(4)

ARL-SYS-TM-101 AR-004 594

AUSTRALIA

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

MELBOURNE, VICTORIA

AD-A215 495

Aircraft Systems Technical Memorandum 101

SINS1 - A MODEL OF A STRAPDOWN INERTIAL NAVIGATION SYSTEM (U)

by

R.B. MILLER

Approved for Public Release

JULY 1989

AR-004-594

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY
MELBOURNE VICTORIA

Aircraft Systems Technical Memorandum 101

SINS1 - A MODEL OF A STRAPDOWN INERTIAL NAVIGATION SYSTEM

by

R.B. MILLER

SUMMARY

SINS1 is a simulation of a strapdown INS, written as a Fortran subroutine to be called at intervals by another program. Sensor errors, system initialization characteristics, and navigation algorithm operation may be selected by the User. Each call corresponds to the passage of a set time interval, and the calling program provides environmental dynamics information about the angular velocity of, and specific force on, the simulated INS. As required by the User, the calling program obtains from SINS1, the INS outputs of position, attitude, velocity, and other data. An example of a calling program is also described.

DSTO ARL MELBOURNE

POSTAL ADDRESS: Director, Aeronautical Research Laboratory,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia

LIST OF CONTENTS

Page

1. INTRODUCTION 1

2 SENSORS SEGMENT 2
- 2.1 Sensor Errors 2

3. NAVIGATOR SEGMENT 3
- 3.1 Split Frame Algorithm 4
- 3.2 Fast Segment Calculations 5
  - 3.2.1 Attitude Updating 5
  - 3.2.2 Body Axes Velocity Accumulation 6
- 3.3 Intermediate Segment Calculations 6
  - 3.3.1 Attitude Updating 6
  - 3.3.2 Velocity Integration 7
  - 3.3.3 Position Update 7
  - 3.3.4 Values for next Intermediate Cycle 8
- 3.4 Slow Segment Calculations 8
- 3.5 One Shot Coarse Alignment 9
- 3.6 Introduction of Tilt Changes During a Run 9
  - 3.6.1 Relative to Body Axes 9
  - 3.6.2 Relative to Geographic Axes 9

4 PROGRAM IMPLEMENTATION 10
- 4.1 Sensors Segment 10
- 4.2 Navigator Segment 11

5 SINS1 OPERATION 11
- 5.1 Initialisation Call 12
- 5.2 In-run Calls 14
- 5.3 Output Returned by SINS1 15
- 5.4 Sensor Characteristics Data File Format 15
- 5.5 Navigator Initialisation Data File Format 17

6 HANDLER PROGRAM EXAMPLE: FPSIN2 18
- 6.1 Program Description 19
- 6.2 Input Files and Initialisation 20
  - 6.2.1 INS Initialisation Data File INSDATA.xxx.yyy 20
- 6.3 Initialisation Call to SINS1 21
- 6.4 Runtime Call to SINS1 21
- 6.5 Output from FPSIN2 22

7 AREAS FOR DEVELOPMENT 22

REFERENCES
TABLES
FIGURES

APPENDICES
1 Notation and Coordinate Frames
2 Direction Cosine and Quaternion Relationships
3 Earth to Geographic to Wander Azimuth Transformation
4 Navigation (G or W) to Body Transformation
5 Frame Rates
6 Gravity Support Correction
7 One Shot Coarse Alignment

DISTRIBUTION
DOCUMENT CONTROL DATA

1. INTRODUCTION

SINS1 is a computer simulation model of a strapdown inertial navigation system (INS). It is written as a Fortran subroutine which is to be called at intervals by some other program, which may be referred to as the handler program. Any simulation of an INS requires a simulated environment for the system, and this arrangement allows the INS simulation to be separated from that of its environment, thus increasing the model's utility. As an example, Section 6 of this document contains a description of a handler program which operates SINS1 using the outputs of a flight profile generator, - a program which simulates an idealised airborne environment.

Each call to SINS1 corresponds to the passage of a set (real world) time interval, and the handler provides environmental dynamics information about the angular velocity of, and specific force on, the simulated INS. At required intervals, the handler obtains from SINS1 the INS outputs such as position, attitude, and velocity.

The simulated INS model has two parts - the sensors segment and the navigator segment. Angular velocity and specific force outputs to SINS1, in nominal INS ("Body") coordinates, are sampled and integrated by the sensors segment, giving "ideal" rate-integrating gyro and incremental-velocity accelerometer outputs. Optionally, SINS1 will accept data as integrated (in Body axes) angular velocity and specific force components. The "ideal" sensor outputs are then corrupted in the sensor segment with sensor characteristics and errors including misalignment, scale factor errors, fixed biases, and quantization.

The navigator segment, which produces the outputs of position, velocity, and attitude, uses a split frame algorithm, partitioned into fast, intermediate, and slow update rate segments, the repetition rates of which may be preset to suit requirements. Quaternions are used for body attitude and angular position relative to Earth, and the navigation axes are wander azimuth/down. Vertical channel navigation may be either pure-inertial or set to follow input altitude. Altitude smoothing is not incorporated. Levelling and alignment of the navigator may be performed either by specifying errors in the initial position, velocity, and attitude, or by using a simple "one shot" coarse alignment routine which uses the (unquantized and stationary) sensor outputs to estimate level and north.

At any time during a simulation run, the calling program may change the sensor error characteristics; it may also make changes ("tilts") to the attitudes of the navigator's analytical axes at any time.

As output, SINS1 returns to the calling program with time tagged attitude, velocity, and position. Attitude is available as Geographic to Body Euler angles, also as Wander to Body quaternion and direction cosine matrix. Position is available as longitude and latitude angles, also as Earth to Wander quaternion and direction cosine matrix. Velocity relative to Earth is available in Wander and Geographic coordinates. See Appendix 1 for coordinate frames and notation.

2. SENSORS SEGMENT

A "sensor cycle" is defined as the process whereby simulated sensor output is generated, updated within SINS1 and made available for sampling by the navigator segment.

Ideally, gyroscope outputs are the integral of angular velocity about the gyroscope sensitive axis, over a set period of time. These have the dimensions of angular increments. Similarly, accelerometer outputs are the integral of specific force along the accelerometer sensitive axis, over the period, and have the dimension of velocity increments. The sensors model is provided with samples of angular velocity and specific force from the calling program. These samples are integrated by the Simpsons Rule method to simulate ideal sensors' outputs. Simpsons Rule requires two samples per integration, which correspond with the mid-point and end of a sensor cycle. A sensor cycle is therefore completed after each second call to SINS1: the calls are for the mid point and end of the cycle. Simpsons Rule also requires a sample for the beginning of the sensor cycle, which is usually the previous cycle's end value: for a time interval from T to (T+h), the integral of a function x(t) is given by:

$$\int_{T}^{(T+h)} x(t)\,dt = \frac{1}{6}h\left[\ x(T) + 4x(T+\tfrac{1}{2}h) + x(T+h)\ \right].$$

SINS1 can accurately accommodate step changes in specific force, provided they are timed to occur at the beginning of a sensor cycle. In this case, an extra call is made with the new initial values, with a flag set so no other processing takes place.

Optionally, SINS1 will accept values of integrated (in Body axes) angular velocity and specific force, in which case the above integration is not performed, and a sensor cycle is completed on every call.

2.1. Sensor Errors

Once per sensor cycle, these "perfect" sensor outputs are corrupted with sensor characteristics and errors, which at present include misalignments, scale factor errors, fixed biases, and quantization.

Misalignments are defined for each sensor set by a 3 x 3 matrix which specifies each individual sensor's response to inputs along the other axes, as well as its own sensitive axis. Responses of gyroscopes to specific forces or of accelerometers to angular velocity or acceleration could be similarly included, but are not at this stage.

Scale factor errors are included by multiplying each sensor output by an error factor. (Alternatively these may be included with the misalignment matrices if required.)

Fixed biases are included by adding to each sensor output an amount corresponding to the zero offset multiplied by the sensor cycle time.

Conservative quantization may be included. The sensor output is truncated to an integer number of quanta, and the residual is saved to be added (before quantization) to the sensor output in the next sensor cycle.

3. NAVIGATOR SEGMENT

The navigator model is based upon a split-frame algorithm described in reference (1), with certain changes and additions. Whereas (1) used the North/East/Down (G) frame for navigation, with position maintained directly in Longitude and Latitude angles, this model maintains its position via the Earth to Navigation axes quaternion, and the Navigation frame is the Wander azimuth/Down (W) frame. Sensor compensation is not included: for present purposes, errors at sensor level are sufficient. The fast segment samples the sensors' outputs when required, updates the attitude quaternion for the body's absolute rotation and accumulates velocity increments caused by specific force, in body axes. This segment has a choice of orders of accuracy for the attitude updating and body axis velocity accumulation calculations. The intermediate segment updates the attitude quaternion for absolute rotations caused by Earth rotation and transport. It also transforms the body axes velocity increment into navigation axes, compensates for velocity increment from gravity, and does the position update. In the slow segment, slowly varying Earth-related quantities are updated.

Since some applications have low dynamics, and may be required to run for extended periods, there are optional higher order algorithms for attitude updating and velocity accumulation calculations. These impose slightly heavier computer loadings per iteration of the navigator algorithm, but allow lower iteration rates for a given accuracy, provided the frequency content of the environment is appropriate.

3.1. Split Frame Algorithm

For the sake of completeness of this report, the algorithm will now be briefly summarized.

The Body frame rotation rate may be several hundred degrees per second - far greater than the Navigation frame rate, even in a high speed aircraft. Changes in gravity, and effects of Earth rotation and curvature are functions of the position of Navigation axes relative to Earth: although they change slowly, their effects on the accelerometers change at Body rate. The split frame mechanisation performs Body axes related calculations at a fast iteration rate (the "Fast Segment") in Body coordinates, and Navigation axes related calculations at a slower rate (the "Intermediate Segment") in Navigation coordinates. At what may be a yet slower rate (the "Slow Segment"), certain slowly varying quantities such as local Earth radii and gravity, are calculated.

The attitude part of the algorithm updates the attitude quaternion in the Fast Segment for attitude changes of the Body relative to the Inertial frame, using gyroscope outputs. In the Intermediate Segment, the attitude quaternion is updated to account for rotation of the Navigation frame relative to the Inertial frame; this update is calculated from position changes and Earth rotation.

The basis of the navigation part of the algorithm is that changes in velocity or position caused by specific force may be calculated separately from changes caused by gravitation (mass attraction). The actual change is the sum of these. Velocity increments caused by specific force are accumulated in the Fast Segment. In the Intermediate Segment these are transformed into Navigation coordinates, and are added to calculated velocity increments caused by gravitation. Position updating is then performed.

Coordinate transformations between Earth, Geographic, and Wander Azimuth frames are described in Appendix 3. Coordinate transformations between the Navigation frame and the Body frame are described in Appendix 4. The relationships in Appendix 4 apply, whatever frame is used as the Navigation frame: in SINS1, the Wander Azimuth frame is used. Angular velocity relationships between Inertial, Earth, Geographic, and Wander frames are derived in Appendix 5.

3.2. Fast Segment Calculations

The "Fast Period" is defined as the time between outputs from the fast segment. The fast segment requires sensor output data at the mid-point and end-point of its period. Sensors' outputs are sampled at each entry to the fast segment, but the calculations are only performed at each second entry. The fast period is therefore twice the time between calls to the fast segment and is an even multiple of the sensor cycle period.

3.2.1. Attitude Updating

Gyroscopes incremental sample sets $\underline{\Theta}_m$ and $\underline{\Theta}_e$ at mid-point and end-point of the fast period are used to estimate the rotation vector $\underline{\Phi}_{IB}$:

$$\underline{\Phi}_{IB} = \underline{\Theta}_m + \underline{\Theta}_e + \frac{2}{3} (\underline{\Theta}_m \times \underline{\Theta}_e)$$

a derivation of this is available in reference (1), appendix (6).

From this is calculated the updating quaternion $\overline{QU}$ for attitude changes relative to inertial space (for example, to 3rd order):

$$\overline{QU} = [1 - \frac{1}{8}(\underline{\Phi}_{IB} \cdot \underline{\Phi}_{IB})], \ [\frac{1}{2} - \frac{1}{48}(\underline{\Phi}_{IB} \cdot \underline{\Phi}_{IB})]\underline{\Phi}_{IB}$$

The attitude quaternion $\overline{QWB}$ is updated (see Appendix 2) for this rotation:

$$\overline{QWB} \leftarrow \overline{QWB}(*)\overline{QU}$$

3.2.2. Body Axes Velocity Accumulation

The equation for velocity $\underline{V}$ caused by specific force $\underline{SF}$ at any instant is:

$$\dot{\underline{V}} = \underline{SF} - [\underline{\omega}_{IB}]x\underline{V}$$

A simple solution for this is available as in reference (1), appendix (9). For this model, an optional 4th order Runge-Kutta solution is also available, offering higher accuracy but for a greater computing burden.

3.3. Intermediate Segment Calculations

The "Intermediate Period" is defined as the time between calls to the intermediate segment. This segment takes the output from the fast segment, plus height data from the calling program, for its input. It updates its output on each call, so the intermediate period may be any integer multiple of the fast period, i.e., an even multiple of the time between calls to the fast segment. The call is defined to be at the end of the segment.

3.3.1. Attitude Updating

Since the previous entry to this segment, the fast segment has updated the attitude quaternion for absolute rotation. Now the quaternion is updated for navigation axes rotations as described in reference (1), appendix 11.

From the previous entry to this segment, there is a (1st order) estimate of Navigation axes rotation vector $\underline{\Phi}_{IW}$ since that time. A third order estimate of the corresponding quaternion is made: the updating quaternion $\overline{QU}$ is the inverse of this. $\overline{QWB}$ is then updated:

$$\overline{QWB} \leftarrow \overline{QU}(*)\overline{QWB}$$

3.3.2. Velocity Integration

The direction cosine matrix C_B^W for the Body to Wander transformation is calculated from $\overline{QWB}$. The velocity increment $\underline{\delta V}_F$ from the fast segment is transformed from B to W coordinates:

$$[\underline{\delta V}_F]^W = C_B^W \; [\underline{\delta V}_F]^B$$

the $[\underline{\delta V}_F]^B$ are then set to zero for the next cycle. From the previous entry to this segment, there is an estimate of the velocity increment due to gravitation $[\delta V_g]^W$, which is added to $[\underline{\delta V}_F]^W$, together with a gravity support correction (see appendix 6), to give the total velocity increment $[\underline{\delta V}_{EW}]^W$ relative to Earth. Total velocity $[\underline{V}_{EW}]^W$ is then updated by adding the increment.

3.3.3. Position Update

Angular velocity of W relative to E frame, in W coordinates is given by

$$[\underline{\omega}_{EW}]^W = D \; [\underline{V}_{EW}]^W$$

where

$$D = \begin{bmatrix} -DX & D2 & 0 \\ -D1 & DX & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The components of D are calculated by the Slow segment. They are functions of Earth radius, Latitude, and Wander angle (see appendix 5).

Position is maintained by the Earth to Wander frame quaternion $\overline{QEW}$. This is updated with the quaternion $\overline{QU}$ representing the rotation of the Wander frame relative to Earth during the intermediate segment.

If T is the intermediate period, rotation of W frame relative to E frame is:

$$[\underline{\Phi}_{EW}]^W = [\underline{\omega}_{EW}]^W \, .T$$

Updating quaternion of above to 1st order is: $\overline{QU} = 1, \frac{1}{2}[\underline{\Phi}_{EW}]^W$

Position update:

$$\overline{QEW} \leftarrow \overline{QEW}(*)\overline{QU}$$

3.3.4. Values for next Intermediate cycle

Absolute rotation of Wander frame during T is:

$$[\underline{\Phi}_{IW}]^W = [\underline{\omega}_{IE} + \underline{\omega}_{EW}]^W.T$$

Velocity increment due to gravitation:

$$[\underline{\delta V}_g]^W = ([g]^W - [\underline{\omega}_{IE} + \underline{\omega}_{IW}]^W x [\underline{V}_{EW} + \frac{1}{2}gT]^W).T$$

Values of $[g]^W$ and $[\underline{\omega}_{IE}]^W$ are calculated in the Slow segment.

3.4 Slow Segment Calculations

This segment is used for calculation of slowly varying quantities related to the systems position relative to Earth. These require parts of the E to W direction cosine matrix CEW, which are calculated from the $\overline{QEW}$.

Quantities updated in this segment are:

Components of the D matrix (see appendix 5),

Angular velocity of E relative to I frame, in W coordinates:

$$[\underline{\omega}_{IE}]^W = C_E^W[\underline{\omega}_{IE}]^E,$$

Gravity support correction (see appendix 6),

Vertical component of gravity g3 (n.b., in a fast aircraft application, this would perhaps be in the Intermediate segment). This is calculated from the relationship:

$$g3 = Ge.(1 + GL.\sin^2(LAT)) - (GA.H)/Re$$

where Ge is gravity at equator, GL is a latitude factor, GA is an altitude factor, H is altitude (metres, +ve up), and Re is Earth equatorial radius, in metres.

3.5 One shot coarse alignment

This facility is described in appendix 7. The simulated sensor outputs, including sensor errors but without quantization effects, are calculated for the system stationary at the true initial attitude. The simulated accelerometer outputs arising from gravity support are then used to calculate the quaternion of the rotation from "level" to Body axes. This is used to calculate the simulated levelled gyro outputs arising from Earth rotation, and these are used to establish the heading of the levelled axes and hence the G to B axes relationship.

This is a rather crude method which would only be useful in a real system for making an initial coarse alignment estimate.

3.6 Introduction of "Tilt changes" during a run

These may be specified relative to either the Body or the Geographical axes in terms of the yaw, elevation, and bank angles, in that order. The effect is a change in the Wander to Body reference quaternion $\overline{QWB}$. Tilts are regarded as positive in the sense <systems previous estimate of Body attitude> to <systems new estimate of body attitude>.

3.6.1 Relative to Body axes

This is effectively equivalent to the reception of spurious gyro data. The quaternion of the tilt $\overline{QT}$ is calculated (as for $\overline{QNB}$ in appendix 4.2) from the angles, and the new $\overline{QWB}$ is obtained:

$$\overline{QWB} \leftarrow \overline{QWB}(*)\overline{QT}$$

3.6.2 Relative to Geographic axes

Here, the tilt is specified as heading, elevation, and bank (of the body) in Geographic axes angles. For example, in a Body heading West with elevation axis level, unit tilt about North would appear as unit tilt about the elevation axis.

In terms of the G to B and the W to B relationships, this is equivalent to an equal and opposite tilt rotation of the G relative to B axes, with the G to W relationship kept constant.

Consider the "true" G reverse tilted to G' and the "true" W reverse tilted to W'. The required Wander to Body quaternion after tilting has the value of $\overline{QW'B}$, given by (see appendix 2.2):

$$\overline{QW'B} = [[\ \overline{QW'G'}(*)\overline{QG'G}\ ](*)\overline{QGW}\ ](*)\overline{QWB}\ ].$$

$\overline{QG'G}$ is obtained from the specified tilt (as $\overline{QNB}$ in appendix 4.2).

$\overline{QGW}$ is rotation through wander angle, and:

$\overline{QW'G'} = \overline{QWG} = (\overline{QGW})^{-1}$ ($\overline{QGW}$ is $\overline{Q3}$ in appendix 3.2).

$\overline{QWB}$ is the previous (before tilt) value.

$\overline{QWB'}$ is the new (after tilt) value for $\overline{QWB}$.

4. PROGRAM IMPLEMENTATION

The executive routine for the program is called SINS1, shown in figure 1. This performs little more than error checking, timekeeping, and making calls to the major subroutines SENSORS and NAVEXEC, shown in figures 2 and 3.

4.1 Sensors Segment

Routine SENSORS generates simulated integrating gyroscope and accelerometer outputs, as described in Part 2. If the entry is at time zero, the sensor initialisation routine SENSINI is called, and no further processing takes place. If sensor errors are to be re-initialised at a particular entry, SENSINI is called, followed by a normal run through the rest of the routine.

Output of SENSORS is in the form of "buffers" containing all the sensor outputs. As they are integrating sensors, the outputs are added to the buffer contents as they are generated: this allows the output sampling to take place at any time, provided the sampling routine resets the buffers contents to zero.

Routine SENSINI calls a routine SENREAD, which reads the sensor characteristics data file, or, if this file is absent, sets the default values. It does some error checking and sets up various initial values as required.

4.2 Navigator Segment

The executive routine for this segment is NAVEXEC, whose function is simply to decide which, if any, of the major navigator routines are to be called on that entry, and to call the routines. These routines are:

NAVINIT, the initialisation routine

NAVFAST, the "fast segment" routine

NAVIMED, the "intermediate segment" routine

NAVSLOW, the "slow segment" routine

NAVOUTS, the output generation routine

Routine NAVINIT is only called on the first, initialisation call to SINS1. It calls a routine NAVREAD, which reads the navigator data file, or, if this file is absent, sets the default values. NAVINIT uses this data, together with the data passed with the call to SINS1, to set up initial values in all the variables used in the navigator segment. If the calling program requires a one-shot alignment, NAVINIT calls the alignment routine SNAPALYN, described in part 3.5.

Routines NAVFAST, NAVIMED, and NAVSLOW perform the fast, intermediate, and slow segment calculations as described in parts 3.2 to 3.4.

Routine NAVOUTS writes the output array. The angular position, wander, and attitude data are obtained from the quaternion elements as shown in appendices 3 and 4. If the latitude or the elevation approaches + or - 90 degrees, a warning flag is set. In either case, valid position and attitude output data is still available from the appropriate quaternion or direction cosine elements.

5. SINS1 OPERATION

SINS1 is a Fortran F77 subroutine which is invoked by:

CALL SINS1(IIO,DIN,DOUT,CIO),

where IIO is an array of input and output integer variables, DIN and DOUT are arrays of input and output double precision (REAL*8) variables respectively, and CIO is an array of character strings for passing filenames. Double precision is used for all real variables in SINS1.

Two data files may be called by SINS1 during initialisation, containing initialisation data for the sensors and the navigator segments. Formats for these files are given below. If required, the files are not called, and default values used. Further data are passed to the routine as arguments to the call, both at initialisation and during the run.

The calling program is to be part of a time stepping simulation, where the time keeping includes an integer counter of fixed period time steps. SINS1 must be called at each increment of this counter, with the appropriate input data. For initialisation it must be called with the counter at zero. Output is available, except as stated below, from any call including at time zero, although there may be some latency in subsequent output data, because this is updated during the intermediate segment of the navigation algorithm.

Elements of IIO, CIO, DIO and DOUT are listed in Tables 1, 2 and 3.

5.1 Initialisation call

IIO(1) - SIMSTEPCTR - must have a value of zero for the initialisation call, i.e. time = 0, as this is used by SINS1 to trigger its initialisation processes.

IIO(2) - SENSFLAG - must have a value of 1 or 2 in this call. This indicates to SINS1 whether input data during the run will be angular velocity and specific force (2), or their integrals (1). In the former case, the sensor cycle time is twice the step time, and every other call will be used only to pass input data at the mid point of the sensor cycle; in the latter case, the sensor cycle is equal to the step time, and sensor output is updated on every call.

IIO(3) - ALYNFLAG - must have a value of 0 or 1 on this call. This is used to tell SINS1 how the Wander to Body quaternion is to be initialised. If it is 1, a coarse one shot alignment will be carried out using the (unquantized and unmoving) sensors' outputs. If it is 0, SINS1 will use the values of heading, bank, and elevation angles passed in DIN.

IIO(4) - HGHTFLAG - must have a value of 0 or 1 in any call. If 0, the value of height passed in DIN will be used by SINS1, and the vertical component of velocity output will be set to zero. If 1, the vertical channel is free inertial (and therefore unstable).

IIO(5) - OUTFLAG - should be 0, 1 or 2 in any call. If 0, no output is provided in DOUT. If 1, partial output is provided, otherwise full output is provided, as described later.

IIO(6) - UNI - is passed as the Fortran unit number for SINS1 to assign to each data file whilst being read. Only one number is required because only one file is open at any time.

IIO(7) is an output warning flag: if (and only if) full output has been requested, IIO(7) is returned set to 1 if the system is near a pole, 2 if the system is near gimbal lock (elevation +/-90 degrees), and 3 if both. If either of these occurs, the undefined angles are set to zero. If neither occurs, IIO(7) is returned set to 0. If full output has not been requested, IIO(7) is undefined.

IIO(8) - SULKS - is an output flag for SINS1 at any call. If operations are normal, it will be set to 0. If a 'fatal' problem is encountered in initialisation or running, SINS1 will write a message to the terminal and set IIO(8) to 1. SINS1 will then go dormant: subsequent calls to SINS1 will be returned with no processing, and no valid output, other than IIO(8) being set to 1.

CIO(1) is a character string (up to 80 characters), which contains the full pathname of the sensor data file from which SINS1 is to read sensor characteristics. If CIO(1) is a null string, SINS1 will default to assume error free sensors.

CIO(2) is a similar string containing the full pathname of the navigator data file. If CIO(2) is a null string, SINS1 will use its own default values.

The DIN are all inputs to SINS1:

DIN(1) to DIN(6) are for Body axes angular velocity and specific force, or their integrals. If SINS1 is to use integrated input data (SENSFLAG = 2), these variables are not used on this call and need not be set. Otherwise, the variables must be set up with their values at time = 0, as initial values for the sensors' integration process.

DIN(7) is HEIGHT at any call.

DIN(8) - SIMSTEPSEC - is the length of time in the simulation corresponding to an increment of SIMSTEPCTR.

DIN(9) to DIN(18) are for initial conditions used by the navigator segment: they are not necessarily the "true" values. DIN(16) to DIN(18), (bank, elevation, heading angles) need not be defined if a one shot alignment is to be used (if ALYNFLAG is 1).

DIN(19) to DIN(28) are allocated for the "true" (or "reference") versions of the above. SINS1 uses some of these for the one shot alignment.

5.2 In-Run calls

IIO(1) contains the current value of SIMSTEPCTR.

IIO(2) may take the value 3, 4, 5 or 6. Other values will be ignored. If 3 to 6 are not required, it is suggested that some value be written, say, 1 or 2, because IIO(2) is tested every time.

If IIO(2) is set to 3, the call may be used as an additional call to accommodate step changes in specific force (step changes in angular velocity are assumed impossible), and the user wishes to reset the specific force recorded for the beginning of the current sensor cycle. In this case the new values of specific force must be in DIN(4) to DIN(6). SINS1 then returns without further processing. Output is not available from this call.

If IIO(2) is set to 4, the sensor error characteristics are reset as from the start of the current sensor cycle. In this case, a new sensor data file is required, and values must be available in IIO(6) and CIO(1) as in the initialisation call.

If IIO(2) is set to 5, a "tilt" is applied to the navigators analytical axes. The tilt is specified by yaw, elevation, and bank changes of the Body axes. Values of these angles must be available in DIN(18), DIN(17) and DIN(16) respectively. SINS1 then returns without further processing.

If IIO(2) is set to 6, a "tilt" is applied as above, except that it is specified as changes relative to Geographic axes.

IIO(3) is not used.

IIO(4) and IIO(5) - same as initialisation call.

IIO(6) is not used, except as above with IIO(2) set to 4.

IIO(7) and IIO(8) are outputs - same as initialisation call.

CIO(1) is not used, except as above with IIO(2) set to 4.

CIO(2) is not used.

The DIN are all inputs to SINS1:

DIN(1) to DIN(6) contain body axes components of angular velocity and specific force, or their integrals.

DIN(7) contains HEIGHT.

DIN(8) and higher are not used, except that DIN(18), DIN(17) and DIN(16) may be used for yaw, elevation, and bank angles respectively for tilt introduction (where IIO(2) is set to 5 or 6).

5.3 Output returned by SINS1

According to the value of IIO(5) - OUTFLAG, SINS1 returns either no output (0), partial output (1), or full output (2). Table 3 lists the output quantities in DOUT.

Partial output fills DOUT(1) to DOUT(28): full output also fills DOUT(29) to DOUT(38). In partial output, DOUT(1) is a time tag for DOUT(2) to DOUT(22), and DOUT(23) is a time tag for DOUT(24) to DOUT(28). In the full output case, DOUT(1) is the time tag for all output.

5.4 Sensors' characteristics data file format

The full pathname of this data file should be placed in CIO(1). If CIO(1) is null string, default values will be used.

An example of contents of the data file is:
(These are the default values, which assume the system is error free.)

```
0                                    /* Gyro misalignment
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
0                                    /* Acc misalignment
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
1.0 1.0 1.0  1.0 1.0 1.0             /* Gyros, Accs, s/factors
```

```
0.0 0.0 0.0  0.0 0.0 0.0         /* Gyros, Accs, biases
0.0 0.0                          /* Gyros, Accs, quant. levels
```

Names of the corresponding variables in the program are:

```
GYMIS
GYMX11    GYMX12    GYMX13
GYMX21    GYMX22    GYMX23
GYMX31    GYMX32    GYMX33
ACMIS
ACMX11    ACMX12    ACMX13
ACMX21    ACMX22    ACMX23
ACMX31    ACMX32    ACMX33
GYSF(1)     GYSF(2)     GYSF(3)     ACSF(1)     ACSF(2)     ACSF(3)
GYBIAS(1)   GYBIAS(2)   GYBIAS(3)   ACBIAS(1)   ACBIAS(2)   ACBIAS(3)
GQLEVEL   AQLEVEL
```

In the above, the first letter being G or A denotes gyro or accelerometer respectively. GYMIS and ACMIS are integer, all others are real.

GYMIS is used to indicate if any misalignments exist in the misalignment matrix GYMX. If it is 0, perfect alignment is assumed, and the values in the components of GYMX are not used, although they must be in the data file. For any other value, the GYMX are used.

GYMX are misalignment factors. GYMXjk means that the gyro on the j axis senses GYMXjk times the integral of body rate about the k axis.

GYSF are scale factor errors. The output of the gyro on the j axis is multiplied by GYSF(j).

GYBIAS(j) are fixed biases, in degrees per hour. These are converted to radians per sensor cycle, and added to the respective gyro's integrated output.

GQLEVEL is the quantization level for all gyros, in arcseconds. This may take a zero value, in which case the quantization level is the resolution (double precision) of the computer used to run the program.

Accelerometer characteristics are similarly described, except that the misalignments operate on the integrals of specific force, and the units for fixed biases are metres per second^2, and quantization is in metres per second.

5.5 Navigator initialisation data file format

The full pathname of this data file should be placed in CIO(2). If CIO(2) is a null string, default values will be used.

An example of contents of the data file is:

6378135.0	298.26	0.0000729211515	9.780333	0.0052884	2.014
2	16	32			
10	10	7	1		

Names of the corresponding variables in the program are:

EQRAD	INVELL	ERATE	EQATGRAV	GRAVLAT	GRAVALT
SSPERFAST	SSPERIMED	SSPERSLOW			
QWBNORM	QEWNORM	QUPORDER	AP9MOD		

The first line consists of Earth quantities. They are all real.

EQRAD	is the equatorial radius in metres
INVELL	is inverse of ellipticity
ERATE	is Earth rate in radians per second
EQATGRAV	is gravity at the equator in metres per second^2
GRAVLAT	is a factor for variation of gravity with latitude
GRAVALT	is a factor for variation of gravity with altitude

The second line are simulation steps (increments of SIMSTEPCTR) per navigation algorithm step. They are all integers.

SSPERFAST is the number of steps per call to NAVFAST. It must be an integer multiple of the sensor cycle period, which is either 1 or 2 simulation steps in length, according to the value of SENSTEP. The Fast period is double the time between calls to NAVFAST.

SSPERIMED is the number of steps per call to NAVIMED. It must be an integer multiple of the fast period, that is, it must be an even multiple of SSPERFAST.

SSPERSLOW is the number of steps per call to NAVSLOW. It should be an integer multiple of SSPERIMED.

The third line quantities are integers.

QWBNORM is the minimum number of times QWB may be updated between normalizations. This occurs in NAVIMED only, so if a large number of fast cycles are performed for each intermediate cycle, QWB may be updated that many more times before normalization.

QEWNORM is the number of times QEW is updated between normalizations.

QUPORDER is the order of calculation of the updating quaternion in NAVFAST. A minimum of 3rd order is always used. If QUPORDER is greater than 3, then 5th order is used; if it is greater than 5, then 7th order is used. The extra computing load for higher orders is almost negligible.

AP9MOD is a flag for higher order body axes velocity calculation in NAVFAST. If it is 0, a simple calculation is performed: otherwise a 4th order Runge-Kutta is used. The extra computing burden may be significant.

Values listed above are the default values for the first and the third lines of data. The default for SSPERFAST is the value of SENSTEP, i.e., IIO(2) on the first call. The default for SSPERIMED is SSPERFAST times 8, which means the intermediate period is 4 times the fast period. The default for SSPERSLOW is equal to SSPERIMED.

6. HANDLER PROGRAM EXAMPLE: FPSIN2

This section contains a description of an example of a handler program known as FPSIN2, which operates SINS1 using the outputs of a flight profile generator program FPG2 (reference 3).

FPG2 generates a file containing a sequence of integrated (in body axes) specific forces and angular velocities at a point in an aircraft executing a sequence of user defined manoeuvres. They are, effectively, "perfect" IMU outputs. It also generates a "reference" file containing a sequence of nominally true position, velocity, and attitude of the aircraft, and a file of initialisation data for use by FPSIN2.

FPSIN2 functions as a time stepping simulation. At each step, it reads the FPG2 file of integrated sensor data and passes these to SINS1. As its output, SINS1 provides the strapdown INS estimates of position, velocity, and attitude - the system state. At specified time intervals, FPSIN2 obtains this output. It also reads the reference file data generated by FPG2. Errors in the INS estimates of system state are then calculated and written to diskfiles and/or the terminal.

6.1 Program Description

The main module of the handler program is called FPSIN2, as shown in figure 4. It performs initialisation, then steps through the simulation.

During initialisation it calls FPSFILES and INISINS1. FPSFILES is a file handler which opens and reads the initialisation data files, opens the FPG2 IMU and reference data files for reading during the run, prepares files for output, and does checking and manipulation of the initial input data. The timing variables and counters are set to zero. INISINS1 is a SINS1 initialisation handler. SINS1 is to use integrated sensor data, and full output from SINS1 is required at the initialisation call, so flags are set, and INISINS1 is called. It sets up the call arguments for the initialisation call to SINS1, calls it and obtains its output.

In each step of the run, the stepcounter is incremented, and routines CLOCK, RUNSINS1, and FPSOUT are called. CLOCK is a timing utility which keeps time in hours, minutes, and seconds. RUNSINS1 is a SINS1 run-time handler. Before this is called, FPSIN2 checks the stepcounter to ascertain if output is required, and sets a flag accordingly. RUNSINS1 is then called. This reads the FPG2 IMU data and sets up the run arguments for the call to SINS1, calls it and gets its output. The output routine FPSOUT is entered on every step, and generates and writes output if required (as determined from the stepcounter).

Finally, when stepping is completed, all open files are closed.

6.2 Input Files and Initialisation

The first file sought by FPSFILES is a command file which contains the following information, one item per line, in the order shown:

Directory name where FPG2 output files are to be found
The filename suffix xxx of the FPG2 files to be used
Second filename suffix yyy to be used for FPSIN2 output
(Existing directory name where FPSIN2 output files are to be written)
Terminal output specifier.

This command file is to have the name FPSRUN, and is to be in the users directory from which FPSIN2 is being executed. If this file cannot be found, FPSIN2 will ask for the same data interactively at the terminal.

Data files required to have been generated by FPG2 (the "flight specific" files), and located in the directory specified above, are:

SIMDATA.xxx	the file of initialisation data for FPSIN2
IMUOUT.xxx	the file of "perfect" IMU sensor data
REFOUT.xxx	*the file of "reference" flight profile data.*

These files should not be altered by the user. SIMDATA.xxx contains initial true values of position, velocity, attitude, and wander angle, plus timing data.

A "run specific" input data file of INS initialisation data is also required. This is prepared by the user, is to be in his directory as above, and has the name INSDATA.xxx.yyy where the xxx and yyy are specified as above.

6.2.1 INS Initialisation Data File INSDATA.xxx.yyy

Contents of this file, using variable names as in the program, are:

ELON ELAT EALPHA EHEIGHT EVNORTH ... (continued)
... EVEAST EVDOWN EBANK EELEV EHEAD

UNI

FNINSSEN

FNINSNAV

ALYNFLAG HGHTFLAG.

The first line is a set of initial error quantities: longitude, latitude, wander angle (degrees); height (metres + up); velocities north, east, down (metres/second); attitudes bank, elevation, heading (degrees). These are used in setting up the simulated INS. The actual value used by the INS for any variable is the reference value (from SIMDATA) plus this error value.

UNI is a Fortran unit number for use by SINS1. It may be any integer greater than 8 available to the user (FPSIN2 uses 1 to 8).

FNINSSEN and FNINSNAV are the pathnames (up to 80 characters each) for SINS1 sensor characteristics data and navigator data files respectively. Contents of these are discussed in a previous section. If either has the value ' ' its default is used.

ALYNFLAG and HGHTFLAG are alignment and height flags respectively, and each may have values 0 or 1. If ALYNFLAG is 1, the coarse one-shot INS alignment will be carried out; if it is 0, the data values of heading, bank, and elevation will be used. If HGHTFLAG is 0, the reference values of height will be used by SINS1, and its vertical velocity output will be set to zero; if it is 1, the vertical channel is free inertial (and therefore unstable).

If a file INSDATA.xxx.yyy cannot be found, FPSFILES looks for a file INSDATA.xxx, and if this cannot be found, it looks for a file INSDATA.

6.3 Initialisation Call to SINS1

INISINS1 sets up the call arguments (see Section 5) for the initialisation call, namely, IIO(1 to 6), CIO(1 and 2), DIN(7 to 15) and DIN(16 to 28) as required. FPSIN2 sets SENSFLAG, for IIO(2) to 1, and OUTFLAG, for IIO(5) to 2 ("Full" SINS1 output is required here), and the stepcounter, for IIO(1), to zero. Other arguments are all available from the initial data files, either directly or, for INS initial estimate conditions, by adding the error and reference quantities.

6.4 Runtime Call to SINS1

RUNSINS1 reads the FPG2 IMU data and sets up the run arguments for each call to SINS1, namely IIO(1,2,4 and 5) and DIN(1) to DIN(7). IIO(1 and 5) are determined by FPSIN2, IIO(2 and 4) are from the initial data files. The DIN values come straight from the FPG2 IMU data file.

6.5 Output from FPSIN2

The following files are written, in the directory specified in the command file or interactive session with FPSFILES:

ERRORS.xxx.yyy	file of INS errors
RANGES.xxx.yyy	maximum and minimum values in above
RECORD.xxx.yyy	record file of all initialisation data.

The file ERRORS.xxx.yyy is an ASCII file of data in columns:
TIME LON LAT ALPHA HEIGHT VNORTH VEAST VDOWN BANK ELEV HEAD.
Each line is the difference between the INS estimate and the reference value of the various quantities at the time.

The file RANGES.xxx.yyy is similar to ERRORS.xxx.yyy, except that only the greatest and least of each quantity is recorded.

The file RECORD.xxx.yyy contains a list of the contents of the SIMDATA.xxx file, the INSDATA.xxx.yyy file, and the contents of the SINS1 sensor data and navigator data files.

If the terminal output specifier is "F", the same data is sent to the terminal as is written in the ERRORS.xxx.yyy file. If the specifier is "P", the only data sent to the terminal are the times at which each FPG2 manoeuvre is completed, and may provide reassurance that the computer is still operating. Otherwise there is no terminal output.

7. AREAS FOR DEVELOPMENT

In its present form, the program may use a rather idealised one shot open loop alignment. In a real INS, with noise and quantization in the sensors and movement of the system, this arrangement would not be satisfactory, except for a coarse levelling and perhaps a rough gyrocompassing. There is scope for inclusion of an alignment phase: this could use much of the existing program structure, probably with the addition of an alignment status in NAVEXEC, which would call an alternative intermediate rate routine to perform the alignment functions.

A realistic alignment routine was not included in this program because it was intended for use as a simulation of a navigating system, in which certain initialisation errors may be included as required. In practice, an alignment routine would in any case be designed to suit the characteristics of a particular system and application.

The vertical channel, if not in pure-inertial mode, relies upon provision of accurate height data to the system, from which vertical velocity is derived. A useful development would be the inclusion of a more realistic vertical channel, which could be inserted in NAVIMED at the point where the vertical velocity is calculated.

Sensor error models could be further developed if required by inclusion of the appropriate characteristic into SENSORS and SENREAD, and if initialisation is required, into SENSINI also. Note that if any additional modelled errors are likely to affect a one-shot alignment, they should also be included in SNAPALYN. Sensor error characteristics were repeated separately in SNAPALYN in order to keep the Sensors segment and the Navigator segment structurally separated in the total program.

REFERENCES

No.	Author	Title
1	R.B. Miller	Strapdown I.N.S.: an Algorithm for Attitude and Navigation Computations. ARL-SYS-REPORT-23, Oct. 1980
2	M. Kayton & W.R. Fried	Avionics Navigation Systems. J. Wiley & Sons, 1964
3	R.B. Miller	FPG2: A Flight Profile Generator Program ARL-SYS-TM-98, Jul 1989

TABLE 1: Integer and Character Arguments to SINS1 Call

	Function	When read or written
Integer Input/Output array IIO.		
Input to SINS1:		
IIO(1)	Step counter	- every call
IIO(2)	Sensors/tilts flag	- every call
IIO(3)	Alignment flag	- first call
IIO(4)	Height flag	- every call
IIO(5)	Output flag	- every call
IIO(6)	Data file unit	- first call or when IIO(2) = 4
Output from SINS1:		
IIO(7)	Output flag	- when IIO(5) = 1 or 2
IIO(8)	Status flag	- every call

Also, IIO(9) and IIO(10) are available but unused.

Character string input array CIO.

CIO(1)	Pathname of sensor data file	- first call or when IIO(2) = 4
CIO(2)	Pathname of navigator data file	- first call only

Also, CIO(3) and CIO(4) are available but unused.

TABLE 2: Double Precision Real Input Arguments to SINS1 Call

Every call:

DIN(1)	Angular velocity component -	axis B1	radians/sec
DIN(2)	ditto	axis B2	ditto
DIN(3)	ditto	axis B3	ditto
DIN(4)	Specific force component -	axis B1	metres/sec^2
DIN(5)	ditto	axis B2	ditto
DIN(6)	ditto	axis B3	ditto

(if integrals of the above are used, units are radians and metres/sec., and they need not be defined for the first call)

DIN(7)	Height	metres (+up)

First call only:

DIN(8)	Simulation time between calls	seconds

Initial conditions for navigator:

DIN(9)	Longitude		radians
DIN(10)	Latitude		radians
DIN(11)	Wander angle	(+ve G to W axes)	radians
DIN(12)	Height	(+ve up)	metres
DIN(13)	North velocity	(+ve N)	metres/sec
DIN(14)	East velocity	(+ve E)	metres/sec
DIN(15)	Down velocity	(+ve D)	metres/sec
DIN(16)*	Bank angle	(+ve G to B axes)	radians
DIN(17)*	Elevation angle	(+ve G to B axes)	radians
DIN(18)*	Heading angle	(+ve G to B axes)	radians

* Only required if IIO(3) = 0

True initial conditions. These are only required if IIO(3) = 1; those marked with * are not required in any case. Units, etc., as above:

DIN(19)*	Longitude
DIN(20)	Latitude
DIN(21)*	Wander angle
DIN(22)	Height
DIN(23)*	North velocity
DIN(24)*	East velocity
DIN(25)*	Down velocity
DIN(26)	Bank angle
DIN(27)	Elevation angle
DIN(28)	Heading angle

DIN(29) to DIN(40) are not used.

Calls to introduce tilt changes:

If IIO(2) is set to 5 or 6, values of Bank, Elevation, and Yaw angles are required in DIN(16) to DIN(18). Units are radians.

TABLE 3: Double Precision Real Output Arguments to SINS1 Call

If IIO(5) = 0, SINS1 does not return any valid output.

If IIO(5) = 1, SINS1 returns partial output:

DOUT(1)	is	WBTIME	Time tag DOUT(2) to (22)		seconds
DOUT(2)	is	VEW1	Wander axes velocity	W1	metres/sec
DOUT(3)	is	VEW2	"	W2	"
DOUT(4)	is	VEW3	"	W3	"
DOUT(5)	is	HEIGHT	Height		metres (+ve up)
DOUT(6)	is	QEW0	E to W quaternion		
DOUT(7)	is	QEW1	"		
DOUT(8)	is	QEW2	"		
DOUT(9)	is	QEW3	"		
DOUT(10)	is	OQWB0	W to B quaternion		
DOUT(11)	is	OQWB1	"		
DOUT(12)	is	OQWB2	"		
DOUT(13)	is	OQWB3	"		
DOUT(14)	is	CBW11	B to W direction cosine		
DOUT(15)	is	CBW12	"		
DOUT(16)	is	CBW13	"		
DOUT(17)	is	CBW21	"		
DOUT(18)	is	CBW22	"		
DOUT(19)	is	CBW23	"		
DOUT(20)	is	CBW31	"		
DOUT(21)	is	CBW32	"		
DOUT(22)	is	CBW33	"		

DOUT(23)	is	EWTIME	Time tag DOUT(24) to (28)	seconds
DOUT(24)	is	CEW13	E to W direction cosine	
DOUT(25)	is	CEW23	"	
DOUT(26)	is	CEW31	"	
DOUT(27)	is	CEW32	"	
DOUT(28)	is	CEW33	"	

If IIO(5) = 2, SINS1 returns full output. DOUT(1) is the time tag for all output. In addition to the above, there is:

DOUT(29)	is	LON	Longitude	Radians	
DOUT(30)	is	LAT	Latitude	"	
DOUT(31)	is	ALPHA	Wander angle	"	
DOUT(32)	is	HEIGHT	Height	metres (+ve up)	
DOUT(33)	is	VNORTH	North velocity	metres/sec	(+ve N)
DOUT(34)	is	VEAST	East velocity	"	(+ve E)
DOUT(35)	is	VDOWN	Down velocity	"	(+ve D)
DOUT(36)	is	BANK	Bank angle	radians (+ve G to B)	
DOUT(37)	is	ELEV	Elevation angle	"	
DOUT(38)	is	HEAD	Heading angle	"	

DOUT(39) and DOUT(40) are not used.

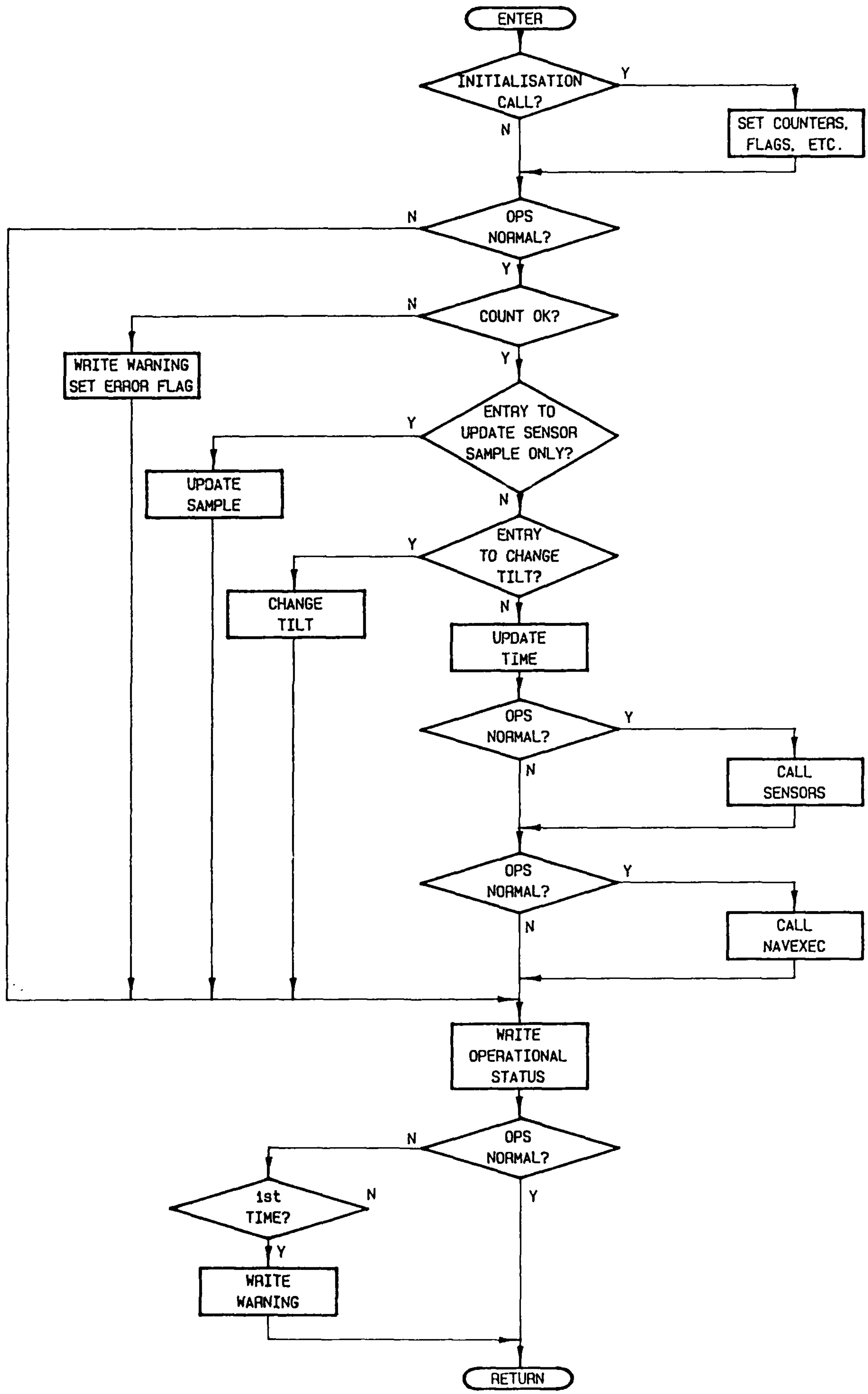

FIG. 1. SINS1 EXECUTIVE

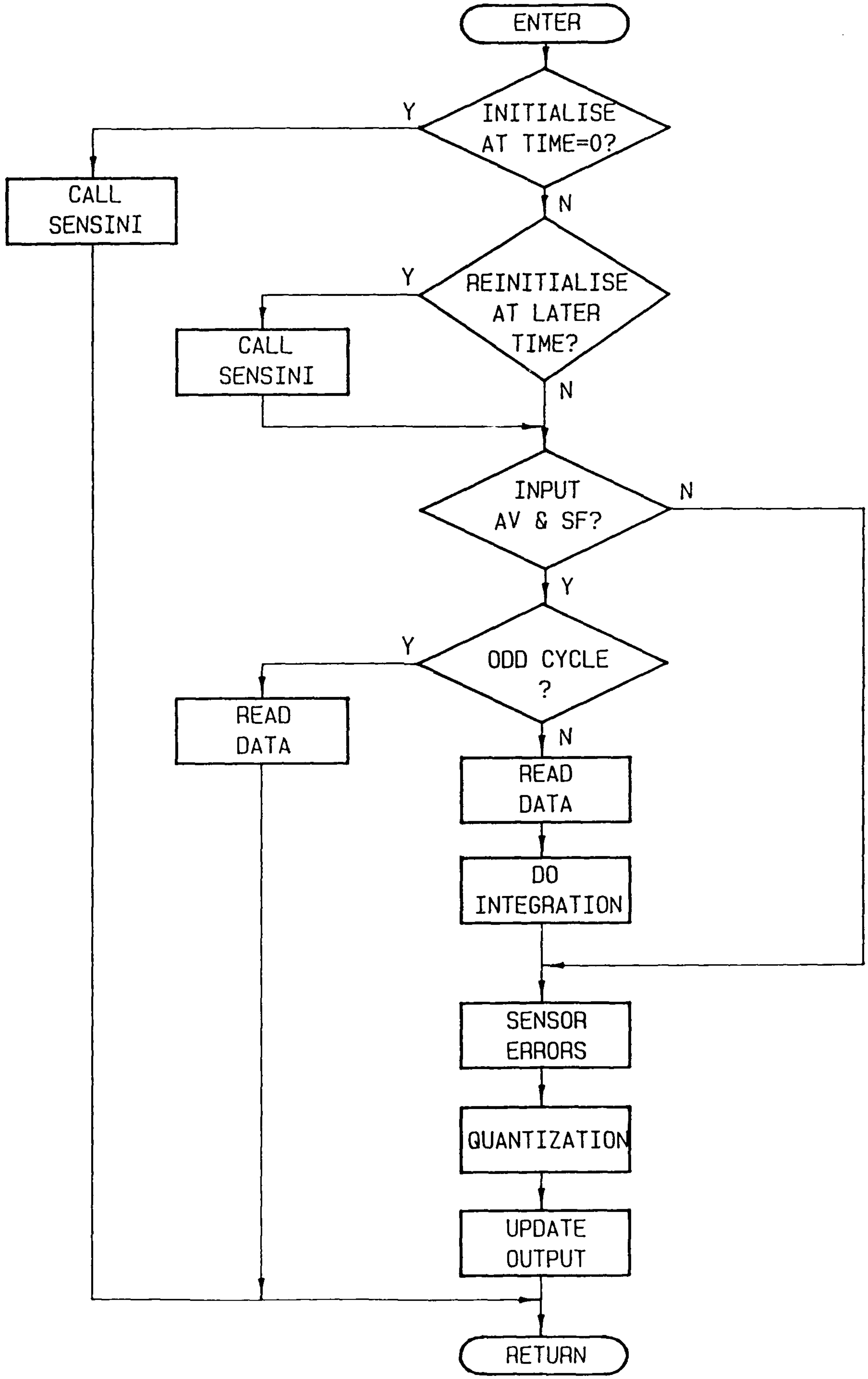

FIG. 2. SENSORS

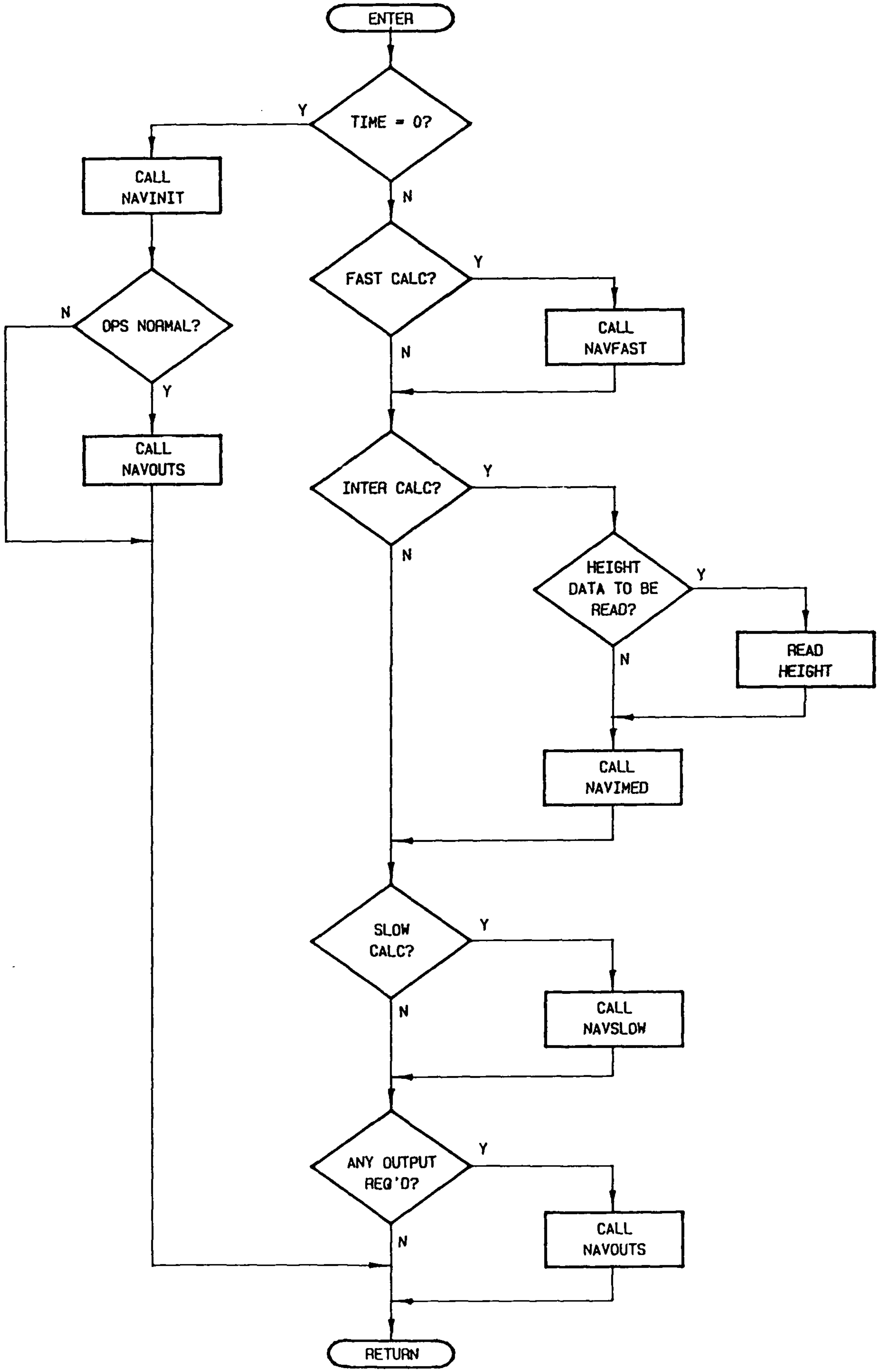

FIG. 3. NAVEXEC

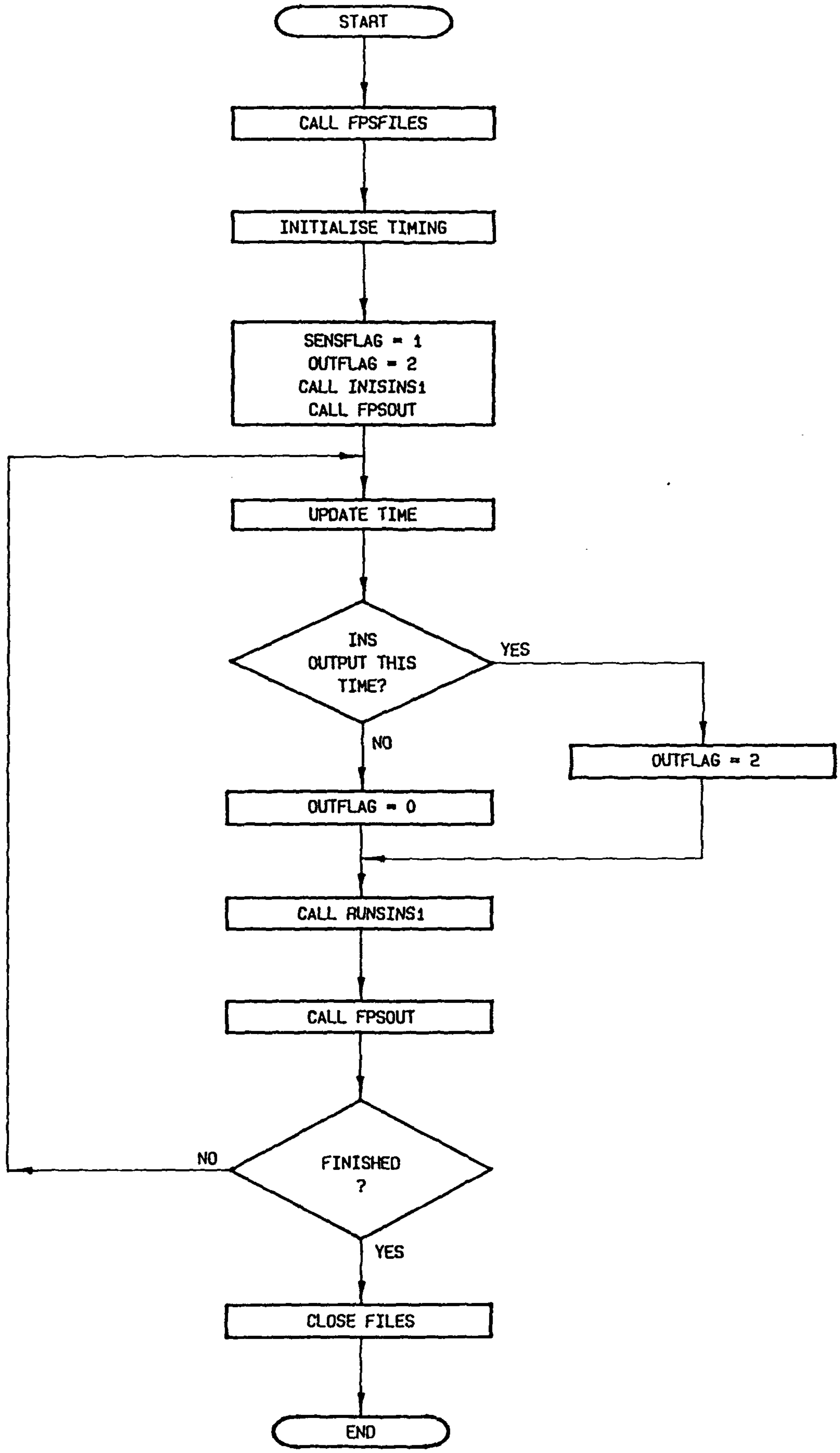
START
CALL FPSFILES
INITIALISE TIMING
SENSFLAG = 1
OUTFLAG = 2
CALL INISINS1
CALL FPSOUT
UPDATE TIME
INS OUTPUT THIS TIME?
YES
NO
OUTFLAG = 2
OUTFLAG = 0
CALL RUNSINS1
CALL FPSOUT
FINISHED ?
NO
YES
CLOSE FILES
END

FIG. 4. FPSIN2

Appendix 1 - Notation and Coordinate Frames

$\underline{U}$	a vector
$[\underline{U}]^X$	vector $\underline{U}$ in X frame coordinates
$\left[\frac{d\underline{U}}{dt}\right]_X$	rate of change of $\underline{U}$ with respect to frame X
$\left[\frac{d\underline{U}}{dt}\right]_X^Y$	is $\left[\frac{d\underline{U}}{dt}\right]_X$ in Y frame coordinates. N.B. $\left[\frac{d\underline{U}}{dt}\right]_X^X = [\dot{\underline{U}}]^X$
$\overline{QXY}$ or $\overline{Q}_{XY}$	quaternion representing rotation from X to Y frame
CXY or C_X^Y	direction cosine matrix: X to Y coordinates
$\underline{\Phi}_{XY}$	"rotation vector": X frame to Y frame
$\underline{\omega}_{XY}$	angular velocity of Y frame relative to X frame
$\underline{V}_{XY}$	velocity of Y frame relative to X frame
$\underline{g}$	gravity vector
Ω_E	scalar Earth rotation rate about polar axis
(*)	quaternion "multiplication" operator

Coordinate Frames

Axes sets are defined as follows:

INERTIAL (I) - having I3 along Earth's axis of rotation. There is no need to define I1 and I2.

EARTH (E) - has E3 along Earth axis of rotation, E1 and E2 are in the equatorial plane, with E1 at 0 degrees longitude.

GEOGRAPHIC (G) - having G1 and G2 as local level North and East, and G3 is Down. The origin is at the INS reference point.

WANDER AZIMUTH (W) - is obtained by a positive rotation of the G set about Down, through the Wander Angle. The origin is at the INS reference point.

BODY (B) - is the set of nominal INS axes: if the INS and its sensors were perfect, the Body axes would be the sensors' sensitive axes. The origin is at the INS reference point.

APPENDIX 2 - Direction Cosine and Quaternion Relationships.

There are many statements of these relationships (see for example, reference 1), they are repeated here for convenience.

A 2.1 Quaternion/Direction Cosines

Consider a quaternion $\overline{QXY}$ representing the rotation from the X to the Y frame, having components Q0, Q1, Q2 and Q3. Consider also the direction cosine matrix CYX, which may be used to transform coordinates of a vector from Y to X axes.

CYX may be written C_Y^X. For a vector $\underline{V}$: $[\underline{V}]^X = C_Y^X[\underline{V}]^Y$.

The components of CYX are:

$$\begin{bmatrix} CYX11 & CYX12 & CYX13 \\ CYX21 & CYX22 & CYX23 \\ CYX31 & CYX32 & CYX33 \end{bmatrix}$$

Given CYX, then CXY is its transpose.

$$\text{If } \overline{QXY} \text{ is } \begin{bmatrix} Q0 \\ Q1 \\ Q2 \\ Q3 \end{bmatrix}, \text{ then } \overline{QYX} \text{ is } \overline{QXY}^{-1} = \begin{bmatrix} Q0 \\ -Q1 \\ -Q2 \\ -Q3 \end{bmatrix}$$

In terms of the $\overline{QXY}$ components, CYX is:

$$\begin{bmatrix} 1 - 2(Q2^2 + Q3^2) & 2(Q1Q2 - Q0Q3) & 2(Q0Q2 + Q1Q3) \\ 2(Q0Q3 + Q1Q2) & 1 - 2(Q1^2 + Q3^2) & 2(Q2Q3 - Q0Q1) \\ 2(Q1Q3 - Q0Q2) & 2(Q0Q1 + Q2Q3) & 1 - 2(Q1^2 + Q2^2) \end{bmatrix}$$

A 2.2 Quaternion "Multiplication"

Consider a body with a quaternion $\overline{A} = (A_0, \underline{A})$ representing the rotation from reference to body axes. Give the body a rotation such that quaternion $\overline{B} = (B_0, \underline{B})$ represents the rotation from old to new body axes. The quaternion $\overline{C} = (C_0, \underline{C})$ representing the rotation from reference to new body axes is:

$$\overline{C} = \overline{A}(*)\overline{B} = (A_0B_0 - \underline{A}.\underline{B}),\ (A_0\underline{B} + \underline{A}B_0 + \underline{A}x\underline{B})$$

if $\overline{A} = \begin{bmatrix} A0 \\ A1 \\ A2 \\ A3 \end{bmatrix}$, and $\overline{B} = \begin{bmatrix} B0 \\ B1 \\ B2 \\ B3 \end{bmatrix}$, then

$$\overline{C} = \begin{bmatrix} C0 \\ C1 \\ C2 \\ C3 \end{bmatrix} = \begin{bmatrix} A0B0 - A1B1 - A2B2 - A3B3 \\ A0B1 + A1B0 + A2B3 - A3B2 \\ A0B2 - A1B3 + A2B0 + A3B1 \\ A0B3 + A1B2 - A2B1 + A3B0 \end{bmatrix}$$

also, $\overline{QRD} = [[[\ \overline{QRA}(*)\overline{QAB}\ ]\ (*)\overline{QBC}\ ]\ (*)\overline{QCD}\ ]$ etc.

Appendix 3 - Earth to Geographic to Wander Azimuth Transformation.

Rotations are positive in the sense E to G to W.

A 3.1 Direction Cosines/Euler Angles.

a) rotate about E3 through Longitude (L):

$$DCM1 = \begin{bmatrix} \cos(L) & \sin(L) & 0 \\ -\sin(L) & \cos(L) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

b) rotate about G2 (East) through -(90 + latitude (l)):

$$DCM2 = \begin{bmatrix} -\sin(l) & 0 & \cos(l) \\ 0 & 1 & 0 \\ -\cos(l) & 0 & -\sin(l) \end{bmatrix}$$

c) rotate about G3 (Down) through wander angle (A):

$$DCM3 = \begin{bmatrix} \cos(A) & \sin(A) & 0 \\ -\sin(A) & \cos(A) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

combining these, the E to G DCM, C_E^G = DCM2 . DCM1 , is :

$$C_E^G = \begin{bmatrix} -\sin(l)\cos(L) & -\sin(l)\sin(L) & \cos(l) \\ -\sin(L) & \cos(L) & 0 \\ -\cos(l)\cos(L) & -\cos(l)\sin(L) & -\sin(l) \end{bmatrix}$$

the E to W DCM, C_E^W = DCM3 . C_E^G is, (putting c for cos and s for sin):

$$C_E^W = \begin{bmatrix} -c(A)s(l)c(L)-s(A)s(L) & -c(A)s(l)s(L)+s(A)c(L) & c(A)c(l) \\ s(A)s(l)c(L)-c(A)s(L) & s(A)s(l)s(L)+c(A)c(L) & -s(A)c(l) \\ -c(l)c(L) & -c(l)s(L) & -s(l) \end{bmatrix}$$

A 3.2 Quaternions/Euler Angles

The quaternion equivalents of the above may be written:

$$\overline{Q1} = \begin{bmatrix} \cos\tfrac{1}{2}(L) \\ 0 \\ 0 \\ \sin\tfrac{1}{2}(L) \end{bmatrix} \qquad \overline{Q2} = \begin{bmatrix} \cos\tfrac{1}{2}(90+l) \\ 0 \\ -\sin\tfrac{1}{2}(90+l) \\ 0 \end{bmatrix} \qquad \overline{Q3} = \begin{bmatrix} \cos\tfrac{1}{2}(A) \\ 0 \\ 0 \\ \sin\tfrac{1}{2}(A) \end{bmatrix}$$

combining these by quaternion "multiplication" (appendix 2):

$$\overline{QEW} = [\ \overline{Q1}(*)\overline{Q2}\ ](*)\overline{Q3} = \begin{bmatrix} \cos\tfrac{1}{2}(90+l)\cos\tfrac{1}{2}(L+A) \\ \sin\tfrac{1}{2}(90+l)\sin\tfrac{1}{2}(L-A) \\ -\sin\tfrac{1}{2}(90+l)\cos\tfrac{1}{2}(L-A) \\ \cos\tfrac{1}{2}(90+l)\sin\tfrac{1}{2}(L+A) \end{bmatrix}$$

From a comparison of the C_E^W equation above with the quaternion expansion in appendix 2, it is found that:

$$\tan(LON) = \frac{-CEW32}{-CEW31} = \frac{-(QEW2.QEW3-QEW0.QEW1)}{-(QEW1.QEW3+QEW0.QEW2)}$$

$$\sin(LAT) = -CEW33 = 1 - 2(QEW1^2 + QEW2^2)$$

and $$\tan(A) = \frac{-CEW23}{CEW13} = \frac{-(QEW0.QEW1+QEW2.QEW3)}{(QEW1.QEW3-QEW0.QEW2)}$$

except at the poles, where LAT and A are undefined.

Appendix 4 - Navigation (G or W) to Body Transformation.

Rotations are positive in the sense Navigation to Body. The wander angle here is defined clockwise positive about Down, so Yaw is defined:

Heading = Wander angle + Yaw

A 4.1 Direction Cosines/Euler Angles.

a) rotate about Down through Yaw (Y):

$$DCM1 = \begin{bmatrix} \cos(Y) & \sin(Y) & 0 \\ -\sin(Y) & \cos(Y) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

b) rotate about new elev. through Elevation (E):

$$DCM2 = \begin{bmatrix} \cos(E) & 0 & -\sin(E) \\ 0 & 1 & 0 \\ \sin(E) & 0 & \cos(E) \end{bmatrix}$$

c) rotate about Bank through Bank (B):

$$DCM3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(B) & \sin(B) \\ 0 & -\sin(B) & \cos(B) \end{bmatrix}$$

combining these, the N to B DCM is $C_N^B =$ DCM3.DCM2.DCM1, and the B to N DCM, C_B^N is the transpose of that (putting c for cos and s for sin):

$$C_B^N = \begin{bmatrix} c(E)c(Y) & s(B)s(E)c(Y)-c(B)s(Y) & c(B)s(E)c(Y)+s(B)s(Y) \\ c(E)s(Y) & s(B)s(E)s(Y)+c(B)c(Y) & c(B)s(E)s(Y)-s(B)c(Y) \\ -s(E) & s(B)c(E) & c(B)c(E) \end{bmatrix}$$

A 4.2 Quaternion/Euler Angles.

The quaternion equivalents of the above may be written:

$$\overline{Q1} = \begin{bmatrix} \cos\tfrac{1}{2}(Y) \\ 0 \\ 0 \\ \sin\tfrac{1}{2}(Y) \end{bmatrix} \qquad \overline{Q2} = \begin{bmatrix} \cos\tfrac{1}{2}(E) \\ 0 \\ \sin\tfrac{1}{2}(E) \\ 0 \end{bmatrix} \qquad \overline{Q3} = \begin{bmatrix} \cos\tfrac{1}{2}(B) \\ \sin\tfrac{1}{2}(B) \\ 0 \\ 0 \end{bmatrix}$$

combining these by quaternion "multiplication" (appendix 2):

$$\overline{QNB} = [\overline{Q1}(*)\overline{Q2}](*)\overline{Q3}$$

i.e.

$$\overline{QNB} = \begin{bmatrix} \cos\tfrac{1}{2}(Y)\cos\tfrac{1}{2}(E)\cos\tfrac{1}{2}(B) + \sin\tfrac{1}{2}(Y)\sin\tfrac{1}{2}(E)\sin\tfrac{1}{2}(B) \\ \cos\tfrac{1}{2}(Y)\cos\tfrac{1}{2}(E)\sin\tfrac{1}{2}(B) - \sin\tfrac{1}{2}(Y)\sin\tfrac{1}{2}(E)\cos\tfrac{1}{2}(B) \\ \cos\tfrac{1}{2}(Y)\sin\tfrac{1}{2}(E)\cos\tfrac{1}{2}(B) + \sin\tfrac{1}{2}(Y)\cos\tfrac{1}{2}(E)\sin\tfrac{1}{2}(B) \\ -\cos\tfrac{1}{2}(Y)\sin\tfrac{1}{2}(E)\sin\tfrac{1}{2}(B) + \sin\tfrac{1}{2}(Y)\cos\tfrac{1}{2}(E)\cos\tfrac{1}{2}(B) \end{bmatrix}$$

From a comparison of the C_B^N equation above with the quaternion expansion in appendix 2, it is found that:

$$\tan(Y) = \frac{CBN21}{CBN11} = \frac{2(QNB0.QNB3 + QNB1.QNB2)}{1 - 2(QNB2^2 + QNB3^2)}$$

$$\sin(E) = -CBN31 = -2(QNB1.QNB3 - QNB0.QNB2)$$

and

$$\tan(B) = \frac{CBN32}{CBN33} = \frac{2(QNB0.QNB1 + QNB2.QNB3)}{1 - 2(QNB1^2 + QNB2^2)}$$

except for Elevation angles of +/- 90 degrees, when Yaw and Bank are undefined. (N above may be G or W)

Appendix 5 - Frame Rates

Let Ω_E be (scalar) Earth rate about its polar axis:

then $[\underline{\omega}_{IE}]^E = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Omega_E,$

also, $[\underline{\omega}_{IE}]^G = C_E^G[\underline{\omega}_{IE}]^E = \begin{bmatrix} \cos(LAT) \\ 0 \\ -\sin(LAT) \end{bmatrix} \Omega_E,$

and, $[\underline{\omega}_{IE}]^W = C_E^W[\underline{\omega}_{IE}]^E = \begin{bmatrix} \cos(A)\cos(LAT) \\ -\sin(A)\cos(LAT) \\ -\sin(LAT) \end{bmatrix} \Omega_E,$

If G moves over the Earth with $\dot{LON}$ and $\dot{LAT}$, then:

$$[\underline{\omega}_{EG}]^G = \begin{bmatrix} \dot{LON}.\cos(LAT) \\ -\dot{LAT} \\ -\dot{LON}.\sin(LAT) \end{bmatrix}$$

If W has (A) and $(\dot{A})$ relative to G, then:

$$[\underline{\omega}_{GW}]^W = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{A},$$

also, $[\underline{\omega}_{EG}]^W = C_G^W[\underline{\omega}_{EG}]^G = \begin{bmatrix} \dot{LON}.\cos(LAT)\cos(A) - \dot{LAT}.\sin(A) \\ -\dot{LON}.\cos(LAT)\sin(A) - \dot{LAT}.\cos(A) \\ -\dot{LON}.\sin(LAT) \end{bmatrix}$

therefore,

$$[\underline{\omega}_{EW}]^W = [\underline{\omega}_{EG}]^W + [\underline{\omega}_{GW}]^W = \begin{bmatrix} \dot{LON}.\cos(LAT)\cos(A) - \dot{LAT}.\sin(A) \\ -\dot{LON}.\cos(LAT)\sin(A) - \dot{LAT}.\cos(A) \\ \dot{A} - \dot{LON}.\sin(LAT) \end{bmatrix}$$

In this particular wander azimuth, $\dot{A} - \dot{LON}.\sin(LAT)$ is set to zero, i.e., the azimuth is rotated with Earth Rate Vertical.

If the G or W frame has a velocity relative to Earth surface, $\underline{V}_E$, then:

$$[\underline{V}_E]^G = \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix} \quad \text{and} \quad [\underline{V}_E]^W = \begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix}$$

A wander azimuth navigation algorithm gives $[\underline{V}_E]^W$: it is necessary to know $[\underline{\omega}_{EW}]^W$ in terms of $[\underline{V}_E]^W$.

If the local prime and meridional Earth radii are Rp and Rm, and vehicle height is H, then:

$$\dot{LAT} = \frac{V_N}{Rm+H} \quad \text{and} \quad \dot{LON} = \frac{V_E}{(Rp+H)\cos(LAT)}$$

$$\text{therefore,} \quad [\underline{\omega}_{EW}]^W = \begin{bmatrix} \dfrac{V_E \cos(A)}{Rp+H} - \dfrac{V_N \sin(A)}{Rm+H} \\ -\dfrac{V_E.\sin(A)}{Rp+H} - \dfrac{V_N.\cos(A)}{Rm+H} \\ 0 \end{bmatrix}$$

$$\text{Now,} \quad [\underline{V}_E]^G = C_W^G[\underline{V}_E]^W, \quad \text{i.e.,} \quad \begin{aligned} V_N &= V_X.\cos(A) - V_Y.\sin(A) \\ V_E &= V_X.\sin(A) + V_Y.\cos(A) \\ (\text{and } V_D &= V_Z) \end{aligned}$$

Putting these values of V_N and V_E into $[\underline{\omega}_{EW}]^W$:

$$[\underline{\omega}_{EW}]^W = \begin{bmatrix} -V_X\left[\dfrac{1}{Rm+H} - \dfrac{1}{Rp+H}\right].\sin(A)\cos(A) + V_Y\left[\dfrac{\cos^2(A)}{Rp+H} + \dfrac{\sin^2(A)}{Rm+H}\right] \\ -V_X\left[\dfrac{\sin^2(A)}{Rp+H} + \dfrac{\cos^2(A)}{Rm+H}\right] + V_Y\left[\dfrac{1}{Rm+H} - \dfrac{1}{Rp+H}\right].\sin(A)\cos(A) \\ 0 \end{bmatrix}$$

This may be written:

$$[\underline{\omega}_{EW}]^W = \begin{bmatrix} -V_X DX + V_Y D2 \\ -V_X D1 + V_Y DX \\ 0 \end{bmatrix}$$

where:
$$D1 = \frac{\sin^2(A)}{Rp+H} + \frac{\cos^2(A)}{Rm+H}$$
$$D2 = \frac{\cos^2(A)}{Rp+H} + \frac{\sin^2(A)}{Rm+H}$$
$$DX = \left[\frac{1}{Rm+H} - \frac{1}{Rp+H}\right].\sin(A)\cos(A)$$

From reference 2,

$$Rm = \frac{Re(1 - \epsilon^2)}{(1 - \epsilon^2\sin^2(LAT))^{3/2}} \quad \text{and} \quad Rp = \frac{Re}{(1 - \epsilon^2\sin^2(LAT))^{1/2}}$$

where Re is Earth equatorial radius, and ϵ^2 is (Earth eccentricity)^2.

D1, D2, and DX may be calculated to whatever degree of accuracy is required; for a first approximation, put:

$$Rm = Re + rm \quad \text{i.e.,} \quad rm = Re.\epsilon^2(\tfrac{3}{2}\sin^2(LAT) - 1), \qquad \text{(approx)}$$

and

$$Rp = Re + rp \quad \text{i.e.,} \quad rp = Re.\epsilon^2.\tfrac{1}{2}\sin^2(LAT). \qquad \text{(approx)}$$

Then,

$$\frac{1}{Rp+H} = \frac{1}{Re}\left[1 - \frac{rp+H}{Re}\right] \qquad \text{(approx)}$$

and

$$\frac{1}{Rm+H} = \frac{1}{Re}\left[1 - \frac{rm+H}{Re}\right] \qquad \text{(approx)}$$

Substituting these into D1, D2 and DX, and rearranging:

$$D1 = \frac{1}{Re}\left[1 - \frac{H}{Re} - \tfrac{1}{2}\epsilon^2(\ \sin^2(LAT) - 2\cos^2(LAT)\cos^2(A)\ )\right]$$

$$D2 = \frac{1}{Re}\left[1 - \frac{H}{Re} - \tfrac{1}{2}\epsilon^2(\ \sin^2(LAT) - 2\cos^2(LAT)\sin^2(A)\ )\right]$$

$$DX = \frac{\epsilon^2}{Re}\sin(A)\cos(A)\cos^2(LAT)$$

In terms of the CEW:

$$D1 = \frac{1}{Re}\left[1 - \frac{H}{Re} - \tfrac{1}{2}\epsilon^2(\ CEW33^2 - CEW13^2\ )\right]$$

$$D2 = \frac{1}{Re}\left[1 - \frac{H}{Re} - \tfrac{1}{2}\epsilon^2(CEW33^2 - CEW23^2\ )\right]$$

$$DX = -\frac{\epsilon^2}{Re}\ CEW13.CEW23$$

Appendix 6 - Gravity Support Correction

This is an extension of Reference (1), appendix 10, which was not included in that publication. Its effect is to eliminate much of the position and velocity error seen in fig. 5 of that publication, which were primarily in the East-West direction.

From that appendix 10 we have: $\delta\underline{V}_{EF} = (1 - \frac{1}{2}T[\underline{\omega}_{IE}x])\delta\underline{V}_{IF}$

Where $\delta\underline{V}_{EF}$ is the velocity increment relative to Earth caused by specific force, and $\delta\underline{V}_{IF}$ is the velocity increment relative to "inertial space" caused by specific force.

Consider that component of $\delta\underline{V}_{IF}$ caused by gravity support:

$$[\delta\underline{V}_{IFg}]^{G \text{ or } W} = \begin{bmatrix} 0 \\ 0 \\ -gT \end{bmatrix}$$

For this only, $$\delta\underline{V}_{EFg} = (1 - \frac{1}{2}T[\underline{\omega}_{IE}x])\delta\underline{V}_{IFg}$$

Now, $$[\underline{\omega}_{IE}]^W = C_E^W[\underline{\omega}_{IE}]^E = \begin{bmatrix} CEW13 \\ CEW23 \\ CEW33 \end{bmatrix} \Omega_E$$

Therefore, $$[\delta\underline{V}_{EFg}]^W = [\delta\underline{V}_{IFg}]^W - \frac{1}{2}T[\underline{\omega}_{IE}]^W x [\delta\underline{V}_{IFg}]^W$$

i.e. $$[\delta\underline{V}_{EFg}]^W = [\delta\underline{V}_{IFg}]^W - \frac{1}{2}gT^2\Omega_E \begin{bmatrix} -CEW23 \\ CEW13 \\ 0 \end{bmatrix} .$$

So, in order to correct for the effects of gravity support, we subtract AX10MOD from the velocity increment in Wander axes, where:

$$\begin{bmatrix} AX10MOD1 \\ AX10MOD2 \\ AX10MOD3 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}gT^2\Omega_E.CEW23 \\ \frac{1}{2}gT^2\Omega_E.CEW13 \\ 0 \end{bmatrix} \quad \begin{matrix} (= \frac{1}{2}gT^2\Omega_E\cos(LAT)\sin(A)) \\ (= \frac{1}{2}gT^2\Omega_E\cos(LAT)\cos(A)) \\ \end{matrix}$$

N.B. In Geographic axes, (A = 0), we only have AX10MOD2 = $\frac{1}{2}gT^2\Omega_E\cos(LAT)$

Appendix 7 - One Shot Coarse Alignment.

Simulated sensors' measurements of gravity support and Earth rate are used to estimate the Wander to Body quaternion $\overline{QWB}$. The system is static relative to Earth, and sensor quantization is not included. (A somewhat idealized situation!) The sequence of operation is:

1) Calculate simulated sensor outputs, using the known "true" attitude of the system and the required sensor errors (misalignments, scale factors, fixed biases).

2) Use simulated accelerometer outputs to calculate the (Body Level) to Body axes quaternion $\overline{QLB}$.

3) Use $\overline{QLB}$ to calculate simulated levelled gyro outputs, and use these to calculate the Geographic to (Body Level) axes quaternion $\overline{QGL}$.

4) Calculate $\overline{QGB} = \overline{QGL}(*)\overline{QLB}$; use required wander angle to get $\overline{QWG}$, then $\overline{QWB} = \overline{QWG}(*)\overline{QGB}$.

A 7.1 Simulated sensor outputs.

From initialization data,

$$[g]^G = \begin{bmatrix} 0 \\ 0 \\ g3 \end{bmatrix}, \text{ and } [\underline{\omega}_{IE}]^G = \begin{bmatrix} \cos(LAT) \\ 0 \\ -\sin(LAT) \end{bmatrix} \Omega_E,$$

and C_G^B may be obtained from the "true" attitude angles.

If Ts is sensor cycle time, then:

$$\text{"perfect" accelerometer output} = -[g]^B.Ts = -C_G^B[g]^G.Ts$$

$$\text{"perfect" gyroscope output} = [\underline{\omega}_{IE}]^B.Ts = C_G^B[\underline{\omega}_{IE}]^G.Ts$$

These may then be corrupted with sensor errors in the same way as is done in the sensors simulation.

A 7.2 Levelling

Consider an axis, perpendicular to the plane containing B3 and the local vertical. The Body Level coordinate frame (L) is defined by rotating the Body set about that axis until B3 is coincident with Down. The angle (b) may be defined as the magnitude of that rotation, positive from Level to Body.

Let the simulated accelerometers outputs be $\begin{bmatrix} A1 \\ A2 \\ A3 \end{bmatrix}$;

these may be normalized to 1 by dividing each component by $\sqrt{A1^2 + A2^2 + A3^2}$;

let the result be $\begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix}$.

From figure A 7.1, it may be seen that cos(b) = -a3.

Consider the plane containing B1, B2, OH, and the levelling axis; as shown in figure A 7.2, the angle between B1 and the levelling axis is (c). The unit vector $[\underline{e}]$ along the levelling axis is, in Body coordinates, given by:

$$[\underline{e}] = \begin{bmatrix} e1 \\ e2 \\ e3 \end{bmatrix} = \begin{bmatrix} \cos(c) \\ \sin(c) \\ 0 \end{bmatrix} .$$

From the diagram, it may be seen:

$$\cos(c) = \frac{-a1}{\sqrt{a1^2 + a2^2}} , \quad \sin(c) = \frac{a2}{\sqrt{a1^2 + a2^2}}$$

From the definition of the quaternion (eg, reference 1, appendix 2):

$$\overline{QLB} = \begin{bmatrix} \cos\frac{1}{2}(c) \\ e1.\sin\frac{1}{2}(c) \\ e2.\sin\frac{1}{2}(c) \\ e3.\sin\frac{1}{2}(c) \end{bmatrix} = \frac{1}{\sqrt{2(1-a3)}} \begin{bmatrix} 1-a3 \\ -a2 \\ a1 \\ 0 \end{bmatrix}$$

A 7.3 Geographic to Body Level Quaternion.

The gyro output vector is normalized to its correct magnitude ($\Omega_E.Ts$) , and is then obtained in Level axes coordinates using $\overline{QLB}$:

$$[\underline{gyros}]^L = \overline{QLB}(*)[\underline{gyros}]^B(*)\overline{QLB}^{-1} = \begin{bmatrix} G11 \\ G12 \\ G13 \end{bmatrix} \qquad (\text{"Gyro level"})$$

Let (d) be the angle from North to (Body Level)1 axis, as in figure A 7.3. It can be seen from the figure that

$$\cos(d) = \frac{G11}{\sqrt{G11^2 + G12^2}}$$

The unit vector [$\underline{e}$] for the rotation from Geographic to Body Level is on the Down axis, i.e.

$$[\underline{e}] = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

From the definition of the quaternion:

$$\overline{QGL} = \begin{bmatrix} \cos\tfrac{1}{2}(d) \\ 0 \\ 0 \\ \sin\tfrac{1}{2}(d) \end{bmatrix} = \begin{bmatrix} \sqrt{\tfrac{1}{2}(1 + \cos(d))} \\ 0 \\ 0 \\ \sqrt{\tfrac{1}{2}(1 - \cos(d))} \end{bmatrix}$$

The Geographic to Body quaternion may now be obtained: $\overline{QGB} = \overline{QGL}(*)\overline{QLB}$.

A 7.4 Wander to Body Quaternion

If (A) is the required initial wander angle, defined as positive from G to W, then:

$$\overline{QWG} = \begin{bmatrix} \cos\tfrac{1}{2}(A) \\ 0 \\ 0 \\ -\sin\tfrac{1}{2}(A) \end{bmatrix},$$

the required result may now be obtained:

$$\overline{QWB} = \overline{QWG}(*)\overline{QGB}$$

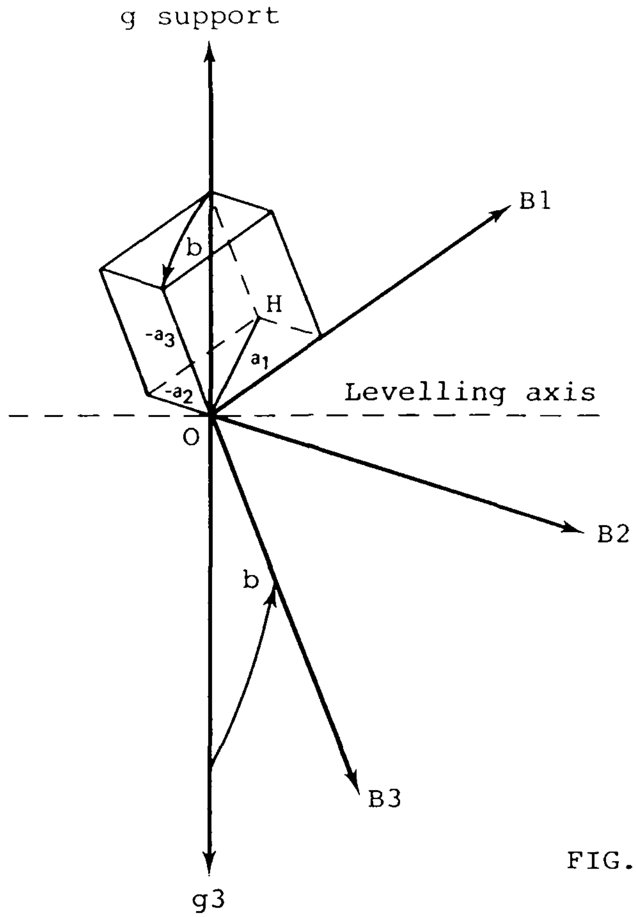

FIG. A7.1

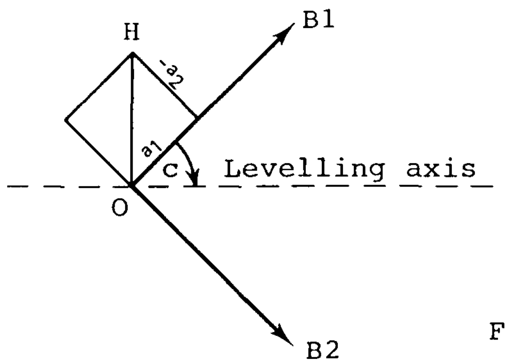

FIG. A7.2

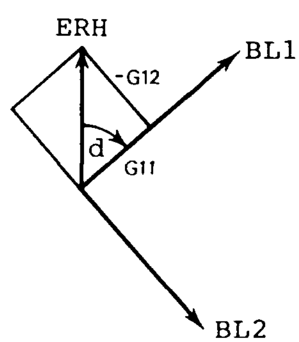

FIG. A7.3

DISTRIBUTION

AUSTRALIA

Department of Defence

Defence Central
- Chief Defence Scientist
- Assist Chief Defence Scientist, Operations (shared copy)
- Assist Chief Defence Scientist, Policy (shared copy)
- Director, Departmental Publications
- Counsellor, Defence Science (London) (Doc Data Sheet Only)
- Counsellor, Defence Science (Washington) (Doc Data Sheet Only)
- S.A. to the Thailand Military R and D Centre (Doc Data Sheet Only)
- S.A. to the DRC (Kuala Lumpur) (Doc Data Sheet Only)
- OIC TRS, Defence Central Library
- Document Exchange Centre, DISB (18 copies)
- Joint Intelligence Organisation
- Librarian H Block, Victoria Barracks, Melbourne
- Director General - Army Development (NSO) (4 copies)

Aeronautical Research Laboratory
- Director
- Library
- Superintendent - Aircraft Systems
- Author: R.B. Miller
- PO/FMG (4 copies)

Materials Research Laboratory
- Director/Library

Defence Science & Technology Organisation - Salisbury
- Library

Air Force Office
- Air Force Scientific Adviser (Doc Data sheet only)

Navy Office
- Navy Scientific Adviser (Doc Data sheet only)

Army Office
- Scientific Adviser - Army (Doc Data sheet only)

Universities and Colleges
- Adelaide
  - Barr Smith Library
- Flinders
  - Library
- La Trobe
  - Library

Melbourne
Engineering Library

Monash
Hargrave Library

Newcastle
Library

New England
Library

Sydney
Engineering Library

NSW
Physical Sciences Library
Library, Australian Defence Force Academy

Queensland
Library

Tasmania
Engineering Library

Western Australia
Library

RMIT
Library

UNITED KINGDOM

Royal Aircraft Establishment, Farnborough
Library
Superintendent, RN1 Division
Mr. R.A. Frampton, RN1 Division

SPARES (10 copies)
TOTAL (64 copies)

AL 149 DEPARTMENT OF DEFENCE

DOCUMENT CONTROL DATA

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

1a. AR NUMBER	1b. ESTABLISHMENT NUMBER	2. DOCUMENT DATE	3. TASK NUMBER
AR-004-594	ARL-SYS-TM-101	JULY 1989	DST 86/064

4. TITLE	5. SECURITY CLASSIFICATION (PLACE APPROPRIATE CLASSIFICATION IN BOX(S) IE. SECRET (S), CONF.(C) RESTRICTED (R), UNCLASSIFIED (U)).	6. NO. PAGES	7. NO. REFS.
SINS1 - A MODEL OF A STRAPDOWN INERTIAL NAVIGATION SYSTEM	U DOCUMENT; U TITLE; U ABSTRACT	51	3

8. AUTHOR(S)	9. DOWNGRADING/DELIMITING INSTRUCTIONS
R.B. MILLER	Not applicable

10. CORPORATE AUTHOR AND ADDRESS	11. OFFICE/POSITION RESPONSIBLE FOR:
AERONAUTICAL RESEARCH LABORATORY P.O. BOX 4331, MELBOURNE VIC 3001	SPONSOR DSTO SECURITY - DOWNGRADING - APPROVAL CSYD

12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT) Approved for Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH ASDIS, DEFENCE INFORMATION SERVICES BRANCH, DEPARTMENT OF DEFENCE, CAMPBELL PARK, CANBERRA, ACT 2601

13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO....
No limitations

13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE [X] UNRESTRICTED OR [ ] AS FOR 13a.

14. DESCRIPTORS	15. DRDA SUBJECT CATEGORIES
Inertial navigation Strapdown Navigation Systems Computer programs	0076D

16. ABSTRACT

SINS1 is a simulation of a strapdown INS, written as a Fortran subroutine to be called at intervals by another program. Sensor errors, system initialization characteristics, and navigation algorithm operation may be selected by the User. Each call corresponds to the passage of a set time interval, and the calling program provides environmental dynamics information about the angular velocity of, and specific force on, the simulated INS. As required by the User, the calling program obtains from SINS1, the

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT.)

INS outputs of position, attitude, velocity, and other data. An example of a calling program is also described.

17. IMPRINT

AERONAUTICAL RESEARCH LABORATORY, MELBOURNE

18. DOCUMENT SERIES AND NUMBER	19. COST CODE	20. TYPE OF REPORT AND PERIOD COVERED
Aircraft Systems Technical Memorandum 101	725326	Work done 1985-86

21. COMPUTER PROGRAMS USED

22. ESTABLISHMENT FILE REF.(S)

23. ADDITIONAL INFORMATION (AS REQUIRED)