV
 R REV
 r_e RE
 T_c^i TI2CO

ERRCOM Labeled Common

ERRCOM - Error state initial conditions

ERRSTATEIC(21)	Initial input error conditions
ERRGYR(3)	Gyro scale factor, bias, random errors, deg/hr
ERRACC(3)	Accelerometer scale factor, bias, random errors, ft/sec**2
ICERR	Error treatment flag, 0 for constant, 1 random
IGRAND	Gyro noise flag, 0 for no noise
IARAND	Accelerometer noise flag, 0 for no noise

Accessing

Subroutines : INESTM INTRU NAVEQ READINP

ERRSTATEIC(21)	U	U	-	S/U
ERRGYR(3)	-	U	U	S
ERRACC(3)	-	U	U	S
ICERR	U	U	-	S/U
IGRAND	-	-	U	S
IARAND	-	-	U	S

Mathematical Symbols

$\tilde{z}(0)$	ERRSTATEIC
\tilde{s}_ω	ERRGYR(1)
$\tilde{\omega}_{bias}$	ERRGYR(2)
$\sigma_{\omega noise}$	ERRGYR(3)
\tilde{s}_f	ERRACB(1)
\tilde{f}_{bias}	ERRACB(2)
$\sigma_{f noise}$	ERRACB(3)

ESTERR Labeled Common

ESTERR - Estimation errors

DX(3)	Position estimation error in feet
DXD(3)	Velocity estimation error in radians
DOM(3)	Tilt estimation error in radians
GDRER(3)	Error in gyro drift rate estimates, deg/hr
ACNBER	Error in accelerometer null bias estimates, ft/s ²
CLKBER	Error in clock bias estimate, nanosec
CKBRER	Error in clock bias rate estimate, nanosec per sec
DNTER(3)	Error in Doppler antenna mounting estimate, deg
ASSFER	Error in airspeed scale factor estimate, unitless
WINDER(2)	Error in wind estimate, feet per sec

Subroutines : ESTMERR OUTPUT

DX	S	U
DXD	S	U
DOM	S	U
GDRER	S	U
ACNBER	S	U
CLKBER	S	U
CKBRER	S	U
DNTER	S	U
ASSFER	S	U
WINDER	S	U

Mathematical Symbols

\tilde{x}	DX		
${}^n\tilde{v}$	DXD		
$\tilde{\theta}$	DOM	$\tilde{s}_{a/s}$	ASSFER
$\tilde{\omega}_d$	GDRER	\tilde{w}_v	WINDER
\tilde{f}_b	ACNBER		
\tilde{c}_b	CLKBER		
\tilde{c}_b	CKBRER		
\tilde{Q}_d	DNTER		

GRVCOM Labeled Common

GRVCOM - Estimated gravity vector in earth-fixed coordinates

GVEC(3) Gravity vector, ft/sec**2

Accessing

Subroutines : AUPT NAVEQ

GVEC(3) U S/U

Mathematical Symbols

\hat{g}_r GVEC

KALM Labeled Common

KALM - Kalman Filter covariance parameters

NST	Number of states total
NFF	Number of forcing functions
IMAXWM	Dimension of WM
UD(253)	U - D covariance matrix - upper stored (Real*8)
WM(21,27)	Time-updated UD matrix with process noise (Real*8)
GAMMA(21,6)	Forcing function sensitivity matrix
QD(6)	Process noise covariance

Subroutines : FILNIT KALFIL MEASUP OUTPUT TIMEUP

NST	S/U	U	U	U	U
NFF	S	-	-	U	-
IMAXWM	S	-	-	-	U
UD	S/U	U	S/U	S/U	U
WM	-	-	-	-	S/U
GAMMA	S	-	-	-	U
QD	S	-	-	-	U

Mathematical Symbols

U,D	UD
W	QD
Φ_n	GAMMA

KALMIN Labeled Common

KALMIN - Kalman filter input parameters

USIGM(10) Input value of measurement weights
PDIAGO(21) Initial state uncertainties
QDIAGO(6) Process noise uncertainties
BIASM(10) Input value of measurement bias
GAUSSM(10) Input measurement value of random noise
BADPCT(10) Input value of measurement points with
 bad values
SCREEN_LIMIT(10) Input value of screening for bad
 measurement data
AMIN(10) Input value of minimum value for taking
 measurements (e.g. airspeed > 30 Kts)

NOTE : For USIGM, BIASM, GAUSSM, BADPCT, SCREEN_LIMIT and AMIN

If I = 1 Measurement is Baro-altimeter
If I = 2 " " Radio-altimeter
If I = 3 " " GPS
If I = 4 " " Doppler
If I = 5 " " Airspeed
If I = 6 " " Position fix

Accessing

Subroutines : FILNIT READINP

Mathematical Symbols

USIGM	U	S/U	R	USIGM
PDIAGO	U	S/U		
QDIAGO	U	S/U	D_0	PDIAGO
BIASM	U	S/U		
GAUSSM	U	S/U	W	QDIAGO
BADPCT	U	S/U		
SCREEN_LIMIT	U	S/U	B	BIASM
AMIN	U	S/U	N	GAUSSM
			K	SCREEN_LIMIT
			y_{min}	AMIN

MESCOM Labeled Common

MESCOM - Measurement-processing parameters (queue for the filter)

MTYPE(10) Measurement type code
 MTYPE = 1 : Baro-altimeter
 MTYPE = 2 : Radio-altimeter
 MTYPE = 3 : GPS pseudo range
 MTYPE = 4,5,6 : Doppler velocity measurement
 MTYPE = 7 : Airspeed measurement
 MTYPE = 8,9 : Position-fix x,y
 H(22,10) Measurement sensitivity vector - upper loaded
 MH(22,10) Identifying array for H (output from PHIX)
 M(22,10) Identifying array for H (input to PHIX)
 YRES(10) Measurement residual array
 SIGMES(10) Measurement weight
 MSTIM(10) Time of measurement
 MINDEX Measurement queue index

Accessing

Subroutines :	AIRSPEED	ALTMTR	DOPPLER	FILNIT	GPS	KALFIL	MEASUP
MTYPE	S	S	S	-	S	U	-
H	S/U	S	S/U	-	S/U	-	U
MH	-	-	-	-	-	-	-
M	S	S	S	-	S	-	U
YRES	S	S	S/U	-	S/U	-	U
SIGMES	S	S	S/U	-	S	-	U
MSTIM	S	S	S	-	S	-	-
MINDEX	S/U	S/U	S/U	S	S/U	S/U	-

	OUTPUT	POSFIX	TIMEUP	Mathematical Symbols	
MTYPE	U	S	-		
H	-	S	S/U	H	H
MH	-	-	S/U		
M	-	S	U		
YRES	U	S	-	y	YRES
SIGMES	-	S	-		
MSTIM	-	S	-	r	SIGMES
MINDEX	U	S/U	-	t_{meas}	MSTIM

MESKED Labeled Common

MESKED - Measurement schedule parameters

MESSTART(10) Measurement start time
 MESINT(10) Measurement interval
 MESSTOP(10) Measurement stop time
 MESTIME(10) Current measurement time for each measurement type :
 MESTIME(1) = Next baro-altimeter measurement time
 MESTIME(2) = " radio-altimeter " "
 MESTIME(3) = " GPS " "
 MESTIME(4) = " Doppler " "
 MESTIME(5) = " Airspeed " "
 MESTIME(6) = " Position-fix " "

Accessing

Subroutines : MEASIM MONTEC READINP

MESSTART	-	U	S/U
MESINT	U	-	S/U
MESSTOP	U	-	S/U
MESTIME	S/U	S	-

Mathematical Symbols

t_{mi}	MESSTART
δt_m	MESINT
t_{mf}	MESSTOP
t_m	MESTIME

MESWTS Labeled Common

MESWTS - Measurement weights and simulated errors

SIGM(10) Measurement weights
 PBIAS(10) Bias of measurement
 PGAUSS(10) Scale for random measurement noise
 PBADPCT(10) Percentage of points with very bad values
 PSCRNLMT(10) Limit for screening out bad data
 PMIN(10) Minimum of acceptable measurements

NOTE : If I = 1 Measurement is Baro-altimeter
 If I = 2 " " Radio-altimeter
 If I = 3 " " GPS
 If I = 4,5,6 " " Doppler
 If I = 7 " " Airspeed
 If I = 8,9 " " Position fix

Accessing Subroutines :	AIRSPEED	ALTMTR	DOPPLER	FILNIT	GPS	POSFIX
SIGM	U	U	U	S	U	U
PBIAS	U	U	U	S	U	-
PGAUSS	U	U	U	S	U	-
PBADPCT	-	-	-	S	-	-
PSCRNLMT	-	-	-	S	-	-
PMIN	U	U	-	S	-	-

Mathematical Symbols

R SIGM
 B PBIAS
 N PGAUSS
 K PSCRNLMT
 y_{min} PMIN

ORBITS Labeled Common

ORBITS - GPS satellite orbit data

NORBS	The number of satellite orbital planes
NSPORB(6)	Number of satellites per orbit
NSATS	Total number of satellites
OINC	Orbital inclination, deg
ASCNOD(6)	Ascending nodes, deg
ANOMLY(6,6)	Mean anomalies, deg
RORB	Orbital radius for circular GPS orbits, km
ORBITAL_RATE	Angular rate for GPS satellite, rad/sec
UM	Gravitational constant, km**3/sec**2
EFEMER(3)	GPS ephemeris errors, ft radial, cross-, along-track
RCA(3,6,6)	GPS ephemeris errors by satellite, ft

Subroutines :	GPS	READINP	SEESAT	TRUSAT
NORBS	-	S/U	U	U
NSPORB	-	S/U	U	U
NSATS	-	-	-	-
OINC	U	S/U	U	U
ASCNOD	U	S/U	U	U
ANOMLY	U	S/U	U	U
RORB	U	S	U	-
ORBITAL_RATE	U	S	U	-
UM	U	S	U	U
EFEMER(3)	-	S	-	U
RCA(3,6,6)	U	-	-	S

Mathematical Symbols

i	OINC
Ω	ASCNOD
ω	ANOMLY
M	ORBITAL_RATE
μ	UM
e	EFEMER, RCA

OUTCOM Labeled Common

OUTCOM - Output file descriptors

FILNM	Output filename is FILNM plus file type
NB	Number of characters in FILNM
TITLE	Title of output file
DATBF	Date of current run
TIMBF	Time of current run
MCPASS	Monte Carlo pass number
FILTRJ	Trajectory filename
NTRJ	Number of characters in FILTRJ

Accessing

Subroutines : NAVSIM OUTPUT READINP TAF COS

FILNM	-	U	S/U	-
NB	-	U	S/U	-
TITLE	-	U	S/U	-
DATBF	-	U	S/U	-
TIMBF	-	U	S/U	-
MCPASS	S	U	-	-
FILTRJ	-	-	S	U
NTRJ	-	-	S	U

PRTCOM Labeled Common

PRTCOM - Extra-output print flags

PRTT	Time update
PRTM	Measurement update

Accessing
Subroutines : MEASUP READINP TIMEUP

PRTT	-	S	U
PRTM	U	S	-

REFCOM Labeled Common

REFCOM - Earth reference parameters (constants for a run)

GHAO	Initial Greenwich hour angle, deg
VLAT	Runway reference latitude - North, deg
VLON	Runway reference longitude - East, deg
RAZ	Runway azimuth clockwise from North, deg

Accessing
Subroutines : INEREF READINP

GHAO	U	S/U
VLAT	U	S/U
VLON	U	S/U
RAZ	U	S/U

Mathematical Symbols

γ_0	GHAO
L_r	VLAT
λ_r	VLON
ψ_r	RAZ

STATCM Labeled Common

STATCM - Current estimates of state

X(3)	Position vector, feet
VN(3)	Velocity vector, ft/sec
TILTS(3)	Tilt vector, radians
GDRE(3)	Gyro drift rate estimate, deg/hr
ACNBE	Accelerometer null bias estimate, ft/sec**2
CLKBE	Clock bias estimate, nanosec
CKBRE	Clock bias rate estimate, nanosec/sec
DNTE(3)	Doppler antenna mounting estimate, deg
ASSFE	Airspeed scale factor estimate
WINDE(2)	Wind estimate, ft per second
TEB(3,3)	Estimated Earth to body transformation

Accessing

Subroutines : AIRSPEED ALTMTR AUPT CHNGX DOPPLER ESTMERR GPS

X	-	U	U	S/U	-	U	U
VN	U	-	U	S/U	U	U	-
TILTS	-	-	-	S/U	-	U	-
GDRE	-	-	-	-	-	U	-
ACNBE	-	-	-	-	-	U	-
CLKBE	-	-	-	S/U	-	U	U
CKBRE	-	-	-	-	-	U	-
DNTE	-	-	-	-	U	U	-
ASSFE	U	-	-	S/U	-	U	-
WINDE	U	-	-	S/U	-	U	-
TEB	-	-	U	-	U	-	-

	INESTM	NAVEQ	POSFIX
X	S	S/U	U
VN	S	S/U	-
TILTS	S	S/U	-
GDRE	S	U	-
ACNBE	S	-	-
CLKBE	S	-	-
CKBRE	S	-	-
DNTE	S	-	-
ASSFE	S	-	-
WINDE	S	-	-
TEB	-	S/U	-

Mathematical Symbols

See ESTERR for X - WINDE

b^T TEB

STATEC Labeled Common

STATEC - States-considered flags

NSC(21) If NSC(I) = 1 : Include Ith state
 If NSC(I) = 0 : Exclude " "
NS(21) Upper loaded array of indices of included states and FF'
NT Total number of states included

Accessing

Subroutines : PHIX READINP TIMEUP

NSC	U	S/U	-
NS	U	S/U	-
NT	U	-	S/U

TIMCOM Labeled Common

TIMCOM - Time counters for program execution

NTIME Time counter at integration rate
 MTIME Time counter at filter rate
 NTMAX Stop time * 20
 DT Integration step size, sec

Accessing

Subroutines : AIRSPEED ALTMTR AUPT DOPPLER FILNIT FILSIM GPS KALFIL

NTIME	-	-	-	-	-	U	-	-
MTIME	U	U	-	U	-	-	U	U
NTMAX	-	-	-	-	-	-	-	-
DT	-	-	U	-	U	-	-	-

	MEASIM	MEASUP	MONTEC	NAVEQ	OUTPUT	READINGP	TIMEUP	TAF COS
NTIME	-	-	S	U	-	-	-	S/U
MTIME	U	U	S	S	U	-	U	-
NTMAX	-	-	-	-	-	S	-	U
DT	-	-	-	U	-	-	-	S/U

TIMEIN Labeled Common

TIMEIN - Case-control times

TAL1	Time to start alignment
TAL2	Time to stop alignment
TFL1	Time to start filter
TSSTOP	Stop time
MNTCRL	Number of Monte Carlo passes

Accessing

Subroutines : NAVSIM READINP SEESAT

TAL1	-	S/U	-
TAL2	-	S/U	-
TFL1	-	S/U	-
TSSTOP	-	S/U	U
MNTCRL	U	S/U	-

TRANS Labeled Common

TRANS - Transition matrix parameters

AP(100) Integral of AK, transition matrix elements (see APFILL)
 AK(100) Sum of the non-zero Elements of the transition matrix
 AM(100) Non-zero Elements of the transition matrix
 NAP(100) State index for the Jth non-zero element of the Ith row
 of the transition matrix
 NRAP(21) Number of non-zero elements in the Ith row of the
 transition matrix
 NBEG(21) Index of the first non-zero element in the Ith row
 of the transition matrix
 ORDER Highest order for transition-matrix updating for PHIX

Accessing

Subroutines : APFILL AUPT PHIX

AP	S	S	U
AK	S/U	S/U	-
AM	-	S/U	-
NAP	-	S	U
NRAP	-	S	U
NBEG	-	S	U
ORDER	-	S	U

Mathematical Symbols

Φ	AP
F_z	AM

TRUORB Labeled Common

TRUORB - True GPS satellite orbital parameters

RSO(3,24) Initial position vector for each GPS satellite, feet
VSO(3,24) Initial velocity vector for each GPS satellite, ft/sec
CB(20,24) Orbital parameters for each GPS satellite

Subroutines :	GPS	TRUSAT
RSO	U	S/U
VSO	U	S/U
CB	S/U	S/U

Mathematical Symbols

$R_s(0)$	RSO
$V_s(0)$	VSO
$E(0)$	CB

TRUTHC Labeled Common

TRUTHC - Simulated true (or "actual") aircraft state

R(3) True position vector in Earth-fixed co-ordinates, ft
 V(3) True velocity in Earth-fixed co-ordinates, ft/sec
 A(3) Specific force vector in aircraft-body co-ordinates, g
 ETB(3,3) Earth to body transformation
 W(3) Angular velocity vector, rad/sec
 WD(3) Angular acceleration vector, rad/sec**2
 TVW(3) True wind velocity, feet per second

Accessing

Subroutines : AIRSPEED ALTMTR DOPPLER ESTMERR GPS INESTM

R	-	U	-	U	U	U
V	U	-	U	U	-	U
A	-	-	-	-	-	-
ETB	-	-	U	-	-	-
W	-	-	-	-	-	-
WD	-	-	-	-	-	-
TVW	U	-	-	-	-	-

	NAVEQ	OUTPUT	POSFIX	TAF COS
R	-	U	U	S/U
V	-	U	-	S/U
A	U	U	-	S/U
ETB	U	-	-	S/U
W	U	-	-	S/U
WD	U	-	-	S/U
TVW	-	-	-	S/U

Mathematical Symbols

\bar{x} R
 \bar{v}_n V
 \bar{f} A
 \bar{T}_b^r ETB

Mathematical Symbols

$\bar{\omega}_b$ W
 $\bar{\omega}_b$ WD
 \bar{v}_w TVW

VISCOM Labeled Common

VISCOM - GPS Visibility parameters

NSEC_CHNG_SATS	Time interval with fixed set of 4 satellites
DESIRED_GDOP	Desired value of GDOP
ELMIN	Minimum elevation angle satellite is visible
LSATS(4,20)	Array of visible satellites
ILSAT	Index for ISATS to select ISAT
ISAT	Current satellite number being processed
GDOPC	Calculated value of GDOP

Accessing Subroutines :	GPS	MONTEC	OUTPUT	READINP	SEESAT
NSEC_CHNG_SATS	U	-	-	S/U	U
DESIRED_GDOP	-	-	-	S/U	U
ELMIN	U	-	-	S/U	U
LSATS(4,20)	U	-	-	-	S/U
ILSAT	S/U	S	-	-	-
ISAT	S/U	-	U	-	S
GDOPC	S	-	U	-	-



Report Documentation Page

1. Report No. NASA CR 177438		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle NAVSIM II A Computer Program for Simulating Aided-Inertial Navigation for Aircraft				5. Report Date February 1987	
				6. Performing Organization Code	
7. Author(s) William S. Bjorkman Analytical Mechanics Associates, Inc. Mountain View, CA				8. Performing Organization Report No.	
				10. Work Unit No. 505-66-11	
9. Performing Organization Name and Address NASA Ames Research Center Moffett Field, CA 94035				11. Contract or Grant No. NAS2-10850	
				13. Type of Report and Period Covered Contractor Report Technical	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				14. Sponsoring Agency Code	
				15. Supplementary Notes Point of Contact: L. A. McGee, Ames Research Center, MS 210-9, Moffett Field, CA 94035 - (415)694-5443 or FTS 464-5443	
16. Abstract NAVSIM II, a computer program for analytical simulation of aided-inertial navigation for aircraft, is described. The description is supported by a discussion of the program's application to the design and analysis of aided-inertial navigation systems as well as instructions for utilizing the program and for modifying it to accommodate new models, constraints, algorithms and scenarios. NAVSIM II simulates an airborne inertial navigation system built around a strapped-down inertial measurement unit and aided in its function by GPS, Doppler radar, altimeter, airspeed and position-fix measurements. The measurements are incorporated into the navigation estimate via a UD-form Kalman filter. The simulation was designed and implemented using structured programming techniques and with particular attention to user-friendly operation.					
17. Key Words (Suggested by Author(s)) AIDED-INERTIAL NAVIGATION STRAPPED-DOWN IMU GPS, DOPPLER RADAR KALMAN FILTER (UD-FORM)			18. Distribution Statement Unlimit.ed Subject Category: 04		
19. Security Classif. (of this report) Unclass.		20. Security Classif. (of this page) Unclass.		21. No. of pages	22. Price