Bellcomm / ~~[illegible]~~

MAR 17 1971

DRA





DR

**TRW**®

**SYSTEMS GROUP**

ONE SPACE PARK • REDONDO BEACH, CALIFORNIA



28

5522.7-71-47

5 March 1971

# DESIGN CONCEPT (MILESTONE 2) FOR SPACE VEHICLE ENTRY AND LANDING NAVIGATION PROGRAMS


Prepared by: L. S. Diamant
Information Systems Analysis Section

Approved by: D. M. Detchmendy, Manager
MSC/TRW Task A-525

Approved by: R. K. Petersburg, Manager
Systems Evaluation Department

Approved by: D. G. Saile, Assistant Project Manager
Guidance and Performance
Mission Trajectory Control Program

ACKNOWLEDGEMENT

The following TRW personnel assisted in the preparation of this document through written and/or verbal contributions: L. H. Bradford, D. M. Detchmendy, R. H. Hoffman, R. H, Kidd, L. A. Phillips. Many helpful comments from MSC personnel have also been received and incorporated.

## 1. INTRODUCTION

This report represents the Milestone 2 Document for the programs being developed by Task A-525, Space Vehicle Entry and Landing Navigation. The function of the Milestone 2 Document is to provide the following information:

(a) The detailed specifications of the programs to be developed and

(b) The functional flow diagrams of the programs.

The information in (a) is necessary to determine if the programs satisfy the requirements of the task order. The information in (b) is necessary to determine if the structure of the program will satisfy the specifications. After insuring positive responses to the above, via the critical design review, then this document will serve as a starting point for the detailed design of the programs.

An additional purpose of this document is to expose the assumptions that were made in the functional design. These assumptions need to be examined, via the critical design review, to insure that any future program requirements expansion will be amenable to the proposed structure.

Section 2 presents the general design criteria, overall functions, and interfaces between the three programs under consideration. The following three Sections provide the information defined by items (a) and (b) for the three programs - External Observations Program, Noisy Observations Program, and Navigation Analysis Program respectively.

## 2. GENERAL DESIGN CRITERIA, FUNCTIONS, AND CONSTRAINTS

The basic philosphy underlying the design of each of the three programs to be described below is that the input trajectory tape modulated by a specified data-select schedule represents the "driving function" or "clock". Thus, after initialization, each program operates by reading in the next epoch point data from the tape, comparing it against the indicated schedule and operating as defined by the input (and program function). The next epoch point is read in and so on. The ground rule under which the navigation analysis capability will be built is that the development of new coding will be minimized by extracting and using the relevent portions of existing programs.

### EXTERNAL OBSERVATIONS PROGRAM

This program will generate the actual (noiseless) observations for the external data types (beacons and altimeter) based upon the input trajectory and data defining the beacon locations, data types, and constraints. The output will be a tape containing the input trajectory data plus the specified external observations and pertinent partials.

The program will operate by generating for each epoch point specified by input all of the external observations (and accompanying partials) which satisfy the constraints. The structure and routines of this program will be derived from the observation link of the HOPE program.

### NOISY OBSERVATIONS PROGRAM

This program will generate the simulated (noise plus biases) observations using the output tape of the External Observations Program as an input. The output will be a tape containing the original trajectory data (state vectors, integrated accelerations, and times) plus the simulated observations at each epoch point defined by the input. The simulated observations will include both the external data types and the platform-related accelerometer data. Inputs will be required to define the statistics of the various error sources.

### NAVIGATION ANALYSIS PROGRAM

This program represents the entry and landing navigation analysis capability. It will operate on the output tape of the External Observations Program, and sharing common modules with the Noisy Observations Program, provide an analysis capability of the sensitivity and Monte Carlo type. Inputs will be required to define types of run and output desired, the statistics of the various error sources, the tracking schedule to be used, the apriori data, and the filter constants. The linear error analysis requirement of the task order will be satisfied by developing techniques and a plan for incorporating the computations into this program. Plans are being firmed to merge the Noisy Observations

Program and the Navigation Analysis Program into one physical package capable of performing each function separately.

In terms of coordinate systems and units, each of the programs will use mean of 1950 and will output in kilometers (km) and kilometers per second (km/sec). The major interfaces between these three programs are shown in Figure 2-1.

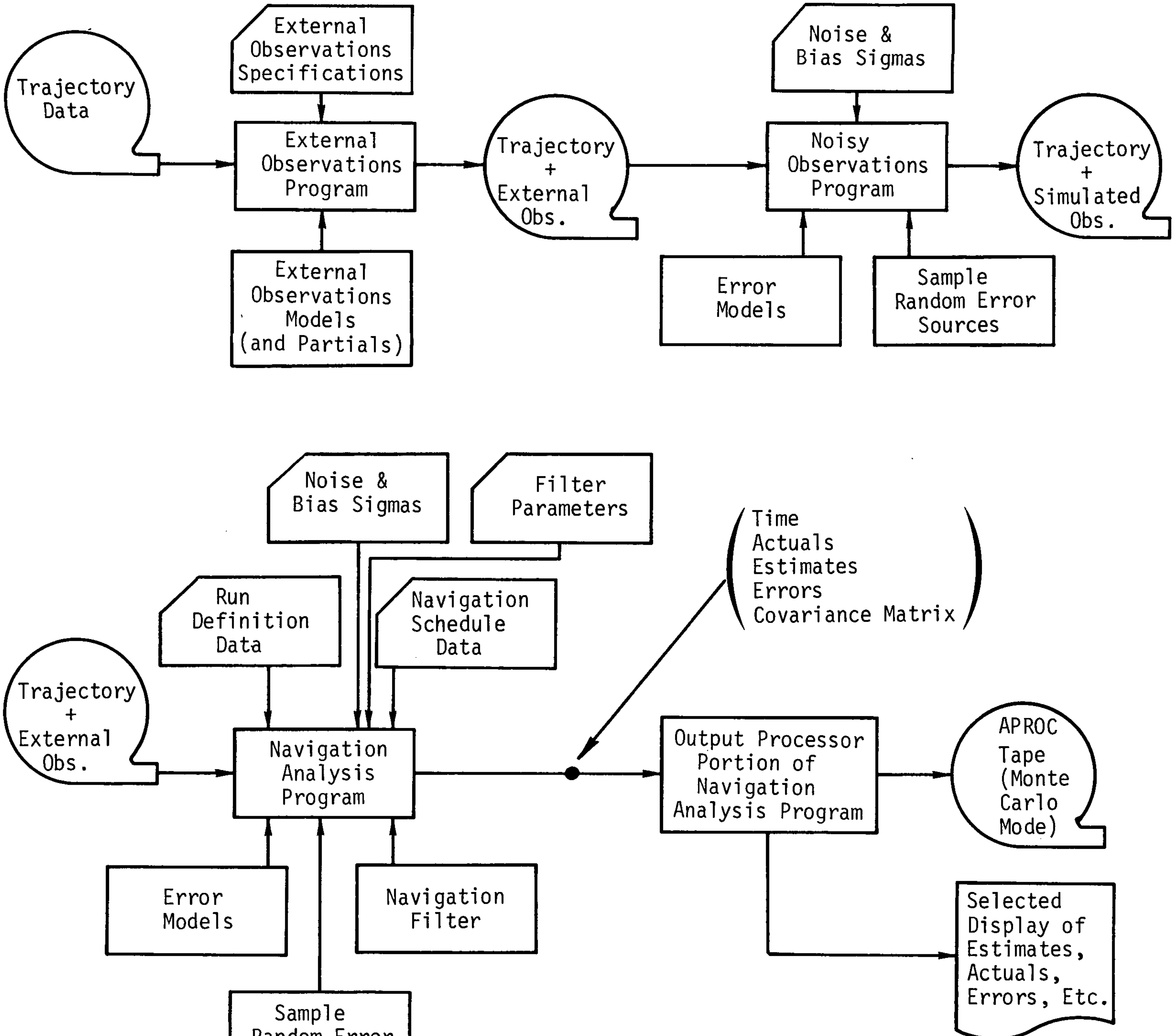


Figure 2-1 BASIC PROGRAM INTERFACES

# 3. EXTERNAL OBSERVATIONS PROGRAM

## 3.1 FUNCTIONAL FLOW

The purpose of this program is to generate the available external observations of the true state of the vehicle as described by the input trajectory tape and as selected by the input data rate scheduled. The observations are error-free; noise and bias errors are generated in the Noisy Observations Program and are added to the error-free observations. In addition to the observations, certain partial derivatives will be generated and output for use in the Noisy Observations Program.

Essentially, this program is an extraction of the "observations link" in the HOPE program and is presented schematically in Figure 3-1. More explicit detail is waived due to the availability of HOPE documentation and the functional status of the routines involved.

Input to the program is of three types:

- Beacon location and special data type identification - the beacons are identified by their location and any special data types as identified by name (e.g. ALTIMETER). At this point, the logic to handle special data types will be built into the program modules.

- Data type specification - the type of data to be taken by the beacons and special data types will be flaged (doppler, range, etc.) and appropriate station parameters will be specified (frequencies, doppler count, etc.). Data rates may also be specified.

- Data constraints - all constraints associated with the tracking instrumentation will be identified and valued (e.g. minimum elevation angle). General constraints such as blackout and antenna occultation will be calculated internally.

The program output will consist of:

- Header data - all those data pertinent to identifying the beacons, special data types and their location.

- Data - the input times, states and accumulated $\Delta V$'s will be recorded as well as the computed observations for each available data type and the pertinent partial derivatives.

Further breakdown of the input/output variables and the input/output format is presented in Section 3.12.

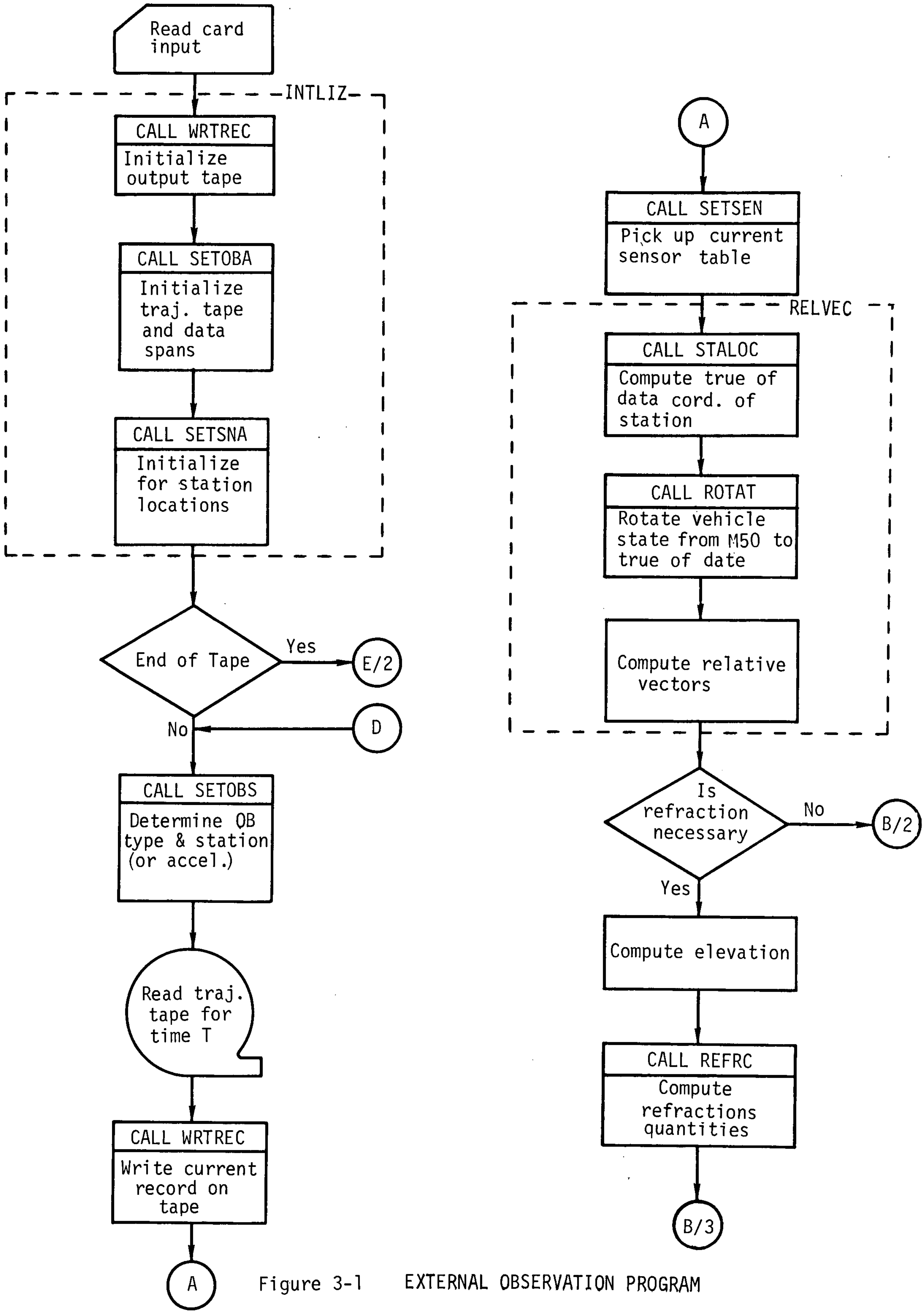


Figure 3-1 EXTERNAL OBSERVATION PROGRAM

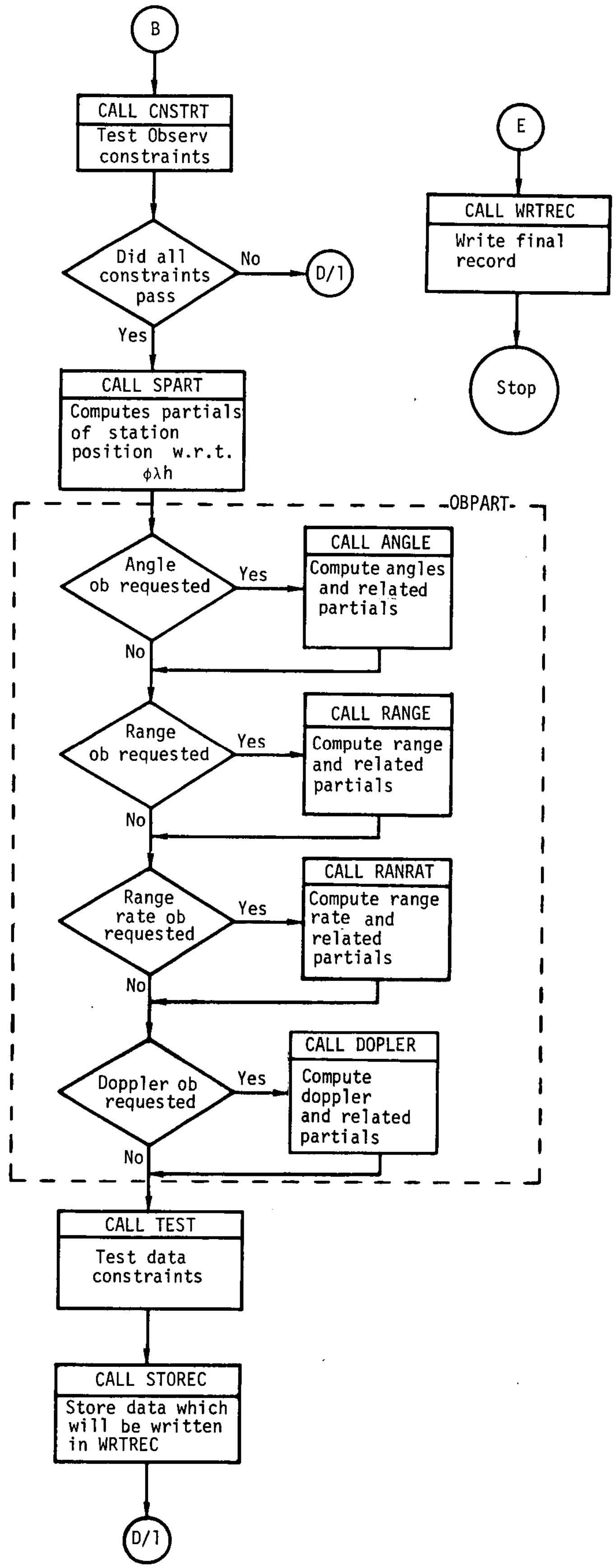


Figure 3-1 (Continued)

Functionally, the flow logic performs the following tasks:

- initialization of all necessary parameters
- positioning of the data tape to the next requested observation or acceleration time (contrary to previous designs, the "clock" is being driven by an input data sampling schedule).
- determination of the station, data type and relative state vectors.
- test for and ignore observability constraint violators (earth occultation, elevation angle, etc.)
- compute observations and partials
- test for and ignore data constraint violators
- record valid observations and accumulated $\Delta V$'s.

Expanded descriptions of the major modules and submodules follows.

## 3.2 INTLIZ

The purpose of this subroutine is to initialize all the pertinent quantities in the External Observations Program. It is composed of three submodules.

### 3.2.1 WRTREC

This module initializes the output tape by recording the Header Data as described above. The number of stations, number of data types, station names and station locations are written in a fixed format to be used in the Noisy Observations Program.

### 3.2.2 SETOBA

This routine initializes the input data tape and data span parameters. This, in essence, is the creation of the "driving function" which will be used to position the data tape at relevent events only.

### 3.2.3 SETSNA

The initialization of all station location calculations occurs in this submodule.

## 3.3 SETOBS

This module represents the "clock" or "driving function" of the program. Provision is made in this module to interpret the input in order to create a schedule of relevent times - those at which some type of data is to be taken or copied. Furthermore, delineation of exactly which data is derived and this schedule are utilized to drive the trajectory tape chronologically to each relevant time tag and create a sensor selection table to be used by module SETSEN in selecting the data generation mode.

## 3.4 WRTREC

This module copies the relevant information from the input trajectory tape to the output tape.

## 3.5 SETSEN

All desired sensors for each station or special data type at the current epoch are catagorized and used to drive the station location, relative state vector and observation generation functions. The outside loop is the station identification which is used to compute the station state. Internal to that loop is the calculation of the relative state between the station and the vehicle (RELVEC) inside of which is the data and partials generation loop (OBPART).

## 3.6 RELVEC

This routine computes the relative state vector between the station and the vehicle. It is assisted by submodules:

3.6.1 STALOC - computes the true of date coordinates of the station.

3.6.2 ROTAT - rotate the station vector into the desired coordinate system (mean of 1950).

## 3.7 REFRC

If it is determined that refraction calculations are to be performed. this routine performs that function. Details of the calculation are available in the HOPE documentation.

## 3.8 CNSTRT

All physical observability constraints based on the vehicle state and vehicle/beacon relative state are checked here to determine whether or not to continue the observations calculations.

This subroutine determines if a navigation measurement is feasible by performing the following tests applicable to the particular navigation system:

1. Propagation blackout test - vehicle altitude and velocity are compared with blackout limits to determine if electromagnetic transmission to the ground can be achieved.

2. Beacon transponder visibility test - to determine if the vehicle-transponder line-of-sight is occulted by the Earth.

3. Beacon transponder elevation angle test - to determine if the vehicle elevation angle relative to the transponder violates a maximum or minimum value.

4. Beacon antenna coverage test - to determine if the vehicle-transponder line-of-sight falls within the coverage limits of one of the on-board antennas.

5. Altimeter antenna coverage test - to determine if the vehicle local vertical falls within the coverage limits of the antenna.

### 3.9 SPART

If all physical constraints are passed, then partials of the cartesian state of each station with respect to its geographic location are computed. These will be later chained with the partials of the observations with respect to the cartesian state to form the desired partial of the observation with respect to the geographic location.

### 3.10 OBPART

The desired observations and the specified partials are generated according to the input parameters. Angle, range, range rate or doppler data may be generated by the subroutines:

#### 3.10.1 ANGLE

#### 3.10.2 RANGE

#### 3.10.3 RANRAT

#### 3.10.4 DOPLER

These subroutines are well documented in the HOPE literature.

### 3.11 TEST

A final test of the computed observations against the input data constraints for validity is made. These constraints are:

1. maximum range or altitude

2. minimum range or altitude

3. maximum range rate

All valid data are sorted and recorded by subroutine STOREC.

### 3.12 Input/Output

Input

| Variable | Dimension | Description |
|---|---|---|
| $INPUT | | |
| NOSTAT | N | Number of regular stations (beacons) |
| NOSPEC | M | Number of special navigational aids (altimeter, radar, etc.) |
| RNGFLG | N | Range measurement flag for each station: 0 = no range data, 1 = range data |
| DOPFLG | N | Doppler (or range rate) measurement flag for each station: 0 = no doppler data, 1 = doppler data |
| MINEL | N+M | Minimum elevation angle constraint for each instrument (may be used to define cone angle limits for special cases) |
| MAXEL | N+M | Maximum elevation angle constraint for each instrument (may be used to define cone angle limits for special cases) |
| MINR | N+M | Minimum range constraint for each device |
| MAXR | N+M | Maximum range constraint for each device. |
| MAXRR | N+M | Maximum range rate constriant for each device. |
| INOUT | 1 | Define input/output units |
| STFREQ | N | Transponder frequency for each regular station |

Input

| Variable | Dimension | Description |
|---|---|---|
| COUNT | N | Doppler count for each station |
| TRFREQ | 1 | Vehicle transmitter frequency |
| DATRAT | N | Data rate for each tracking interval |
| ACCRAT | 1 | Rate at which sensed $\Delta V$'s are to be stored on output tape |

$END

| Device | | Station Longitude | | Station Latitude | | Station Altitude | |
|---|---|---|---|---|---|---|---|
| 1 | 12 | 13 | 27 | 28 | 42 | 43 | 57 |

Example

```
$INPUT

    NOSTAT = 2
    NOSPEC = 1
    RNGFLG = 1,1,1
    DOPFLG = 1,0,1
    MINEL  = 5.,5.,0.
    MAXEL  = 90.,90.,15.
    MINR   = 0.,0.,0.
    MAXR   = 500.,500.,10.
    MAXRR  = 5.E3,0.,500.
    INOUT  = 3 (arbitrary at this point)
    STFREQ = 4.E9,4.E9
    COUNT  = .08
    TRFREQ = 4.E9
    DATRAT = 40., 40., 40.
    ACCRAT = .5

$END
```

| | | | |
|---|---|---|---|
| BEACON ONE | 80.17723 | 28.766 | 526. |
| BEACON TWO | 7.19702 | 4.6773 | 26. |
| ALTIMETER | | | |

OUTPUT

No of Beacons
No of Special Types
Beacon Name or Code
Longitude
Latitude
Altitude
o
o
o

for each beacon

last beacon

Altimeter
Other Special Types

Time
State Vector
Accumulated $\Delta v$'s
Data
range
doppler
Partials
range wrt station location
doppler wrt stations location
Data

for each beacon

for special types

# 4. NOISY OBSERVATIONS PROGRAM

## 4.1 FUNCTIONAL FLOW

The purpose of this program is to compute observations which represent the real world with error due to random noise and biases.

This program recognizes three types of errors:

1. Random - variable at each observation
2. Time invariant bias - statistically described by a distribution, but fixed for the duration of a given flight
3. Time variant bias - variable from observation to observation but predictable as a function of an initial value.

The program also has provision for input of a nonzero mean value of a bias for studies requiring such.

The input to the Noisy Observation Program mainly consists of the sigmas/means for all of the considered error sources. They are in three general catagories:

1. platform errors (includes input REFSMAT)
2. accelerometer errors
3. beacon and altimeter errors.

The program output is identical to that of the External Observations Program except the observation values are now corrupted. Further delineation of these input/output parameters is presented in Section 4.5.

For sensing errors, the modeling in NOBER is segmented into two main sections - the errors in platform alignment, as determined in PLTFRM and the accelerometer errors, as determined in VSINSD. NOBER returns the resultant sensed velocity considering the interactions of the misaligned platform and the erroneous accelerometer readings.

The modeling in XOBER is extremely simplified. At this point the ideal values for the observations have been calculated, and in the main, the computations of XOBER are concerned with adding the various contributions of error onto the original measurement. Those partial derivatives which have been transmitted via the input tape will be used to compute certain of the observational data errors.

It should be mentioned at this point that the Noisy Observations Program has been provided with the potential to further cull the input data. This capability, besides rendering the program more general, will allow a future merging of the Noisy Observations Program with the Navigational Analysis Program (Section 5.). Eventually, this program will be only a link in the analysis program. A functional flow diagram is presented in Figure 4-1.

## 4.2 RUREDY

This module is not truly a complete entity in itself, but a series of submodules designed to read the input data-select schedule, position the tape to the first valid point and to (for every subsequent record) position the tape for all selected data retrieval. The submodules used to accomplish this are named in the following sections.

### 4.2.1 POINTR

This submodule generates key pointer words to identify the desired data records by using the input data-select schedule and the tape heading record.

### 4.2,2 POSTAP

This routine merely positions the tape to the start of the trajectory segment or first record thereafter.

### 4.2.3 FINOBS

This routine finds the desired observations and partials using the pointer words generated by POINTR. Contrary to the other submodules in RUREDY, FINOBS is utilized for every input observation time.

## 4.3 XOBER

The purpose of XOBER is to use the input observations and external observation error model to calculate observations which are representative of real world measurements. For the most part XOBER operates on the assumption that the various sources of error are uncorrelated at a given time. In the case of a time-variant bias, the correlation of the bias at two different times is modeled by a variety of time-functional expressions.

The program first tests for the presence of valid data. If none is found, control is returned to XOBER. The module uses the value of a counter to determine the type and source of the particular observation.

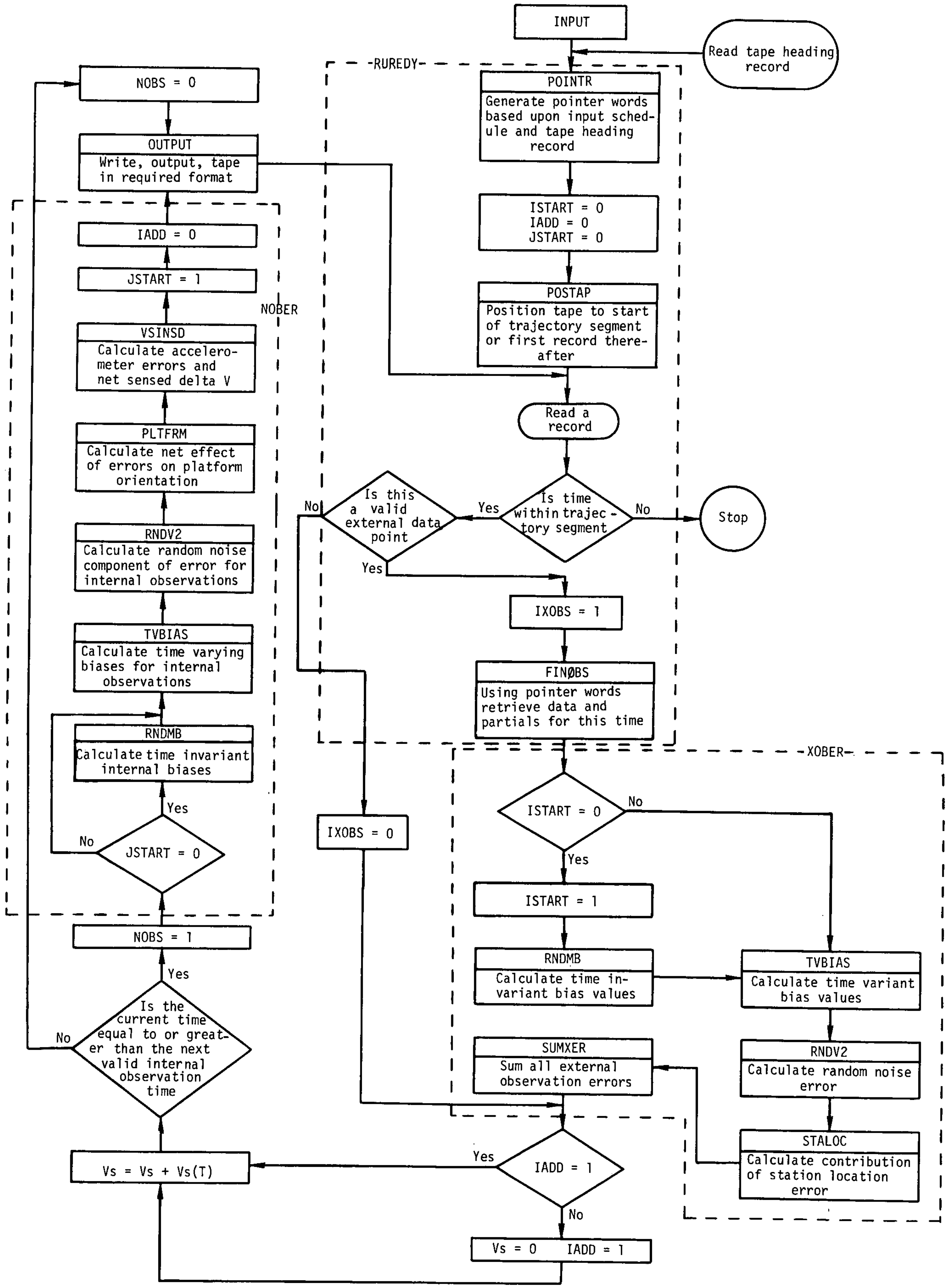


Figure 4-1 NOISY OBSERVATIONS PROGRAM

The program then interrogates the table of time-invariant biases generated by RNDMB. If a match is found then the bias value and/or bias mean are added directly to the input value. If no match is found then the source and data type must be located in the time-variant bias table generated by TVBIAS. A pointer will be found in this table to determine the particular function to be used in calculating the bias value. The current bias value and time are provided to the appropriate function and the sample value returned.

### 4.3.1 RNDMB

This module generates a table of time-invariant biases for all those data designated. The process involves the generation of a random number selected from a unit-variance, zero-mean gaussian distribution which is multiplied by the input bias standard deviation.

### 4.3.2 TVBIAS

The purpose of this routine is to generate the time-varying biases defined. These sample values for the biases will be generated sequentially as the result of a first order Gauss-Markov process. The required inputs are:

- Previous value and time
- Current time
- Time constant for the random process
- Standard deviation for the white noise.

### 4.3.3 RNDV2

The purpose of this submodule is to compute the sample noise values for the external observations. The approach is straightforward - a random number is calculated in accordance with a normal distribution of unit variance and multiplied by the standard deviation noise for a particular data type.

### 4.3.4 STALOC

The purpose of this module is to compute the contribution of station location errors to the observation value. A test will be made to see if the station location errors for this station are zero. If so the input observation to STALOC is returned unchanged.

Since the format of the output tape from the External Observations Program includes the partial derivative of the observation with respect to the station then all that is required is to multiply this vector by the sample station location errors.

## 4.4 NOBER

The purpose of this routine is to control the calculation of the sensed delta-v's. Time-varying biases and random noise values are calculated for each appropriate component of error at each time.

### 4.4.1 RNDMB

See Section 4.3.1

### 4.4.2 TVBIAS

See Section 4.3.2

### 4.4.3 RNDV2

See Section 4.3.3

### 4.4.4 PLTFRM

Subroutine PLTFRM will calculate the coordinate transformation matrix (Tp) which relates the true inertial platform orientation to the hypothetical platform axes orientation which is assumed to be inertially fixed by the input REFSMAT. The matrix (Tp) will be updated each time step ($\Delta T$) to account for platform rotation caused by gyro drifts. The equation for updating (Tp) will be a closed form solution which assumes a constant platform angular drift vector over the time interval $\Delta T$. The platform drift vector will be calculated as the average of the total gyro drift vectors computed for the present and previous time step. The total gyro drift for each axis will be composed of a randomly varying bias drift, a g-sensitive drift, and a $g^2$-sensitive (anisoelastic) drift.

PLTFRM will compute the initial transformation matrix $(Tp)_0$ using the input misalignment vector $\underline{\phi}_0$.

Inputs include:

- Initial platform misalignment vector $\underline{\phi}_0$
- g-sensitive and $g^2$-sensitive drift coefficients for each axis
- $\Delta \underline{V}$ vector from trajectory tape

- Bias drifts (each time step & initial)
- Time

Output will be:

- Platform misalignment matrix (Tp)
- Platform misalignment vector $\underline{\phi}$

This subroutine will be coded from existing equations.

### 4.4.5 VSINSD

Subroutine VSINSD will calculate the delta-velocity vector ($\Delta\underline{V}$) measured by the accelerometers in the true inertial platform coordinate system. The reference delta-velocity vector ($\Delta V$) from the trajectory tape is first transformed to the true platform coordinates using misalignment matrix (Tp) generated in subroutine PLTFRM. The delta-velocity error due to accelerometer errors is then added to the transformed $\Delta\underline{V}$ to produce the sensed velocity vector $\Delta\underline{V}$. The accelerometer errors for each axis will consist of a randomly varying bias, a scale factor error, a scale factor nonlinearity, an input axis misalignment, and a random output noise.

Inputs inlude:

- Inertial platform misalignment matrix (Tp)
- Reference $\Delta\underline{V}$ from trajectory tape
- Accelerometer scale factor error, scale factor
- Nonlinearity, and input axis misalignment coefficients.
- Accelerometer bias each time step
- Accelerometer noise

and the output will be:

- Sensed delta-velocity vector $\Delta\underline{V}$

This subroutine will be coded from existing equations.

## 4.5 INPUT/OUTPUT

### Input

```
$IN
   NOSTAT
   NOSPEC
$END
$PLTFRM         Platform - related errors (sigmas & means)

                    Biases                    Random

                    gyro bias                 noise
                    gyro g-sensitive          drift rate
                         coeff.
                    etc.                      etc.

   REFSMAT
$END
$ACCL           Accelerometer - related errors (sigmas & means)

                    Biases                    Random

                  accel. misalignment         noise
                  accel. bias
                  etc.                        etc.

$END
$BEACON         Beacon errors (sigmas & means)

                    Biases                    Random

                    Range                     noise
                    Doppler
                    etc.                      etc.

$END
$ALTMTR         Altimeter - related errors (sigmas & means)

$END
$SPEC
                Other special data type errors

$END
$NAVSCH

   STATNO               M             Station Number
   T1                   M             On times for station
   T2                   M             Off times for station
   ΔT                   M             Data rate
   R                    M             Range flag for each time span
   DOP                  M             Doppler flag for each time span
   ALT
$END
```

$ACCCYL

| | | |
|---|---|---|
| T1 | K | On times for acceleration sensing |
| T2 | K | Off times for acceleration sensing |
| $\Delta T$ | K | Data rates for each interval |

$END

Output

Same as External Observations Program except selected data is noised up and there are no partials.

# 5. NAVIGATION ANALYSIS PROGRAM

This program represents the entry and landing navigation analysis capability. It will utilize the output of the External Observations Program to perform various types of linear error analysis.

## 5.1 FUNCTIONAL FLOW

Figure 5-1 presents the functional flow diagram for the Navigational Analysis Program in terms of its major modules. This particular version allows for single case or multiple case (Monte Carlo) simulations to be performed through the input KMAX (maximum number of cycles).

An input navigation schedule (see 5.10) controls which portions of the observation data are to be processed with the navigation filter. The control of the data by the navigation schedule is performed by the POINTR module which is described in Section 4.2.1. The external observations taken from the input tape are noised up by a streamlined version of Noisy Observation Program (Section 4) before being filtered by the module. The sigmas for all error sources are input and provision is made for initial state vector uncertainty through an input state error covariance matrix.

The program output may take several forms, but for the present is assumed to be either 1) a fixed format for single-cycle simulations or 2) in an APROC compatable format (on tape) for multi-cycle Monte Carlo runs.

## 5.2 POINTR

See Section 4.2.1.

## 5.3 POSTAP

See Section 4.2.2.

## 5.4 FLTR00

This module will be used to initialize the navigational filter. Further definition of this function will be available as the filter being developed under Task A-524 is documented.

## 5.5 FINOBS

See Section 4.2.3.

## 5.6 XOBER

See Section 4.3.

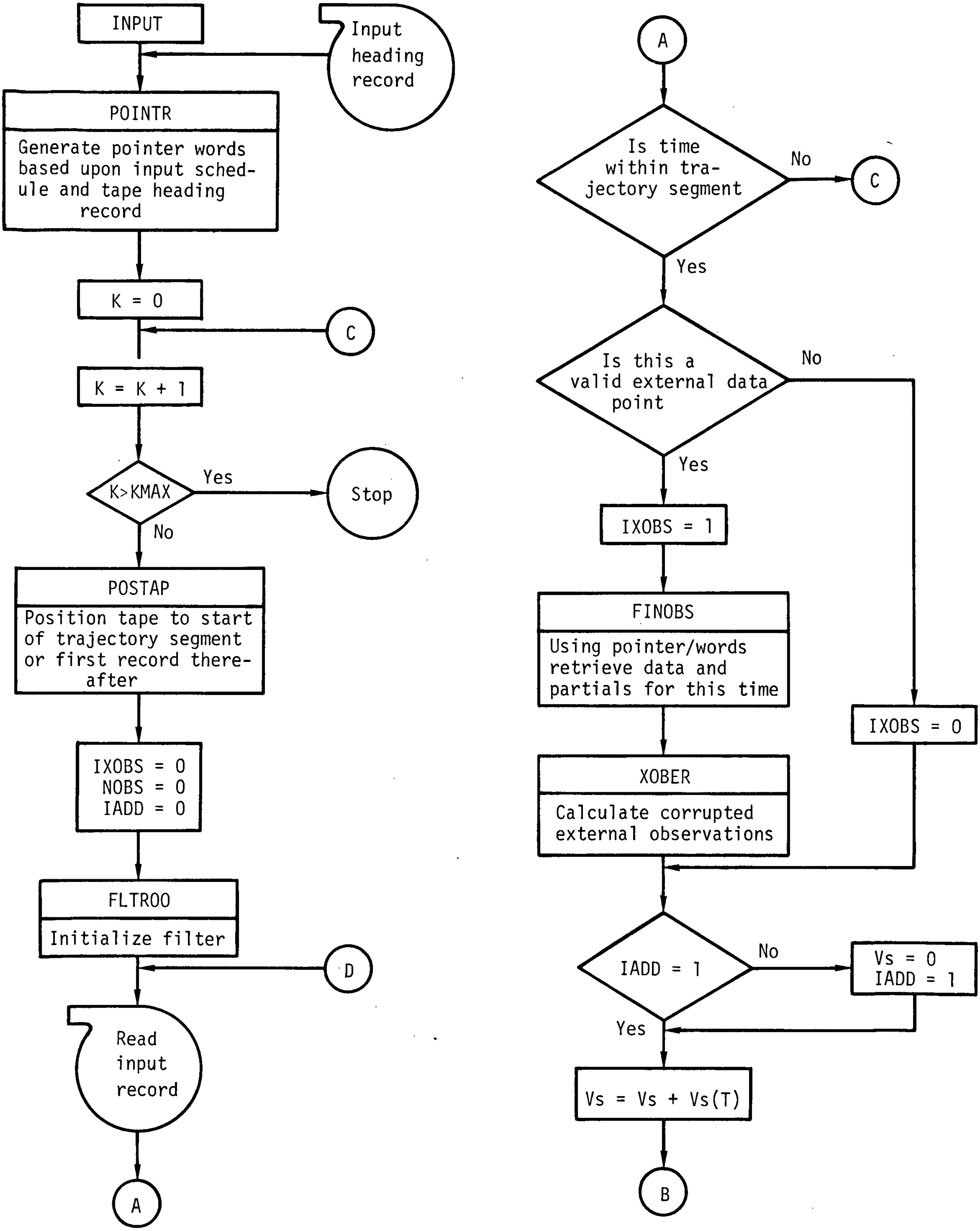


Figure 5-1 NAVIGATION ANALYSIS PROGRAM

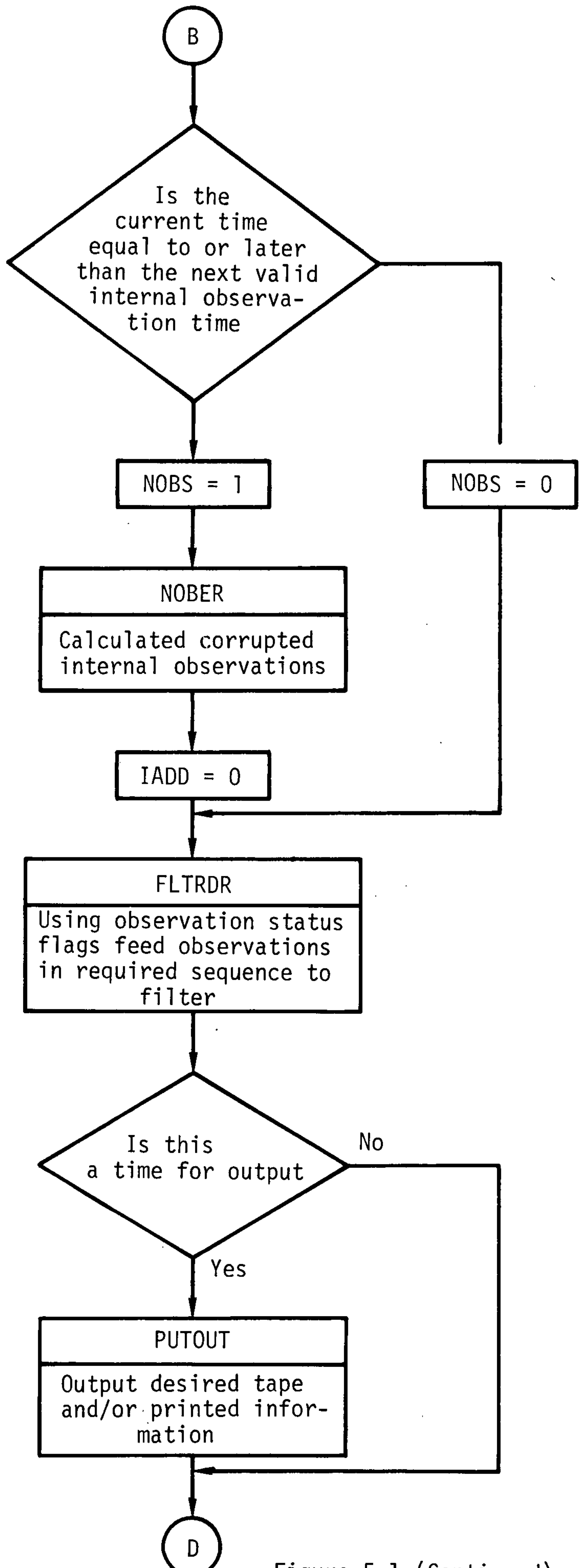


Figure 5-1 (Continued)

## 5.7 NOBER

See Section 4.4.

## 5.8 FLTRDR

This module is visualized as containing the Shuttle Navigation Filter and all the necessary logic to drive the filter in the presence of external data. Appropriate coding of the incoming data and all the required interface parameters will be set. Further discussion of the filter itself is not possible due to insufficient detail at this time.

## 5.9 PUTOUT

The control of the output type will be handled here. Single cycle data, multi-cycle Monte Carlo output or certain selected parameters will be options to be controlled by input.

## 5.10 INPUT/OUTPUT

### Input

| | Variable | Dimension | Description |
|---|---|---|---|
| $RUNDEF | | | |
| | NCYCLE | 1 | Number of cycles to be executed |
| | APROCT | 1 | Write on APROC tape for processing |
| | PRINTF | 1 | Print frequency |
| | COORDP | 1 | Coordinate system for output |
| $END | | | |

$NOP

same as Noisy Observations Program

$END

| | Variable | Dimension | Description |
|---|---|---|---|
| $INITST | | | |
| | COV | n,n | Input state covariance matrix |
| | TSTART | 1 | Time associated with COV |
| $END | | | |

$FILTER

specific filter parameters including REFSMAT

$END

### Output

To be determined