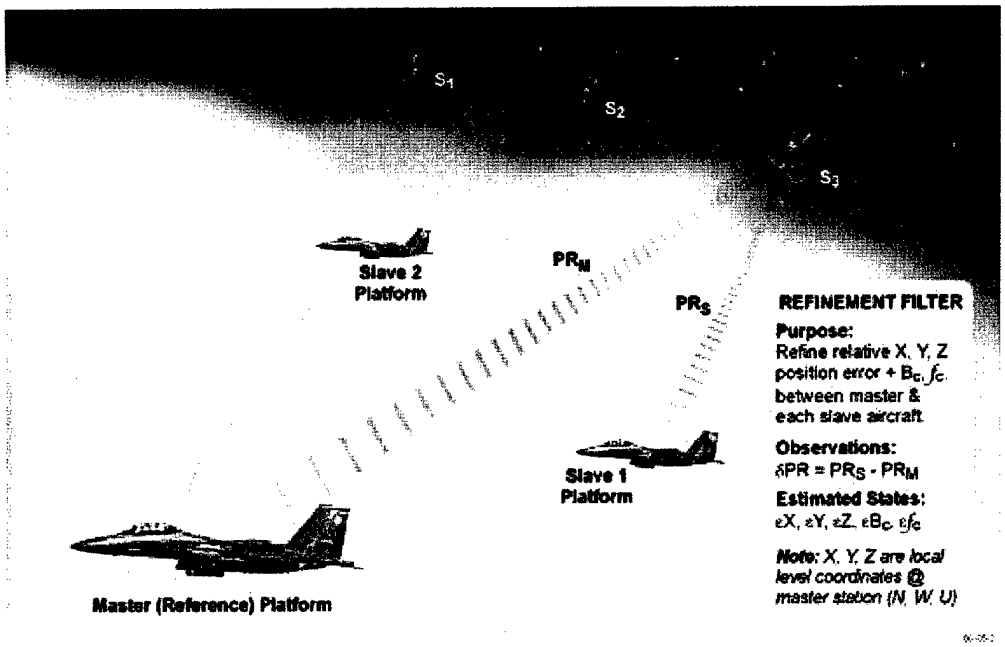
BAE SYSTEMS



Refinement Filter Analysis & Design Report:

Application of Differential GPS Processing to Precision Link -16 Navigation and Synchronization Interim Report

Joel Reiss
Tina Harwani
Jessica Hsu

BAE SYSTEMS, CNIR Division
Wayne, New Jersey

30 September 2003

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

Prepared for:
Program Officer
Office of Naval Research
Ballston Tower One
800 North Quincy Street
Arlington, VA, 22217

Attn: Dr. John Kim, Code 313, Telephone: (703) 696-4214
Ref: Contract No: N00014–02–C-0418 CLIN 0002 CDRL A002

UNCLASSIFIED

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2003	Scientific

4. TITLE AND SUBTITLE

Interim Report (Oct 2002 – Sept 2003)
Application of Differential GPS Processing to Precision Link-16
Navigation and Time Synchronization

5. FUNDING NUMBERS

C - N0014 - 02 - C 0418

6. AUTHOR(S)
J. Reiss
J. Hsu
T. Harwani

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSE(S)

BAE Systems CNIR Division
164 Totowa Road
Wayne, NJ 07470

8. PERFORMING ORGANIZATION REPORT NUMBERS

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESSE(S)
Dept of the Navy
Office of Naval Research
800 N. Quincy Street
Room 704
Arlington, VA 22217
Supervised by: Dr. John H. Kim, ONR-02

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
An innovative algorithm has been developed and evaluated which utilizes Differential GPS techniques to enhance Link 16 navigation and time synchronization performance. Relative positioning accuracies within a mobile airborne community on the order of one foot, and time synchronization to less then one nanosecond have been demonstrated. The algorithm is structured as an Extended Kalman Filter mostly these position errors, clock bias, and frequency drift. Plans are described for laboratory demonstration using Link-16 terminals by Spring 2004

14. SUBJECT TERMS
Link – 16
GPS
Differential GPS
Kalman Filtering

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

Abstract

An innovative algorithm has been developed and evaluated which utilizes Differential GPS techniques to enhance Link 16 navigation and time synchronization performance. Relative positioning accuracies within a mobile airborne community on the order of one foot, and time synchronization to less then one nanosecond have been demonstrated. The algorithm is structured as an Extended Kalman Filter mostly these position errors, clock bias, and frequency drift. Plans are described for laboratory demonstration using Link-16 terminals by Spring 2004.

Table of Contents

Table of Figures

Table of Tables

1. Summary

The prime technical objective of the work described in this report is to identify, evaluate, and demonstrate algorithms based upon Link-16 organic navigation capabilities aided by Differential GPS pseudorange measurements which can provide enhanced performance in relative navigation and time synchronization to arbitrary mobile communities. The performance goals of this technology are to achieve a relative positioning accuracy, relative to a designated reference platform, of one meter or less, a relative velocity accuracy of 0.1 ft/second or less, and a relative time base synchronization to a precision of one nanosecond or better. The algorithms which have been developed to satisfy these requirements are collectively designated as the **Differential GPS Refinement Filter (DGPS RF).**

The work described herein was initiated under the Office of Naval Research (Code ONR-02) BAA Next Generation Concepts, Devices, and Systems in Navigation and Timekeeping Technologies, under the technical supervision of Dr. John S. Kim. This study was performed under defining contract number C-N0014-02-C0418, dated October 2002, which included the period October 2002 through March 2004. Results presented in this report cover the period October 2002 through September 2003, which is referred to as Phase 1 of this project.

Phase 1 is designated as the **Refinement Filter Analysis Study,** whose scope is defined as follows:

> The contractor (BAE SYSTEMS, CNIR Division, Wayne NJ) shall perform analyses of not less than two alternative configurations of the Refinement Filter (RF) algorithm, which represent alternative models of the position and time errors resulting from Link-16 nominal navigation performance. The contractor shall exercise each of these configurations on its simulation software package under scenarios designed to stress the operation of the RF, and shall determine the parameters of each version which provide optimum system performance. Among these parameters shall be the required system GPS pseudorange observation rate, covariance rate limiting, and error budgets of the platform inertial navigation units.

The primary tasks undertaken under Phase 1 of the project included:

- Delivery of analysis report
- Design and Code of Real Time Demonstration RF software
- Rehosting of Simulation Software on Laptop or equivalent
- Integration of Full RF Demonstration System.

Phase 2 of this program, commencing in October 2003, is designated as the **DGPS RF/Link-16 Demonstration.** It covers the integration of the DGPS RF Algorithm in real time executable code within a selected Link-16 terminal asset and demonstration of the DGPS RF

algorithm performance on BAE SYSTEMS' Terminal ATP Test Set. It will culminate in a real time demonstration of performance in operationally representative scenarios during Spring 2004.

The intent of the activity performed during Phase 1 of this study was to go beyond the specific tasks identified for this period by scheduling early design work that would allow a seamless transition to Phase 2 laboratory integration and demonstration. These "pre-Phase 2" tasks included preliminary design for the TATS and OCP modifications which wpuld be required immediately upon the start of the second phase of this program. A brief summary of the activities and results of the work performed between October 2002 and September 2003 is given below. For each technical area, a reference to specific paragraphs of this report is provided.

Development of Fundamental Algorithm Architecture (Paragraph 4.1)

The development of the basic architecture of the DGPS RF algorithm was guided by a simple idea. We sought to develop a design which, with minimal modifications, could operate in any of the three environments which the program would have to operate, namely (a) Laptop or PC environment (Version 1.0), (b) Link-16 Terminal /TATS Laboratory operation (Version 2.0), or (c) Airborne testing with real GPS receiver (Versions 3.0 and 4.0). The central algorithm routines were designed to be essentially common for all four versions, while the interface routines with the host and Link-16 OCP were changed as required to satisfy specific requirements of each operating environment. The result of this design was a flexible and responsive algorithm capable of working in multiple simulation and operation environments.

Test Bed 1 Simulation Activities (Paragraph 4.2)

In order to quickly establish the basic operating performance of the DGPS RF algorithm, a simplified test bed, designated as **Test Bed 1** was utilized, in which a simple trajectory generator and error model mimicked Link-16 hybrid navigation performance for the participating aircraft. Initial runs on this simulator utilized what we refer to as Closed Loop operation, in which the error estimates generated by the DGPS RF algorithm were removed from the simulated Link-16 hybrid navigation solution. DGPS RF state values were then zeroed. This proved to yield quick convergence to small error values, but ultimately resulted in long-term instability (Figure 1-1). When the DGPS RF error states were uncoupled from the Link-16 navigation (Open Loop Operation) stable and reliable operation meeting all performance goals was realized. (Figure 1-2).

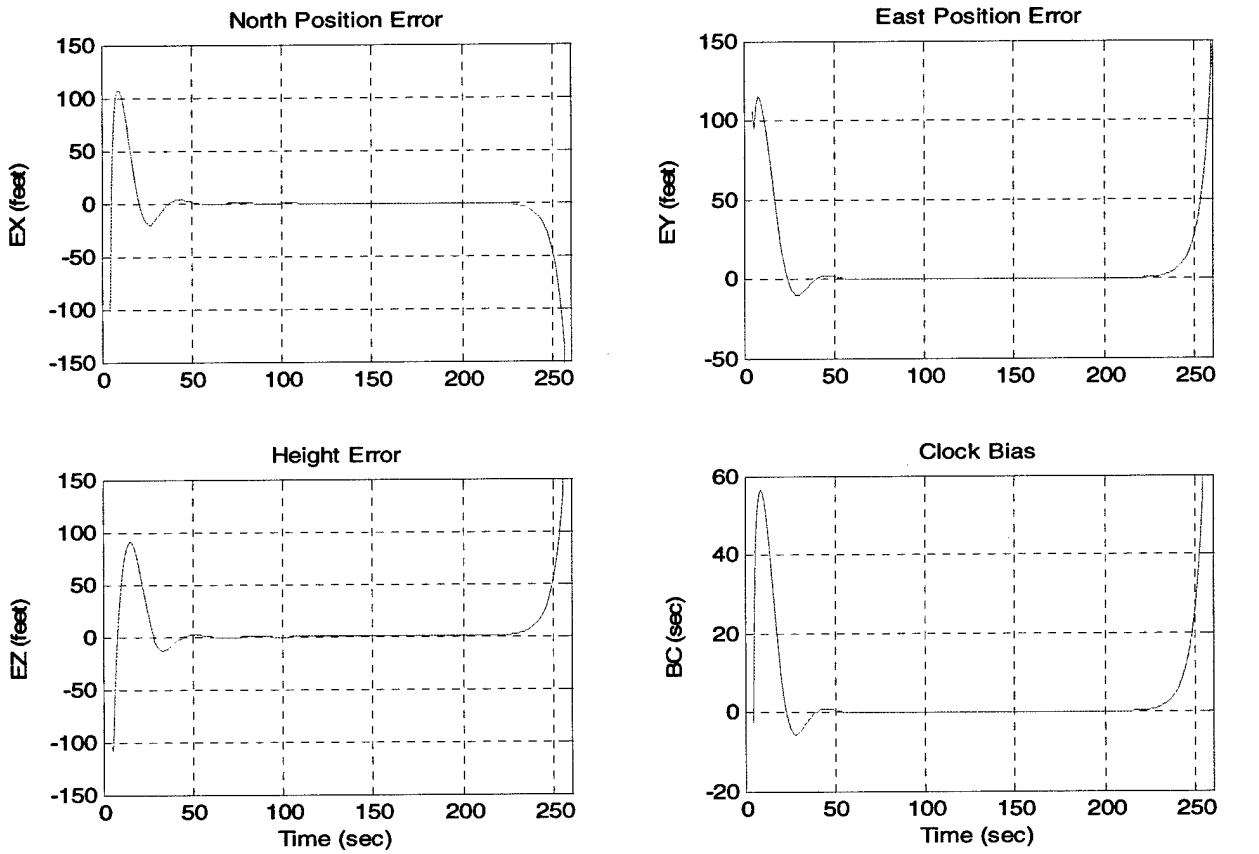


Figure 1-1: Instability in Closed Loop operation

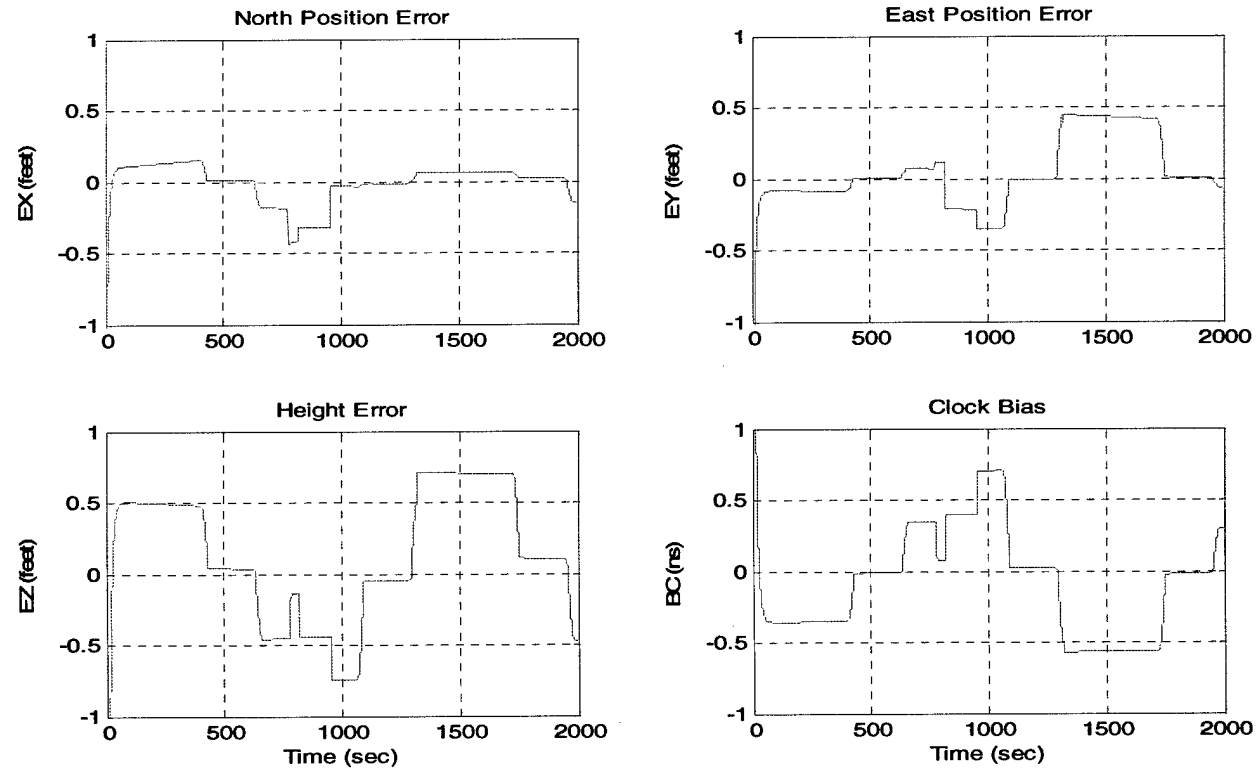


Figure 1-2: Open Loop configuration provides stable solution

Test Bed 2 Simulation Activities (Paragraph 4.3)

Test Bed 2 was the designation given to the simulation package in which the Link-16 navigation model was represented by the Link-16 Navigation Simulator (LNS). This BAE SYSTEMS proprietary software package represents the standard for modelling Link-16 navigation performance and is the most accurate such reference. The purpose of Test Bed 2 simulation activity was to verify that the DGPS RF would work successfully in open loop mode with the LNS. The results of these tests were uniformly successful, and verified that excellent navigation and time synch performance was produced by the algorithm in all test scenarios. (Figure 1-3).

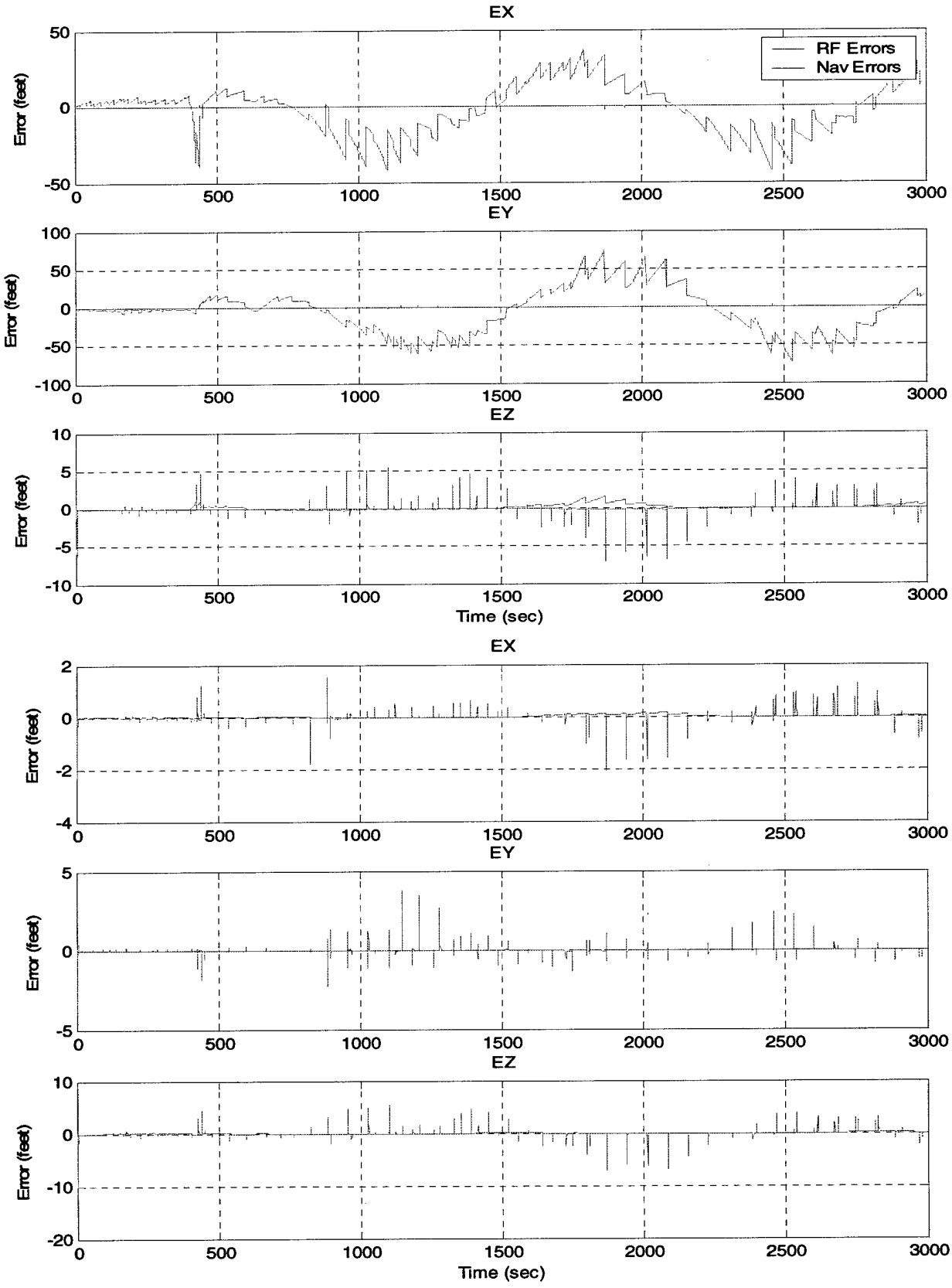


Figure 1-3: Test Bed 2 proved DGPS RF performance with LNS

DGPS RF Operational Software Design (Paragraph 4.4)

The detailed design of the DGPS RF operational real time software is presented in this section of the report. The selected architecture divided the software into seven basic functions, as follows:

(a) RF_EXEC_CTRL - (RF Executive Control): Represents the executive element in the operational software. Controls and synchronizes the execution of all subsidiary routines

(b) RF_SOURCE_DATA_PROCESSING – (RF Source Data Processing: Prepares and synchronizes all required observation data for processing within the RF Kalman Filter code.

(c) RF_KALMAN_PROCESSING – (RF Kalman Processing): Implements the five state Extended Kalman Filter algorithm which is the heart of the algorithm

(d) HOST_INTERFACE_PROCESSING- (Host Interface Processing): Provides the two way interface processing controlling all data provided by or supplied to the host computer. This data is composed primarily of control signals and GPS pseudorange observation data.

(e) OCP_INT_PROC – (OCP Interface Processing): Provides two way interface controlling all data provided by or supplied by the Link-16 Operational Computer Program.

(f) BUILD_MASTER_MESSAGE – (Build Master Message): Invoked only when own platform is designated as a master. This routine constructs the Master Reporting Message containing master-measured GPS pseudorange observations, as well as own position data, for transmission via Link-16 to all other members.

(g) BUILD_RF_RSLTS – (Build RF Results): This routine generates a summary report to host containing all output data generated by the algorithm, including error states, covariances, and status variables.

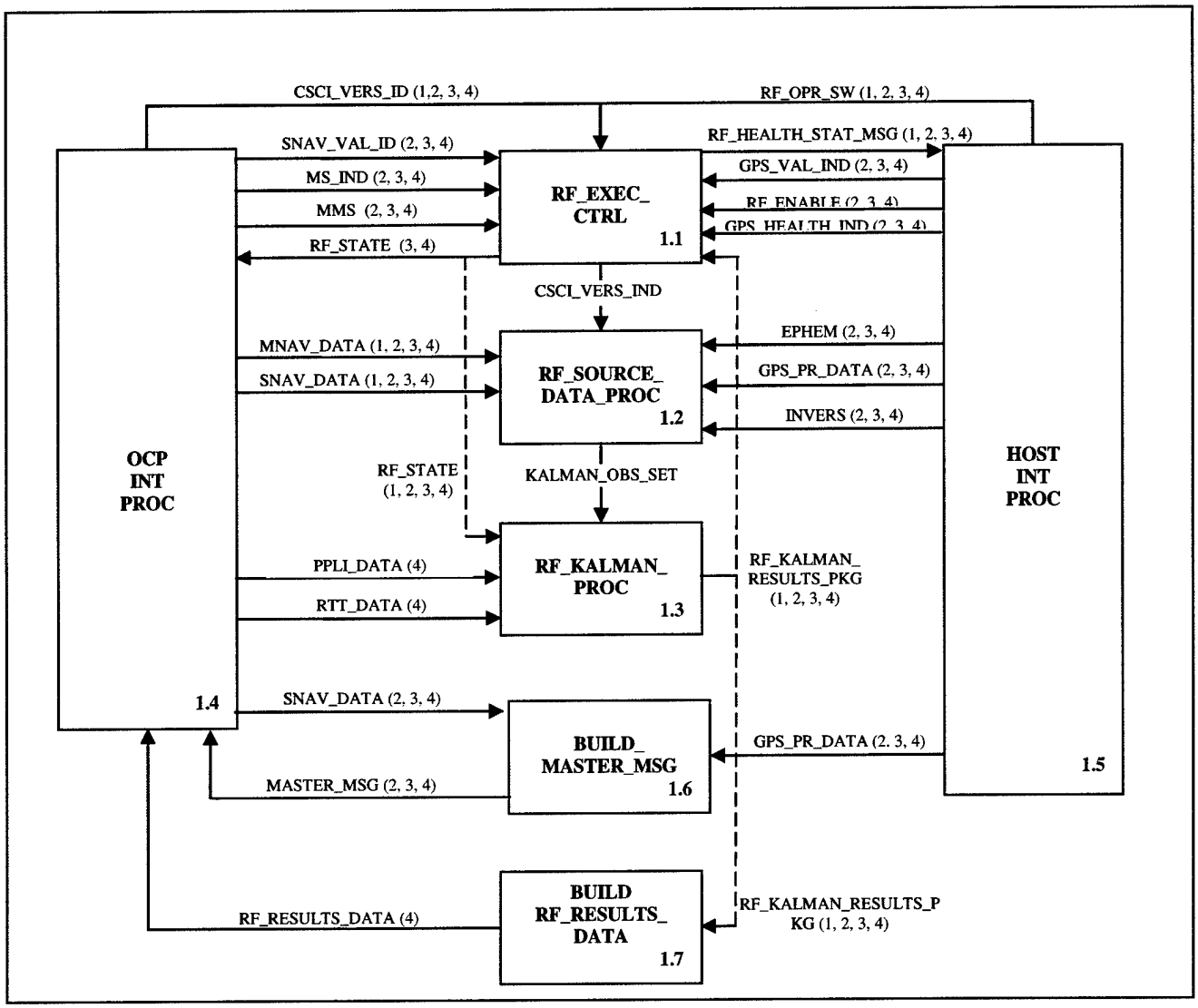


Figure 1-4: DGPS RF Software Architecture

The real time software architecture described in this section is designed for operation in Version 1.0 (Laptop-hosted preliminary demonstration), as well as for Version 2.0 (Laboratory demonstration on Link-16 terminal assets, and Version 3.0 (Airborne flight demonstration). The simulation results presented in paragraphs 4.2 and 4.3 were generated using the code specified in this section.

Laboratory Testing Requirements and OCP Design Modifications (Paragraph 4.5)

The laboratory testing environment planned for the Phase 2 demonstration will require significant changes to both the Link-16 Operational Computer Program (OCP), as well as modifications to the existing Terminal ATP Test Set operational software. The required modifications to the Link-16 OCP includes establishment of a working interface with Link-16 navigation processing in order to provide the DGPS RF algorithm with the necessary self-

navigation data. It will also be necessary to provide a direct interface with the host via the 1553 MUX to supply the real or simulated GPS pseudorange measurements from all satellites within view. Finally, a new TADIL-J message, designated by J28.7, will be required to support the master platform reporting of its measured GPS pseudorange data. With regard to the Terminal ATP Test Set (TATP), provision must be made to supply the terminal under test with synthetic GPS pseudorange measurements corresponding to the satellite ephemeris and the position of the slave platform. Master messages must also be transmitted to the slave platform via the test terminal. Finally, new control and initialization inputs required to invoke and control the operation of the DGPS algorithm will also be required. The TATP will not be required to generate these inputs; each of them will be precomputed and fed to the appropriate terminal input, either RF or MUX, by the TATP at the required intervals of time.

The work reported on within this section represents modification of both the terminal and TATS software performed during Phase 1, which will allow a seamless and efficient transition to laboratory implementation beginning in October 2003.

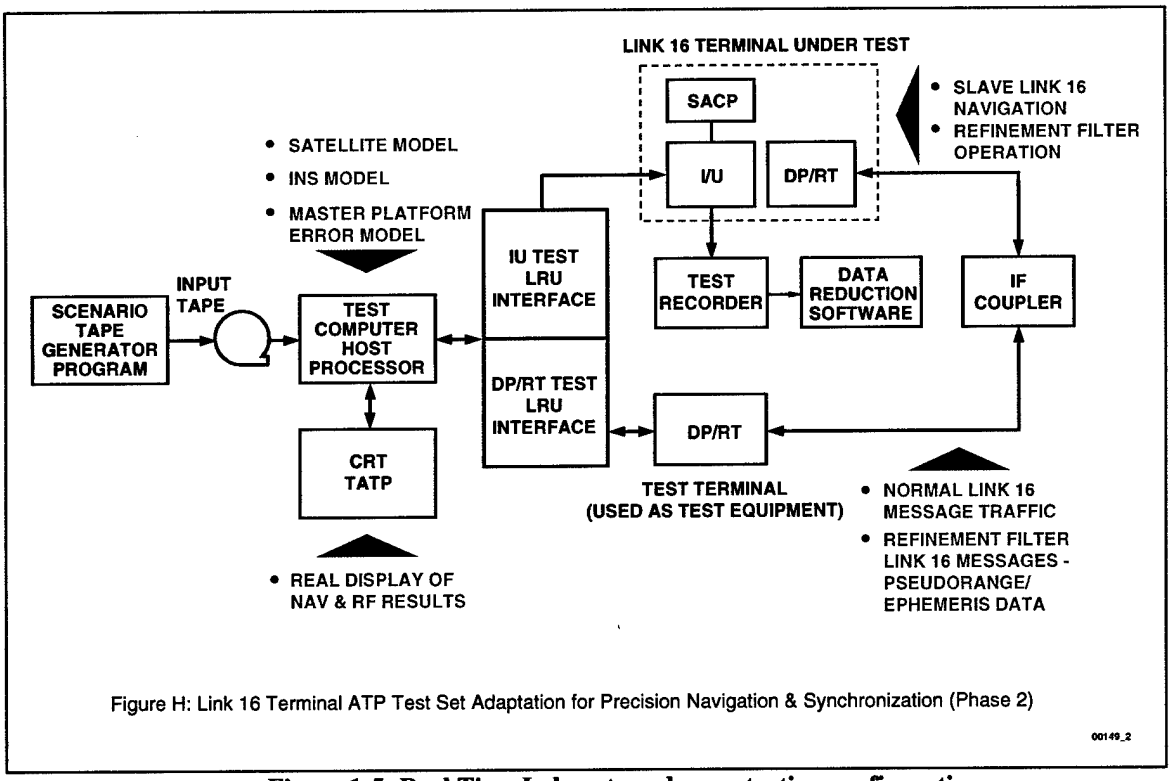


Figure 1-5: Real Time Laboratory demonstration configuration

Results and Discussion (Paragraph 5.0)

The Phase 1 effort was a successful one in all respects, with all goals met or exceeded. Most importantly, the intrinsic accuracy of the DGPS RF algorithm was established under a wide range of potential operating conditions. Only under extreme navigation errors in the underlying Link-16 navigation, far beyond normal operating levels, was the DGPS RF algorithm made to fail. Of equal importance, advance design work on required modifications to both the Link-16

operational software and the TATS was carried out, thus paving the way to a smooth transition to the Phase 2 laboratory demonstration phase.

Conclusion and Recommendations (Paragraph 6.0)

Two primary recommendations are set forth in this section. The first is a proposal for the support of a flight test of the DGPS RF algorithm based on a test plan implemented for testing of BAE's sensor registration algorithm at Wallops Island, Va. The second recommendation is that the operation of the DGPS RF be combined with a parallel effort for atmospheric refraction calibration to produce a robust navigation solution even in the event of GPS jamming for periods of time. It is shown that these algorithms can be mutually supporting, with the DGPS RF providing accurate locations needed for RF calibration. Conversely, if GPS is shut off, Link-16 navigation can be improved via the excellent estimation accuracy of the range between members afforded by the atmospheric filter operation.

2. Introduction

Mission requirements related to electronic warfare and precision strike impose unprecedented requirements on community relative positioning and time synchronization. The achievement of precision navigation and synchronization among members of a mobile community is crucial to geolocation algorithms needed to localize hostile emitters. These algorithms often utilize combinations of pulse time difference of arrival (TDOA) and/or frequency difference of arrival (FDOA) as measured between participating cooperating platforms. The need for precision navigation and synchronization in support of geolocation relates directly to the need to perform radar pulse deinterleaving and leading-edge pulse detection. Link-16 has often been considered as a potential contributor to this mission by way of its embedded hybrid-inertial navigation capability. The integrity of Link-16 navigation has been demonstrated by US and allied services in many thousands of operational and test flight hours since 1985. Performance levels characteristic of organic Link-16 navigation are shown in Figure 2-1. While navigation and synchronization performance of Link-16 is of demonstrably high quality, relative positioning accuracy typically falls within 10-30 meters, and thus falls short of the extreme precision desired for geolocation. Similarly, interplatform time synchronization may fall below 10 nanoseconds, but is internal to the terminal itself and not readily available to geolocation sensors needing this level of synchronization. What would be most desirable is an approach that builds on Link-16 navigation and time synch performance, but enhances it via additional observation processing. The development of such an algorithm is the subject of this report.

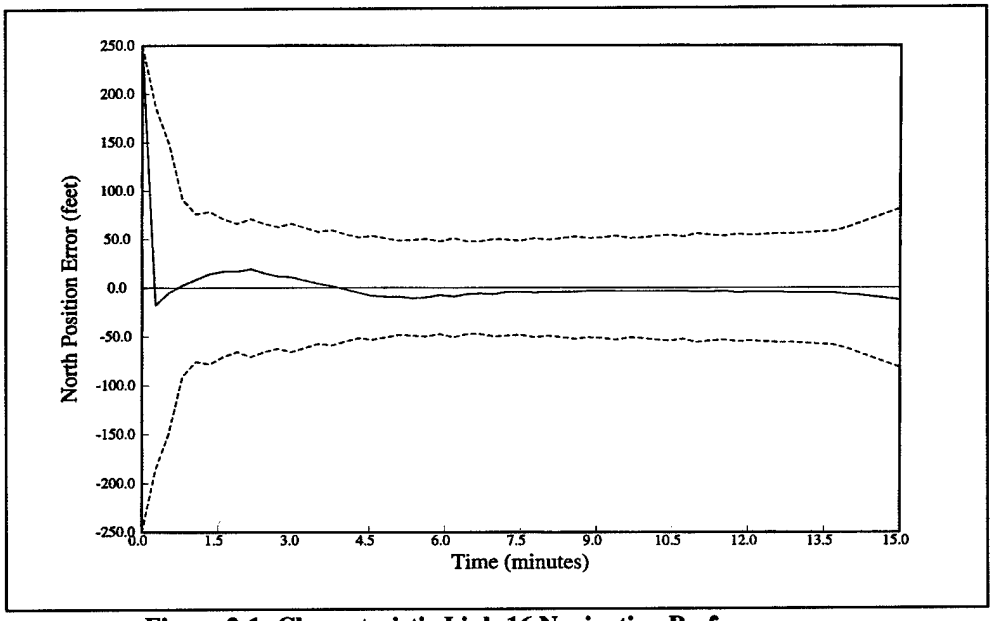


Figure 2-1: Characteristic Link-16 Navigation Performance

In a response to an Office of Naval Research (ONR) Broad Area Announcement, BAE SYSTEMS proposed a research study to investigate the joint application of Link-16 embedded navigation and Differential GPS (DGPS) techniques to the goal of precision community navigation and time synchronization. In May 2002, BAE SYSTEMS was selected for FY 2003 and 2004 funding to develop this technology. The FY03 phase of this study was dedicated toward algorithm development, simulation, and documentation, while the FY04 phase is aimed at laboratory demonstration of precision navigation and synchronization using actual Link-16 terminal assets. This report constitutes the primary deliverable document for FY03 research under BAA Solicitation 02-007.

The expected payoff of this study is the identification and demonstration of an algorithm, designated as the **DGPS Refinement Filter (DGPS RF)** which is capable of reducing position and timing errors of a mobile community to extremely low levels. The operational goals of this algorithm are (1) relative community positioning accuracy to a precision of 1 meter or less, (2) relative velocity accuracy of better than 0.1 ft/second, and (3) common time synchronization to a level of one nanosecond or better. Because this approach requires no modification to the government's Link-16 terminal assets beyond the addition of certain software algorithms, it provides an economical growth path to reaching the stated performance goals.

Link-16 utilization of GPS data for navigation updates and synchronization has been a fact for many years. The use of the one PPS time discrete as a measurement of UTC time results in Link-16 network time synchronization to that standard. In the case of navigation data, GPS PVT (position, velocity, and time) measurements are processed by any Link-16 terminal as a direct observable for platform position, and thus contribute to the accuracy of the navigation solution. The navigation accuracy illustrated in Figure 2-1 is indicative of the results of this GPS position aiding. Our approach for the DGPS RF is not to ignore this very accurate navigation solution, but to utilize it as a foundation, which can then be further refined by processing additional measurements. What are these additional measurements? We have decided to utilize the raw pseudorange measurements available from all visible GPS satellites as the source of new observations, which the DGPS RF requires to refine residual position and time errors remaining from the existing Link-16 navigation solution.

We will be seeking a relative solution, in which all mobile members estimate their position and time errors with respect to a selected community member, designated as a Master Platform. Any platform, static or mobile may be so designated. All remaining members are thus "slaves", who seek to determine their position and timing errors relative to the master. Should the master be accurately located in absolute, i.e. WGS-84 coordinates; this relative correlation automatically will converge to an absolute one in the WGS-84 frame of reference. The task of the DGPS RF, which is implemented on each platform, is thus to estimate the five-element error vector which is the residual error of the Link-16 navigation. The constituents of this error vector are three components of relative position error, a relative clock bias and relative frequency between each platform and the master. As will be derived in paragraph 3.0, the configuration of the DGPS RF is that of an Extended Kalman Filter (EKF) which models the five error components described above. The question remains as to how the GPS pseudorange information should be presented to the DGPS RF for processing. We have selected a differential mode, that is, a vector of pseudorange differences as measured between the master and all slaves.

The logic for this choice is as follows. GPS pseudorange errors are directly affected by both tropospheric and ionospheric propagation delays, which basic GPS processing algorithms attempt to compensate for. Differential operation can reduce these errors to small values by taking advantage of the high spatial correlation of these terms within 100-200 Km on the earth's surface. This is the basis of conventional DGPS operation, which usually involves the placing of a GPS receiver at a known (well-surveyed) position on the earth, and comparing received pseudorange with values compatible with known satellite ephemeris. Here, we have no such surveyed position. The steady state error behavior of Link-16 is well known from both simulation and flight testing, and may be characterized by a Markov process with correlation time measured in hundreds of seconds, that is, it is basically a slowly varying quantity. If we can provide measurements with time constants at least an order of magnitude less than this, we can eliminate the need for a static and surveyed reference point. This is the fundamental principle on which the DGPS RF relies. The measurements we use are those of the differential satellite pseudoranges (PR) themselves (that is, master minus slave measurements of each satellite PR)

This project represents an effort extending over FY03 and FY04 to develop the DGPS RF technology with a goal of demonstrating its performance within an actual Link-16 terminal in a laboratory environment by 2Q FY04. It has always been the intention of BAE SYSTEMS that the DGPS RF operational software used within this laboratory demonstration be configured as closely as possible to flight ready and operational software. This would pave the way toward early flight testing and possible operational use. For this reason, the scope of the FY03 project effort extends beyond the analysis and proof of performance of the DGPS RF algorithm to include rigorous design and documentation of a near-operational code version. In addition, the inclusion of new (to Link-16) observations and interfaces require specification of code changes needed to current Link-16 software and terminal test equipment.

During this project, the analysis and development of the DGPS algorithm initially utilized simple community navigation error models, and then tested its performance within a rigorous representation of Link-16 navigation code known as the Link-16 Navigation Simulator. Extensive trials were performed to optimize algorithm performance by varying certain parameters, such as state configuration, tuning constants, update rates, initial navigation errors and community configurations. The end result was an extremely robust and flexible algorithm that provides excellent position and time error recovery equaling or bettering our initial performance goals. Detailed analysis and results of each of the candidate algorithm configurations is included within this report.

The design and specification of the final derived algorithm had to take into consideration the fact that several different versions of the software are required. Version 1.0 is designed for early demonstration on a PC or laptop environment using simulated satellite data based upon a circular orbital configuration. It also interfaces with the Link-16 Navigation Simulator. This version will be utilized for the interim demonstration scheduled for September 2003. Version 2.0 constitutes the laboratory version, designed for integration within the Link-16 operational software. This version utilizes a full elliptical satellite model and mimics GPS input data characteristic of a Canadian Marconi GPS satellite receiver. Version 3.0 is reserved for flight-ready software capable of using arbitrary GPS receiver units. The core processing routines of

each of these versions are nearly identical, and differ only in the interface processing routines dealing with the navigation software and host interface.

Since the laboratory demonstration of the DGPS RF must operate within a Link-16 terminal, certain unavoidable changes are required to the current operational terminal software to allow the integration with the new algorithm. For technical reasons relating to solution stability, no direct feedback of the results of the DGPS RF solution may be allowed to affect the current LINK-16 navigation algorithm. Nonetheless, many changes to the existing Link-16 software are required to accept the satellite pseudorange measurements and to output results of the new configuration. In addition, a new Link-16 message, designated as the Master Message, is required to convey pseudoranges measured at the master platform to all other units. Although detailed coding of these required changes will not commence until FY04, the design of these software modifications was initiated within the scope of the FY03 activity and reported on in this report.

The requirement for laboratory testing also imposes new and different requirements on the laboratory test set (Terminal ATP Set – TATS) on which the recoded terminal will be exercised. The task of the TATS is to replicate at both the RF and host terminal interfaces exactly the type and format of data required by the DGPS RF algorithm. None of these data types is currently implemented in Link-16 test equipment. One example of this data is the need to replicate the simulated pseudorange data in a format characteristic of a modern GPS receiver. Another example is the generation of Master Messages containing precisely regulated pseudorange data coming from the simulated master platform. Still another example is the need to accept new precision navigation results from the DGPS RF for data recording and analysis to prove its performance within the terminal environment.

The activity summarized above constitutes the scope of work performed during the FY03 period, and is reflected in the structure of the remainder of this document, as follows:

Paragraph 3.0 – Development of Fundamental Equations

This section contains the derivation of the fundamental mathematical relationships of the DGPS RF, including the specification of the related Kalman Filter modeling expressions on which it depends. This section also describes two alternative methods of specifying the GPS satellite motion (circular and elliptical) which are required for both simulation and terminal implementation of the DGPS RF algorithm.

Paragraph 4.0 – Methods and Procedures

This section presents a detailed description of the development methodology of the DGPS RF algorithm, a narrative of significant development and simulation studies performed and a full analysis of results. It also includes comprehensive software design specifications of the developed RF algorithm and implementation plans for FY04 on both Link-16 operational

software and changes required for the laboratory test equipment. The contents of this section shall be as follows:

4.1 Development Plan for Simulation, Laboratory, and Flight Test Versions of the DGPS RF software

4.2 Algorithm Development, Test, and Analysis

4.3 Software Design for DGPS RF Algorithm

- 4.3.1 Test Bed 1 Covariance Based Navigation Model
- 4.3.2 Test Bed 2 LNS Navigation Model
- 4.3.3 Algorithm Simulation Study Summary and Conclusion

4.4 DGPS RF Operational Software Design

4.5 Laboratory Testing Requirements for DGPS RF Algorithm – TATS and OCP Design Modifications

Paragraph 5.0 – Results and Discussion

Paragraph 6.0 – Recommendations

Recommendations for further algorithm development and demonstration.

Paragraph 7.0 – Conclusions

Summary of significant project results for FY03

Paragraph 8.0 References

3. Development of Fundamental Equations

3.1 DGPS as a Correction to Link-16 Navigation Solutions

The extremely precise level of accuracies we are seeking immediately suggests the application of differential GPS measurements on pseudorange data provided by the GPS satellite constellation. Differential positioning accuracy of two members with conventional timing on the C/A or P(Y) code heretofore has required the establishment of a **surveyed reference point**, which can provide "truth" measurements of pseudorange to the satellites. Using the truth estimates, tropospheric and ionospheric refraction corrections can be derived, which are then data-linked to mobile members to reduce the effect of these terms[1]. Since the warfighter mission of emitter location may take place over territory, which has no such surveyed reference point available, it may appear at first glance that DGPS techniques cannot be applied to our problem. In fact, DGPS measurements can be successfully processed without a surveyed reference point if an alternative reference not dependent on such a static point can be found.

Let us take a step back from the problem to consider the following considerations. We are essentially seeking a **relative solution** between a designated master and another slave platform. That is to say, we wish to accurately estimate their position differences in some convenient frame of reference, not necessarily WGS-84. We note that Link-16 navigation, particularly when GPS aided, already provides such a solution, although not to the required accuracy. This solution enables us to compute a (time-varying) baseline between any two mobile platforms. Furthermore, even though the length and orientation of the moving baseline may change extremely rapidly for fast moving aircraft, the **errors or perturbations in this solution do not**, particularly if the platforms are equipped with well calibrated inertial navigation systems (INS). More specifically, Link-16 navigation integrates inertial navigation measurements within a **hybrid-inertial mechanization** whose characteristic errors are represented within Figure 2-1. These errors are slowly varying with time constants measured in tens or hundreds of seconds. Let these errors be represented by εx, εy, and εz in some local level reference frame. An algorithm, to be described in the next paragraph, will provide accurate and rapid measurements of these errors within a period that is short compared to their time constants, and reduce them to small values

3.2 Derivation of DGPS RF State and Measurement Equations

The potential measurements at our disposal for this problem are (a) PPLI messages provided by the Link-16 system, and (b) GPS pseudorange measurements provided by a military-quality GPS receiver (Please note that GPS PVT measurements are assumed to be already processed by Link-16 as part of its normal hybrid navigation function). Figure 3.2-1 represents the relative positions of a designated master platform (A) and a slave platform (B). We consider the measured pseudoranges, S_i from a given GPS satellite i, where i ranges from 1 to NV (NV is equal to the number of visible satellites common to both platform A and platform B). The measured pseudorange to A is designated by S_i/A, and that to B by S_i/B. We select as a

representative frame of reference the **Local Level Coordinate System (X, Y, Z)** centered at the master platform and moving with it. This is an orthogonal reference frame with X pointing true North from the master, Y true West, and Z Up.

The choice of pseudorange **differences** as a potential observation rather than the pseudoranges themselves is a straightforward one. The time delays attendant with S_i/A and S_i/B , both ionospheric and atmospheric, can be assumed to be **nearly equal** due to the fact that the range between A and B is small compared to the distances to the GPS satellites themselves (approximately 11000 NM). By differencing pseudoranges from all visible satellites, the common time delays drop out of the range equation and may be safely ignored. A second choice in the use of the pseudorange differences is the use of either (a) P(Y) code phase measurements or (b) carrier phase measurements, both of which are usually available from military GPS receivers. Use of the carrier phase measurements, as is often done in precision DGPS measurements using surveyed reference points can provide centimeter accuracy on positioning, but results in a modulo 2π phase ambiguity in carrier measurement since relative positioning errors may add up to many carrier wavelengths. There are various methods available for estimation of the integer wavelength differentials, but most of these require extended processing time[2], which may exceed the position error time constants of our navigation solution. We were unable to find any references in the technical literature for methods of phase ambiguity resolution fast enough for our purposes. We therefore decided to proceed using code phase measurements instead, anticipating that processing of enough data would compensate for the coarser resolution of the pseudorange measurement data as compared with carrier phase operation. Simulation results presented later in this report seem to bear out this assumption.

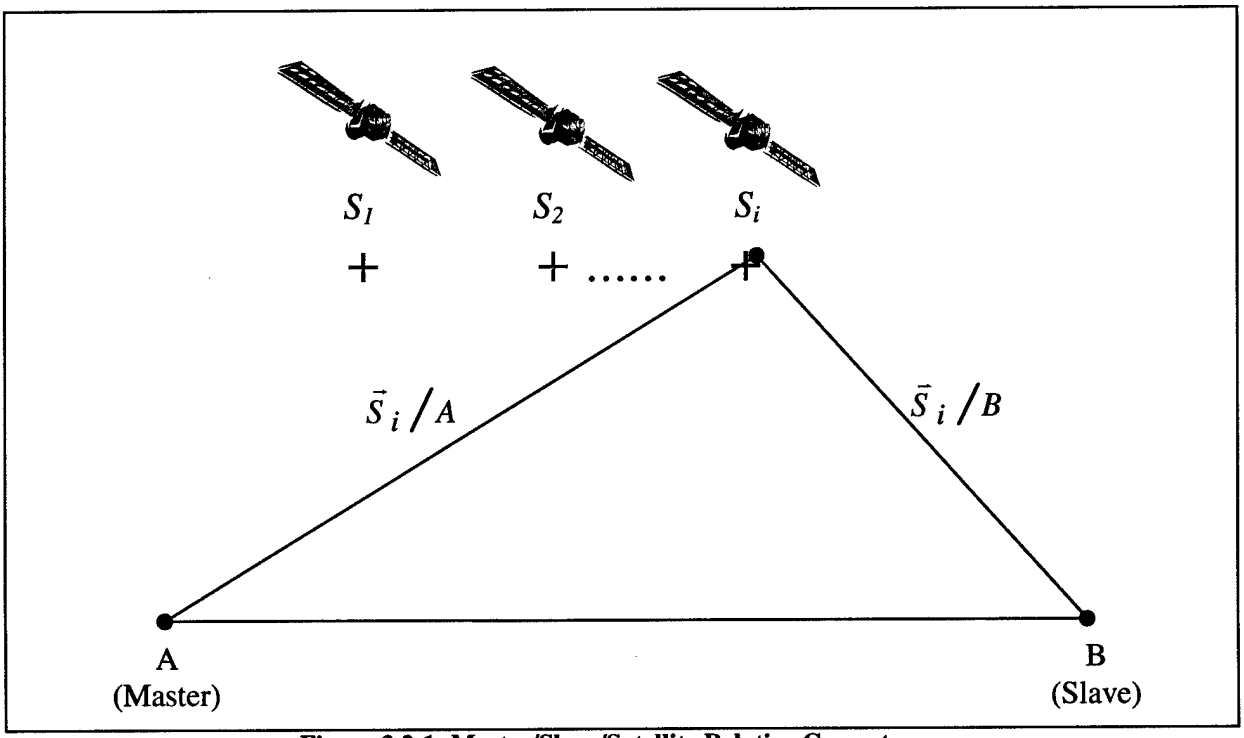


Figure 3.2-1: Master/Slave/Satellite Relative Geometry

In Figure 3.2-1, above, the pseudorange difference quantity S_i /A – S_i /B is expressed in Equation (1) as:

$$\delta PR = \delta \text{ Pseudorange A/B} = \delta x \, Cxsi + \delta y \, Cysi + \delta z \, Czsi + Bc * Cs \qquad (1)$$

where δx = Absolute * difference between A & B

$$= \underbrace{\Delta x_N}_{\substack{\text{nominal difference based upon navigation solution}}} + \underbrace{\varepsilon x}_{\text{error term}}$$

δy = Absolute y difference between A & B

$= \Delta y_N + \varepsilon y$

δz = Absolute z difference

Cxs_i, Cys_i, Czs_i = direction cosines between S_i/A and X, Y, Z axes

B_c = Clock Bias between A & B

C_s = Speed of light (ft/ns) = 0.983569 ft/ns

Based upon the definition of terms in Equation 1, we note that potential variables of interest, which could be estimated via the DGPS RF, are:

- Three elements of relative position error(εx, εy, εz)
- Three time derivatives of the relative position errors (εx, $\varepsilon \dot{y}$, $\varepsilon \dot{z}$)
- Clock Bias (B_c)
- Derivative of Clock Bias or frequency drift between platforms (f_c)

The presence of a Clock Bias term in Equation (1) indicates that the measurement differential is expressed in terms of **both time and position errors.** These error terms are separable only on the basis of a common estimation process, and cannot be treated independently. Furthermore, the Clock Bias term in Equation (1) represents the time difference reflected in the GPS time base, rather than the similar term reflecting the difference between the Link-16 terminal time bases. Timing errors within the GPS time base will vary depending on the number of satellites processed and the quality of the navigation solution. A typical value of the steady state relative time differences within the GPS time base might be as high as 10-15 nanoseconds, a significant term compared to the accuracies that we are seeking. Furthermore, there might still remain a level of oscillator drift, resulting in a time derivative term of the relative clock bias, Bc.

Based upon the above considerations, the Extended Kalman Filter (EKF) formulation of the DGPS RF state vector, X, could be as large as 8 elements, defined as:
If velocity, error terms are modeled as constant.

$$
\begin{aligned}
x_1 &= \varepsilon x \\
x_2 &= \varepsilon \dot{x} \\
x_3 &= \varepsilon y \\
x_4 &= \varepsilon \dot{y} \\
x_5 &= \varepsilon z \\
x_6 &= \varepsilon \dot{z} \\
x_7 &= B_c \quad \text{(Relative Bias Between A, B GPS Sets)} \\
x_8 &= f_c \quad \text{(Relative Frequency Drift between A, B GPS Sets)}
\end{aligned}
\tag{2a}
$$

Time Propagation of these terms is thus -

$$
\begin{aligned}
\dot{x}_1 &= x_2 \quad \dot{x}_5 = x_6 \\
\dot{x}_2 &= 0 \quad \dot{x}_6 = 0 \\
\dot{x}_3 &= x_4 \quad \dot{x}_7 = x_8 \\
\dot{x}_4 &= 0 \quad \dot{x}_8 = 0
\end{aligned}
\tag{2b}
$$

The relative range between master and slave, δr, may be expressed in terms of the variables defined above:

$$
\begin{aligned}
\delta R &= [(\Delta x_N + \varepsilon x)^2 + (\Delta y_N + \varepsilon y)^2 + (\Delta z_N + \varepsilon z)^2]^{1/2} \\
&\cong [(\Delta x_N{}^2 + 2\Delta x_N \cdot \varepsilon x) + (\Delta y_N{}^2 + 2\Delta y_N \cdot \varepsilon y) + (\Delta z_N{}^2 + 2\Delta z_N \cdot \varepsilon z)]^{1/2} \\
&\text{if square terms in } \varepsilon x, \varepsilon y, \varepsilon z \text{ are neglected}
\end{aligned}
\tag{3}
$$

Two potential observation types for the DGPS RF are considered, namely:

y1 = Range between Master and Slave – Provided by Link 16 PPLI Message (Note that there will be one measurement per PPLI message where NC is the number of suitable satellites commonly visible to both master and slave platforms)

y2 = Pseudorange Differences to each satellite from master and slave. (Note that there will be NV such measurements at each update)

Based on Equations (1 – 3), the derivation of the observation partial derivatives for each of these observation types is a straightforward exercise. The results are:

$$
\begin{aligned}
H(1,1) &= \frac{\partial \delta R}{\partial \varepsilon x} = \frac{\Delta x_N}{\delta R} \\
H(1,3) &= \frac{\partial \delta R}{\partial \varepsilon y} = \frac{\Delta y_N}{\delta R} \quad \text{Range Observation Derivatives} \\
H(1,5) &= \frac{\partial \delta R}{\partial \varepsilon z} = \frac{\Delta z_N}{\delta R}
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
H(j,1) &= \frac{\partial \delta PR}{\partial \varepsilon x} = Cxs_j \\
H(j,3) &= \frac{\partial \delta PR}{\partial \varepsilon y} = Cys_j \quad \text{PseudoRange DifferenceDerivatives} \\
H(j,5) &= \frac{\partial \delta PR}{\partial \varepsilon z} = Czs_j \\
H(j,7) &= \frac{\partial \delta PR}{\partial B_c} = Cs \\
j &= 2, NV+1 \\
i &= j \quad 1
\end{aligned}
$$

Where δR is the nominal range between members $A, B = (\Delta x_N{}^2 + \Delta y_N{}^2 + \Delta z_N{}^2)^{1/2}$

$[\ H\ ]$ is the Kalman Filter Observation Matrix
$(NV+1)\times 8$

The direction cosine terms in the above equation shall be generated as functions of member platform positions and the ECEF positions of the satellites, as computed via satellite ephemeris data. Details are presented in paragraphs 3.3. The observation matrix, H, will be required for the construction of the DGPS RF Kalman Filter algorithm, as detailed in paragraph 3.4.

3.3 Prediction of Satellite Pseudorange Measurements

The use of GPS pseudorange data for the DGPS RF Kalman Filter requires representative models of the satellite constellation orbital mechanics. The fully operational GPS constellation includes 24 or more (28 as of March 2000) satellites approximately uniformly dispersed around six (nearly) circular orbits each with four or more satellites. The orbits are inclined at approximately 55 degrees relative to the equator, and are separated from each other by multiples of 60 degrees right ascension. The orbits are nongeostationary and approximately circular, with radius of 26,560 Km, and orbital periods of one-half sidereal day (approximately 11.967 hours).

For purposes of DGPS algorithm development and simulation, modeling of the satellite mechanics utilized a simple circular orbit model, since this is an excellent representation of the actual GPS satellite orbital dynamics. This modeling was utilized for Version 1.0 of the developmental software simulation and testing. The operational use of GPS satellite pseudorange data, however, requires the implementation of the true elliptical satellite model and

solving the nonlinear Kepler equation using ephemeris data for each of the participating satellites. All laboratory and flight test versions of the DGPS RF software will use the elliptical satellite orbital models. Both circular and elliptical satellite models have a single purpose – that of predicting to high accuracy the coordinates of each satellite within the Earth Centered, Earth Fixed (ECEF) coordinate frame at specified times.

The mechanization of the DGPS algorithm involves two coordinate frames of reference, namely: (a) ECEF, and (b) Local Level – North, West, Up, as well as the ability to transform specific vector quantities between them. ECEF coordinates, by definition, have their origin at the earth's center. The ECEF X-axis lies in the equatorial plane and passes through the prime meridian at Greenwich. The Z axis passes through the north pole, while the Y-axis is orthogonal to X and Z such that XYZ represents a right hand system. The Local Level frame of reference used will have axes parallel to local North/West/Up directions specified at a the master's position. The transformation between these coordinate systems is an orthogonal one, and is represented by matrix $[T_{LE}]$ such that (Equation 5):

$$[T_{LE}]\ [V_{ECEF}] = [V_{LL}] \qquad (5)$$

where : $[T_{LE}]$ is an orthogoanl transform ation matrix given by :

$$[T_{LE}] = \begin{bmatrix} -\sin\lambda\ \cos\varphi & -\sin\lambda & \sin\varphi\ \cos\lambda \\ \sin\varphi & \cos\varphi & 0.0 \\ \cos\lambda\ \cos\varphi & \cos\lambda\ \sin\varphi & \sin\lambda \end{bmatrix}$$

λ = Platform latitude

φ = Platform longitude

V_{ECEF} = Vector in ECEF Coordinate s

V_{LL} = Vector in Local Level Coordinate s

3.3.1 Circular Satellite Dynamic Model

A simple circular orbit model is useful for approximating the elliptical orbit of an actual GPS satellite constellation [3]. The ECEF positions of each of the satellites when the circular model is used may be specified in terms of five parameters, namely:

- Right Ascension Angle (Ω)
- Orbital Position (ν)
- Orbital Inclination (α)
- Nominal Orbital Radius (R_K) = 87,051,108 feet
- Time in Seconds from Epoch

The parameter Ω represents the longitude where the orbital plane intersects the earth equatorial plane as the satellite crosses from the Southern Hemisphere to the Northern Hemisphere. The orbital inclination for all GPS satellites is approximatly 55 degrees. For GPS satellite orbits, the angle ν changes at a nearly constant rate of about $1.4584 \times 10 - 4$ radians/second, and a period of about 43,082 seconds (half a day) .

The nominal satellite position of any given satellite in ECEF coordinates is then given as (Equation 6):

$$\begin{aligned} ECEFXS &= R_K [\cos \nu \cos \Omega - \sin \nu \sin \Omega \cos \alpha] \\ ECEFYS &= R_K [\cos \nu \cos \Omega + \sin \nu \sin \Omega \cos \alpha] \\ ECEFZS &= R_K [\sin \nu \sin \alpha] \end{aligned} \quad \textbf{(6)}$$

$\nu = \nu_o$ (specified for each satellite) + $(t - t_0)$ 360/43,082 degrees
$\Omega = \Omega_o$ (specified for each satellite) - $(t - t_0)$ 360/43,082 degrees
R_K = Nominal Satellite Radians (ft) = 87,051,108 ft

Note: Although actual satellite ephemeris data is expressed in terms of meters, the DGPS RF state elements will be expressed in English units (ft). Thus, conversion of satellite ECEF coordinates must be converted to English units

3.3.2 Elliptical GPS Satellite Model

The corresponding computation of satellite ECEF positions for actual GPS satellites in elliptical orbits is a good deal more complex than the circular model. This solution requires the solution of Kepler's equation and the availability of satellite ephemeris data for each satellite. The detailed algorithm for this process is well known (See, for example, reference[3]), and is implemented in all operational GPS receivers.

GPS satellite ephemeris data for all GPS satellites is transmitted by the constellation within the 50 bps data stream. Each satellite will transmit current ephemeris data not only for itself, but for all other satellites within the constellation. This data can be represented by a 28 x 23 matrix EPHEM (k, 23), where k represents the individual satellite ID. Within the Canadian Marconi NORTHSTAR unit, a representative GPS receiver, the received ephemeris data is outputted at selected intervals of time within a specific message, denoted as **Message 22.** The detailed format of this interface is presented in Reference[4].

The contents of the Ephemeris Matrix shall be as presented in Table 3.3-1. The processing algorithm utilized for the generation of the Satellite ECEF coordinates is given in Figure 3.3-1 and Figure 3.3-2.

VAR	EMPEM ARRAY ID	DESCRIPTION
SVNO	= EPHEM(K,1)	Satellite vehicle number
WEEKNO	= EPHEM(K,2)	Week No.
TGD	= EPHEM(K,3)	Time of Week
TOC	= EPHEM(K,4)	Ephemeris Reference Time
AF2	= EPHEM(K,5)	2^{nd} Order Clock Correction Coefficient
AF1	= EPHEM(K,6)	1^{st} Order Clock Correction Coefficient
AF0	= EPHEM(K,7)	0^{th} Order Clock Correction Coefficient
CRS	= EPHEM(K,8)	Amplitude of sine harmonic correction term to orbit radius [m]
DELN	= EPHEM(K,9)	mean motion difference [semicircle/s]
M0	= EPHEM(K,10)	Mean anomaly at reference time [smcir]
CUC	= EPHEM(K,11)	Amplitude of cos harmonic correction term to the argument of latitude [rad]
ES	= EPHEM(K,12)	Eccentricity
CUS	= EPHEM(K,13)	Amplitude of sine harmonic correction term to the argument of latitude [rad]
SAS	= EPHEM(K,14)	Square root of semi-major axis [m^1/2]
TOE	= EPHEM(K,15)	Ephemeris reference time (from Epoch)
CIC	= EPHEM(K,16)	Amplitude of cos harmonic correction term to the angle of inclination [rad]
OMEGA0	= EPHEM(K,17)	Longitude of ascending node of orbit plane at weekly epoch
CIS	= EPHEM(K,18)	Amplitude of sine harmonic correction term to the angle of inclination [rad]
IOTA	= EPHEM(K,19)	Inclination angle at reference time [semicircle]
CRC	= EPHEM(K,20)	Amplitude of cos harmonic correction term to the orbit radius [m]
W	= EPHEM(K,21)	Argument of perigee [semicircle]
OMEGAD	= EPHEM(K,22)	Rate of right ascension [semicircle/s]
IDOT	= EPHEM(K,23)	Rate of inclination angle [semicir/s]

Table 3.3-1: Ephemeris Message 22 Elements

```
MU = 3.986008 E14 [m^3/s^2]                    Earth's uni gravitational param
OMEGAE = 7.292115167 E-5 [rad/s]               Earth's rotation rate

TK = PGPST - TOE                               Time from epoch
IF (TK > 302400.0) THEN
      PGPST = PGPST - 604800.0
IF (PGPST < 302400.0) THEN
      PGPST = PGPST + 604800.0

1AS = SAS^2                                    Semi-major axis
N0 = SQRT(MU/AS^3)                             Computed mean motion
N = N0 + DELN                                  Corrected mean motion
MK = M0 + N*TK                                 Mean anomaly

EKNEW = MK                                     Newton-Raphson estimation
FOR JX:1:100 DO
      EKOLD = EKNEW
      EKNEW = MK + ES*SIN(EKOLD)               Kepler's EQ for eccentric
END                                            anomaly
EK = EKNEW

FK = ACOS((COS(E)-1)/(1-ES*COS(E)))            ***
PHI = FK + W                                   Argument of latitude

DELR = CRC*COS(2*PHI) + CRS*SIN(2*PHI)         Radius correction
DELI = CIC*COS(2*PHI) + CIS*SIN(2*PHI)         Correction to inclination
DELM = CUC*COS(2*PHI) + CUS*SIN(2*PHI)         Argument of latitude correction

RK = AS(1 - ES*COS(EK)) + DELR                 Corrected radius
MU = PHI + DELM                                Corrected argument of latitude
IOTAK = IOTA + DELI + IDOT*TK                  Corrected inclination

OMEGAK = OMEGA0 + OMEGAD*(TK - TOE) -          Corrected longitude of
OMEGAE*TOE                                     ascending node

XK1 = RK*COS(MU)                               Position in orbital plane
YK1 = RK*SIN(MU)

XK = XK1*COS(OMEGAK) - YK1*COS(IOTAK)*SIN(OMEGAK)
YK = XK1*SIN(OMEGAK) + YK1*COS(IOTAK)*COS(OMEGAK)
ZK = YK1*SIN(IOTAK)
PR = SQRT((XS-XP)^2 + (YS-YP)^2 + (ZS-ZP)^2)

PRSEC = PR/C                                   Pseudorange in seconds

IPGPS = PGPST                                  Integer value of PGSTM
FPGPS = PGPST - FLOAT(IPGPS)                   Fractional GPS

CDPHASE = (FPGPS - PRSEC)*CR                   Codephase derivation
```

Figure 3.3-1: Code Phase Derivation from Ephemeris Message 22

```
For K = 1, 24
Compute mean motion
DELN = EPHEM(INVERS(ISVNO),9)          mean motion difference
                                       [semicircle/s]
SAS = EPHEM(INVERS(ISVNO),14)          Square root of semi-major axis
                                       [m^1/2]
AS = SAS^2                             Semi-major axis
N = DSQRT(MU/AS^3) + DELN              Computed mean motion

Compute mean anomaly
M0 = EPHEM(INVERS(ISVNO),10)           Mean anomaly at reference time
                                       [smcir]
MK = M0 + N*(TC - TOE)                 Mean anomaly

Solve for eccentric anomaly            E + MK + ES*SIN(EOLD) [Newton-
                                       Raphson]
ES = EPHEM(INVERS(ISVNO),12)           Eccentricity
EKNEW = MK                             Kepler's equation for Eccen.
                                       anomaly
FOR JX:1:100 DO
      EKOLD = EKNEW
      EKNEW = MK + ES*SIN(EKOLD)
END
E = EKNEW                              Eccentricity anomaly

Compute time correction term
DEKTR = F*ES*SAS*SIN(E)                Relativistic correction term

AF0 = EPHEM(INVERS(ISVNO),7)           7 -
AF1 = EPHEM(INVERS(ISVNO),6)           6 -
AF2 = EPHEM(INVERS(ISVNO),5)           5 -
TGD = EPHEM(INVERS(ISVNO),3)           3 -
TOC = EPHEM(INVERS(ISVNO),4)           4 -
DELT = AF0 + AF1*(TC - TOC) +
      AF2*(TC - TOC)^2 + DELTR -       Transit time
TGD

T = TC - DELT                          GPS time of transmission correction
R = AS*(1.0 - ES*SIN(E))               Satellite average radius R

Compute satellite true anomaly, NU
NUM = SQRT(1.0 - ES*ES)*SIN(E))/       Sin(true anomaly)
            (1.0 - ES*COS(E))
DENOM = (COS(E)-ES)/(1.0-              Cos(true anomaly)
ES*COS(E))
NU = ATAN2(NUM, DENOM)                 Four quadrant inverse tangent
W = EPHEM(INVERS(ISVNO),21)            Argument of perigee [semicircle]
PHI = NU + W                           Argument of latitude

Compute correction terms from harmonics
CUS = EPHEM(INVERS(ISVNO),13)          Amplitude of sine harmonic
                                       correction term to the argument of
```

```
                                        latitude [rad]
CUC = EPHEM(INVERS(ISVNO),11)           Amplitude of cos harmonic
                                        correction term to the argument of
                                        latitude [rad]
CRS = EPHEM(INVERS(ISVNO),8)            Amplitude of sine harmonic
                                        correction term to orbit radius [m]
OMEGA0 = EPHEM(INVERS(ISVNO),17)        Longitude of ascending node of
                                        orbit plane at weekly epoch
CIS = EPHEM(INVERS(ISVNO),18)           Amplitude of sine harmonic
                                        correction term to the angle of
                                        inclination [rad]
CRC = EPHEM(INVERS(ISVNO),20)           Amplitude of cos harmonic
                                        correction term to the orbit radius
                                        [m]
CIC = EPHEM(INVERS(ISVNO),16)           Amplitude of cos harmonic
                                        correction term to the angle of
                                        inclination [rad]
IOTA = EPHEM(INVERS(ISVNO),19)          Inclination angle at reference time
                                        [semicircle]
IDOT = EPHEM(INVERS(ISVNO),23)          Rate of inclination angle
                                        [semicir/s]

C2PHI = COS(2*PHI)                      Double angle
S2PHI = SIN(2*PHI)

Second harmonic perturbations
DELPHI = CUS*S2PHI + CUC*C2PHI          Argument of latitude correction
DELR = CRS*S2PHI + CRC*C2PHI            Radius correction
DELI = CIS*S2PHI + CIC*C2PHI            Corrected inclination

PHI = PHI + DELPHI                      Corrected argument of latitude
R = R + DELR                            Corrected radius
IOTA = IOTA + DELI + IDOT*(T-TOE)       Corrected inclination

OMEGAD = EPHEM(INVERS(ISVNO),22)        Rate of right ascension
                                        [semicircle/s]
OER = OMEGA0 + OMEGAD*(T-TOE) -         Corrected longitude of ascending
            OMEGAE*T                    node

Compute satellite ECEF vector
(meters)
ECEFXS(L) = R*COS(OER) *COS(PHI)- R*SIN(OER)*COS(IOTA)*SIN(PHI)
ECEFYS(L) = R*SIN(OER) *COS(PHI)+ R*COS(OER)*COS(IOTA)*SIN(PHI)
ECEFZS(L) = R*SIN(IOTA)*SIN(PHI)
```

Figure 3.3-2: Derivation of Satellite ECEF Coordinates from Ephemeris Data

3.3.3 Computation of Direction Cosine Terms

The observation matrix partial derivatives relating to the error states, as described in Equation (4), are, in fact, the direction cosines Cxs_i, Cys_i, Czs_i between the local level coordinate axes and the position of each GPS satellite indexed by i. Computation of the direction cosines is facilitated by transforming the platform position and that of each satellite to local level coordinates, from which the desired results may be computed from their difference vector. The computation of platform ECEF coordinates (ECEFXP, ECEFYP, ECEFZP) takes into account the elliptical WGS-84 geodetic model as follows (Equation 7):

$$
\begin{aligned}
R_{EW} &= A_R / \sqrt{1 \quad e^2 \quad \sin^2 \lambda} \\
ECEFXP &= (REW + H) \quad \cos\lambda \quad \cos\varphi \\
ECEFYP &= (REW + H) \quad \cos\lambda \quad \sin\varphi \\
ECEFZP &= ((1 - e^2) \quad REW + H) \quad \sin\lambda
\end{aligned}
\tag{7}
$$

where A_R = Equatorial Earth Radius(ft) = 2.09257382×10^7 ft

e^2 = Earth's Ellipticity Squared = 6.74499984×10^3 (dimensionless)

H = Platform Altitude(ft)

Now that both satellite and platform coordinates in ECEF are computed, we transform both to Local Level coordinates by multiplying by the transformation matrix T_{LE} defined in Equation 5, and compute their difference vector DIFF, as shown in Equation 8:

$$
\underline{LLSAT} = \begin{matrix} LLSAT\ (1) \\ LLSAT\ (2) \\ LLSAT\ (3) \end{matrix} = [T_{LE}] \begin{matrix} ECEFXP \\ ECEFYP \\ ECEFZP \end{matrix}
$$

$$
\underline{LLPLAT} = \begin{matrix} LLPLAT\ (1) \\ LLPLAT\ (2) \\ LLPLAT\ (3) \end{matrix} = [T_{LE}] \begin{matrix} ECEFXP \\ ECEFYP \\ ECEFZP \end{matrix} \tag{8}
$$

$$
\underline{DIFF} = \underline{LLSAT} \quad \underline{LLPLAT}
$$

The computation of the desired direction cosine terms will then be in terms of elements of the DIFF vector defined above (Equation 9):

$$\begin{aligned} CXS_i &= DIFF(1) / SATRANGE \\ CYS_i &= DIFF(2) / SATRANGE \\ CZS_i &= DIFF(3) / SATRANGE \end{aligned} \quad \textbf{(9)}$$

$$\text{where } SATRANGE = [DIFF(1)^2 + DIFF(2)^2 + DIFF(3)^2]^{/2}$$

It should also be noted that the elevation of the satellite above the horizon may also be computed from these vector components. This result will be utilized to eliminate all satellites that do not achieve at least five degrees of elevation from filter consideration:

$$\text{Satellite Elevation Angle} = \tan^{-1} \frac{DIFF(3)}{\sqrt{DIFF(1)^2 + DIFF(2)^2}} \quad \textbf{(10)}$$

3.4 Operation of DGPS RF in Simulator or Terminal Environment and RF Kalman Filter Design

The integration of the DGPS RF within a simulator or terminal environment is based on the availability of a stable Link-16 community navigation solution, since the RF is designed to compute corrections to this solution, and cannot operate without it. In the case of actual **Link-16 terminal integration**, the navigation solution is provided via interface with embedded terminal navigation software. In the case of a **simulator environment**, the navigation solution is provided by one of two Link-16 navigation representations: (a) A simple covariance representation of a community navigation solution having constant but arbitrary position and time errors (COV50) or (b) a comprehensive simulation of Link-16 navigation performance known as the Link-16 Navigation Simulator (LNS). Option (a) provides an easily modified representation of a mobile community, which, though not completely modeling all error sources, nevertheless provides a simple test bed for preliminary RF design and integration studies. Option (b) embodies the actual Link-16 OCP navigation code exactly as implemented within the terminal, and has been proven over many years to exactly replicate community error dynamics. Final performance estimates of DGPS RF operation always use the LNS as the primary navigation reference. (Please refer to Paragraph 4.1 for details on the navigation reference software used in each of the program versions of the DGPS RF.) In the remainder of this section, we review the operational integration of the DGPS RG algorithm with the navigation solution and the host DGPS input data provided via a real or simulated GPS receiver. Also presented herein is the formal definition of the RF Kalman Filter design utilized within this program.

3.4.1 DGPS RF Algorithm – Data Specification

Based on the preceding sections, the input data required for initialization and operation of the DGPS RF may be summarized as follows:

(a) GPS pseudorange information for all selected satellites at a rate commensurate with the RF Kalman Observation rate (1 Hz, 2Hz, 4Hz)
(b) Ephemeris Data, also provided by the GPS Receiver (possible refresh at intervals of approximately 30 minutes)
(c) Terminal Initialization Data (Master/Slave Indication) – One time only
(d) Self Navigation Data (Provided via L16 OCP or Simulator) – 4 Hz Rate
(e) PPLI messages from Master - Nominal Rate One per 12 Seconds
(f) Master Reported Navigation Data and Measured Pseudorange Data
(g) Time Reference for Time Tagging Observation Data

The requirement for differencing self-measured pseudorange data and that of the master platform dictates that a special TADIL-J message, designated as the **Master Message** must be transmitted at intervals by the master platform to provide the data set denoted by (f) to all slave platforms. The interval for this transmittal is subject to tradeoffs between algorithm performance and additional Link-16 bandwidth usage. At this writing, it is believed that a suitable compromise for the rate of Master Message transmission will be at a 2 Hz rate, with either two or

four measurement cycles worth of position and pseudorange data being packed within each message. This rate will provide the fastest convergence of the RF algorithm, measured in seconds. If, however, operational usage of the DGPS RF can tolerate convergence rates in the order of tens of seconds, the rate of the Master Message transmission can be slowed to 1 Hz, and thus incur only 50% of the L16 bandwidth loss. A final decision on this question will be deferred to FY- 04. The current baseline for simulation and laboratory testing remains at a 4Hz RF Kalman Cycle. It should also be mentioned that the usage of PPLI information as one of the two observation types is periodic, but in general asynchronous with the processing of pseudorange differences. The operation of the DGPS RF is fundamentally based upon the **Kalman Cycle,** nominally 4 Hz, but reducible to 2 Hz or 1 Hz as desired.

Figure 3.4-1 indicates a simplified representation of the data flow required to support the operation of the DGPS RF. It should be noted that, with the single exception of a request to transmit a Master Message (shown dotted) and exercised for a master platform only, no data from the DGPS RF is fed back to the OCP. The results of the DGPS RF estimation process are provided only to the host, and are, likewise, not fed back to the existing Link-16 navigation computations. Those operate precisely as they are currently implemented within Link-16 terminal assets.

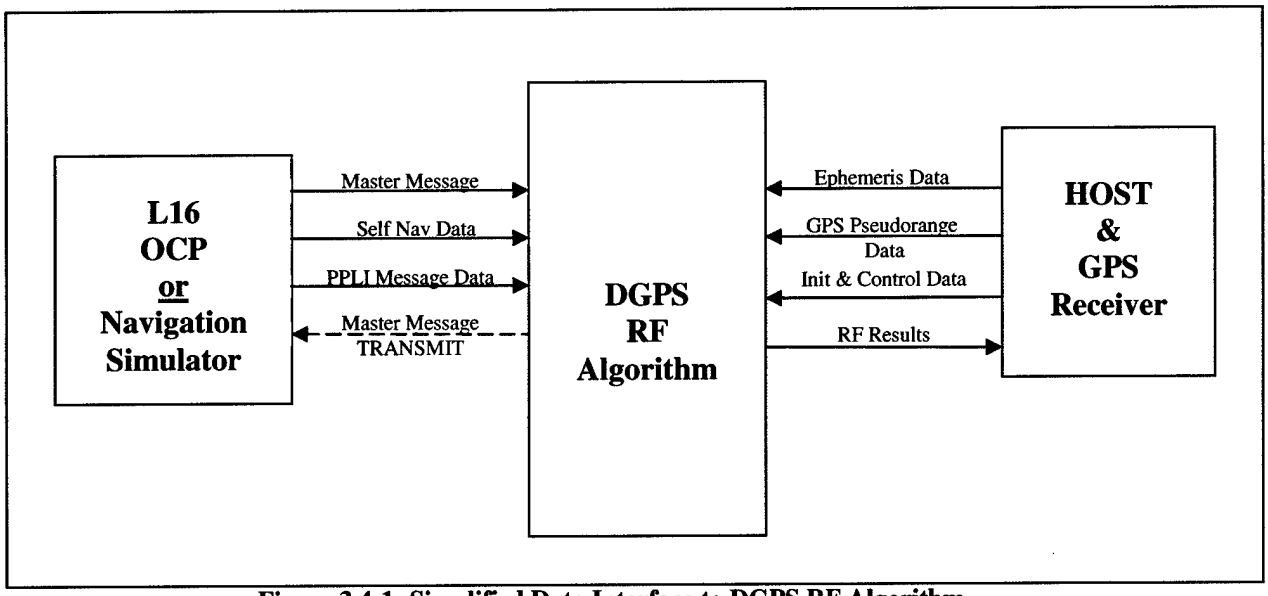


Figure 3.4-1: Simplified Data Interface to DGPS RF Algorithm

3.4.2 DGPS RF Kalman Processing Overview

The basis of the DGPS RF is that of a discrete Extended Kalman Filter (EKF), operating at a nominal Kalman Observation Cycle of 250 milliseconds. The theory of discrete Kalman Filtering is available in many references. See, for example reference (5). A summary of the fundamental processing equations for a discrete Kalman filter is given in Figure 3.4-2 extracted from reference[5].

System Model Measurement Model	$x_k = \Phi_{k\ 1} x_{k\ 1} + w_{k\ 1}, \quad w_k \sim N(0, Q_k)$ $z_k = H_k x_k + v_k, \quad v_k \sim N(0, R_k)$
Initial Conditions Other Assumptions	$E[x(0)] = \hat{x}_0, E[(x(0)] \quad \hat{x}_0)(x_0 \quad \hat{x}_0)^T] = P_0$ $E[w_k v_j^T] = 0$ for all j, k
State Estimate Extrapolation Error Covariance Extrapolation	$\hat{x}_k(\ ) = \Phi_{k \quad 1} \hat{x}_{k \quad 1}(+)$ $P_k(\ ) = \Phi_{k\ 1} P_{k\ 1}(+) \Phi_{k\ 1}^T + Q_{k\ 1}$
State Estimate Update Error Covariance Update Kalman Gain Matrix	$\hat{x}_k(\ ) = \Phi_{k \quad 1} \hat{x}_{k \quad 1}(+)$ $P_k(\ ) = \Phi_{k\ 1} P_{k\ 1}(+) \Phi_{k\ 1}^T + Q_{k\ 1}$

Figure 3.4-2: Summary of Discrete Kalman Filter Equations

The design of the discrete Kalman equations to the DGPS RF centers on the selection of the appropriate observation matrix (H), the computation of the corresponding transition matrix, Φ_A, the provision of representative error models for the two types of potential observation (R matrix), and appropriate process noise models (Q matrix). We have previously specified the configuration of the maximum state vector, X, in Equation (2). The error velocity terms (states 2, 4, and 6) were initially included in the DGPS RF design and evaluated for effectiveness (See Paragraph XXX for details). The configuration of the transition matrix, Φ_A, can thus have two possible forms, one not including the velocity states, and one where the states are included. The resultant form for matrix Φ_A in both cases are given in equation 11:

The elements of the observation matrix (H) have previously been specified in Equation 4. The H(1,1),H(1,3),and H(1,5) terms applied to range processing derived from PPLI measurements will be set to zero for all Kalman cycles for which a PPLI from the master has not been received. For cycles in which a PPLI is present, those terms will be computed using the expressions in Equation 4. The terms H(k,1), H(k,3), H(k,5), and H(k,7) will usually be nonzero for all Kalman cycles for all k over the range (2 ... NVS + 1), where NVS is number of processed satellites in any given cycle.

The detailed specification of the Initial state covariance matrix (P), the Observation Noise Matrix R, and the Process Noise Matrix (Q) may be found in the DGPS RF Software Requirements Specification, presented in Appendix A.

$$\Phi_A = \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}$$

Transition Matrix - No Velocity Terms

$$\Phi_A = \begin{matrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}$$

Transition Matrix - Velocity Terms Present

(11)

4. Methods and Procedures

The purpose of this section is to present a comprehensive account of the design studies and analyses conducted during the FY03 phase of the DGPS RF contract. The primary focus of these efforts centered on the analysis and simulation of the basic DGPS RF algorithm on a series of simulation test bed programs. The result of this development process was a robust and effective algorithm capable of meeting the performance goals set at the project's beginning. However, the FY 03 program had goals beyond the definition and development of the algorithm itself. Since the FY04 program anticipates laboratory demonstration of the algorithm integrated within a Link-16 terminal and exercised on the Terminal ATP Test Set (TATS), a significant amount of study and effort was expended on the formal specification of real time RF software via a Software Design Document (SDD). The SDD was a necessary first step for the generation of Operational Code suitable for inclusion within the terminal, as well as the design of code changes to the existing OCP needed for integration with the RF. Finally, modifications to the TATS itself to accommodate the satellite pseudorange measurements and the generation of the Master Message were also required. The objective of all these efforts was a comprehensive implementation plan for near-operational level software design and test equipment capabilities to enable coding to begin immediately after FY04 begins in October 2003.

At the inception of this project, it became apparent that several different versions of the software algorithm were necessary. The first of these, Version 1.0, was designed to function entirely within a simulation environment (laptop or PC). Therefore, all algorithm interfaces with the supporting community navigation solutions and GPS pseudorange measurements were provided via software generation of the needed interface data. The laboratory test version of the DGPS RF algorithm, Version 2.0, was expressly designed for inclusion within the terminal. Community navigation inputs required by the DGPS RF algorithm will now be derived directly from the terminal operational software, rather than via digital simulation. Furthermore, in the interest of testing a near-operational version of the software, the laboratory TATS pseudorange inputs will be derived from a full elliptical satellite model, rather than the circular model used in V.1. The projected Version 3.0 of this software will be a fully flight qualified design that will differ from V.2 only in the details of the GPS host pseudorange data transfer. Integration with a real vs. simulated GPS receiver will take place for V3.0. It is significant to note that with the exception of the OCP and navigation interface design, all three versions of the algorithm share common code for the algorithm implementation.

4.1 Description of DGPS RF Software Versions

Each of the software versions of the DGPS RF was designed for a specific operating environment and was tailored to support specific simulation and testing objectives. Version 1.0

was designed to operate within a PC or laptop environment and had as its primary objective the determination of basic algorithm performance and operating characteristics. Version 2.0 is the version of the test software tailored for terminal operation within a laboratory test environment. A projected Version 3.0 also operates within a terminal environment, but will be designed to function as a flight test enabler using an engineering pallet within an actual test aircraft. The structure and organization of each of these versions is described below.

4.1.1 DGPS Version 1.0

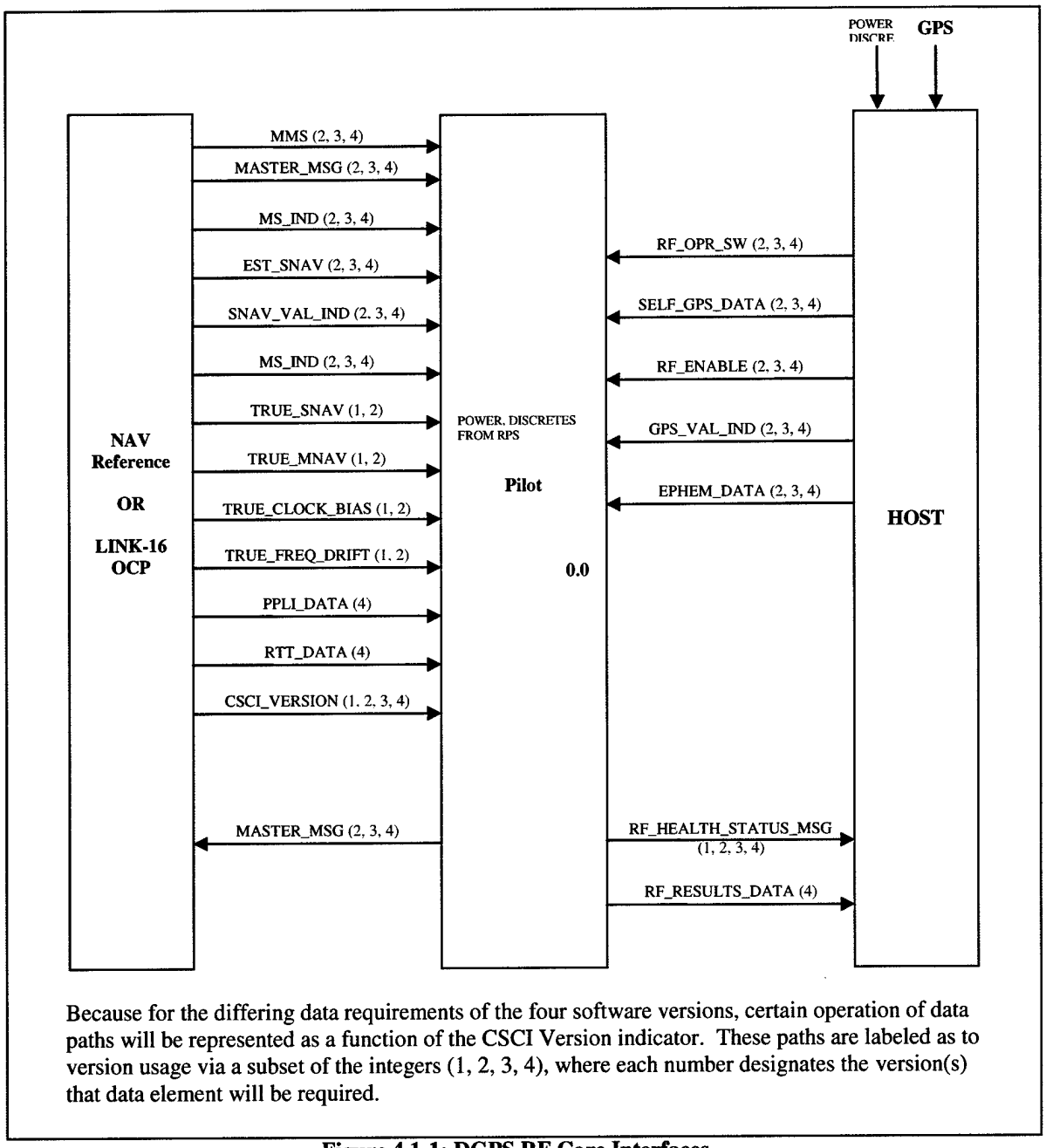
The architecture of the DGPS RF algorithm, in all program versions, is built around seven core Computer System Components (CSCs), whose titles and general functionality are presented in Table 4.1.1-1. A detailed design description of each of these CSCs is discussed in paragraph 4.3.

CSC	Description
RF_EXEC_CTRL	Controls the data flow within Refinement Processing depending on inputs to the system.
RF_SOURCE_DATA_PROCESSING	Prepares and synchronizes all data for Kalman filtering.
RF_KALMAN_PROCESSING	Applies Kalman filtering to the data passed from the RF_SOURCE_DATA_PROCESSING unit.
HOST_INTERFACE_PROCESSING	Interfaces with host and processes data that enters this system from the host, mainly control signals and GPS data.
OCP_INTERFACE_PROCESSING	L16 OCP interface from which all OCP data enters the system, mainly navigation solution including position, velocity, and time.
BUILD_MASTER_MESSAGE	Called if the platform is designated the master of the network from the host. The master message is sent at a 2 Hz rate to all the slaves in the network.
BUILD_RF_RESULTS_MESSAGE	Outputs DGPS RF solution, status, and covariance to host computer.

Table 4.1-1: DGPS RF Computer Software components and description

Operation of the core modules within the DGPS RF algorithm requires the establishment of both Operational Computer Program (OCP) and host interfaces. The OCP interface comprises the (simulated or real) availability of pseudorange measurements and master navigation solution provided by a (simulated or real) Master Message transmitted from the master, and the navigation solution of the host platform. The host interface requires the provision of (simulated or real) GPS own platform pseudorange measurements, GPS ephemeris data, and certain control inputs.

A comprehensive interface description of the data transfer across these interfaces for all software versions is presented in Figure 4.1 -1



Because for the differing data requirements of the four software versions, certain operation of data paths will be represented as a function of the CSCI Version indicator. These paths are labeled as to version usage via a subset of the integers (1, 2, 3, 4), where each number designates the version(s) that data element will be required.

Figure 4.1-1: DGPS RF Core Interfaces

Within the DGPS RF Version 1.0, the OCP interface is provided by community navigation simulation software that generates the estimated and true navigation solutions of the simulated community. The community navigation capability was provided by one of two navigation software packages designed and developed by BAE SYSTEMS. The first, and simplest of these was a navigation simulator designated as **COV50**, which was a covariance-

based software package developed prior to this project as a "quick-look" estimator of community navigation performance. This program was used in the early development stages of the project due to its ease of use and modification, as well as its non-proprietary status. The second, and more complex navigation model used to provide the community navigation solution was the **Link-16 Navigation Simulator (LNS).** This proprietary BAE SYSTEMS product was used for detailed performance studies due to the fidelity with which it represents the true error characteristics of a mobile LINK-16 community. Appendix B provides a listing of COV50, as well as a description of the primary attributes and capabilities of the LNS. It should be noted at this point that the LNS is mandated to track the operational navigation code implemented within Link-16 assets, and must be considered the primary reference in all questions of Link-16 community navigation performance.

The GPS satellite dynamic profile in Version 1.0 was provided by the simple circular satellite prediction algorithm described in paragraph 3.3.1. For operation of the DGPS RF Version 1.0, we note that either the COV55 program or the LNS was used to provide the community navigation solutions required by the algorithm. When the COV55 program was utilized, we will designate the composite program as **Test Bed 1.** When the full LNS is installed, we designate the composite program as **Test Bed 2.** Figures 4.1-2 and 4.1-3 present simplified schematics of the interfaces implemented within Test Beds 1 and 2.

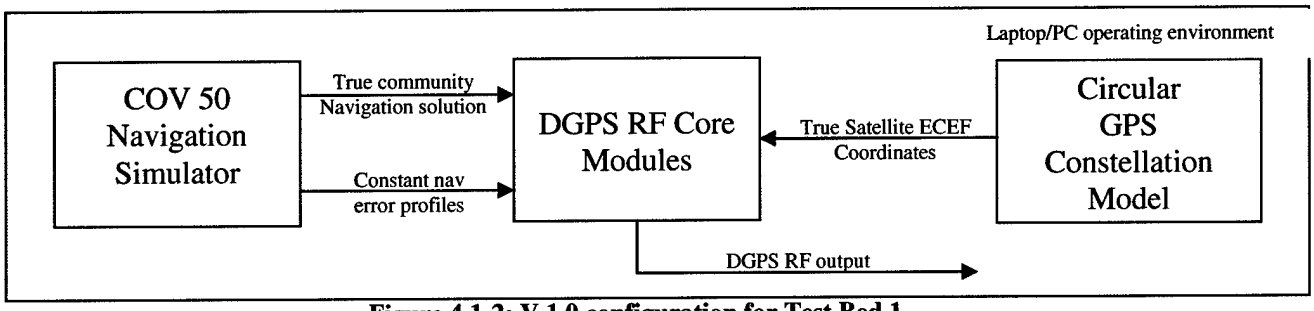


Figure 4.1-2: V.1.0 configuration for Test Bed 1

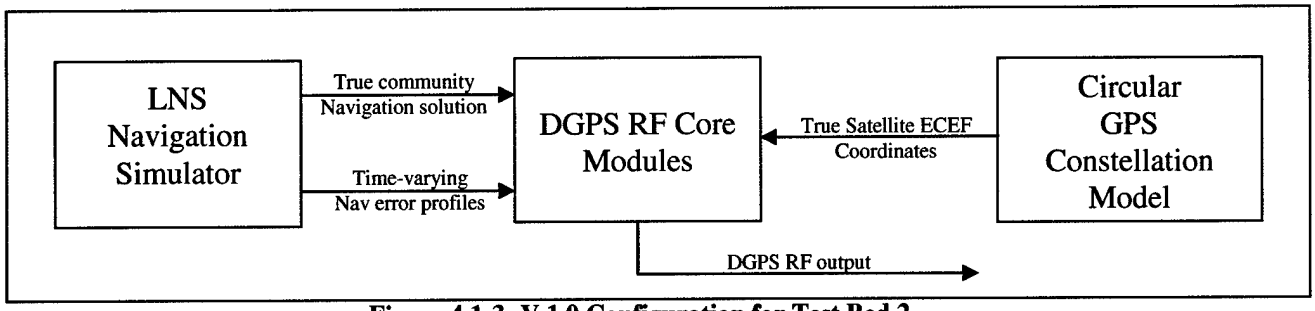


Figure 4.1-3: V.1.0 Configuration for Test Bed 2

Test Bed 1, which provides dynamic community error trajectories but constant navigation error profiles was used during the early months of the project for fundamental algorithm design, test, and experimentation. In contrast, Test Bed 2, which provided realistic community navigation error profiles, was used to determine algorithm performance levels under a wide variety of conditions. Paragraph 4.2 provides full details of the testing and analysis performed using these two programs. Test Bed 2 is the basis for the interim PC demonstration scheduled for September 2003.

4.1.2 DGPS RF Version 2.0

Version 2.0 of the DGPS RF is designed for integration and real-time execution within an actual Link-16 terminal in a laboratory environment. Version 2.0, integrated with the Link-16 Operational Computer Program (OCP) will operate on BAE SYSTEMS Terminal ATP Set (TATS), which provides a real time testing environment for Link-16 terminal assets. The TATS environment, more fully discussed in paragraph 4.5, supports Link-16 testing by providing the Terminal Under Test (TUT) with host and RF inputs that such a unit would see in actual flight operation. Under the control of a scenario tape (now in CD format), the supporting test terminal provides actual message traffic that would be present in a test community. The LRU interface provides simulated host inputs, such as control and INS acceleration data needed to support the terminal's normal operation. For the operation of the DGPS RF program, however, certain inputs not presently supported by the TATS must be developed and supplied to the terminal. These inputs include pseudorange data provided by a GPS satellite model, control inputs, and the generation of Master Messages from the simulated master platform. Note that the TUT will always be representing a slave unit in this environment.

Unlike V1.0, in which community and self-navigation is provided by either the COV55 or the LNS navigation programs, Version 2.0 is fully integrated within the Link-16 CORE software package, and receives all necessary interface information from the embedded software package in which it operates. (Figure 4.1-4) Please note that during the testing of the DGPS RF code, the TUT is executing its navigational function, processing received PPLI and RTT messages as part of its normal operation.

Figure 4.1.2-1 presents a simplified block diagram of the MIDS LVT1 terminal, and shows the location of the OCP within the terminal data processor. In the expansion diagram of the Link-16 OCP, it is seen that the DGPS RF V2.0 will be embedded within this software load, and will include both RF OCP and RF host interface code. Host data that is required by the RF will pass through the physical host interface, located in the (unchanged) Tailored I/O Module. RF interface data passing through this processor will be routed to the (software) Host I/O processing for later submission to the RF algorithm.

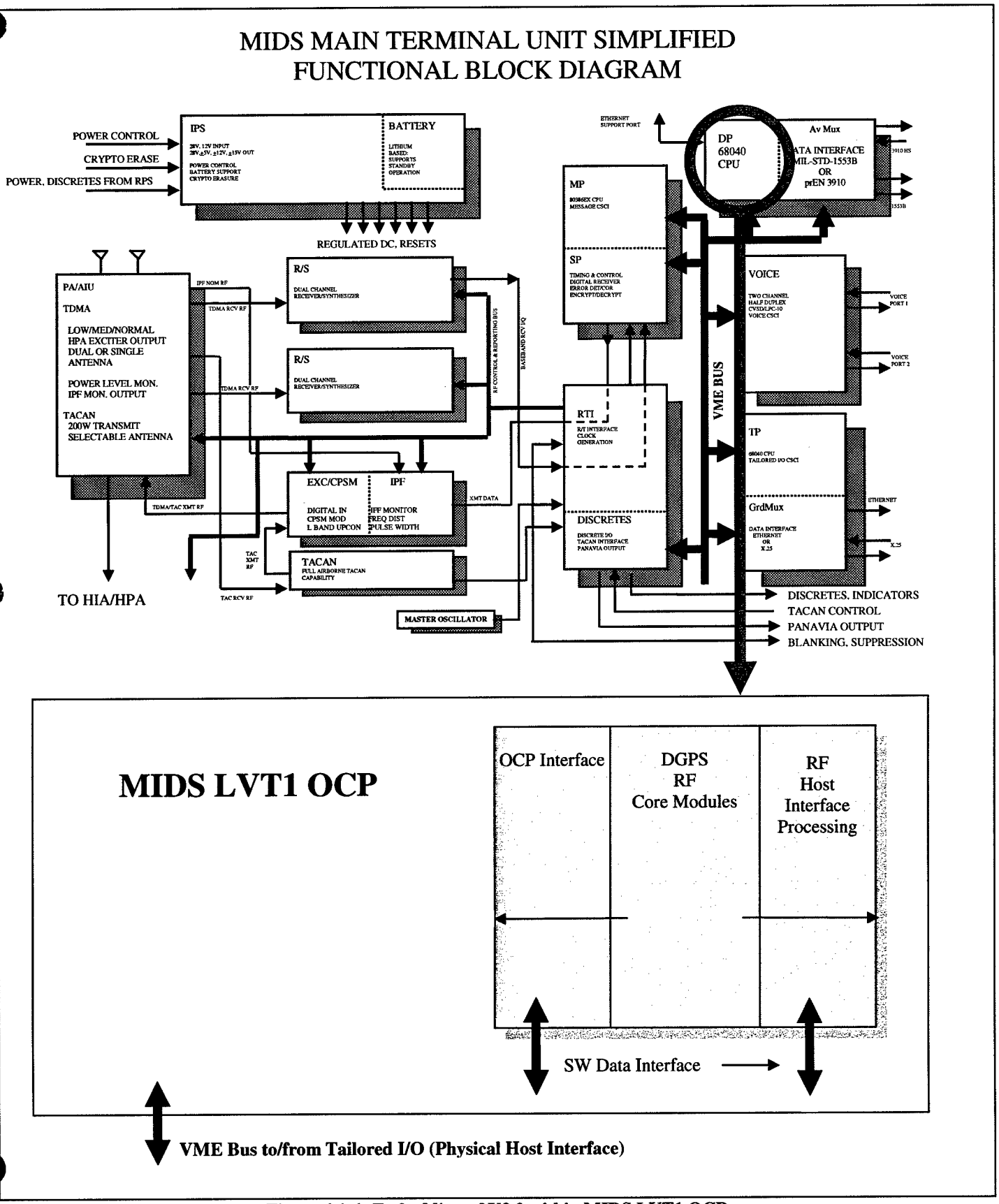


Figure 4.1-4: Embedding of V2.0 within MIDS LVT1 OCP

In contrast to Version 1.0, which uses a circular satellite model, Version 2.0 is designed to use the **full elliptical satellite model**, whose data must be supplied by the TATS equipment as described in paragraph 4.5. This simulated pseudorange data will be furnished to the TUT in the format of a typical military GPS receiver, in this case, the Canadian Marconi NORTHSTAR unit. In this regard, Version 2.0 is very close to being operable with actual satellite units. Version 2.0, as implemented within a Link-16 terminal on the Terminal ATP Test Set will be used for the final FY04 phase demonstration in March 2004.

4.1.3 DGPS Version 3.0

Version 3.0 is the designation for a potential flight-qualified version of the DGPS RF software, should the scope of this project be expanded to incorporate flight-testing. As such, it will be designed to operate with an arbitrary GPS receiver set, a (provisional) data link processor connecting the Link-16 terminal, GPS receiver, and a data recording device. Nominal configurations of both ground and airborne equipment suites are indicated in Figure 4-5.

In contrast with the test configuration of V2.0, which derives its INS acceleration measurements, GPS pseudorange, and PPLI messages from the TATS, V3.0 will interface with actual INS and GPS units, and all messages utilized will be physically generated via the Link-16 terminals. As was the case with Version 2.0, both Link-16 terminal units will be supporting their navigational functions just as they do in normal operation. DGPS RF solution data will be supplied to the recording apparatus for further analysis, but in no event will such solutions be fed back to the Link-16 navigation processing.

4.1.4 DGPS Version 4.0

A parallel research program performed by BAE SYSTEMS for ONR, called Improvement of Link-16 Navigation via Real-Time Atmospheric Modeling is directed at improving Link-16 navigation through improved estimation of atmospheric propagation effects. To this end, a new algorithm designated as the **Atmospheric Filter (AF)** was designed and tested. The goal of this model was to yield an order of magnitude or better of estimation accuracy for PPLI message TOA measurements. The designation DGPS V4.0 is a placeholder for a potential software package that could combine the capabilities of DGPS V3.0 with the AF algorithm range measurement improvement. The goal of this combination would be to exploit the synergy of these two algorithms to provide superior navigation capability to a mobile community in the event that GPS capability was jammed or otherwise unavailable. This hybrid configuration is not currently under consideration by ONR or other agencies.

4.2 Test Bed 1 Simulation Activities

Overview and Rationale

The overall objective of the series of simulations performed with Test Bed 1 was to provide a baseline analysis of the capabilities of the DGPS RF algorithm, and, more specifically, to determine if the performance goals of the DGPS program (i.e. < 1 meter positioning, < 1 ns time synchronization) were supportable. Within the FY03 activities, the Test Bed 1 simulation study was the first systematic research performed into the sensitivities of the DGPS RF algorithm. Although this was the first formal study of the DGPS RF algorithm, it was, in fact, preceded by related studies performed during April – August 2001under the DARPA-sponsored PAT3 program, in collaboration with BAE SYSTEMS IEWS of Nashua, NH. The goal of this program was to combine extremely precise community positioning and time synchronization with innovative signal processing and geolocation algorithms for accurate location of hostile radar emitters.

The initial design of the DGPS algorithm was performed for PAT3 under severe financial and time constraints, which provided no opportunity to conduct a systematic study of the algorithm's capabilities. Using an early version of the COV50 navigation simulator, a small number of quick reaction simulations produced promising early results that indicated the potential of DGPS for reducing position and time errors to small levels. Figure 4.2-1 is an example of the results obtained during PAT3 simulation studies. Under the terms of the PAT3 program, BAE SYSTEMS was directed to evaluate the performance of the DGPS algorithm using recorded GPS satellite pseudorange data collected by SAIC in Arizona during flight tests performed in spring 2000. These studies were the first application of the use of the elliptical satellite ECEF algorithms for the DGPS, and resulted in mathematically stable and consistent position error determinations. Although no reference solution accurate enough to verify these results was available, the resulting position solutions agreed with recorded pseudorange data to well within a meter. (Figure 4.2-2)

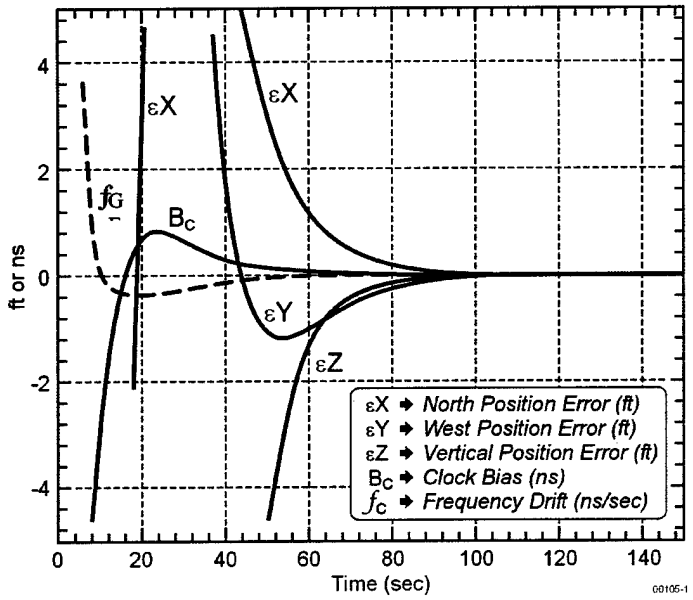


Figure 4.2-1: Refinement Filter nominal performance

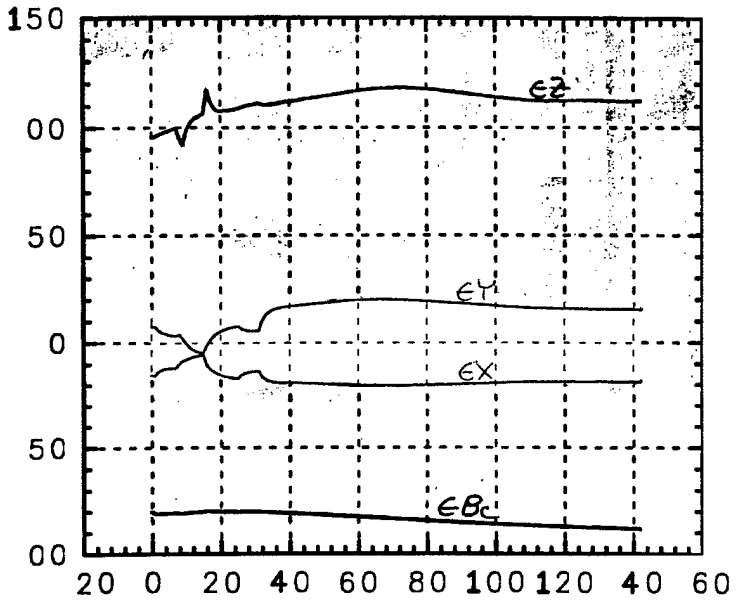


Figure 4.2-2: Refinement Filter Performance with Recorded GPS Data - Mission 10/Dwell 11

The above results, although encouraging, provided very few reference points to determine algorithm performance under varying conditions of initial errors, observation rates, and Kalman Filter tuning conditions. It is for this reason that the Test Bed 1 series was designed to evaluate performance of the algorithm under carefully controlled error characteristics.

Parameter Test Space for Test Bed 1 Studies

The Test Bed1 simulation series was designed to investigate performance of the DGPS algorithm while varying key algorithm parameters. A list of the key parameters that were varied and their test limits is presented in Table 4.2-1

Parameter Type	Allowable Ranges	Comments
a) Link-16 Position Errors	$\varepsilon x, \varepsilon y, \varepsilon z$ = 2, 20, 200,2000	Normal L16 Position Errors < 100 ft.
b) Link-16 Velocity Errors	$\varepsilon V_x, \varepsilon V_y$ = 0,1,2 ft/sec	Normal L16 Velocity Errors < 1 ft/sec
c) Time Base Clock Bias & Frequency Drift	εB_c = 0, 200 ns εf_c = 0, 10 ns/sec	Normal GPS time errors < 40 ns
d) DGPS Kalman Filter Observation Rate	1 Hz, 2 Hz, 4 Hz	-
e) DGPS Kalman Filter Covariance (initial)	Pos. States = $(1000 \text{ ft})^2$ Vel. States = $(1 \text{ ft/sec})^2$ Clock Bias = $(100 \text{ ns})^2$ Freq. Drift = $(10 \text{ ns/s})^2$	Exceptions as noted for large initial errors
f) DGPS Kalman filter Process Noise	Position = $(1000 \text{ ft})^2$ Velocity = $(0.2 \text{ ft/s})^2$ When velocity states present	-
g) Use of Velocity States	N/A	States 2, 4, 6 included where indicated

Table 4.2-1: Range limits of RF test parameters

Performance of the DGPS algorithm when the above parameters were varied was evaluated against the desired position and time estimation results. In addition, the solution stability and robustness, though more difficult to quantify precisely, was a further metric against which performance in each case was graded.

Measurement Variance Values

Values assumed for all test conditions for the measurement observation variances were:

R(1,1) = Variance of PPLI Range Measurement = (21.0 ft)**2
R(2,2) = Variance of GPS Pseudorange = (4.0 ft)** 2

Link-16 Community and Navigation Solution

The Link-16 community for these runs was restricted to two members, namely the master and slave platforms, since the DGPS RF only considers the relative errors between these two platforms. A fundamental requirement for the initiation of a DGPS RF solution is the prior establishment of a stabilized Link-16 navigation solution within the participating community. It is assumed that the Link-16 navigation algorithm operates in a conventional manner, that is, PPLI messages at a rate of once per 12 seconds are transmitted by the master, and processed by the slave. In addition, it is also assumed that both members utilize GPS PVT data at intervals of 12 seconds as inputs to their Navigation Kalman Filter.

The trajectory assumed for the participating community is depicted in Figure 4.2.1 – 3, below. Both aircraft are assumed to fly at a constant speed of 800 feet per second. The master aircraft begins the trajectory at a position of 45 degrees north latitude, 45 degrees west longitude, and traverses the racetrack pattern in a clockwise manner. The slave aircraft begins at a position of 44.8 degrees latitude, and 45 degrees longitude, and traverses the flight pattern in a counter-clockwise pattern.

For Test Bed 1 simulation, navigation errors are varied for each run, and include constant position error and/or constant velocity errors. Please note that these are the relative position errors in the Link-16 solution that the DGPS is designed to estimate.

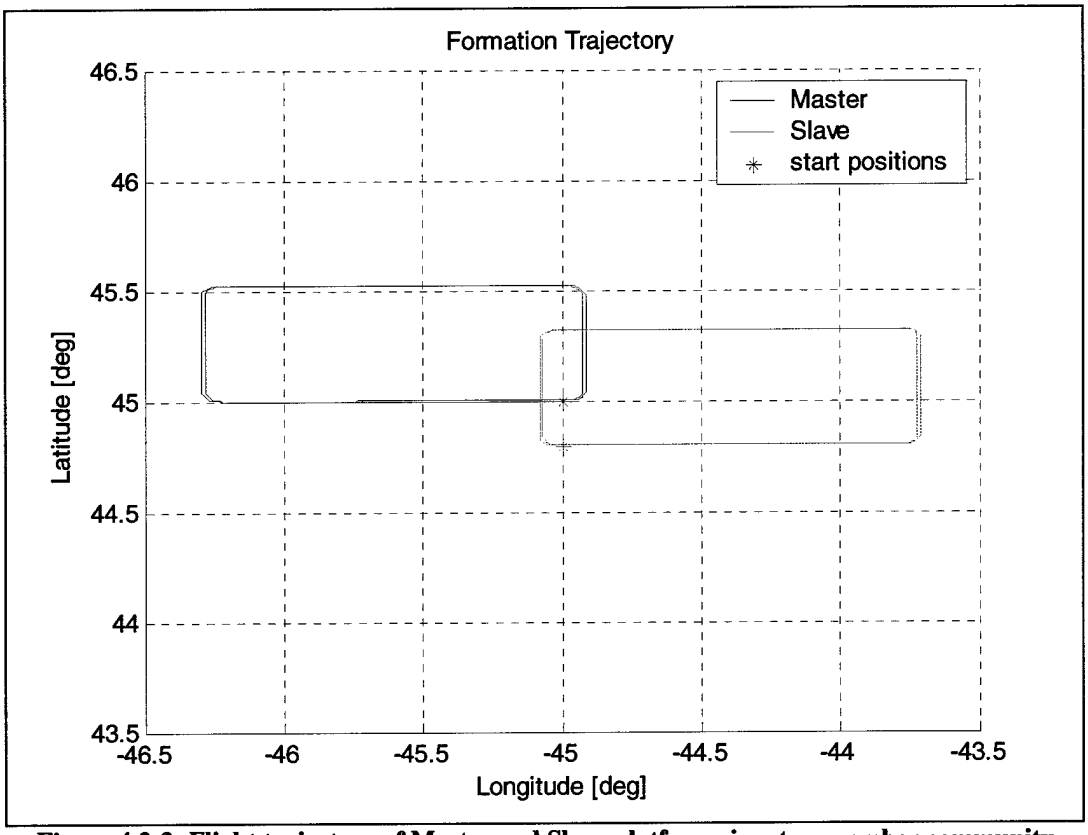


Figure 4.2-3: Flight trajectory of Master and Slave platforms in a two member community

Application of DGPS RF Results – Open vs. Closed Loop

There are two possible approaches to application of the results of the DGPS RF algorithm, which we refer to as **Open Loop** and **Closed Loop**. In the case of Open Loop, we maintain the best estimates of the Link-16 navigation errors exclusively within the DGPS RF Kalman Filter in the estimated state values, and produce a "best composite estimate" of member navigation outside the two filters in the host computer. In the case of Closed Loop, we periodically pass the DGPS corrections back to the Link-16 Navigation Kalman Filter, and zero the corresponding error state estimates of the DGPS _RF. Both of these approaches were evaluated within the Test Bed 1 simulations, and the approach that was selected for use was the Open Loop. There are many reasons for this choice. As will be fully explained in later sections, the Closed Loop approach results in long-term solution instability, which is not present when using the Open Loop approach. In addition, by combining the estimates of two Kalman Filters in anything less than a fully optimal fashion, serious Link-16 navigation stability issues are likewise raised. Therefore, for logistical and technical reasons we would prefer to leave the operation of the trusted and reliable Link-16 navigation algorithm untouched, and combine the results of both filters off-line from both estimators.

Preview of Test Bed 1 Simulation Results

Execution of the Test Bed 1 studies ultimately resulted in a stable, robust algorithm that satisfied all performance goals. During the course of the study, several significant design rules for the DGPS RF algorithm were formulated and verified, as follows:

(a) Use of Closed Loop application of RF corrections appears to yields long-term instability in nearly all cases. In some cases, the solution instability takes hundreds or thousands of seconds of simulation time to appear. Use of Open Loop application always yields a stable and reliable solution.

(b) The use of the three velocity error states within the DGPS RF is ineffective in both Closed Loop and Open Loop operation. In the former case, highly oscillatory solutions can result, while in the latter case, no significant difference was observed in performance whether the velocity states are present or not.

(c) The resulting DGPS RF algorithm was virtually insensitive to the magnitude of the Link-16 navigation position error. Excellent convergence resulted even in the case of initial errors an order of magnitude or more (i.e. > 1000 feet/axis) then nominal Link-16 performance. Unmodeled velocity errors under two feet/second per axis (large compared to nominal Link-16 hybrid performance) similarly had little or no effect.

(d) The DGPS algorithm was virtually insensitive to the magnitude of the initial clock bias and frequency drift errors up to 200 ns and 10 ns/second provided the initial covariance values covered the actual error.

(e) Convergence of the DGPS RF was shown to be a function of observation rate, although in a highly nonlinear fashion. Best performance, of course, was achieved at a 4Hz pseudorange observation rate with convergence in a few seconds the norm. Conversely, stretching the observation rate to 1 Hz increased convergence time to 50-80 seconds, depending on size of initial errors.

(f) Little sensitivity was noted with respect to initial KF state covariance values provided they were large enough to exceed true state errors.

In general, filter worked well when covariance values exceeded actual errors by large margins. Corollary: High process noise modeling kept filter window open in all cases and resulted in stable operation.

Full discussion of all of the above conclusions is provided in the analyses of each of the test cases, which constituted this portion of the design study.

4.2.1 Test Case 1-1 (Closed Loop) Baseline Test – Small Initial Position Errors

The objective of this test case was to attempt to replicate the error trends established during previous research as shown in Figure 4.2-1. For this case, the observation rate matched that of the reference run, with a 1 Hz DGPS RF update rate. No velocity states were utilized in this run.

Initial Conditions

Link-16 Position Errors: (ERRX, ERRY = 2 feet)
Clock Bias Error (50.0 ns)
Frequency Error (0.0 ns/sec)

Graphical Results

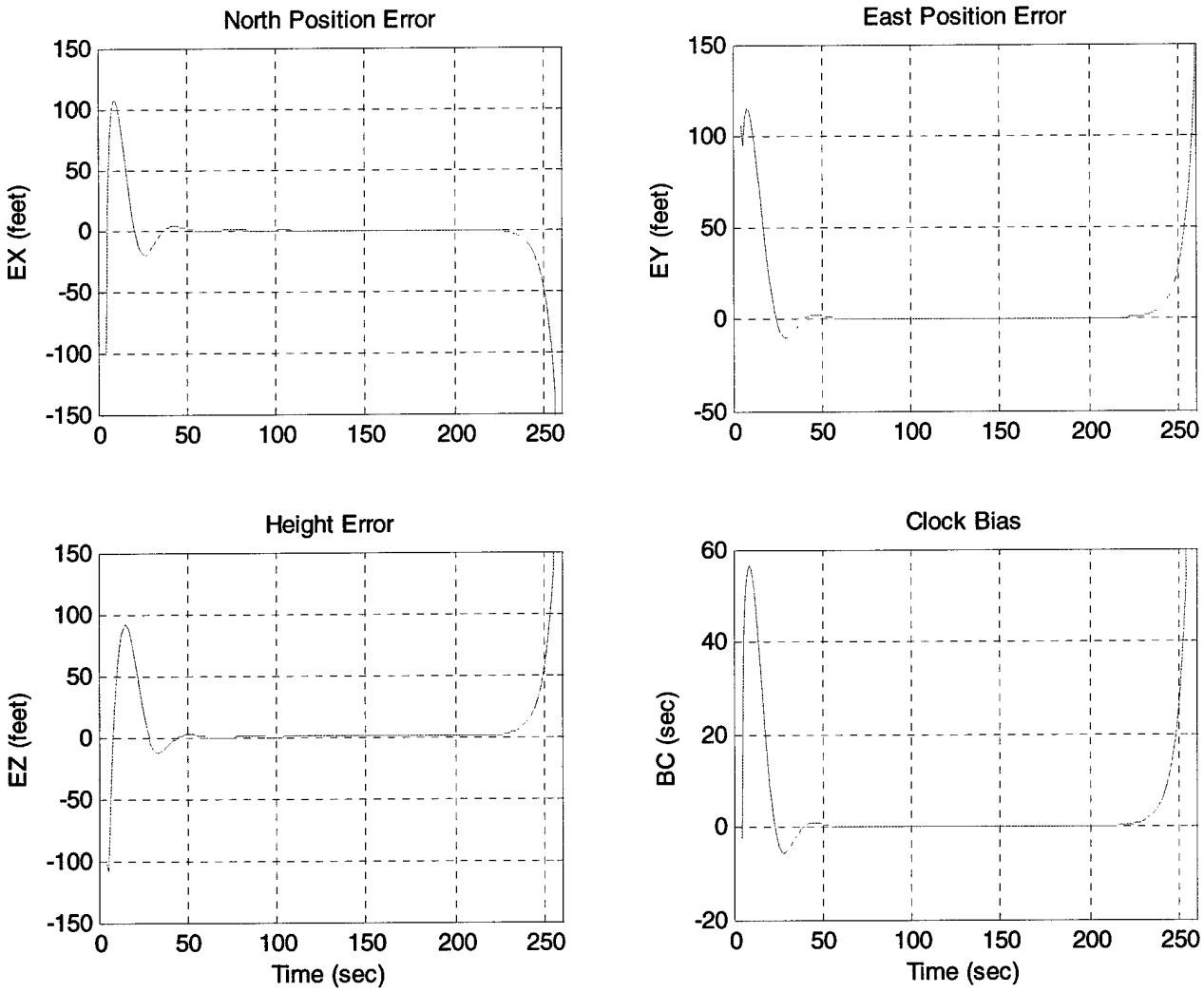


Figure 4.2-4: Closed Loop Baseline Test

Analysis of Results

The solutions shown above bear a close resemblance to the results presented in Figure 4.2-1 during the first 200 seconds with respect to both the position errors and clock bias errors. These results indicated a filter convergence time constant for a 1 Hz update rate of approximately 50 seconds, which is the case here as well. In the reference case, however, the solution extended only to a time of 150 seconds, so any potential solution instability did not have time to occur. In the present case, it is clear that there is a condition of instability that causes the divergence of the solution to occur at approximately 230 seconds. The probable cause of this instability was the closed loop feedback of the DGPS estimated errors into the Link-16 error values. Note that the Link-16 Kalman Filter was not involved in this process, since only a covariance model of the position errors was in use. However, the presence of this solution feedback within the DGPS RF clearly has the potential to destabilize the solution. We wish next to investigate whether an increase in the observation rate from the 1 Hz value used here to 2Hz or 4 Hz can stabilize the solution. As the results presented in the next paragraph indicate, however, the problem will persist as long as closed loop operation of the DGPS RF is maintained.

4.2.2 Test Case 1-2 (Closed Loop) 2Hz, 4Hz Update Rates

The objective of this test case was to investigate whether or not higher update rates (2Hz, 4Hz) would stabilize RF operation in a closed loop mode. No velocity states were used in these runs.

Initial Conditions

Link-16 Position Errors (ERRX, ERRY = 2 feet)
Velocity Errors: 0 ft/sec.
Clock Bias Error (50 ns)
Velocity States - Not Present

Graphical Results

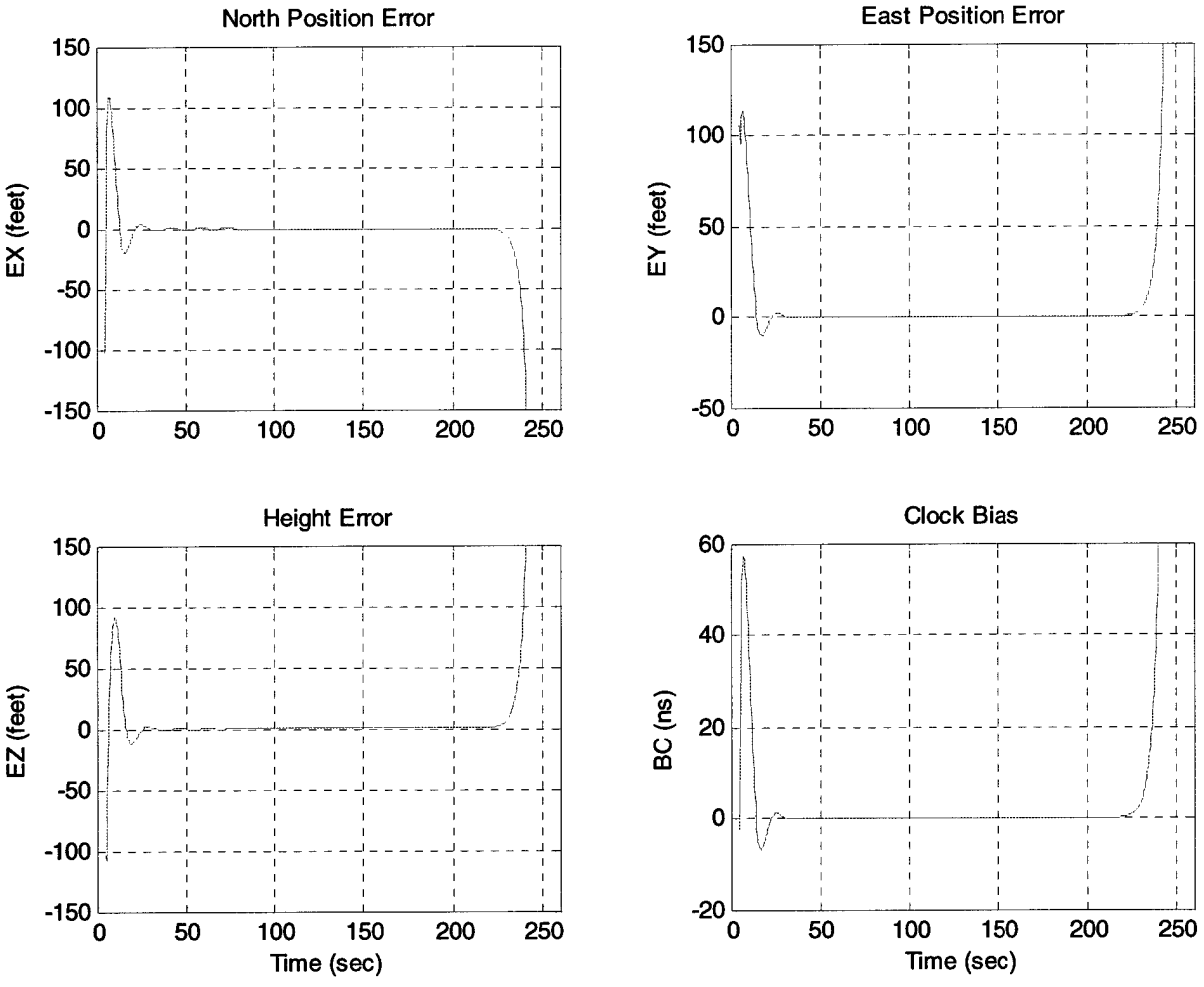


Figure 4.2-5: 2Hz update rate

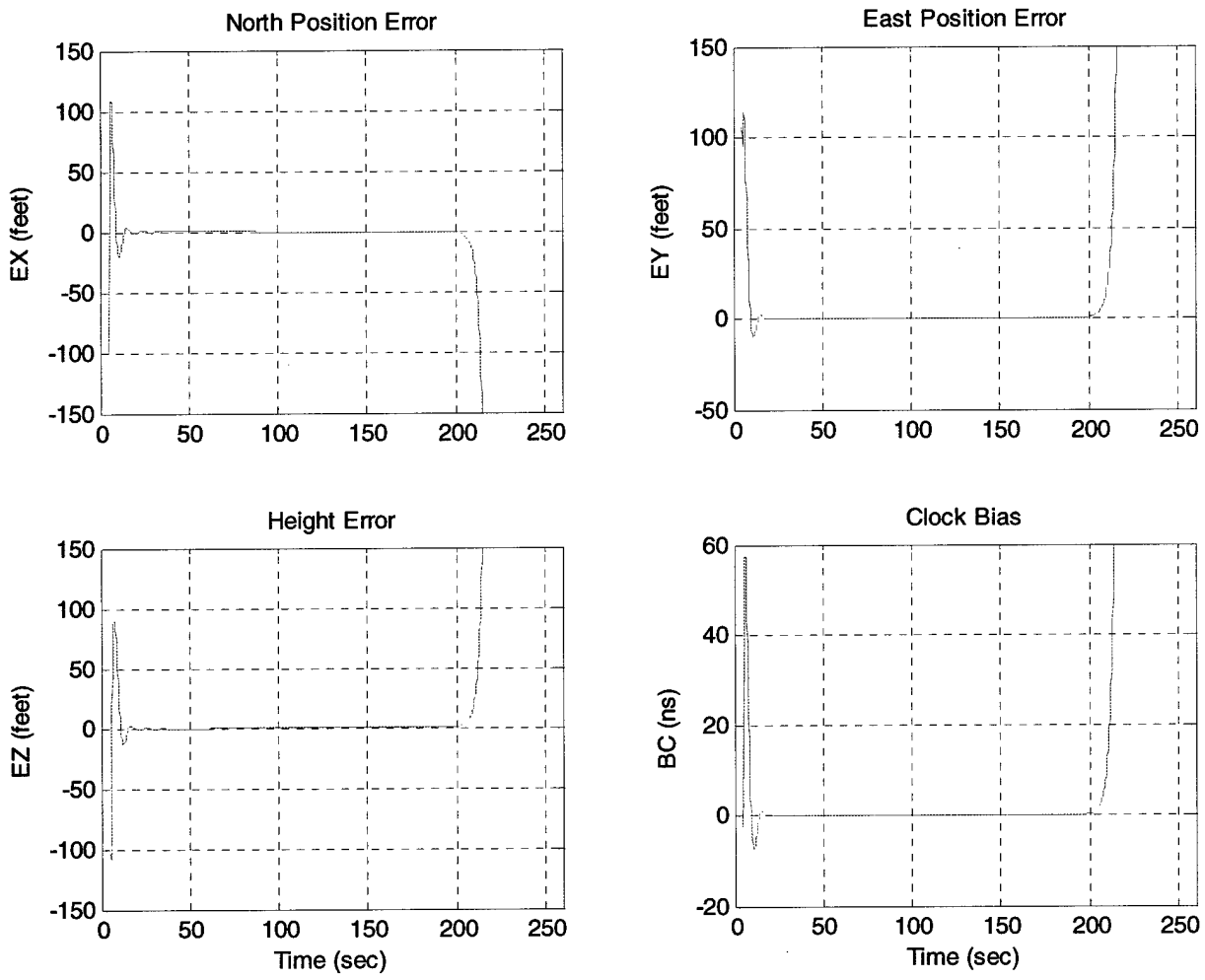


Figure 4.2-6: 4Hz update rate

Analysis of Results

The effect of increasing the observation rate to 2 Hz (Figure 4.2-5), and 4 Hz (4.2-6) indicate that more rapid updates increase the rate of filter convergence. In the case of 2 Hz updates, the estimation errors are driven to near zero within about 20 seconds, while in the case of 4 Hz updates, the convergence time is 5-6 seconds. However, in both cases, increasing the rate of observations does not solve the closed loop stability problem, and results diverge at nearly the same time (230 seconds) that was noted in the previous test case. It is therefore clear, that increasing observation rate alone does not solve the stability problem noted in closed loop operation.

4.2.3 Test Case 1-3 (Closed Loop) Effect of unmodeled Velocity Errors, No Velocity States

The objective of this test case is to investigate the effect on solution stability of unmodeled Link-16 velocity errors when no velocity states are included within the DGPS RF model. For this case, we provide north and east velocity errors of 1 foot per second in the Link-16 navigation solution, which represents an extremely high value not likely to be experienced during normal operation. A 1 Hz observation rate is assumed for this test case.

Initial Conditions

Link-16 Position Errors (ERRX, ERRY = 2 feet)
Link-16 Velocity Errors (EVN, EVE = 1 foot/sec)
Clock Bias Error (50 ns)
Frequency Error (0.0 ns/sec.)
Velocity States: Not Present

Graphical Results

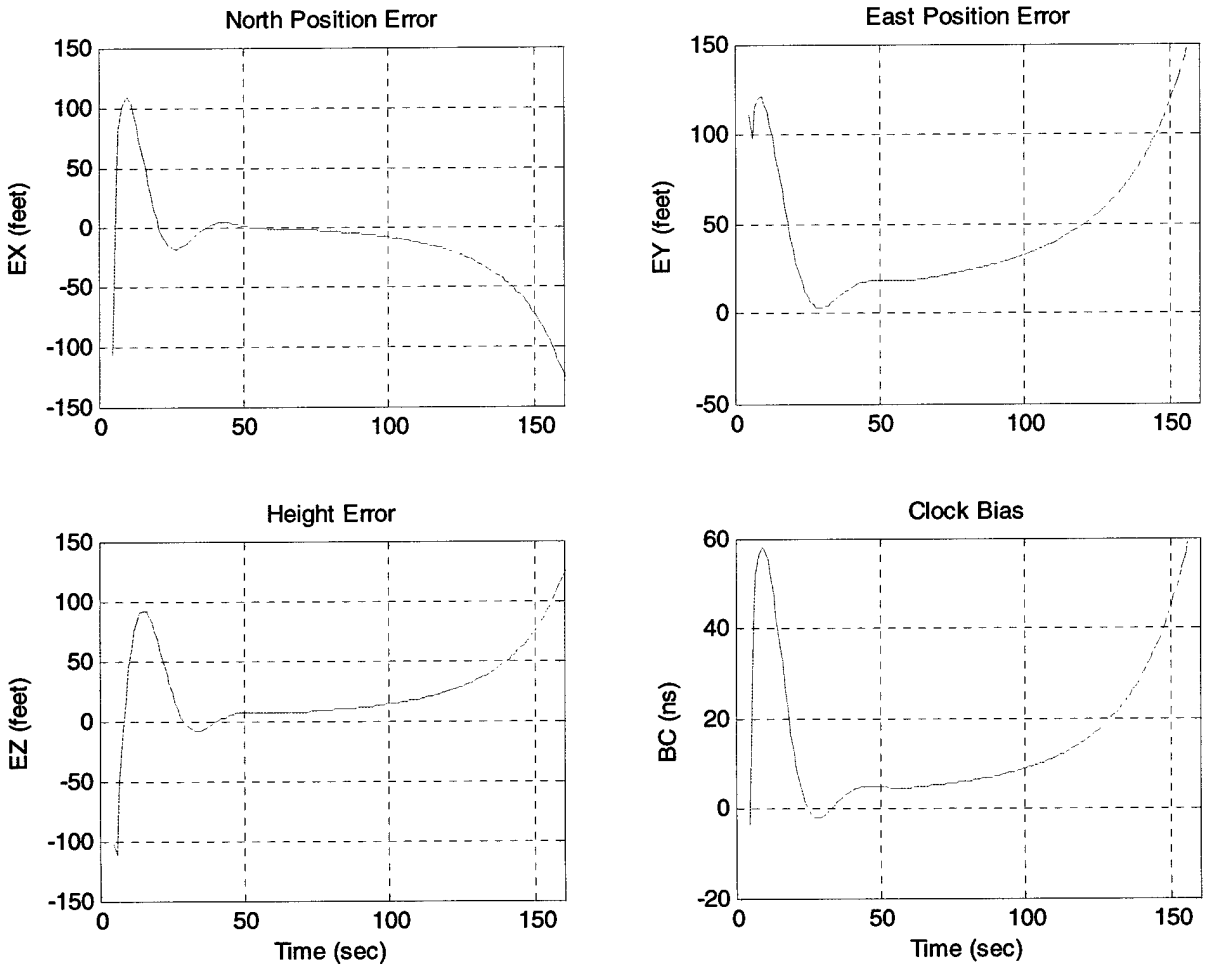


Figure 4.2-7: Unmodeled velocity errors, no velocity states

Analysis of Results

The destabilizing effect of unmodeled velocity errors in closed loop operation is obvious from the results shown above. Although the filter drives the errors to near zero values as in Test Case 1-1 within 50 seconds, these errors are seen to grow almost immediately , and the solution diverges dramatically thereafter. The presence of unmodeled velocity errors in closed loop operation without velocity states is thus noted to have a severe destabilizing effect. Attempts to reduce this error by increasing the observation processing rate to 2 Hz and 4Hz (graphs not shown) did not materially affect the results. The following test case (1-4) will attempt to rectify this situation by adding the three velocity error states to the Kalman Filter. The presence of the velocity states in closed loop, as described in paragraph 4.2.1.4 , will further destabilize the solution by causing an oscillatory response.

4.2.4 Test Case 1-4 (Closed Loop) Effect of Unmodeled Velocity Errors (Velocity States Present)

The objective of this test case was to investigate the effectiveness of adding the three velocity states to the Kalman Filter model. This case was identical to the previous case except for the velocity states. Three observation rates were investigated, namely 1 Hz, 2Hz, and 4 Hz.

Initial Conditions

Link-16 Position Errors (ERRX, ERRY = 2 feet)
Link-16 Velocity Errors (EVN,EVE = 1 foot/second)
Clock Bias Error (50 ns)
Frequency Error (0 ns/sec)
Velocity States - Present

Graphical Results

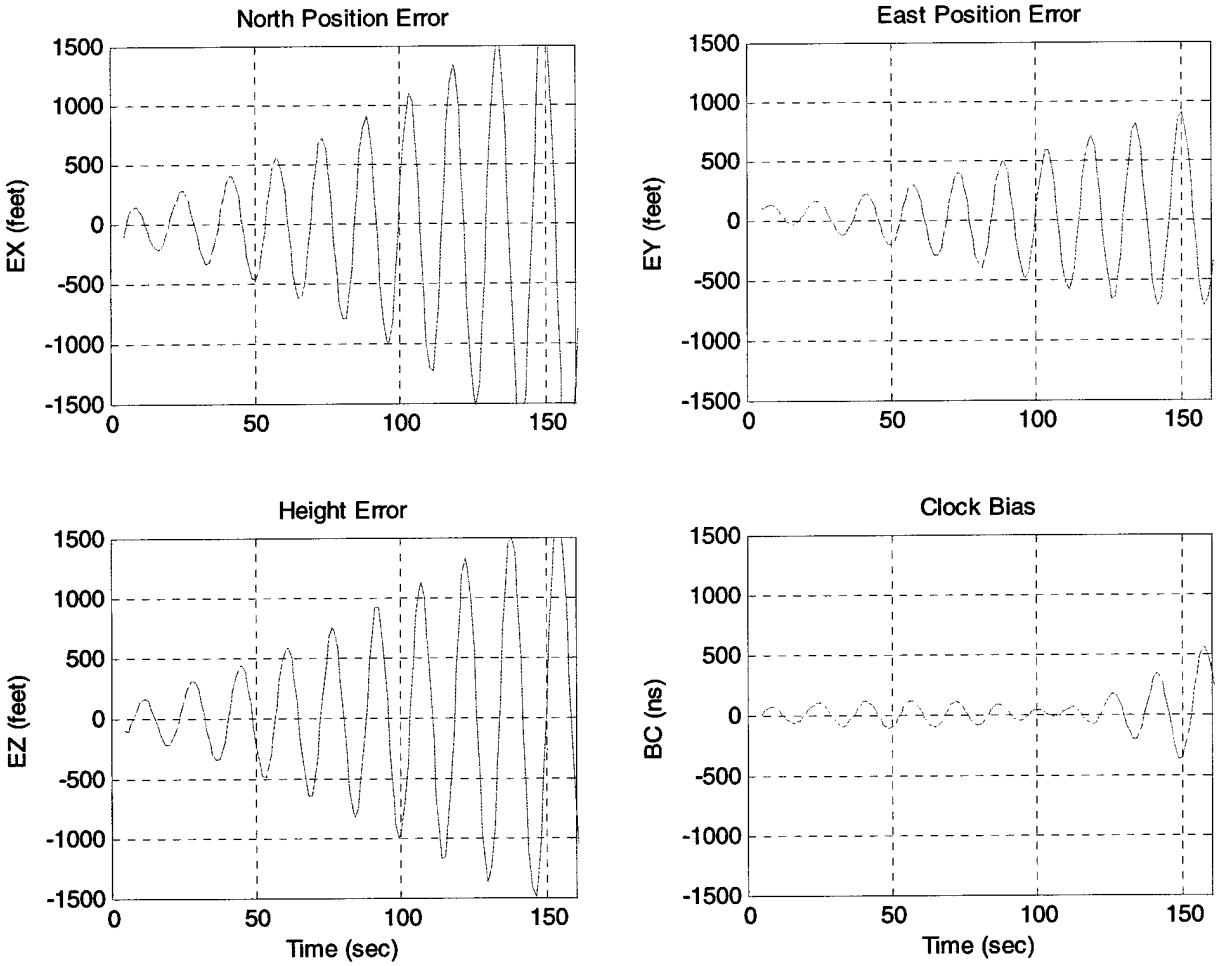


Figure 4.2-8: Velocity states, 1Hz update rate

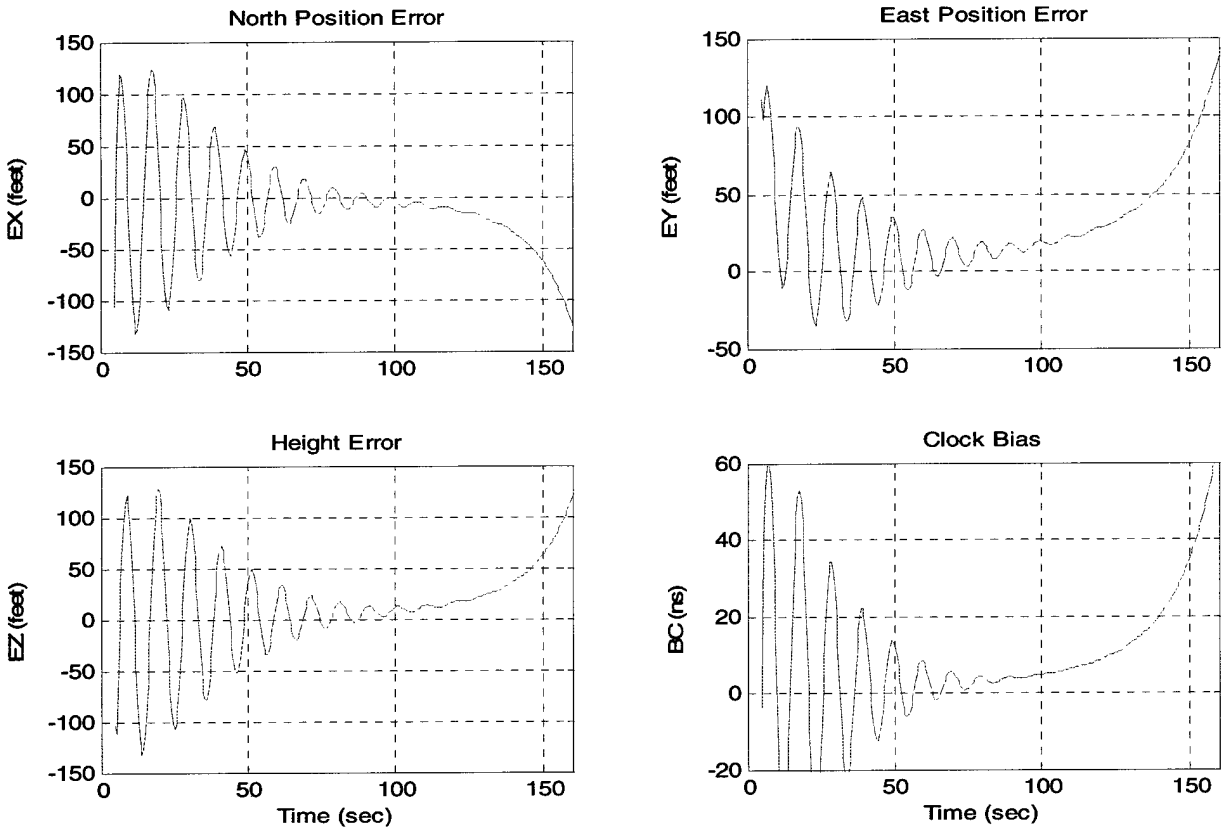


Figure 4.2-9: Velocity states, 2Hz update rate

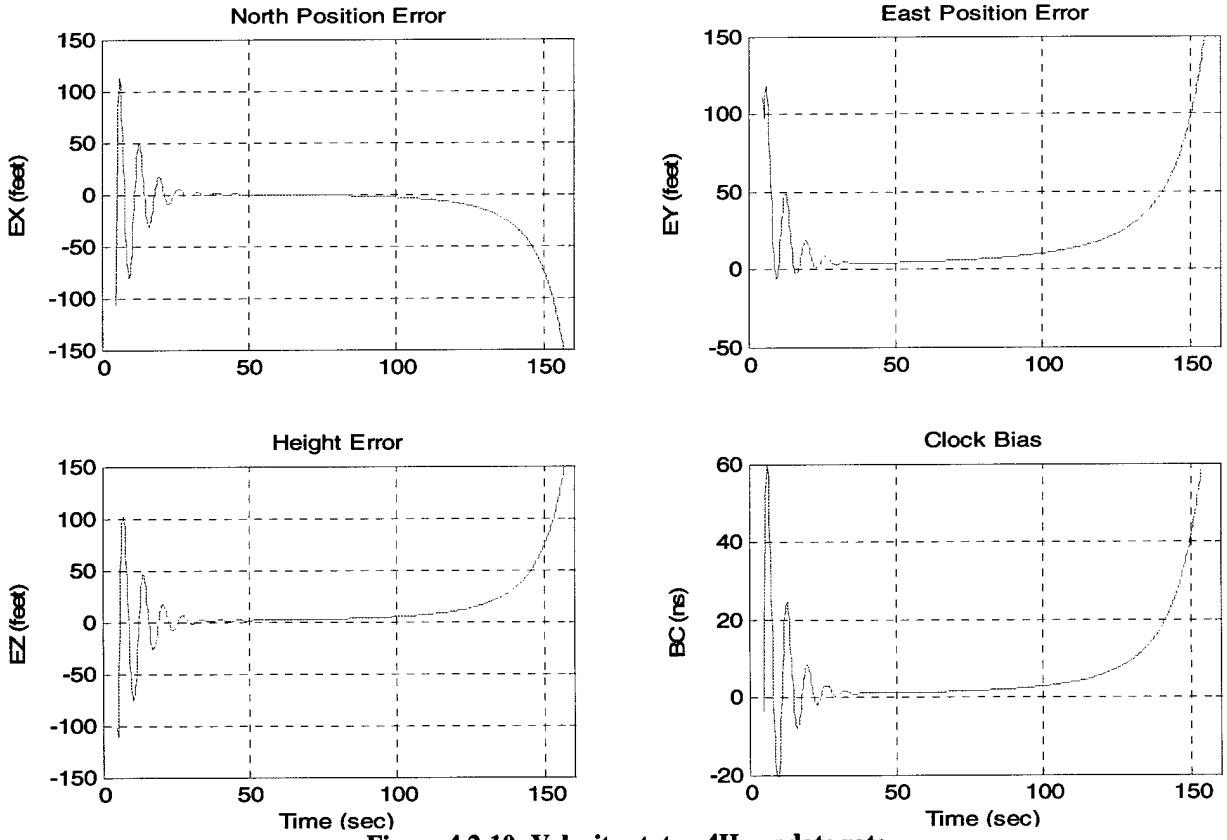


Figure 4.2-10: Velocity states, 4Hz update rate

Analysis of Results

For all observation rates, the inclusion of the velocity states induces a highly oscillatory mode of behavior, with a period of approximately 17 seconds. In the case of the 1 Hz observation rate (Figure 4.2-8), the oscillations continue to increase in magnitude throughout the run. With a 2 Hz update rate (Figure 4.2-9), the oscillations damp out until 100 seconds, at which point a non-oscillatory excursion of the state errors begins, leading to a divergent solution. For the 4 Hz measurement rate, the oscillations damp out more quickly, and the errors appear to be settling toward small values. However, the errors diverge, as before, at approximately 100 seconds. The conclusion is that in the presence of unmodeled velocity errors during closed loop operation, the solution cannot be stabilized even by increasing the measurement rate. Based upon Test Cases 1-1 through 1-4, the closed loop solution must be considered unstable under all conditions. The source of this difficulty lies in the **feedback nature** of the correction process, in which the filter corrections systematically correct the navigation error terms, at which point they are zeroed. There appears no simple or obvious way to correct this problem without breaking the solution feedback loop. As a result, we will now consider the performance of the algorithm in the "Open Loop" mode, in which the values of the state variables are maintained separately from the Link-16 navigation solution, and not zeroed at the time of application. Operationally, this means that the computation of the best estimates of member position must be computed outside both the Navigation Kalman Filter and the DGPS RF Kalman filter by combining their outputs without filter reset.

4.2.5 Test Case 1-5 (Open Loop) Baseline Test – Effect of Position Errors

The objective of this test case is to evaluate the performance of the DGPS RF algorithm working in an Open Loop mode, while allowing controlled variation of Link-16 position errors and observation update rates. For this test case, we set unmodeled velocity errors to zero, and delete the velocity states from the Kalman Filter. We are also interested to determine the practical limits, if any, on the initial Link-16 position errors. As the basic DGPS algorithm is an Extended Kalman Filter, we expect that there might be a region of convergence, within which, the errors will be reduced to small values, but outside which, divergence results. As it turns out, such a limit does appear to exist in the vicinity of 760 feet per axis of position error.

Initial Conditions

Link-16 Position Errors (ERRX,ERRY = 2/200/760/765 feet)
Link-16 Velocity Errors (EVN, EVE = 0.0 ft/second)
Clock Bias Error (50 ns)
Frequency Error (0.0 ns/sec)
Velocity States – Not Present
Measurement Update Rate – 1 Hz

Graphical Results

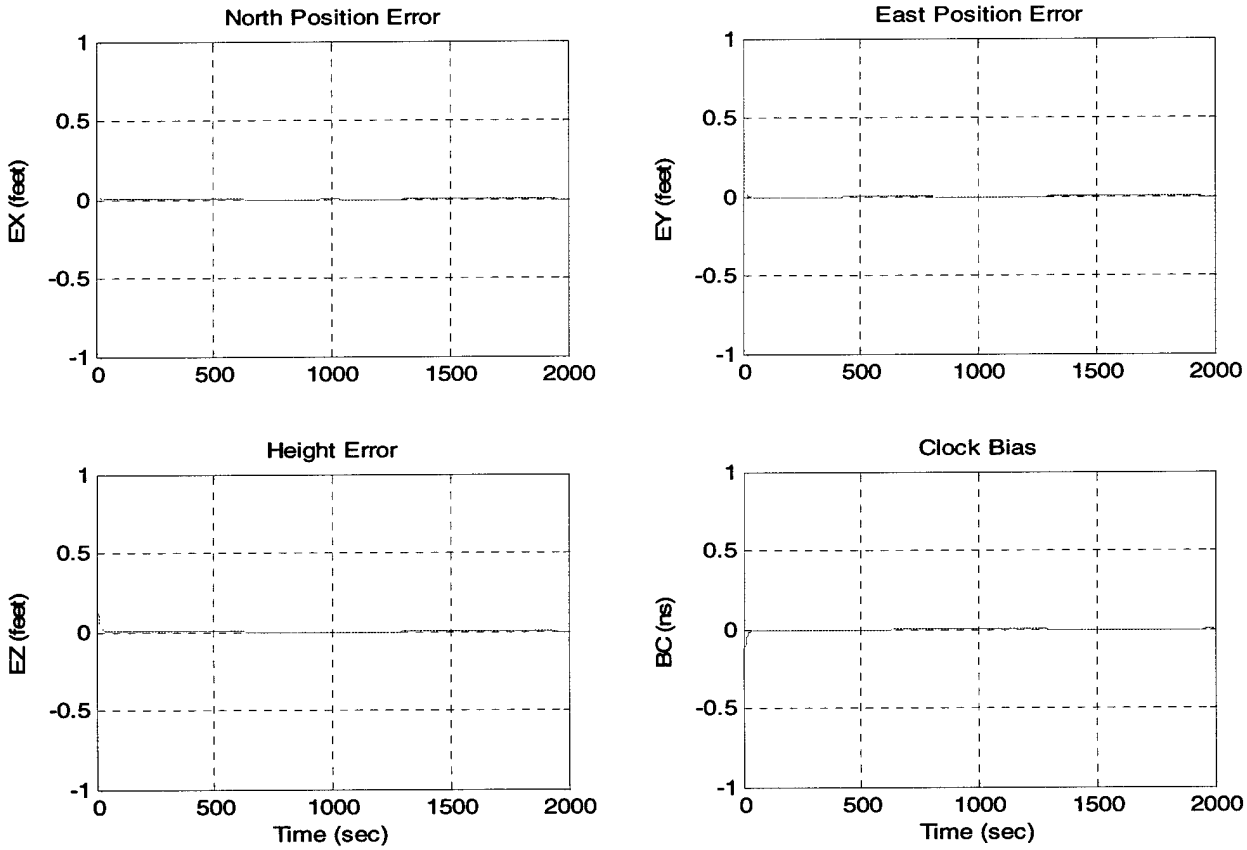


Figure 4.2-11: Open loop, small position errors

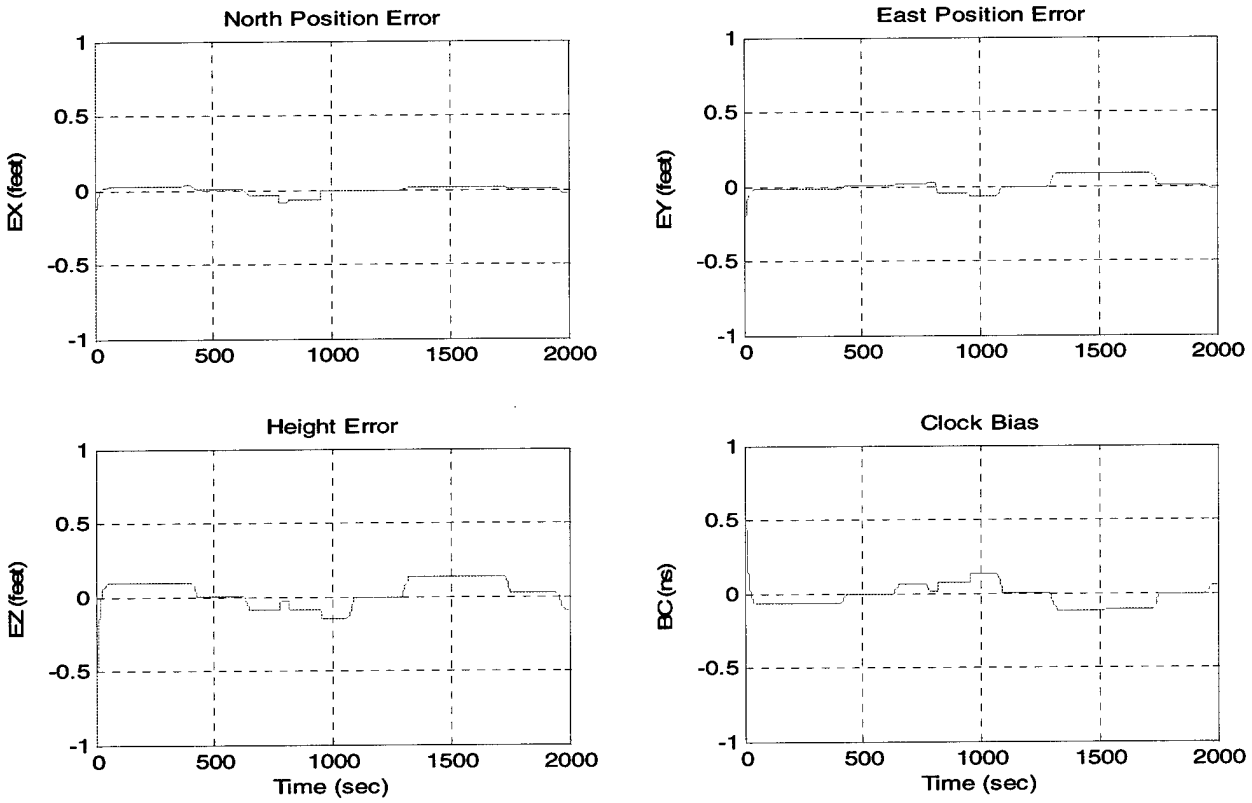


Figure 4.2-12

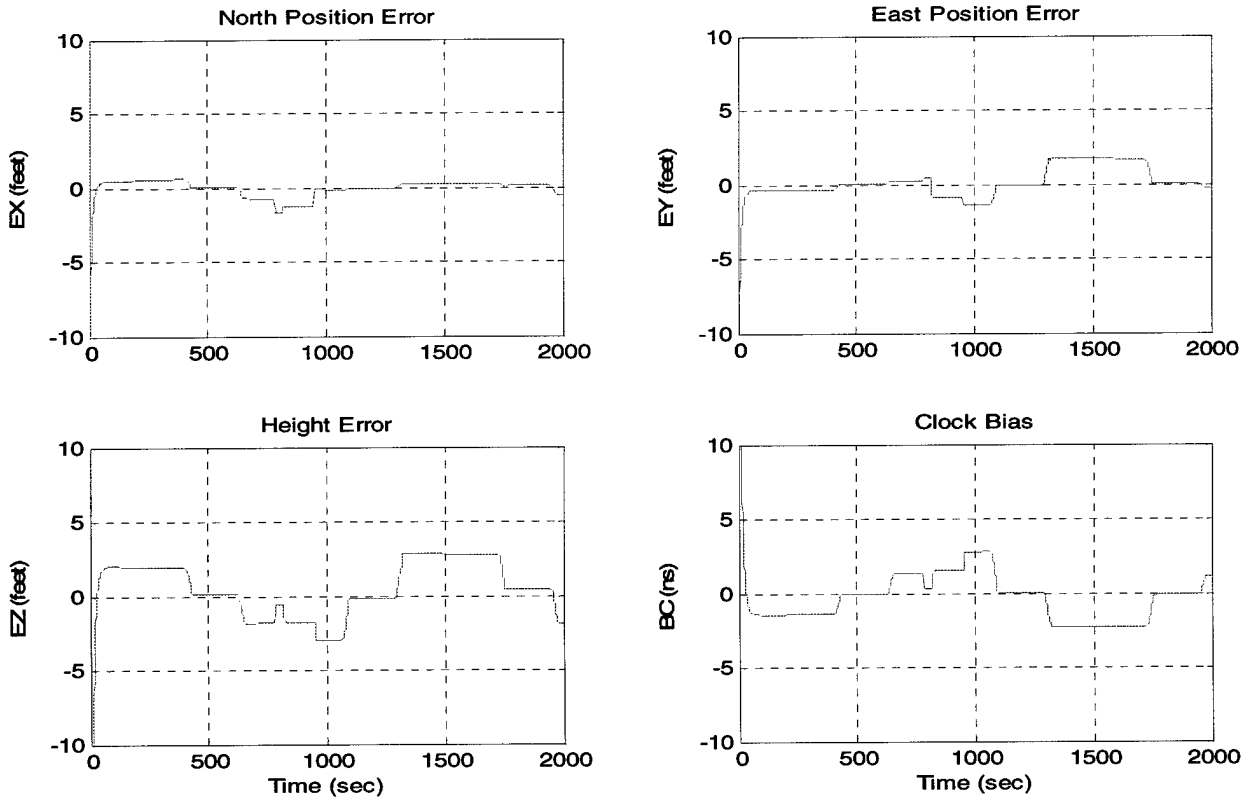


Figure 4.2-13

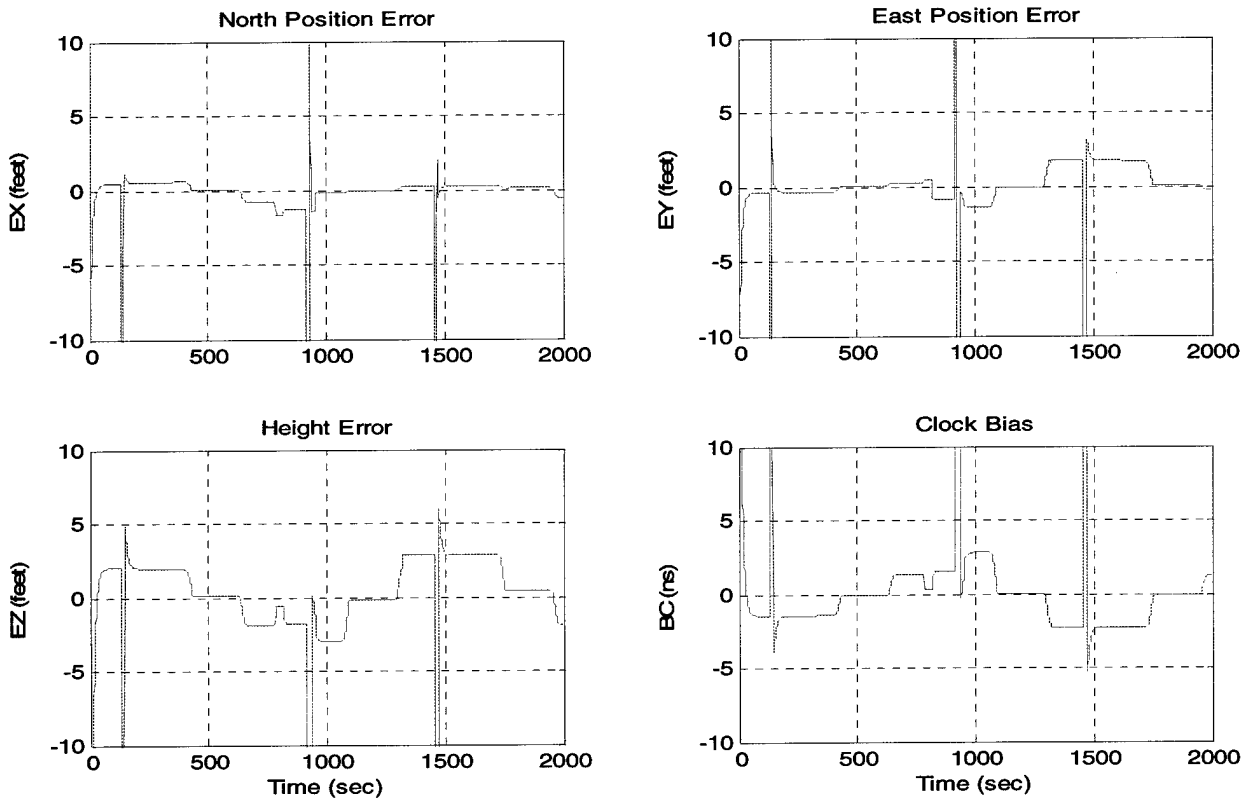


Figure 4.2-14

Analysis of Results

The most significant result of the switch to Open Loop operation is the successful achievement of stable algorithm operation while driving the position and clock bias errors to extremely small levels. The simulation process was allowed to run for 2000 seconds with no sign of stability problems. Figure 4.2-11 (2 Ft position errors) produces excellent results with negligible errors. Increasing position errors to 200 feet per axis (Figure 4.2-12) shows slight increases in error levels, but still well within acceptable levels. The highest acceptable level of position errors is reached at 760 feet per axis (Figure 4.2-13). Increasing the initial position errors by a small amount to 765 feet per axis (Figure 4.2-14) brings out the first signs of solution instability, with periodic spikes in error magnitudes in evidence. The appearance of these sharp, but short lived errors coincides with the change of direction of the participating platforms when they execute their racetrack trajectories. Further attempts to stabilize the solution for large position errors by increasing the initial covariance and process noise levels to 2000 feet were unsuccessful. It thus appears that these large Link-16 position errors represent a practical limit of filter operation. It should be noted that these position errors represent a level far in excess of a stable Link-16 solution, whose maximum errors are in the range of 50-100 feet. Also note that we do not allow the DGPS RF filter to even begin operation until the Link-16 navigation solution has stabilized and converged to their normal small steady state values.

4.2.6 Test Case 1-6 (Open Loop) Effect of Observation Update Rate

The objective of this test case is to assess the effect of increased observation rate to the baseline open loop results discussed within the last paragraph. In addition, we wish to determine if the higher update rates can increase the convergence envelope for the Link-16 navigation errors.

Initial Conditions:

Link-16 Navigation Position Errors (ERRX, ERRY = 2 feet, 760 feet)
Link-16 Velocity Errors (EVN,EVE = 0.0 ft/sec)
Clock Bias Error (50 ns)
Frequency Error (0.0 ns/sec)
Velocity States – Not Present

Graphical Results

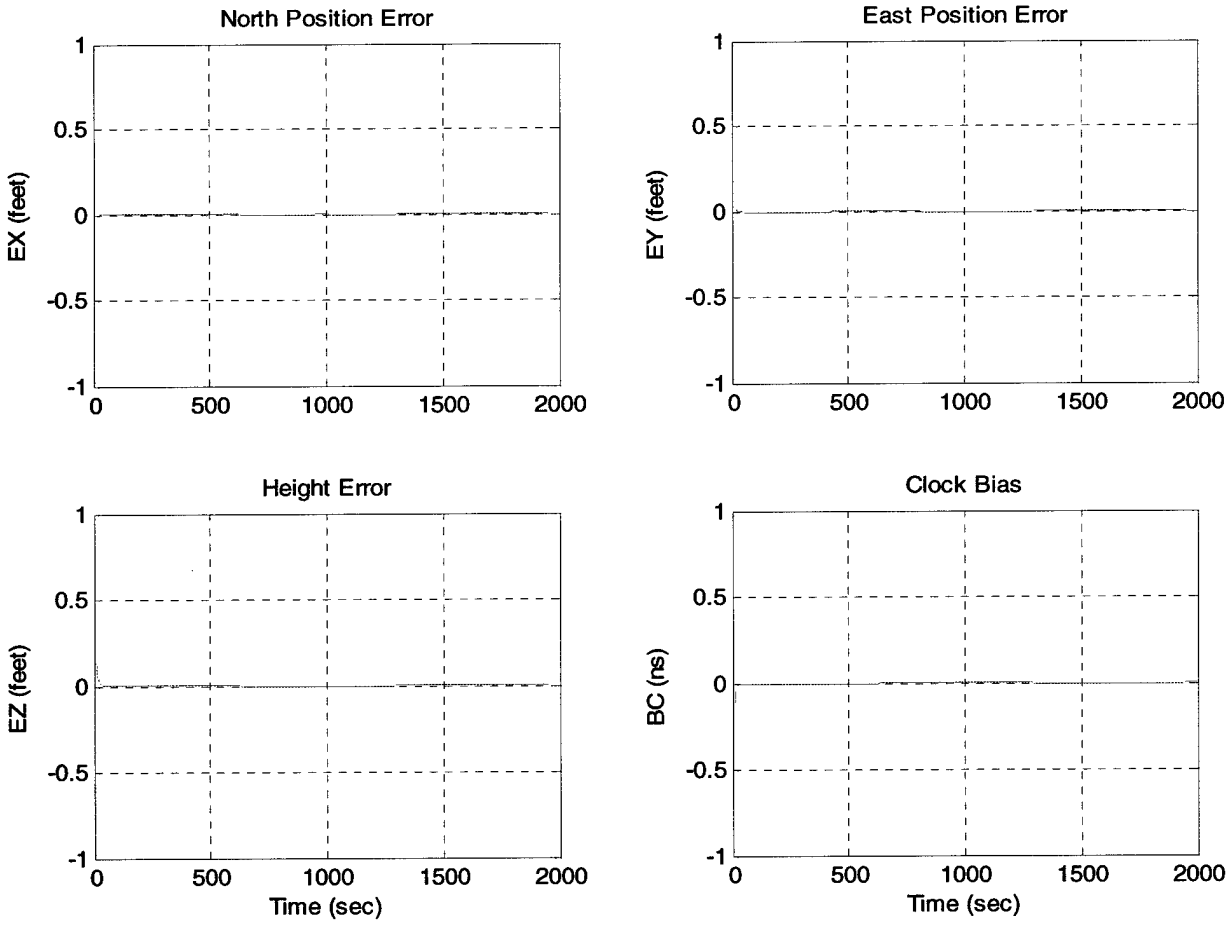


Figure 4.2-15: Open Loop, 2Hz update rate

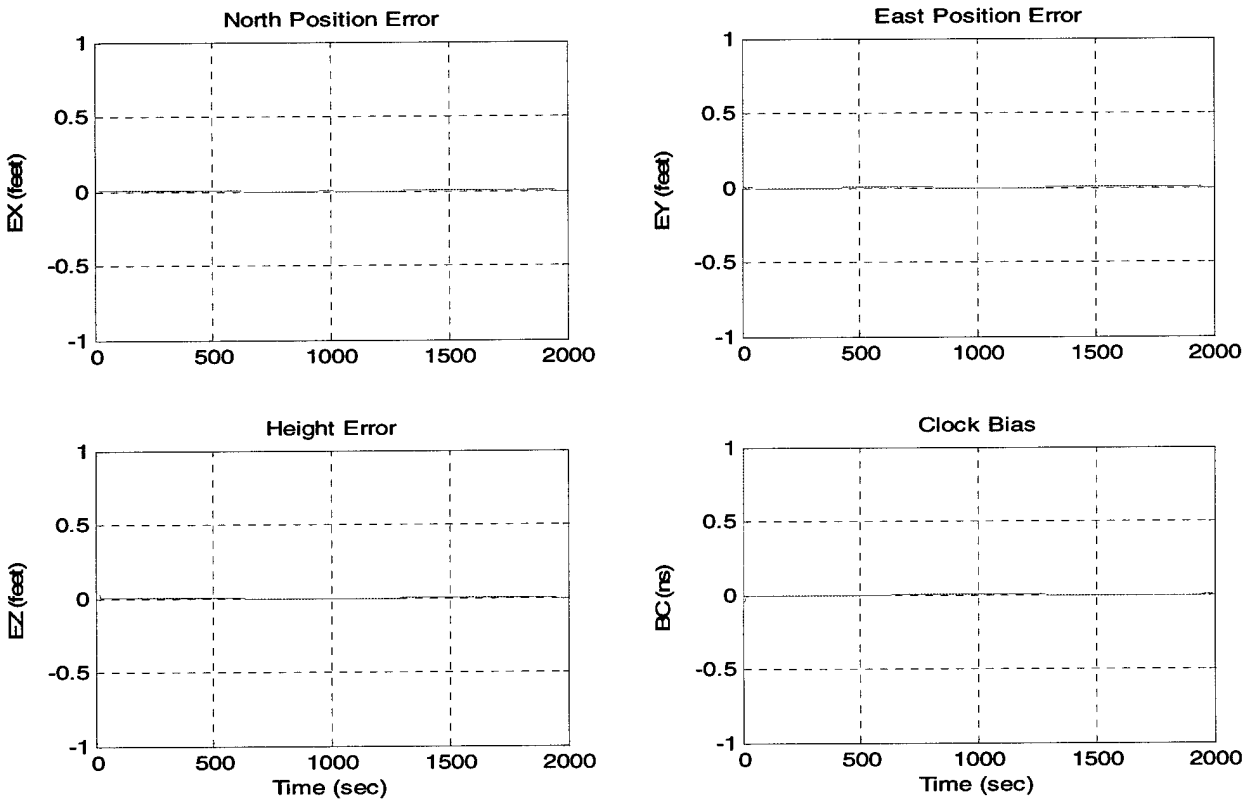


Figure 4.2-16: Open Loop, 4Hz update rate

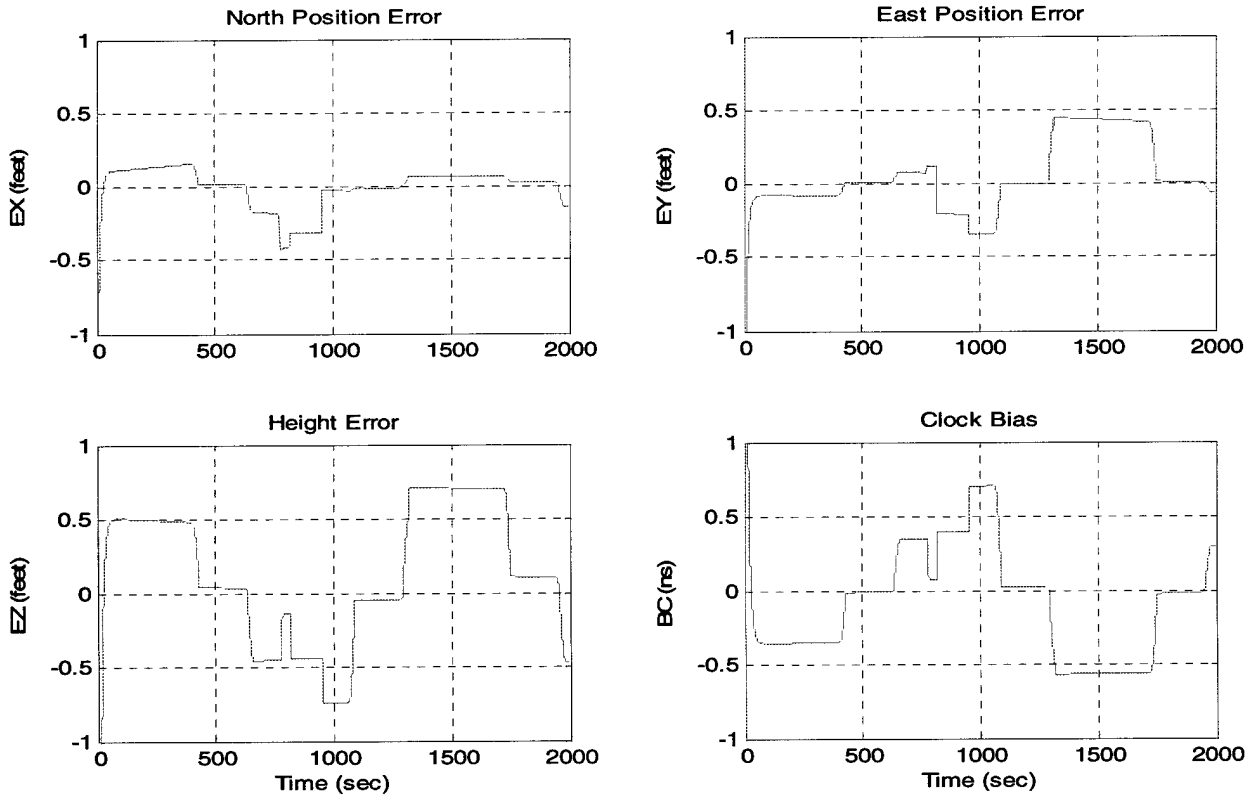


Figure 4.2-17: 4Hz update rate, 760 ft position error

Analysis of Results

Figure 4.2-15 presents the results of operating with a 2 Hz update rate. The results are nearly indistinguishable from those of the previous test case because of the scale. The corresponding result for a 4 Hz update rate is shown in Figure 4.2-16, and drives errors to extremely small values with a convergence time constant of 1-2 seconds or better. Figure 4.2-17 maintains the 4 Hz update rate, but increases the initial position error to the critical level of 760 feet. Once again, this figure proves to be nearly the outer limit of allowable position error for system operation. The errors shown in that figure are just marginal with respect to the performance goals of the algorithm.

4.2.7 Test Case 1-7 (Open Loop) – Effect of Unmodeled Velocity Errors

In prior testing of the effect of velocity errors during closed loop in paragraphs 4.2.3 and 4.2.4, the destabilizing effect of velocity errors was unmistakable. This test case reevaluates the results when open loop operation is used. The algorithm performance under open loop conditions indicates a remarkable improvement as compared with prior closed loop operation.

Initial Conditions

Link-16 Navigation Position Errors (ERRX, ERRY = 730 ft)
Link-16 Velocity Errors (EVE, EVN = 1.0 ft/sec)
Clock Bias (50 ns)
Frequency Error (0.0 ns/sec.)
Velocity States – Both Included and not included

Graphical Results

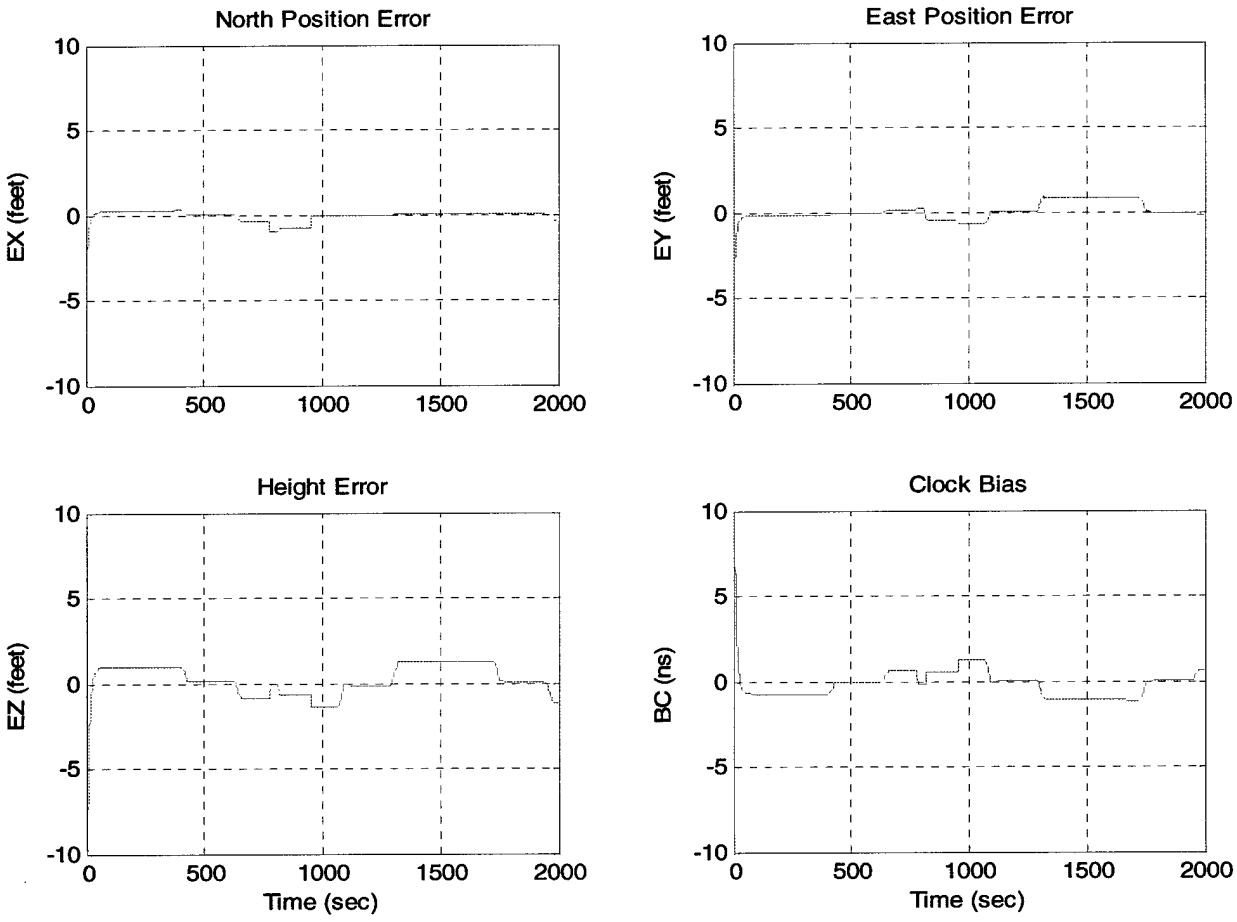


Figure 4.2-18: Position errors = 730 ft, 2Hz update rate

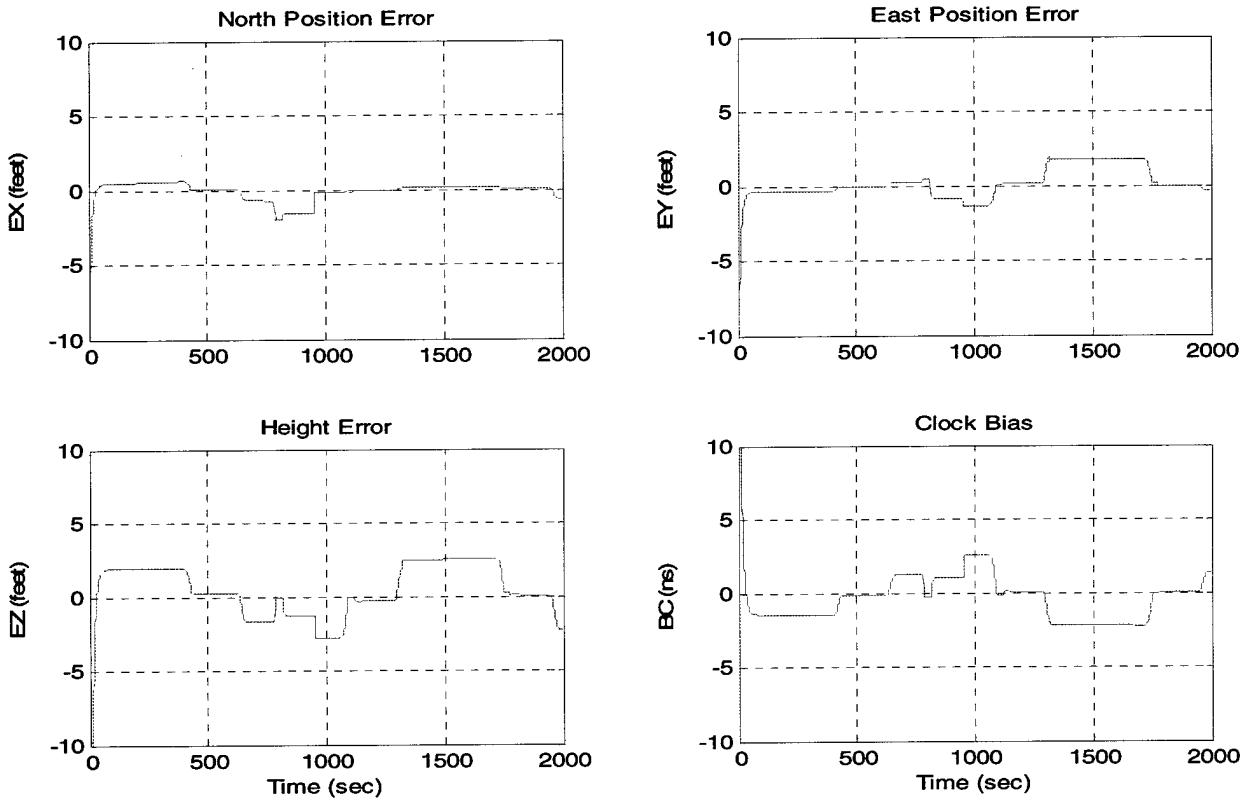


Figure 4.2-19: Position errors = 730 ft, 1Hz update rate

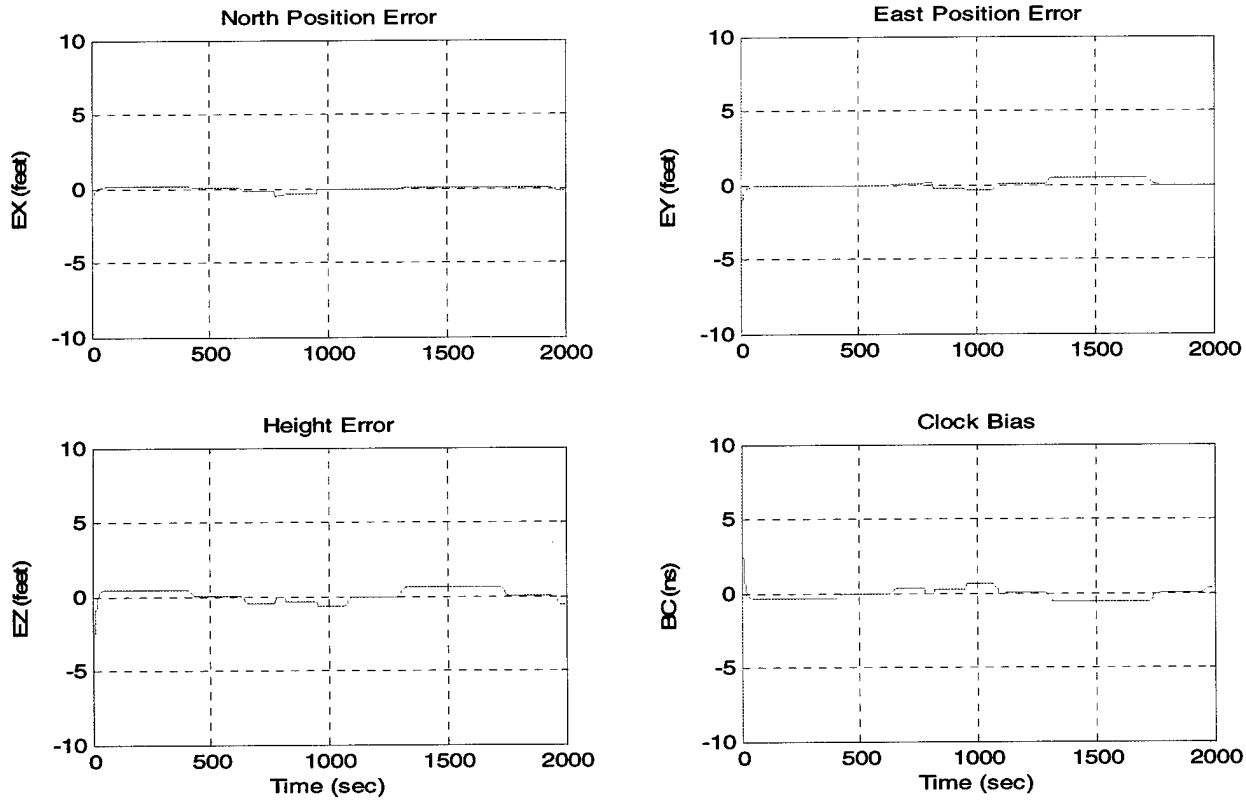


Figure 4.2-20: Position errors = 730 ft, 4Hz update rate

Analysis of Results

The most significant results of these runs is that, in open loop operation, the presence or absence of the velocity states has no measurable effect on algorithm performance. Stable reliable operation is achieved in either case with position errors as large as 730 feet per axis, and with update rates of 1 Hz, 2 Hz, and 4 Hz. Figure 4.2-18 is run at a 2Hz update rate with position errors of 730 feet per axis. Figure 4.2-19 is run at an update rate of 1 Hz and 730 feet per axis, and is only slightly worse than the previous run, but still within allowable limits. Finally, Figure 4.2-20 is run at 4 Hz, and presents the smallest errors of all three runs.

4.2.8 Test Case 1-8 (Open Loop) Effect of Clock Bias and Frequency Drift

In this test case, we concentrate on the performance of the clock bias and frequency estimation in open loop mode. In previous runs, we operated with a constant clock bias error of 50 ns and zero frequency drift. Here, we increase the clock bias initial error to 200 ns, with a frequency drift of 10 ns/second. We emphasize that these error values are much larger than would be expected from the normal operation of a GPS receiver. However, the ability to rapidly and effectively calibrate these errors has great value if the DGPS community is called upon to function as a time transfer agent, using external time references to drive the GPS receiver element.

Initial Conditions

Link-16 Position Errors – (ERRX, ERRY = 200 feet)
Link-16 Velocity Errors – (EVN,EVE = 1 ft/Sec.)
Clock Bias Error – (200 ns)
Frequency Error – (10 ns/sec)
Observation Rate: 1 Hz

Graphical Results

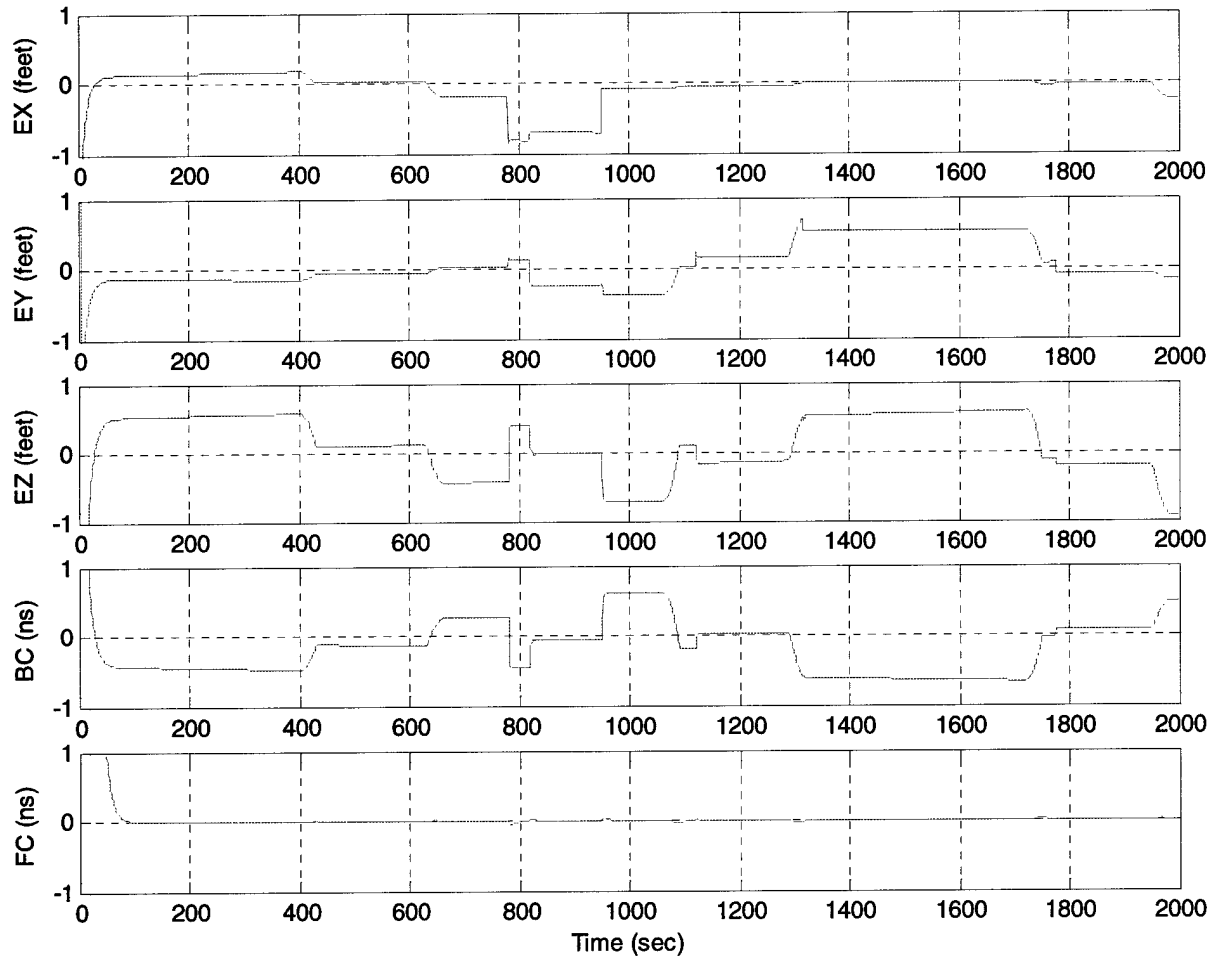


Figure 4.2-21: Clock Bias / Frequency Error Estimation

Analysis of Results

As noted in Figure 4.2-21, the algorithm is demonstrated to operate effectively and stably even with large offsets in both time and frequency. Excellent performance is noted for all three position terms as well as clock bias and frequency drift. Frequency drift, in particular is observed to reflect the same time constant as all other states when 1 Hz observation rate is used. In other runs, not shown here, the same conclusion is reached when update rates are increased to 2 Hz and 4 Hz. In the latter case, frequency drift is estimated within 1-2 seconds of filter operation. These results are unaffected whether or not the velocity states are included within the model.

4.2.9 Test Case 1-9 (Open Loop) Effect of Process Noise on Solution

The quantity of measurement information available from the multiple satellites tends to drive down the level of the RF covariance matrix. The purpose of this test case is to optimize the level of process noise to be applied to the RF Kalman Filter position states.

Initial Conditions

Link-16 Navigation Position Errors (ERRX,ERRY = 2 feet)
Velocity Errors (EVN,EVE = 0.0 ft/sec)
Clock Bias Error (50 ns)
Frequency Error (0.0 ns/sec)
Measurement Rate (4 Hz)

Graphical Results

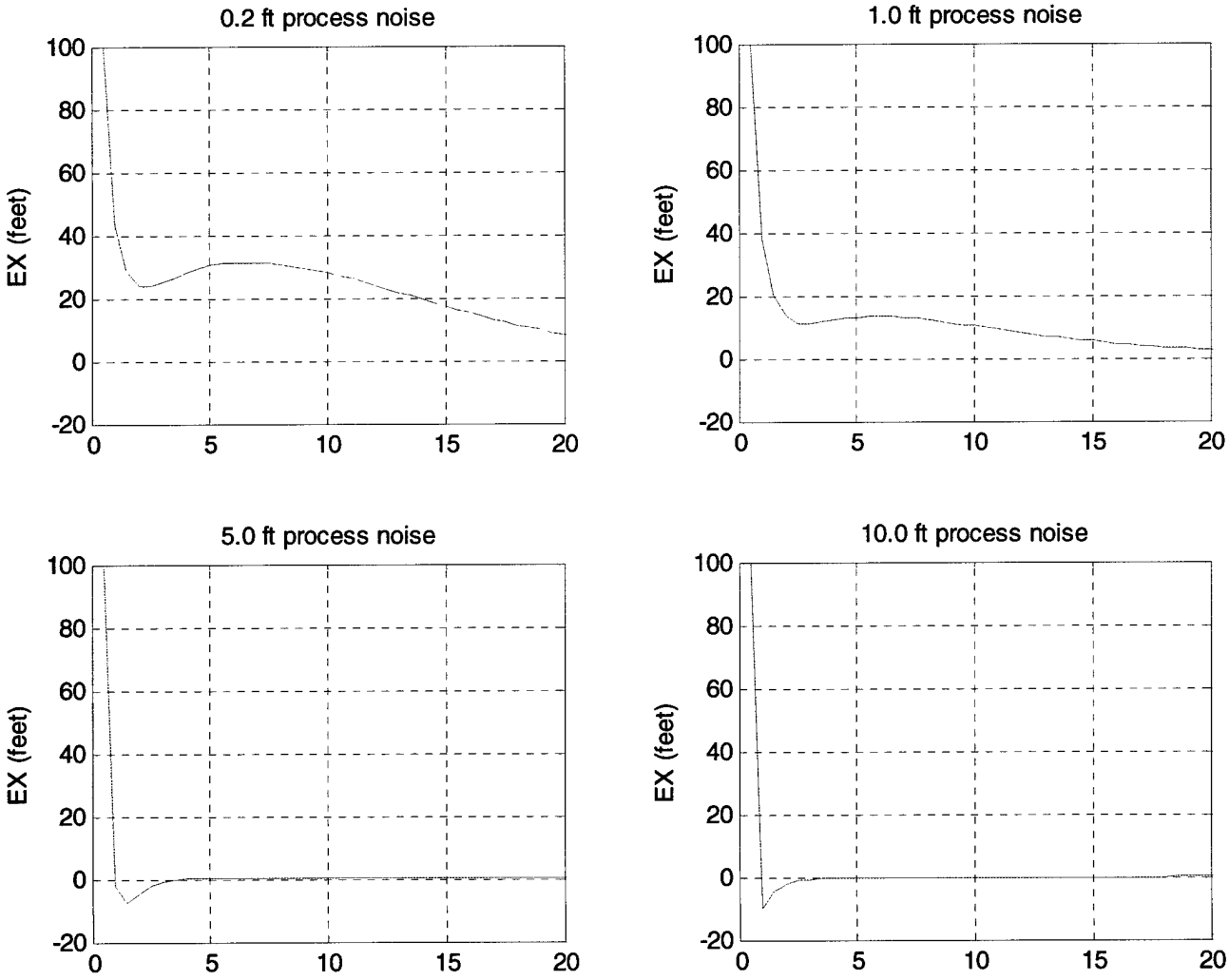


Figure 4.2-22: Process Noise Tradeoffs

Analysis of Results

The results summarized in Figure 4.2- 22 provide clear evidence of the need for high levels of process noise to be applied to the RF Kalman Filter. The position estimation performance when process noise of 10.0 feet per axis is noticeably better than that at higher values. This figure appears to be the lowest level of process noise that provides good performance. Addition of still higher values provides no additional benefit to position estimation performance.

4.2.10 Test Bed 1 Operation – Summary and Conclusions

The eight preceding test cases summarize the simulation activity and results obtained using Test Bed 1. The primary goal of this simple but versatile tool was to provide a capability for isolating the effect on algorithm performance of individual error terms under carefully controlled conditions. The most significant conclusion of this portion of the study is that the closed loop operation always results in unstable operation, even though the instability may take a long time of operation to make itself evident. Another important conclusion is that the use of the three velocity states is ineffective, causing oscillatory behavior in closed loop operation, and not preventing divergence in any application. In open loop operation, the velocity states have virtually no effect at all, for better or for worse. Yet another important conclusion is that an error limit of about 760 feet/axis has been identified for the magnitude of the Link-16 navigation error. The bad news is that this limit exists at all; the good news is that it far exceeds normal operation of Link-16, and should not present any operational issues.

The open loop operation of the algorithm has proven to produce stable and reliable operation that is virtually insensitive to the magnitude of errors as long as the initial covariance and process noise is sufficient to cover these errors. (Exception – L16 position error, as already noted). The basic health of the algorithm depends on maintaining the level of the covariance via process noise modeling, because of the excess of information provided by up to 10-11 satellites per observation cycle. We do not wish to allow the covariance matrix to be anything other than positive definite.

The promising performance of the DGPS algorithm is welcome and reassuring, but we note that the error profiles provided by the Link-16 navigation in Test Bed 1 were simplistic and not necessarily representative of the actual navigation and bias errors experienced with a real Link-16 community. The purpose of Test Bed 2, built around the reliable Link-16 Navigation Simulator, is to provide a realistic navigation environment for the DGPS algorithm. That study is presented in the following paragraph, 4.3.

4.3 Test Bed 2 Simulation Activities

Overview and Rationale

The analysis and simulation described in paragraph 4.2 using Test Bed 1 was designed to determine the fundamental performance limitations and sensitivities of the DGPS RF algorithm under carefully controlled conditions. It was well understood that the test conditions, such as the Link-16 constant error profile were, to some extent, artificial and did not truly reflect the actual error dynamics of a realistic Link-16 mobile community. These dynamics can only be represented by replacing the covariance approximation of the COV50 software with the **Link-16 Navigation Simulator (LNS)** as a driver of the DGPS RF algorithm.

The LNS is a proprietary computer program developed by BAE SYSTEMS which embodies the actual operational navigation code executing on each and every Link-16 terminal. Under development for nearly thirty years, the LNS has become a trusted surrogate which accurately represents Link-16 navigation performance under all operational conditions. The combination of the LNS and the DGPS RF source code plus a circular GPS satellite routine constitutes Test Bed 2, which will be used in all further simulations presented within this study. The LNS permits the navigation design engineer to analyze both community and individual platform performance in the areas of Navigation, Time Synchronization, and (in special versions) Sensor Registration. The LNS is written in VAX Fortran 77, and comprises 45 source code modules which are always configured **to replicate exactly** the terminal operational code. (Figure 4.3 – 1) Thus, no approximations or assumptions need be made in interpreting navigation performance; it faithfully represents the performance that a real terminal would deliver under simulation conditions and trajectories. A full description of the structure and organization of the LNS is presented in Appendix B.

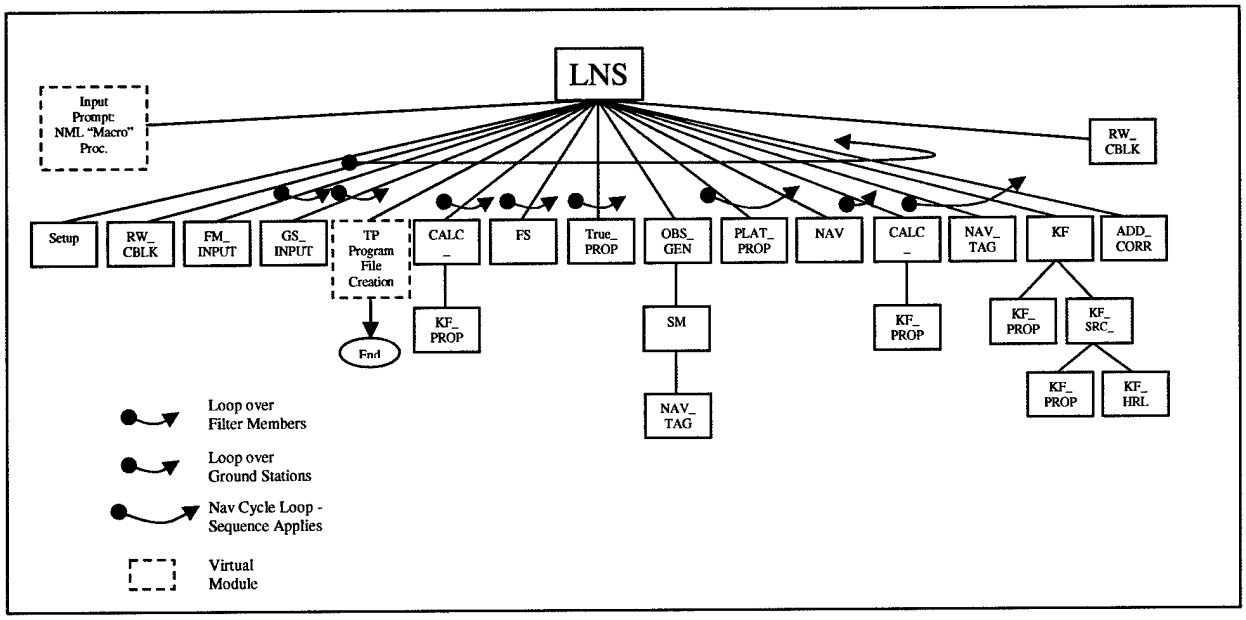


Figure 4.3-1: LNS Architecture

Based upon the studies documented using Test Bed 1, all runs executed herein will only operate using the **Open Loop** mode, since closed loop operation has been determined to be unstable. A firm ground rule has been established for these simulations, as follows. The DGPS RF algorithm will not be permitted to begin its execution unless and until the Link-16 navigation performance, as constructed by the LNS, has been allowed to reach steady state operation. This rule will be adapted for any further laboratory and /or flight test activity of the DGPS RF algorithm. The quality of the Link-16 navigation performance will be varied considerably during the Test Bed 2 simulations, always keeping in mind that earlier Test Bed 1 simulation has uncovered regions within the space of position errors which may lead to instability. Even in the test cases where larger than normal Link-16 position and velocity errors are imposed, these errors will never be larger than 50% of the limiting position errors (760 ft/axis)

Community Navigation Trajectory and Error Budgets

The trajectories of the two participating platforms are as they were presented in Figure 4.2-3. The nature of the Link-16 navigation errors of the two platforms are functions of many variables, including the error budgets of their included inertial platforms, the presence or absence of GPS PVT processing, and the PPLI transmission rate between master and slave platforms. Table 4.3-1, below, lists the critical navigation error budgets of the participants, as well as the DGPS RF Kalman Filter tuning constants.

The navigation error budgets of the mobile members are characteristic of the high quality inertial navigation units typically installed on modern military fighter aircraft. In the case of Link-16, terminals are designed to accept position, and time (PVT) updates from installed GPS units, if available. Position updates are usually scaled to represent 10 meters CEP position quality, while time updates are assumed to be accurate to 50 ns RMS error. These error budgets ,assigned to the mobile members shall be considered the **Nominal** navigation error budgets.

The DGPS RF Kalman Filter error parameters are also listed in Table 4.3-1, and represent the Nominal values assigned to this algorithm. In the simulation studies listed herein, these nominal values for both navigation and DGPS RF shall be used, except for special cases, as noted, where study requirements require variations in these terms.

VARIABLE	SYMBOL	NOMINAL VALUE
Initial platform and NAV position errors		
North error West error Alt error	EN_INIT EW_INIT EZ_INIT	1500.0 [ft] 1500.0 [ft] 0.0 [ft]
Initial platform and NAV velocity errors		
Vx error Vy error Vz error	EVX_INIT EVY_INIT EVZ_INIT	1.0 [ft/sec] 1.0 [ft/sec] 0.0 [ft/sec]
Initial tilt errors = true platform tilts	ETX_INIT ETY_INIT ETZ_INIT	4.848e-3 [deg] 4.848e-3 [deg] -1.907e-1 [deg]
Accelerometer bias	BX_A BY_A BZ_A	29.0 [ug] 11.0 [ug] 26.0 [ug]
Accelerometer scale factor	SFX_A SFY_A SFZ_A	2.9e-2 [percent] 3.6e-2 [percent] 2.2e-2 [percent]
Accelerometer non-orthogonalities	XZ_A ZY_A ZX_A	61.0 [arcsec] 23.0 [arcsec] -40.0 [arcsec]
Gyro "bias" (constant portion of drift rate)	BX_G BY_G	1.5e-3 [deg/hr] 6.8e-3 [deg/hr]

	BZ_G	1.3e-3 [deg/hr]
Gyro scale factor	SFX_G SFY_G SFZ_G	-4.1e-2 [percent] 3.2e-2 [percent] 4.4e-2 [percent]
Gyro non-orthogonalities	XZ_G XY_G YZ_G YX_G ZY_G ZX_G	32.0 [arcsec] 19.0 [arcsec] 17.0 [arcsec] -71.0 [arcsec] -10.0 [arcsec] 63.0 [arcsec]
Gryo mass unbalance	MUX_G MUY_G MUZ_G	-0.023 [deg/hr/g] -0.076 [deg/hr/g] 0.023 [deg/hr/g]
Gyro anisoelasticity	ANX_G ANY_G ANZ_G	0.004 [deg/hr/g**2] 0.002 [deg/hr/g**2] 0.001 [deg/hr/g**2]
Initial values for geodetic state covariances	GEO_POS_UNC ALT_BIAS_UNC AZIM_UNC VEL_UNC TILT_UNC ALT_SF_UNC	1006.0 [feet] 405.6 [feet] 0.5 [degrees] 7.5 [ft/sec] 0.0417 [degrees] 3.0 [percent]
Initial values for geodetic state process noise values. (NICP standard values)		
(NICP =0.12 [ft**2/sec]) (NICP =0.25 [ft**2/sec]) (NICP =1.D-4)	GEO_POS_PN ALT_BIAS_PN ALT_SF_PN	0.12 [ft**2/sec] 0.25 [ft**2/sec] 1.D-4
F/A-18-RLG Values		
(NICP = 2.597D-13 [rad**2/sec]) (NICP = 1.648D-4 [ft**2/sec**3]) (NICP = 2.467D-13 [rad**2/sec])	AZIM_PN VEL_PN TILT_PN	8.965D-14 [rad**2/sec] 7.914D-5 [ft**2/sec**3] 8.954D-14 [rad**2/sec]
PPLI measurement noise value	PPLI_MEAS_NOISE	400 [ns**2]
Covariance Matrix		
North Position Covariance West Position Covariance Altitude Covariance Clock Bias Covariance Freq. Drift Covariance	PA(1,1) PA(3,3) PA(5,5) PA(7,7) PA(8,8)	1000000.0 1000000.0 1000000.0 10000.0 100.0
Process Noise Covariance Matrix		

North Position Process Noise West Position Process Noise Altitude Process Noise Clock Bias Process Noise Freq. Drift Process Noise	QA(1,1) QA(3,3) QA(5,5) QA(7,7) QA(8,8)	1000000.0 1000000.0 1000000.0 100.0 1.0
Observation Noise Covariance Matrix		
	RA(1,1) RA(2,2) RA(I,I) for I=3,12	441.0 156.25 16.0

Table 4.3-1: Navigation Error Budgets

Preview of Test Bed 2 Simulation Results

The DGPS RF algorithm architecture and tuning constants developed in paragraph 4.2 were evaluated in Test Bed 2 with the full Link-16 community navigation error model provided by the LNS code. A full range of parameter variations were applied, including observation update rate, number of processed GPS satellites, and quality of the Link-16 navigation solution. In all cases, the DGPS algorithm provided a robust, and accurate estimate of position and clock errors for the mobile community. Highlights of the most significant results are summarized below:

(a) Good to excellent performance was produced using variable observation update rate covering the range 4 Hz, 2Hz, 1Hz, 0.25Hz. Although some dilution of performance is noted for the lowest of these rates, the application of low pass filtering of the error terms shows beneficial results.

(b) Sensitivity of the DGPS estimation error with respect to raw Link-16 navigation errors appears to be extremely low. Use of degraded INS navigation models causes large increases in raw platform navigation error, but minimal effect on the performance of the DGPS RF algorithm.

(c) In the event of partial jamming of the GPS constellation, the performance level of the DGPS algorithm could be significantly affected. Simulation shows virtually no degradation of performance with as few as six visible satellites, although total collapse of the solution is noted when the number of processed satellites is reduced to four.

(d) The Link-16 Kalman Filter position update processing causes discontinuities in raw position error, resulting in much smaller discontinuities in the DGPS RF error results. The DGPS RF discontinuities coincide with Link-16 position updates, but disappear within one observation cycle thereafter. Even these small variations can be reduced by introduction of simple low pass filtering on the DGPS results

Full discussion of all of the above conclusions is provided in the analyses of each of the test cases which constituted this portion of the study.

4.3.1 Test Case (2-1) Performance Sensitivity to Observation Update Rate

This series of simulations is designed to evaluate DGPS RF performance against the realistic LNS Link-16 navigation error model, while varying the basic pseudorange update rate over the range 4 Hz, 2 Hz, 1Hz, and 0.25Hz. Unlike the simulations performed within the Test Bed 1 software where navigation position and velocity errors were **constant**, the position and velocity errors of the two platform configuration present a **dynamic** variation. This dynamic error profile is the resultant of the inertial error budgets of the two platforms, their trajectories, and the measurements processed by the Link-16 Navigation Kalman filter. The observation processing utilized for Link-16 navigation is precisely that utilized in normal terminal operation, that is PPLI measurements transmitted from master to slave at a 12 second rate, plus GPS PVT measurements incorporated at an identical 12 second rate. No feedback or adjustment of this hybrid inertial solution from the DGPS RF is implemented. The Link-16 navigation algorithm is completely untouched.

In Equation (4), we defined the DGPS RF observation matrix H to accept PPLI measurements, as well as GPS pseudorange measurements. These measurements, at a rate of one per 12 seconds were incorporated in the simulations performed as part of Test Bed 1. No attempt was made in that study to evaluate the effectiveness of these observations, as compared with the GPS pseudorange measurements. That evaluation was conducted as part of Test Bed 2 operation, however. The result of that analysis was the unsurprising conclusion that when measured against the information content of 8-12 satellite pseudoranges, the PPLI range measurements had **no measurable effect** on the RF solution. The results presented in this section of the report therefore deleted the use of PPLI range measurements by the DGPS RF Kalman Filter, and relied exclusively on pseudorange measurements. There are conceivable circumstances, such as partial jamming of the GPS satellite constellation, where it may pay to revisit the use of these range measurements. This point is further discussed in paragraph 4.3.X.

Test Conditions

Link-16 Navigation Error Budget:/ Measurement Schedule - Nominal
DGPS Configuration – Nominal
Psuedorange Update Rates: 4Hz, 2Hz, 1Hz, 0.25Hz

Graphical Results

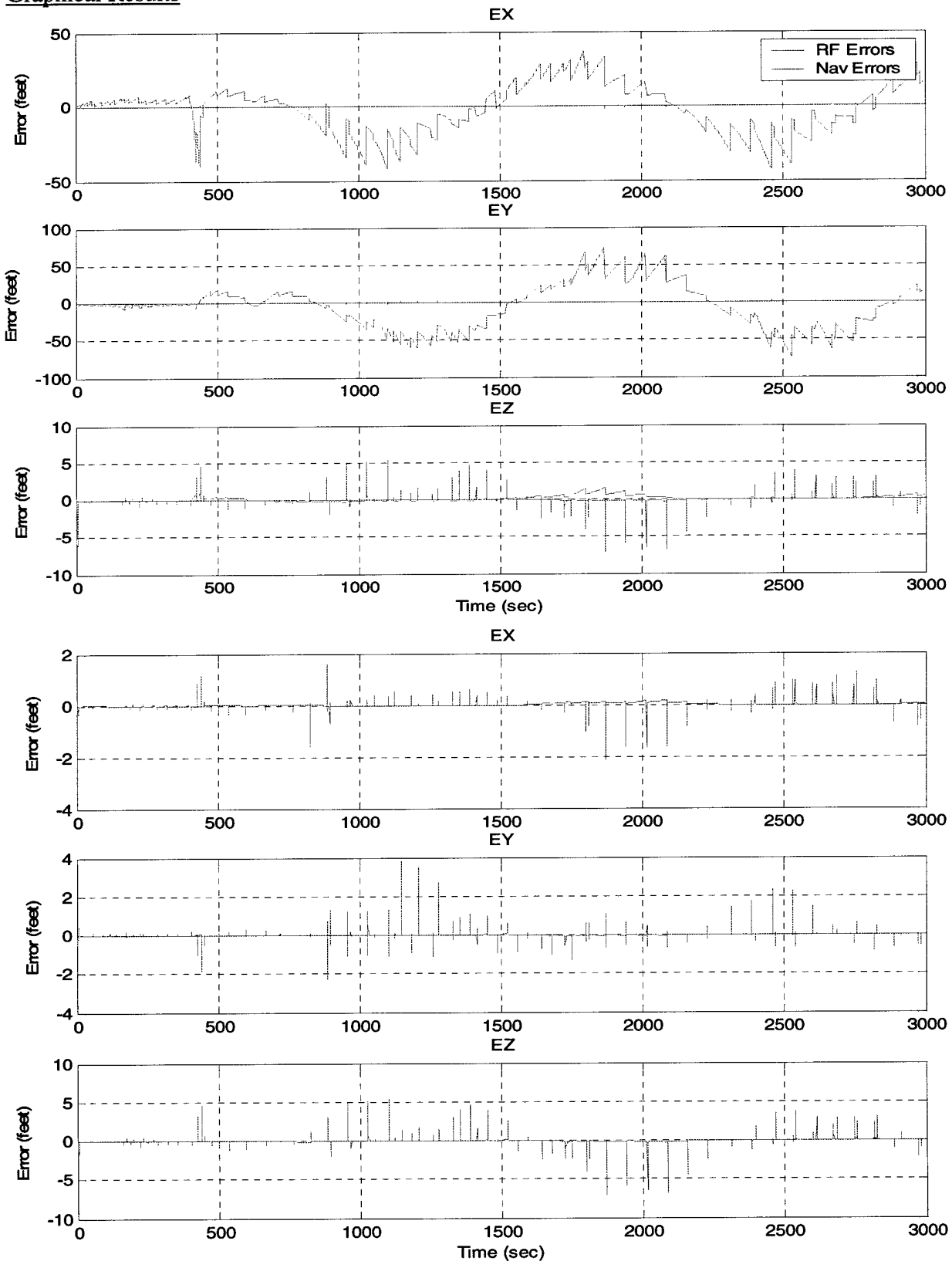


Figure 4.3-2: Test Bed 2 DGPS Performance – 4Hz update rate

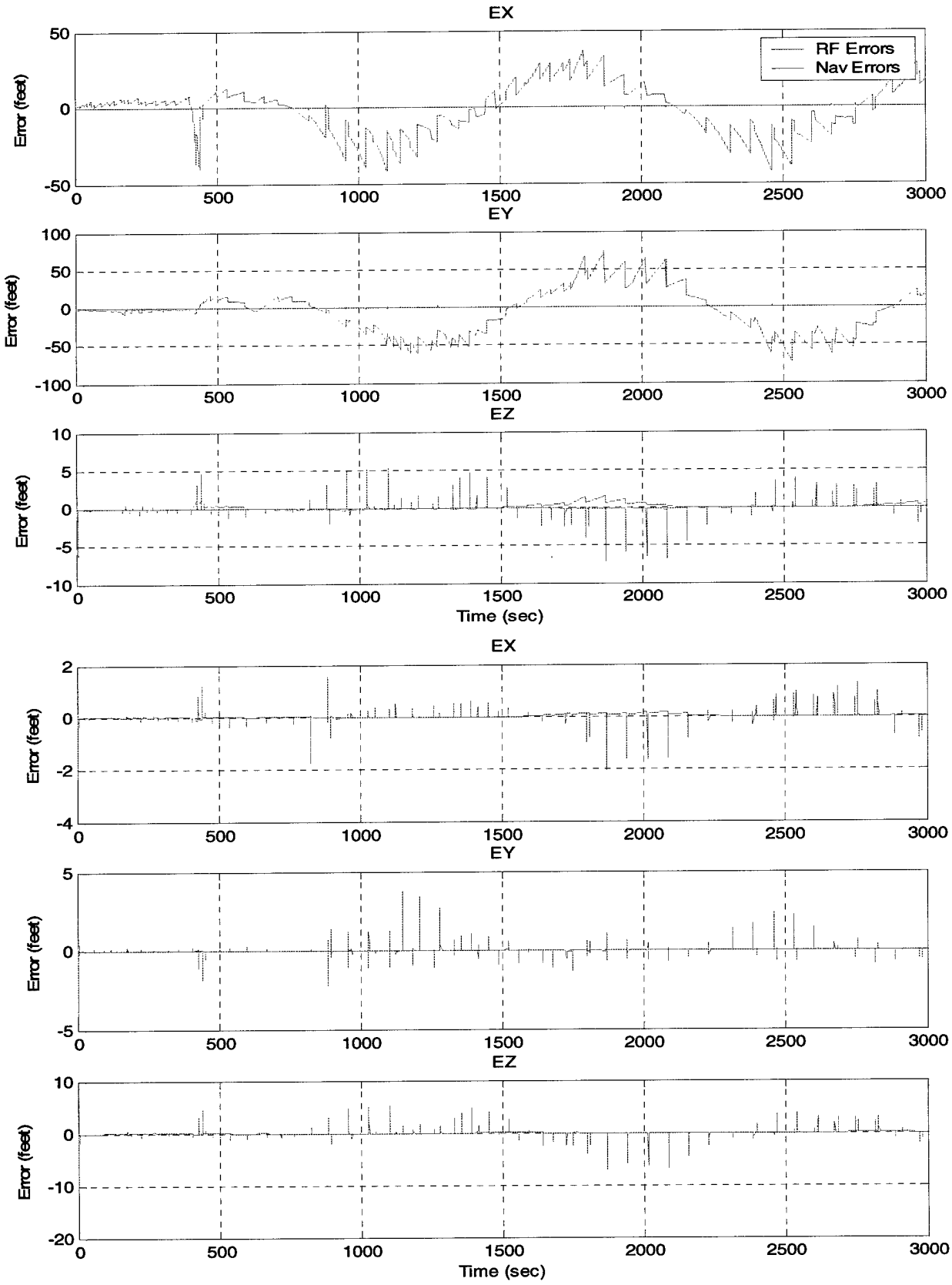


Figure 4.3-3: 2Hz update rate

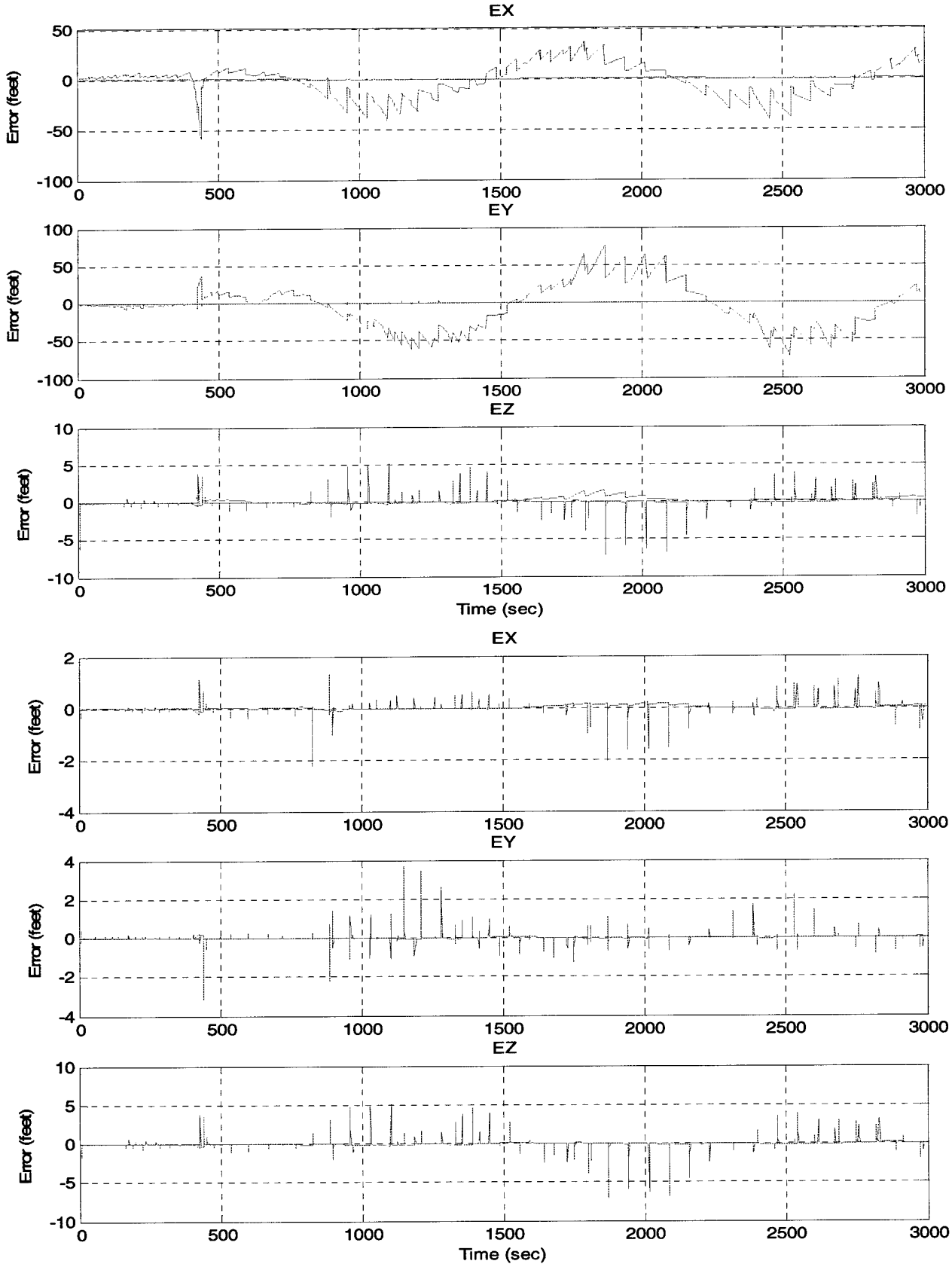


Figure 4.3-4: 1Hz update rate

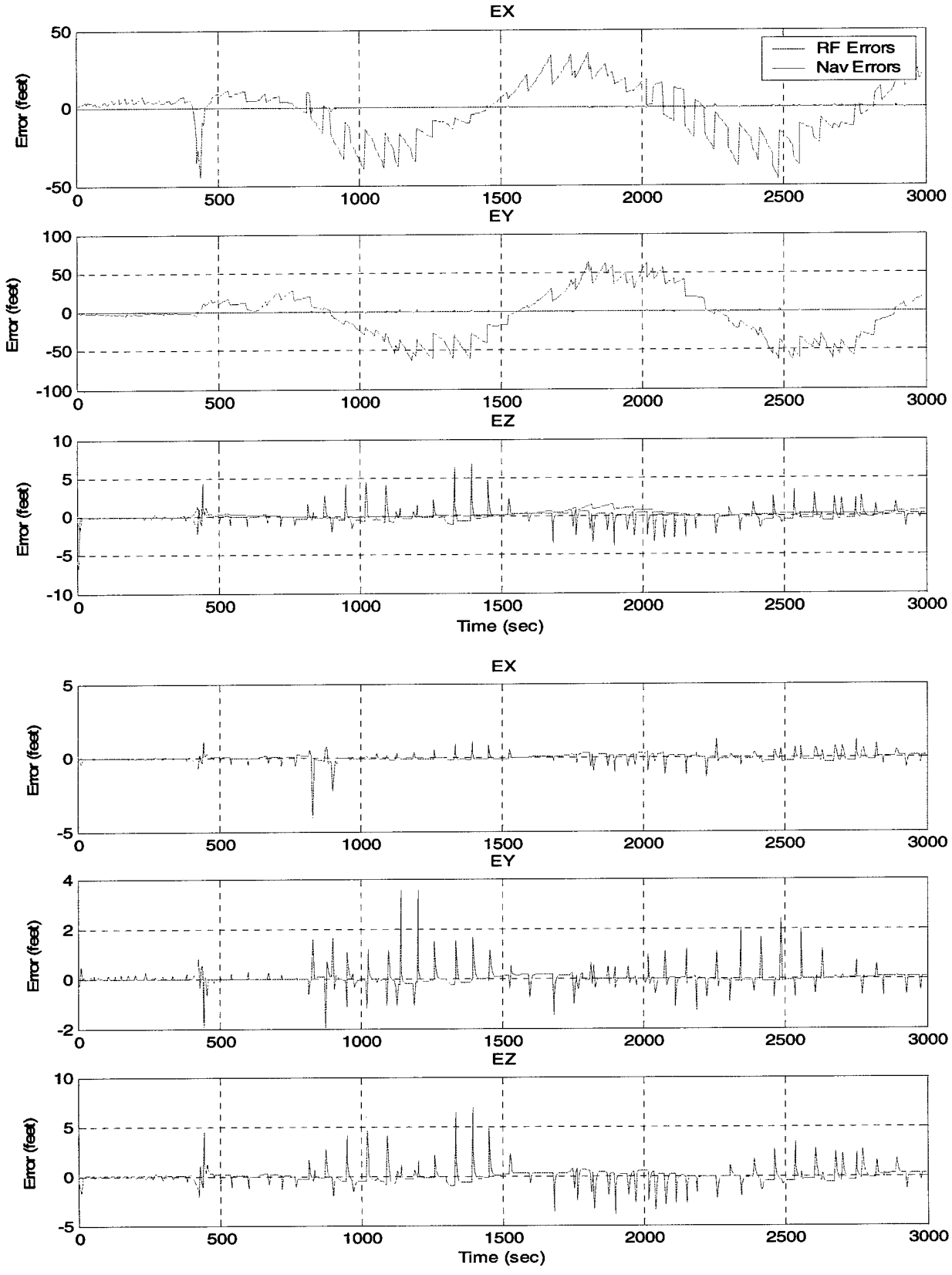


Figure 4.3-5: 0.25Hz update rate

Analysis of Results

Figure 4.3-2 presents the RF position errors developed during a test for which the pseudorange observation update rate is 4 Hz. In this figure (upper three curves), the relative navigation position error of the two mobile members using nominal error budgets is displayed in red. The trend of this position error is nearly sinusoidal in nature, with an average error per horizontal axis of approximately 40-50 feet. The jagged breaks in the position error terms are caused by the GPS 12 second measurement updates, which create step changes in these position error terms. The corresponding DGPS RF errors in estimating these relative navigation error terms appear in blue, and appear "spiky" in nature. Note that the appearance of these spikes coincide with step changes in the relative position error of the navigation solution. The spike duration is no more than the 0.25 second DGPS RF Kalman Filter cycle time, at which point, each error collapses nearly to zero. The sense (positive or negative) of each spike is of the opposite sign to that of the corresponding navigation error. Both the appearance and the characteristics of the spikes are explained by the fact that there is no direct connection between the L16 navigation position updates and that of the DGPS filter. Any step change in the former must, by definition, cause a step change in the latter error source. Please note that the short duration of these spikes is caused by the fact that multiple satellite pseudorange measurements become available to the DGPS filter immediately after the step change occurs, and these spikes are nearly perfectly negated by further measurements. Thus, there is virtually no "steady state" DGPS error, which averages near zero over time with the exception of those instants where the reference solution changes sharply. This fact is borne out by the lower three graphs, which rescale the DGPS errors to make them easier to see, A little consideration indicates that the "true" relative position error between the mobile members, as opposed to that estimated by the Navigation Kalman Filter, must be a smoothly varying function of time. Since the user of this precision navigation would prefer a smoothly varying function, we will later process the DGPS output data through a low pass filter to smooth the spiky error profile and more closely reflect the true error characteristics. Please refer to Paragraph 4.3.2 for details of this computation and results.

In Figure 4.3-3, we decrease the pseudorange observation rate to 2 Hz, and interpret the results exactly as for Figure 4.3-2. A comparison of the 2 Hz results with the 4 Hz results shows virtually no difference between the two charts. The 1 Hz results, given in Figure 4.3-4 still show little change from the 4Hz case. It is only when we decrease the update rate to one per four seconds (Figure 4.3-5), that a slight degradation of results appears. In general, for all four simulations, it is noted that the vertical channel, eZ, produces errors of a larger magnitude then those of the horizontal axes. This can be explained by the fact that most of the satellites used are relatively low on the horizon, and thus provide less information along the vertical axis.

Because of the time scale of these results, the transient performance of the DGPS is not visible on any of the four graphs, which illustrate steady state performance only. Since there is essentially no steady state performance advantage between the observation rates of 1, 2, and 4 Hz, we would presumably seek the lowest rate for operational use of this technology provided there were no pressing reason to insist on filter convergence within a few seconds. Even for the 1 Hz rate, this convergence occurs within no more than 50-60 seconds. We shall further discuss the optimal choice of DGPS Kalman Filter cycle time in paragraph 5.0.

4.3.2 Test Case (2-2) Performance Sensitivity to Low Pass Filtering of DGPS RF Error Terms

In the previous paragraph, the discontinuities of the Link-16 hybrid navigation solution were noted to produce spiky discontinuities of the DGPS estimation error terms. We now investigate the effect of applying a low pass first order digital filter to these error terms in an attempt to smooth out these discontinuities and more accurately represent the steady state performance of the filter. In this series of tests, we repeat the runs performed in 4.3.1 with Kalman Filter cycles of 4, 2, and 1 Hz with a low pass filter time constant of 1 second.

Test Conditions

Link-16 Navigation Error Budget:/ Measurement Schedule - Nominal
DGPS Configuration – Nominal
Psuedorange Update Rates: 4Hz, 2Hz, 1Hz

Graphical Results

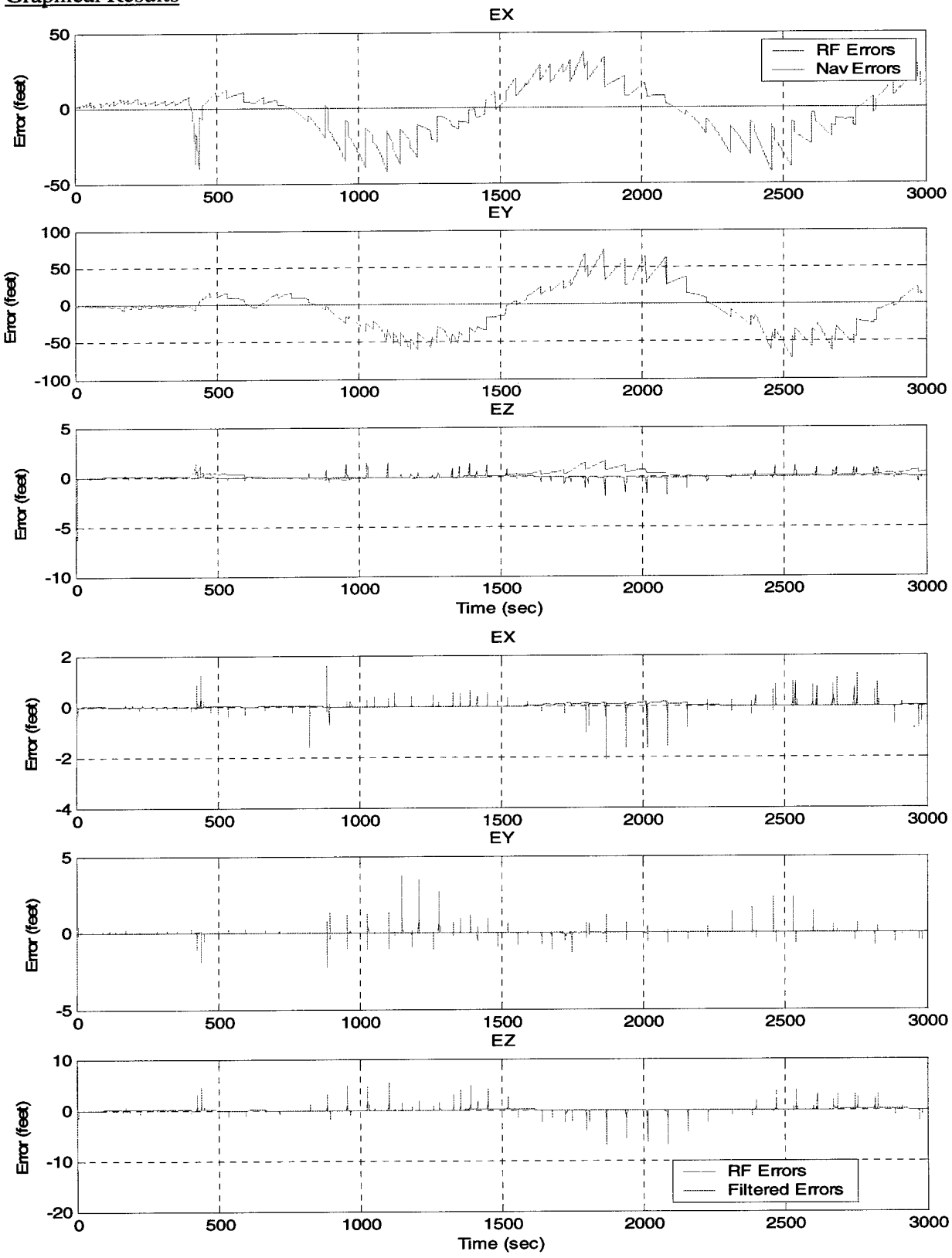


Figure 4.3-6: Low pass filtering – 4Hz

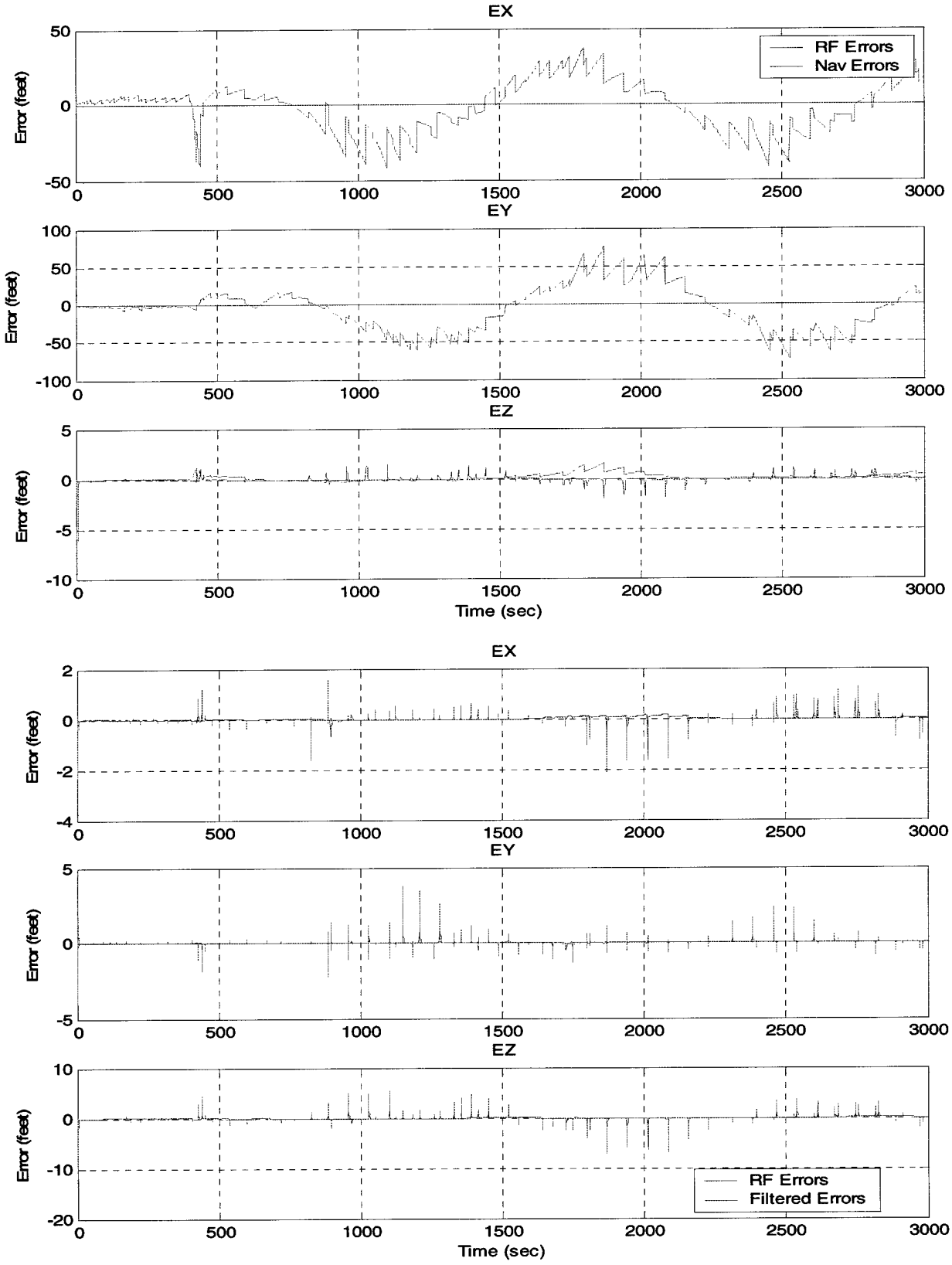


Figure 4.3-7: Low pass filtering – 2Hz

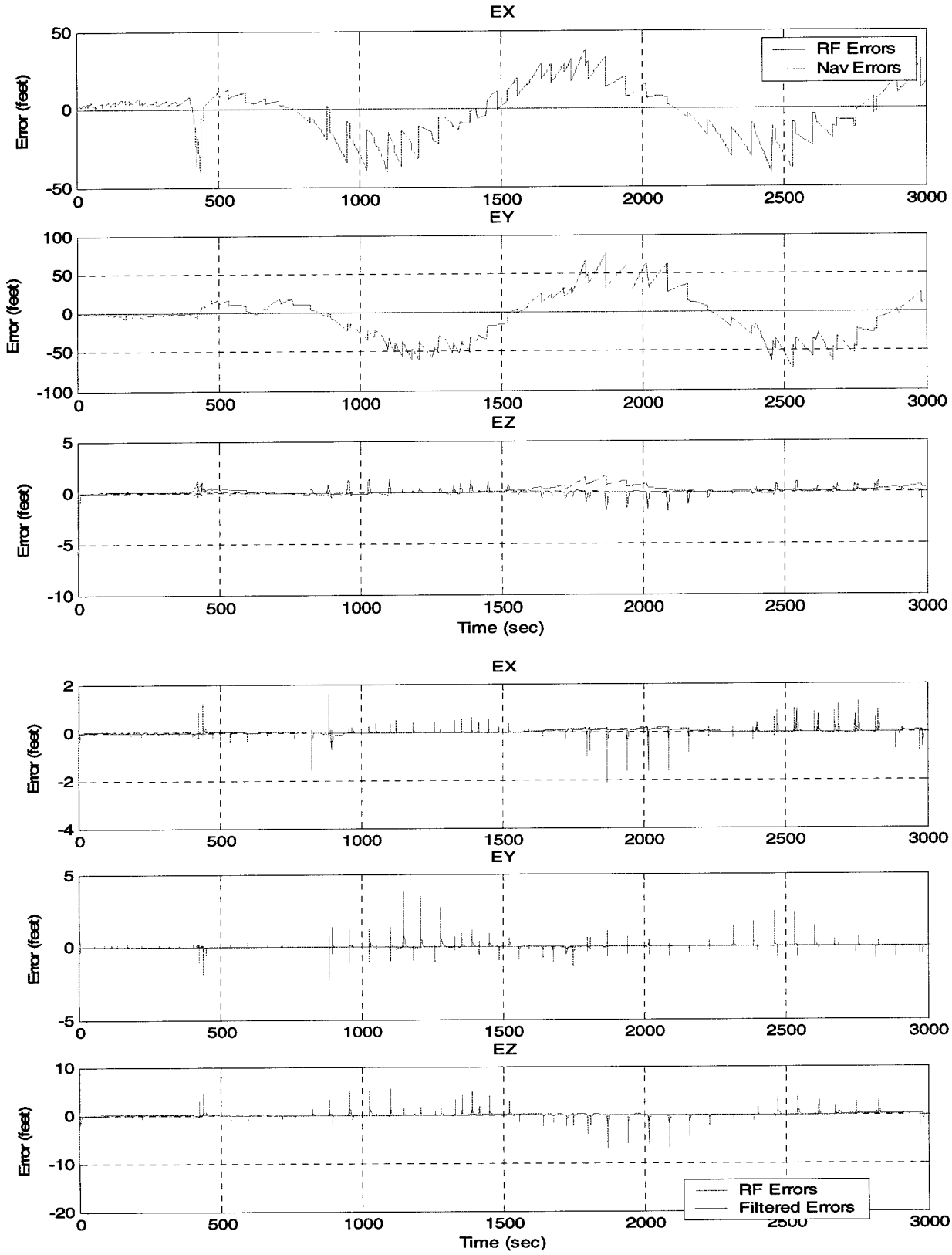


Figure 4.3-8: Low pass filtering – 1Hz

Analysis of Results

The application of a low pass digital filter with time constant tau of 1 second to the DGPS error quantities brings the steady state error of each of the position states down to well under the desired one meter for each of the observation rates tested. The meaning of the discontinuities that were removed, however, is not that the DGPS filter estimates themselves were spiky, but in fact, the hybrid navigation solution itself. Therefore, an important lesson to keep in mind when we wish to combine the results of the Link-16 navigation and the DGPS is not to add the DGPS corrections to the L16 hybrid solution itself, but to a low pass filtered version of the Link-16 hybrid navigation solution.

In conventional usage of the Link-16 hybrid navigation solution, the step changes due to Kalman corrections are normally too small to worry about. In precision navigation, however, these step changes are large compared to the desired performance levels, and must be treated as stated above. We shall return to this important distinction in the conclusions discussed in Paragraph 5.0.

4.3.3 Test Case (2-3) Performance Sensitivity to Link-16 Navigation Position and Velocity Errors

In paragraphs 4.2.3 and 4.2.5, we examined the effect of unmodelled Link-16 velocity and position errors on DGPS RF performance. We return to this study using Test Bed 2 by imposing larger than normal Link-16 navigation dynamic position and velocity errors. We achieve these errors by derating the IMU model to provide a level of performance which is worse than would normally be tolerated on a modern high performance fighter aircraft. Specifically, we have chosen to increase the gyro drift terms (constant portion of gyro drift rate) by an order of magnitude. As will be noted below, this will have an effect of nearly doubling the magnitude of the position and velocity terms of the relative Link-16 navigation solution over that provided by a nominal error budget, as shown in the previous two paragraphs. While these position errors exceed 125 feet per horizontal axis, they remain well below our computed limit of approximately 760 feet per axis for DGPS operation. As will be noted below, a minimal degradation in DGPS RF performance will result, which, in turn, can be well damped via use of the low pass filtering technique introduced in 4.3.2.

Test Conditions

Link-16 Navigation Error Budget:/ Measurement Schedule - Nominal, except for Gyro drift bias terms increased to:

BX_G = 1.5E-2 [degrees/hr]
BY_G = 6.8E-2 [degrees/hr]
BZ_G = 1.3E-2 [degrees/hr]

DGPS Configuration – Nominal
Pseudorange Update Rate 1Hz

Graphical Results

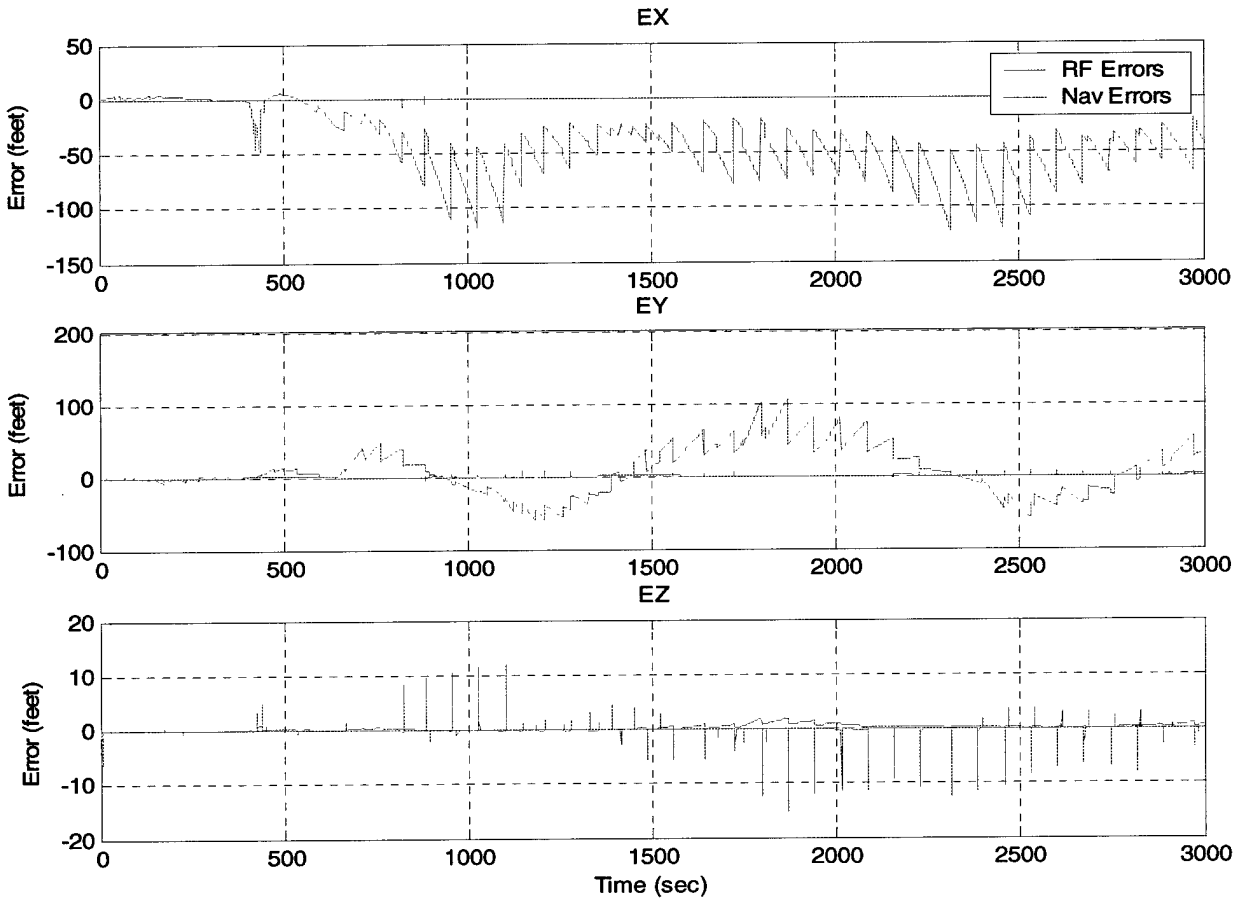


Figure 4.3-9: Sensitivity to large L16 navigation errors

Analysis of Results

As is clearly indicated in Figure 4.3-9, the Link-16 relative navigation solution has been severely impacted by the increase in gyro bias of the respective inertial measurement units, with position swings nearly tripling to the 130 feet/ axis. By comparison, the degradation of the DGPS RF solution is restricted to increases in the spikes of less than 12 feet. As was the case in the previous section, the addition of low pass filtering can easily reduce these to acceptable values. We thus have demonstrated that the DGPS RF algorithm can easily accommodate much poorer than nominal platform navigation performance, and still recover the majority of these errors.

4.3.4 Test Case (2-4) Sensitivity to Number of Satellites Processed

Although the DGPS RF algorithm will normally attempt to process as many satellites as are visible and above 5 degrees elevation, one can conceive of hostile jamming situations where some percentage of these satellites are not usable. It is therefore useful to study the performance of the DGPS RF algorithm where the number of satellites processed per update period is restricted to less than the normal 10-12 number, In this test case, we have capped the number of processed satellites over the range { 8,6,5, and 4} to investigate the performance of the DGPS RF filter.

Test Conditions

Link-16 Navigation Error Budget:/ Measurement Schedule - Nominal
DGPS Configuration – Nominal
Psuedorange Update Rates: 1Hz

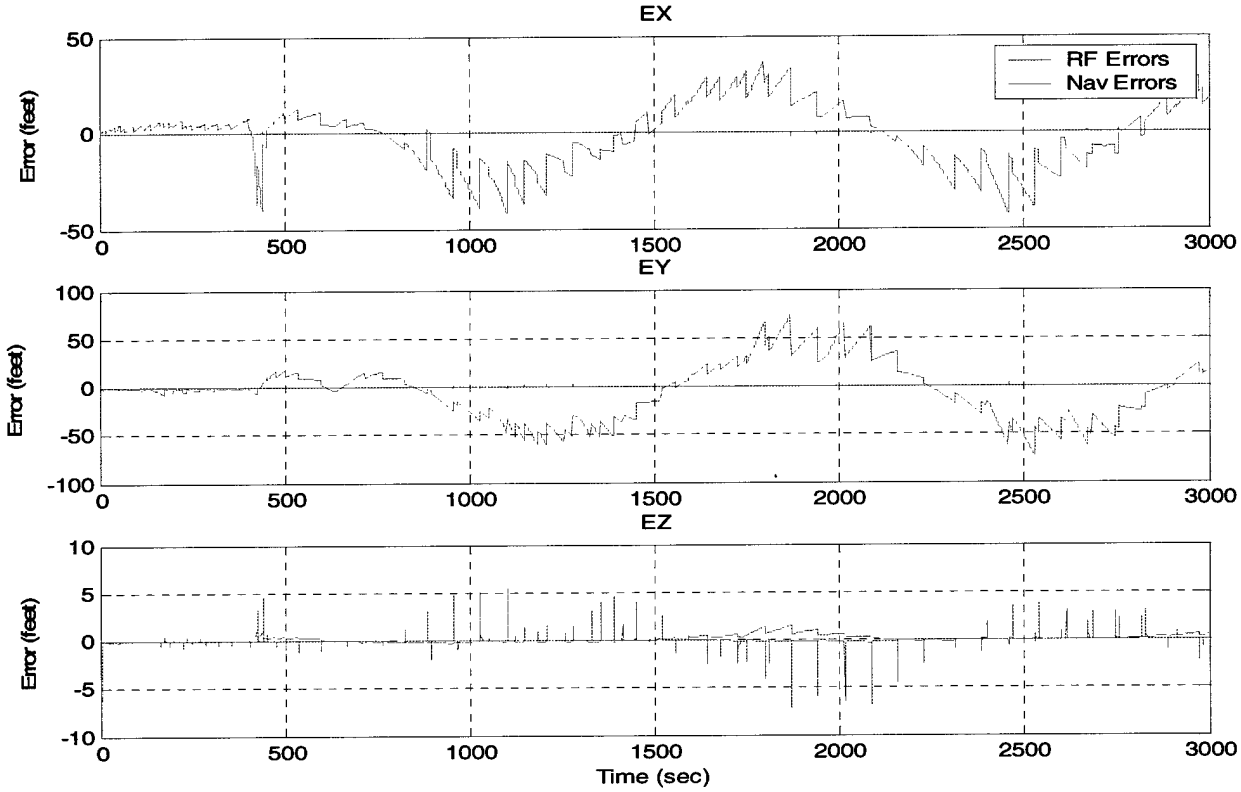


Figure 4.3-10: 8 satellite constellation

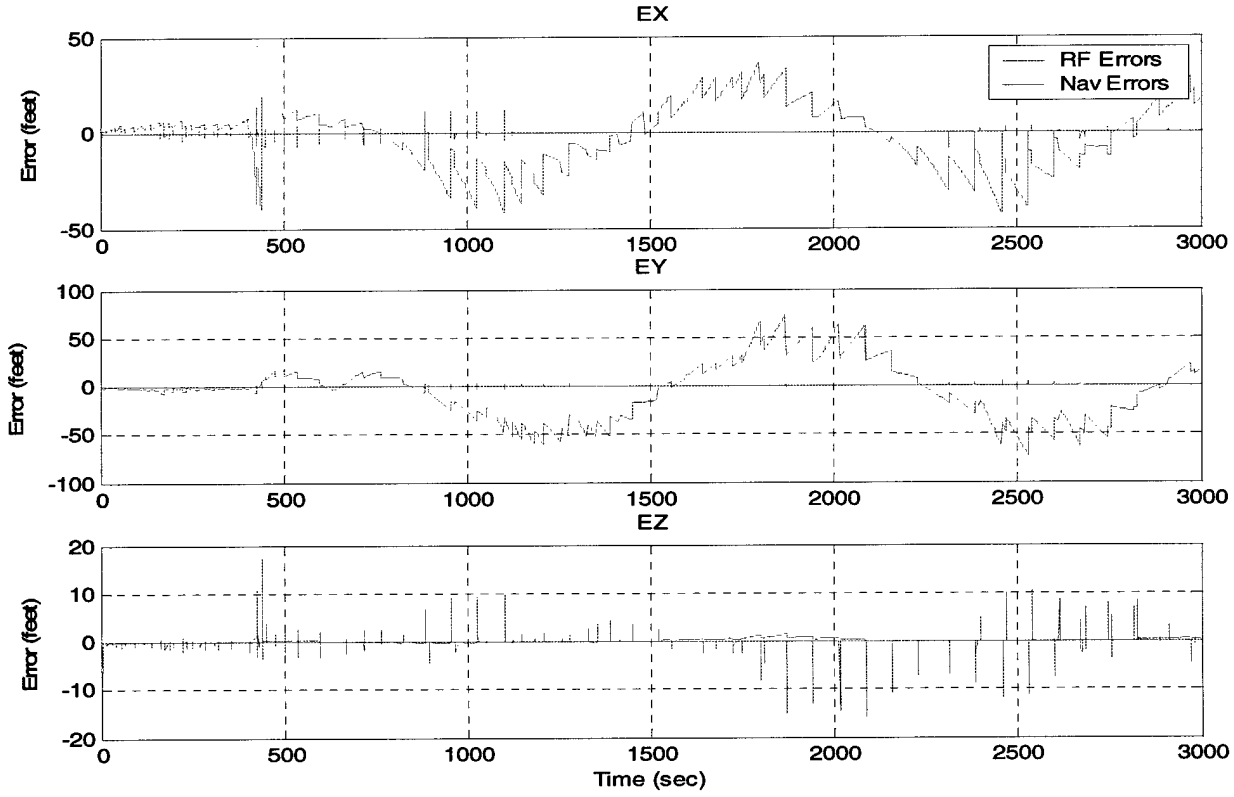


Figure 4.3-11: 6 satellite constellation

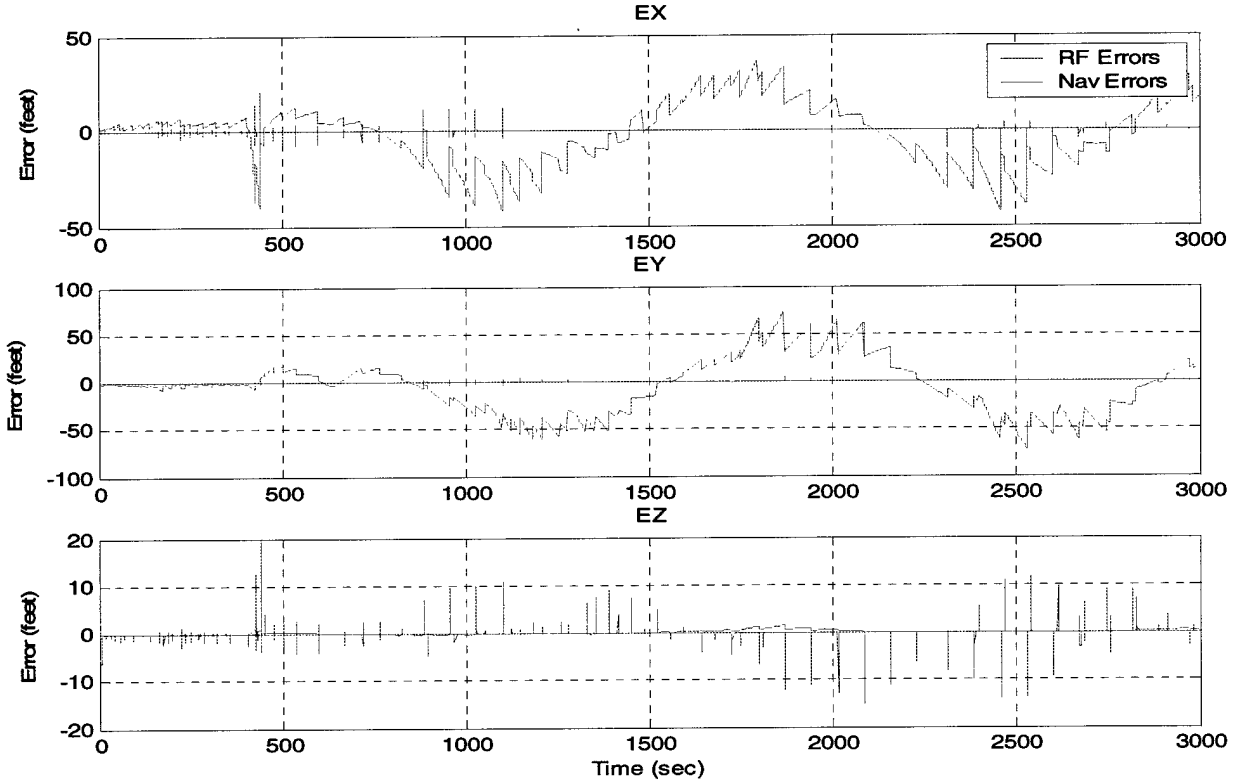


Figure 4.3-12: 5 satellite constellation

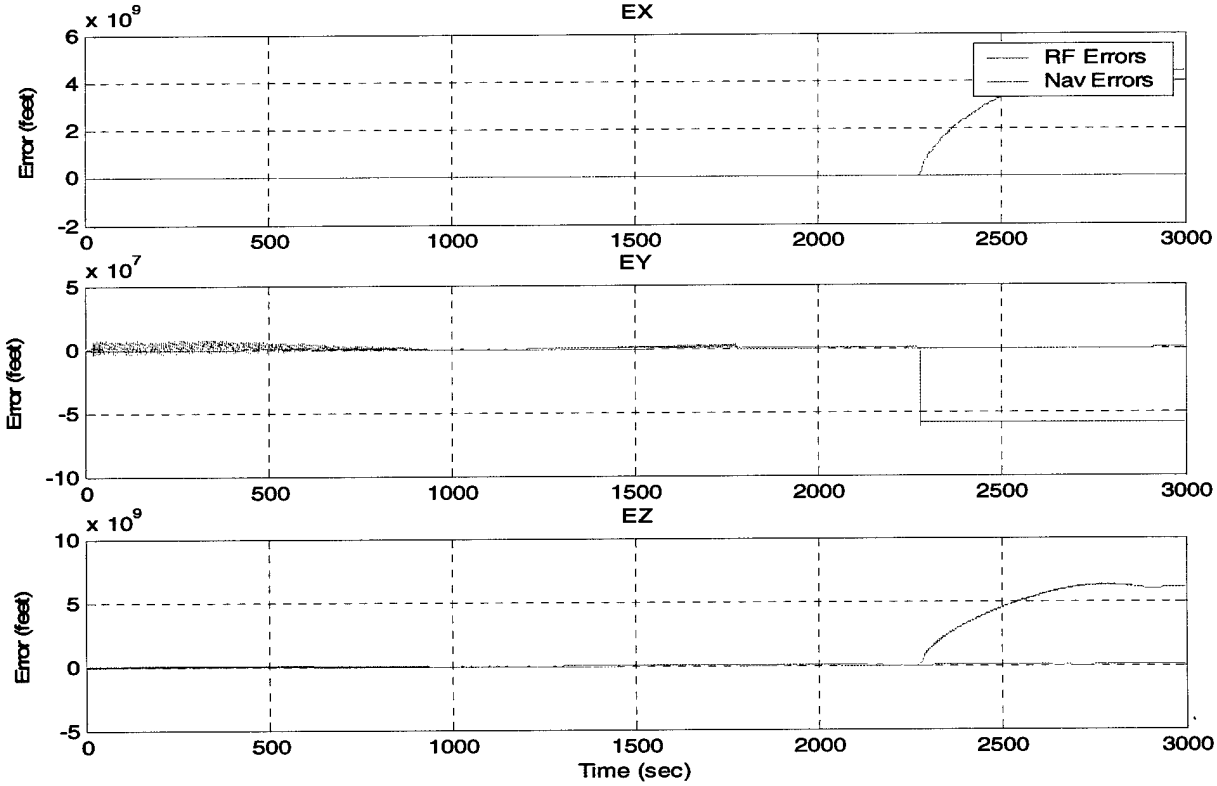


Figure 4.3-13: 4 satellite constellation

Analysis of Results

Figure 4.3-10 illustrates the performance of the DGPS RF filter when only 8 satellites are processed per update cycle. Virtually no difference is noted between these results and that shown in Figure 4.3-4, where 10 satellites are processed. Further reducing the number of satellites to 6, as shown in Figure 4.3-11 shows only slightly higher spike errors as compared with the previous case. Restricting the number of satellites observed to 5, as shown in Figure 4.3-12, shows somewhat greater degradation, but these errors can be easily smoothed by low pass filtering, as was discussed in paragraph 4.3.2. It is only when the number of processed satellites decreases to 4, which is the bare minimum for normal GPS PVT solutions, that the DGPS RF filter solution collapses completely. We can therefore conclude that the DGPS RF algorithm is highly robust with respect to the number of satellites processed. An additional observation is that the largest effect of a reduction in satellites processed is observed in the vertical axis. We note, however, that this axis usually showed the largest errors, due to the fact that most of the processed satellites are relatively low on the horizon, and thus provide less information for the vertical axis

4.3.5 Test Case (2-5) Sensitivity to Clock Bias and Frequency Errors

The objective of this test case is to revisit the DGPS RF solution sensitivity to large clock bias and frequency errors, a case that was first studied in Test Bed 1, Paragraph 4.2.8. In the former case, the Link-16 position errors were essentially constants, whereas in this test case, the full dynamic navigation error model has been applied. We again consider large clock bias and frequency errors ,namely 200 ns of GPS receiver bias, coupled with a 10 ns/seconds frequency drift term. We again note that these are very large values unlikely to occur in modern GPS receiver units.

Test Conditions

Link-16 Navigation Error Budget:/ Measurement Schedule - Nominal
DGPS Configuration – Nominal
Pseudorange Update Rate: 4 Hz
GPS Clock Bias: 200 ns
GPS Frequency Drift: 10 ns/sec

Graphical Results

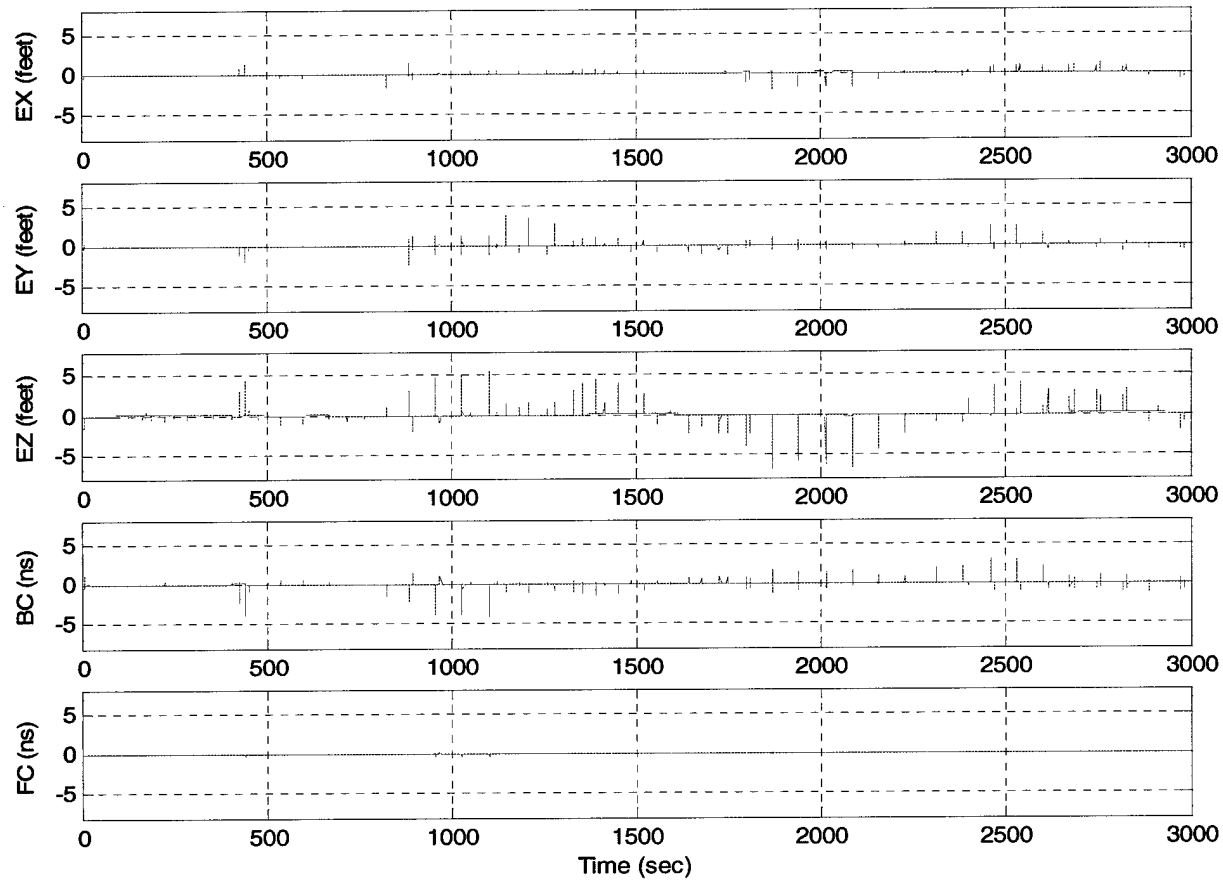


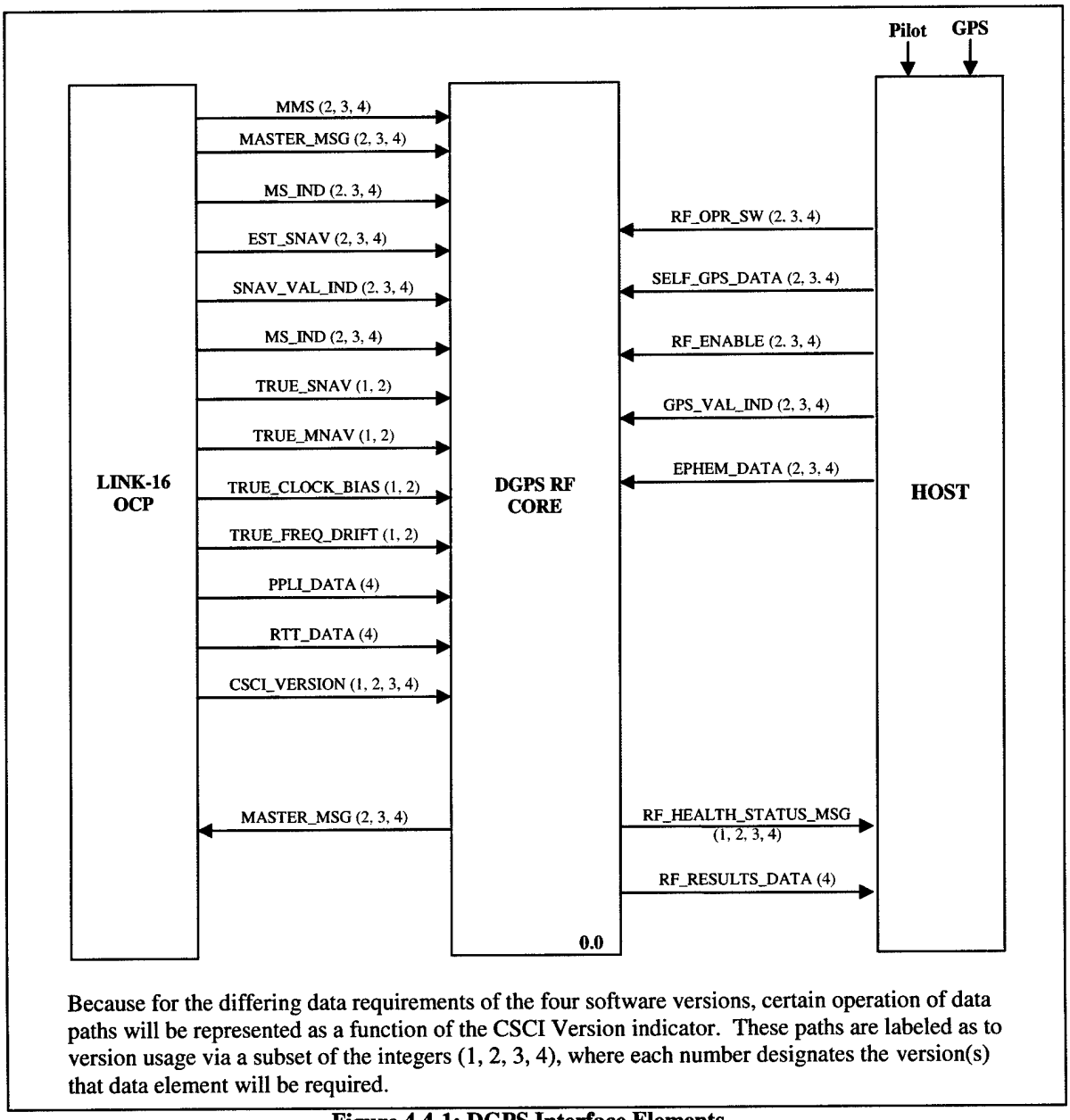
Figure 4.3-14: Test Bed 2 – clock bias/frequency error performance

Analysis of Results

The successful estimation by the DGPS RF algorithm of these large clock bias and frequency terms noted in Paragraph 4.2.8 is replicated here as well. The performance levels of the three position states is equivalent to earlier Test Bed 2 results, while the Clock Bias (B_c) and Frequency (f_c) results are likewise stable and well behaved. In short, the filter has been demonstrated to produce robust performance even in the case of severe clock bias and frequency offsets. In other runs, not reproduced here, equivalent performance was demonstrated even at slower update rates of 1 and 2Hz, at the expense of a slightly slower filter convergence rate.

4.4 DGPS RF Operational Software Design

This section provides an overview of the software implementation of the DGPS RF algorithm as an extended version of the existing Link-16 operational computer program (L16 OCP). The core processing of the DGPS RF interfaces with the L16 OCP and the host. The L16 OCP provides critical self-navigation data derived from the organic navigation algorithm of L16. Also provided by the L16 OCP are received Master Message data. The host provides GPS ephemeris data, GPS pseudorange data, and control data. Figure 4.4-1 illustrates the DGPS interface elements. All algorithm inputs and outputs are presented, respectively, in Tables 4.4-1 and 4.4-2.



Because for the differing data requirements of the four software versions, certain operation of data paths will be represented as a function of the CSCI Version indicator. These paths are labeled as to version usage via a subset of the integers (1, 2, 3, 4), where each number designates the version(s) that data element will be required.

Figure 4.4-1: DGPS Interface Elements

Inputs	Version	Definition	Description
MMS	2,3,4		Master Message Signal - indicates master message is present and valid
MASTER_MSG	2,3,4	MNAVTOV, MLAT, MLON, MALT, MEVEL, MNVEL, MALTR, GPS_HEALTH_IND, MGPSTOV, UMSAT, UMPRR	Master's Navigation solution, along with master GPS data
SNAV_VAL_IND	2,3,4		Self Navigation Validity Indicator
EST_SNAV	1,2,3,4	SLAT, SLON, SALT, SEVEL, SNVEL, SALTR, SNAVTOV	Navigation solution from L16 OCP
MS_IND	2,3,4		Master/Slave indicator
TRUE_SNAV	1,2	TSLAT, TSLON, TSALT, TLATR, TLONR, TSALTR, TSNAVTOV	TRUE SELF navigation solution
TRUE_MNAV	1,2	TMLAT, TMLON, TMALT, TMLATR, TMLONR, TMALTR, TMNAVTOV	TRUE MASTER navigation solution
TRUE_CLOCK_BIAS	1,2	BCTRUE	TRUE clock bias
TRUE_FREQ_DRIFT	1,2	FCTRUE	TRUE frequency drift
PPLI_DATA	4		Requested PPLI
RTT_DATA	4		Requested RTT
CSCI_VERSION	1,2,3,4	CSCI_VERS_ID	Denotes operational version
RF_OPR_SW	2,3,4		Reset/Initiate signal from Host pilot
SELF_GPS_DATA	2,3,4	USSAT, USPRR, SGPSTOV, GPS_HEALTH_IND	Received GPS data from HOST
RF_ENABLE	2,3,4		ON/OFF switch
GPS_VAL_IND	2,3,4		GPS validity Indicator
EPHEM_DATA	2,3,4	EPHEM(30,23)	Ephemeris satellite data received from the host for pseudorange calculations

Table 4.4-1: DGPS RF Inputs

Outputs	Version	Definition	Description
MASTER_MSG	2,3,4	MNAVTOV,MLAT, MLON, MALT, MEVEL, MNVEL, MALTR, GPS_HEALTH_IND, MGPSTOV, UMSAT, UMPRR	Master's Navigation solution, along with master GPS data
RF_RESULTS_DATA	4	XA, PA	Refinement Filter corrections and covariance message sent to L16 OCP for AF
RF_HEALTH_ STATUS_MSG	1,2,3,4	RF_STATE, RF_STATUS_IND, XA, PA, SCOUNT, SATVIS, SNAVTOV	Indicates health of the refinement filter to the host to determine whether the filter should be reset, switch to backup mode, or system normal.

Table 4.4-2: DGPS RF Outputs

The DGPS RF algorithm is comprised of seven CSCs whose identification and general description are summarized in Table 4.4-3, below.

CSC	Description
RF_EXEC_CTRL	Controls the data flow within Refinement Processing depending on inputs to the system.
RF_SOURCE_DATA_PROCESSING	Prepares and synchronizes all data for Kalman filtering.
RF_KALMAN_PROCESSING	Applies Kalman filtering to the data passed from the RF_SOURCE_DATA_PROCESSING unit.
HOST_INTERFACE_PROCESSING	Interfaces with host and processes data that enters this system from the host, mainly control signals and GPS data.
OCP_INTERFACE_PROCESSING	L16 OCP interface from which all OCP data enters the system, mainly navigation solution including position, velocity, and time.
BUILD_MASTER_MESSAGE	Called if the platform is designated the master of the network from the host. The master message contains master navigation data and is sent at a 2 Hz rate to all the members in the network.
BUILD_RF_RESULTS_MESSAGE	Outputs DGPS RF solution, status, and covariance to host.

Table 4.4-3: DGPS RF 7 main Computer Software Components

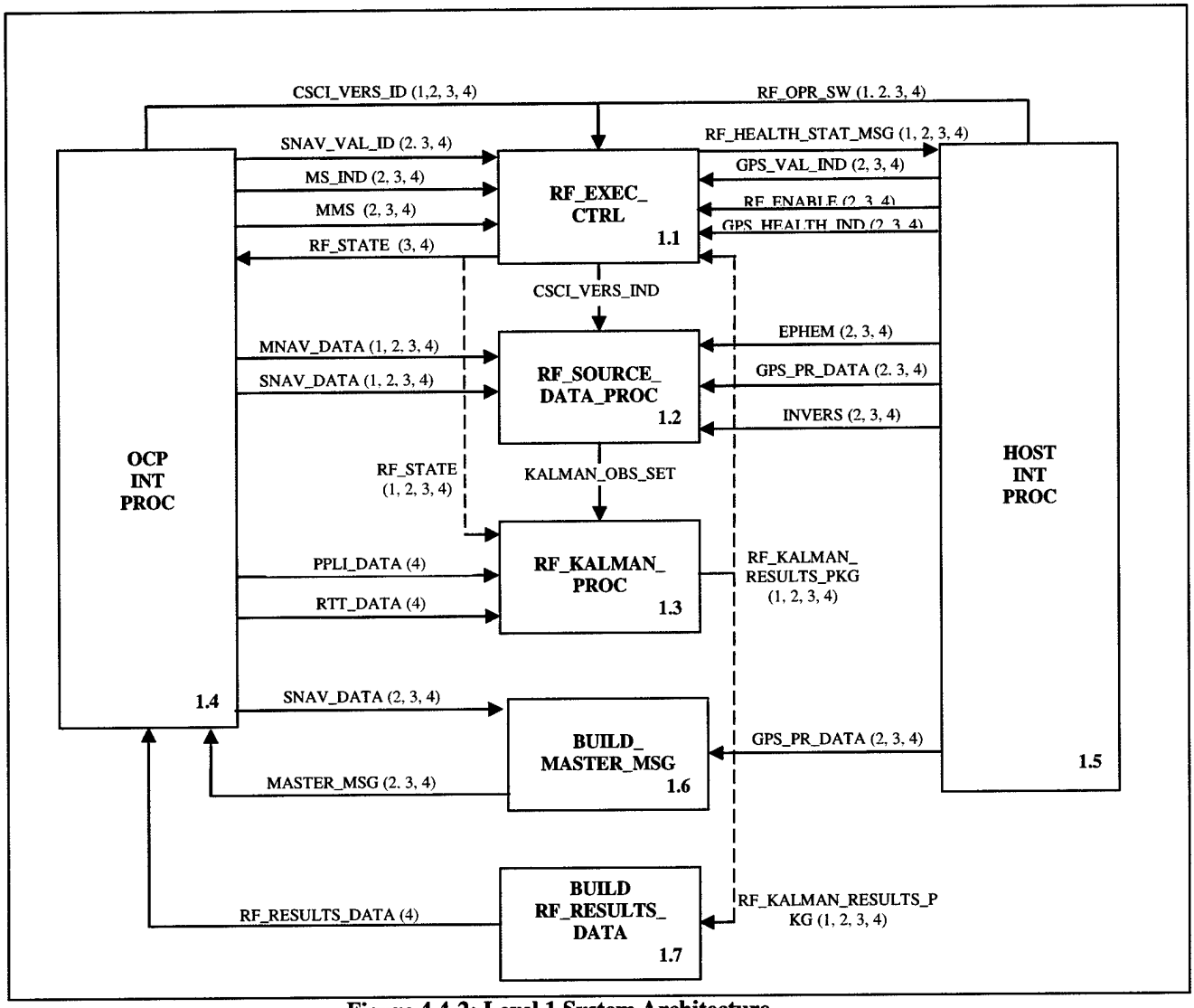


Figure 4.4-2: Level 1 System Architecture

The following sections provide brief technical descriptions of each of the seven CSCs. A Software Design Document for the DGPS RF CSCI is presented in Appendix A.

4.4.1 RF_EXEC_CTRL

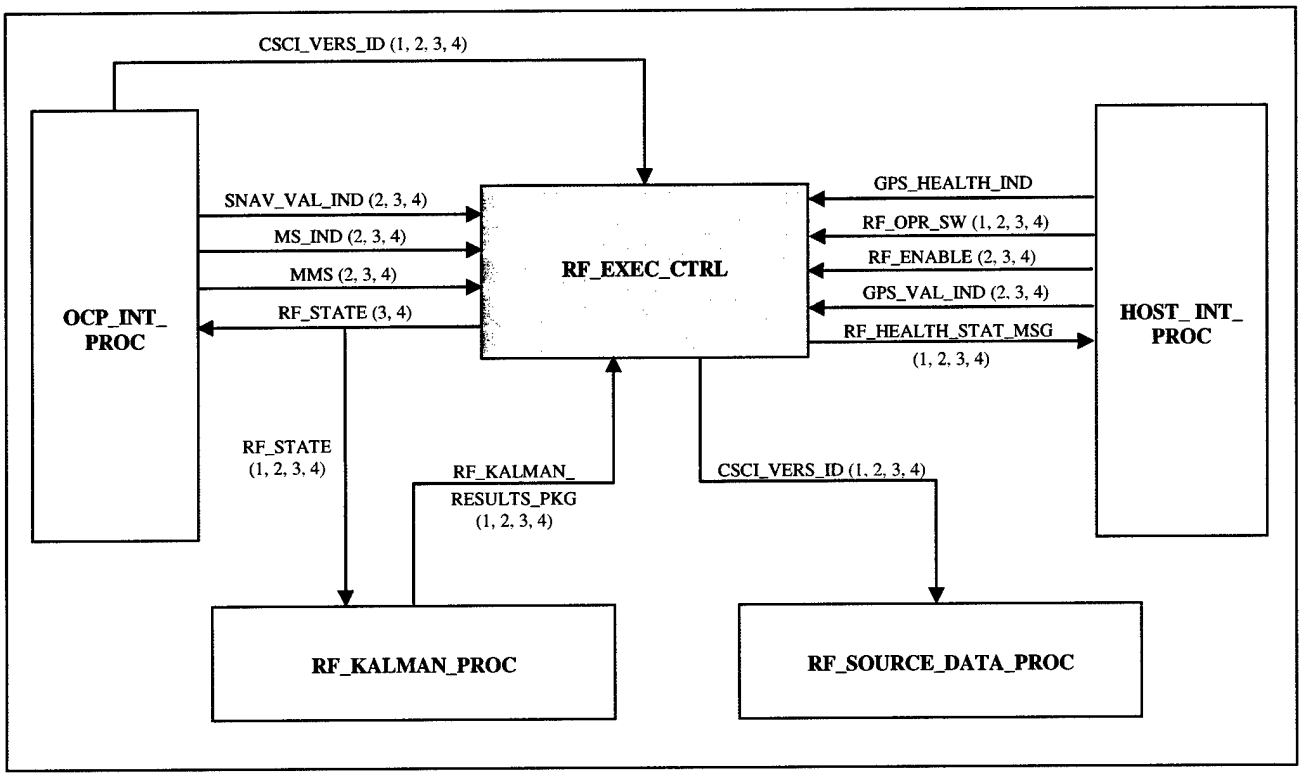


Figure 4.4-3: RF_EXEC_CTRL Level 1 Breakdown

The Refinement Filter Executive Control is the primary control element of the DGPS RF algorithm. A fundamental requirement of the algorithm is the pre-existence of a stable community navigation solution, as provided by conventional Link-16 navigation. The required indicators for this condition are: (a) Self-navigation valid and (b) Master navigation valid, both of which are provided by OCP_INT_PROC. Also required for algorithm operation is the presence of valid GPS pseudorange and ephemeris data, as well as algorithm selection by the operator. All of the above conditions must be satisfied prior to the initialization and execution of the algorithm.

When conditions exist to begin algorithm execution, RF_EXEC_CTRL shall invoke the various sub-elements of the program, as appropriate.

RF_STATUS_INDICATOR (RSI)

The RSI indicates the status of the filter and locates the source of RF operation failure.

RSI	Status	Conditions
0	RF Inoperative	Off Selected
1	RF Inoperative	SNAV_VAL_IND = invalid
2	RF Inoperative	MNAV_VAL_IND = invalid
3	RF Inoperative	GPS failure
4	RF Operative	Normal Mode / Non Steady State
5	RF Operative	Normal Mode / Steady State
6	RF Operative	Backup / Non Steady State
7	RF Operative	Backup / Steady State
8	RF Master Mode	
9	RF Observation Failure Reset Command	

Table 4.4-4: RF_STATUS_INDICATOR description

RF_STATE

The RF_EXEC_CTRL is modeled as a finite state machine, directly corresponding to the actual state of the Refinement Filter, which can be in one of 7 states. The RF_EXEC_CTRL determines the state of the filter according to validity checks and the health of data and observations. The states are defined as follows:

State 0 – Initial startup state, RF reset
- If the filter should fail any validity check, become instable, or obtain deficient observations, it returns to this state

State 1 – Master designation
- Requires no Kalman Processing

State 2 – Slave designation / not yet initialized
- Initializes filter variables
  - PA, PX, QA, RA
- RTT_REQUEST = OFF

State 3 – Slave Designation / GPS Pseudorange - steady state not achieved
- Normal mode
- Filter working, not yet steady state

State 4 – Slave Designation / GPS Pseudorange - steady state achieved
- state at which RF will most likely exist

State 5 – Slave Designation / RTT, PPLI – backup mode initialization (not steady state)

- GPS problem
  - Insufficient signal strength
  - Number of satellites in view fall under minimum requirement
- RTT_REQUEST = ON
- Using PPLI/RTT in Kalman Processing
- Reconfigure RF: 5 → 8 states
- Steady state not yet achieved

State 6 – Slave Designation / backup mode – steady state achieved

- Backup until GPS becomes available

RF_EXEC_CTRL switches between states when the state of the data changes. If RF_EXEC_CTRL resides in State 4, and then GPS becomes weak due to the number of satellites available, or if the health indicator has fallen below the minimum requirement, then RF_EXEC_CTRL falls into State 5 and begins requesting PPLI transmissions from OCP_INT_PROC CSC.

The backup mode, indicated by State 6, is used when GPS data becomes temporarily unavailable. In this mode, PPLI data from the master platform is used as an observation to the DGPS Kalman Filter to maintain the error state solution.

Figure 4.4-4 on the next page illustrates the 7-state state transition conditions for the RF.

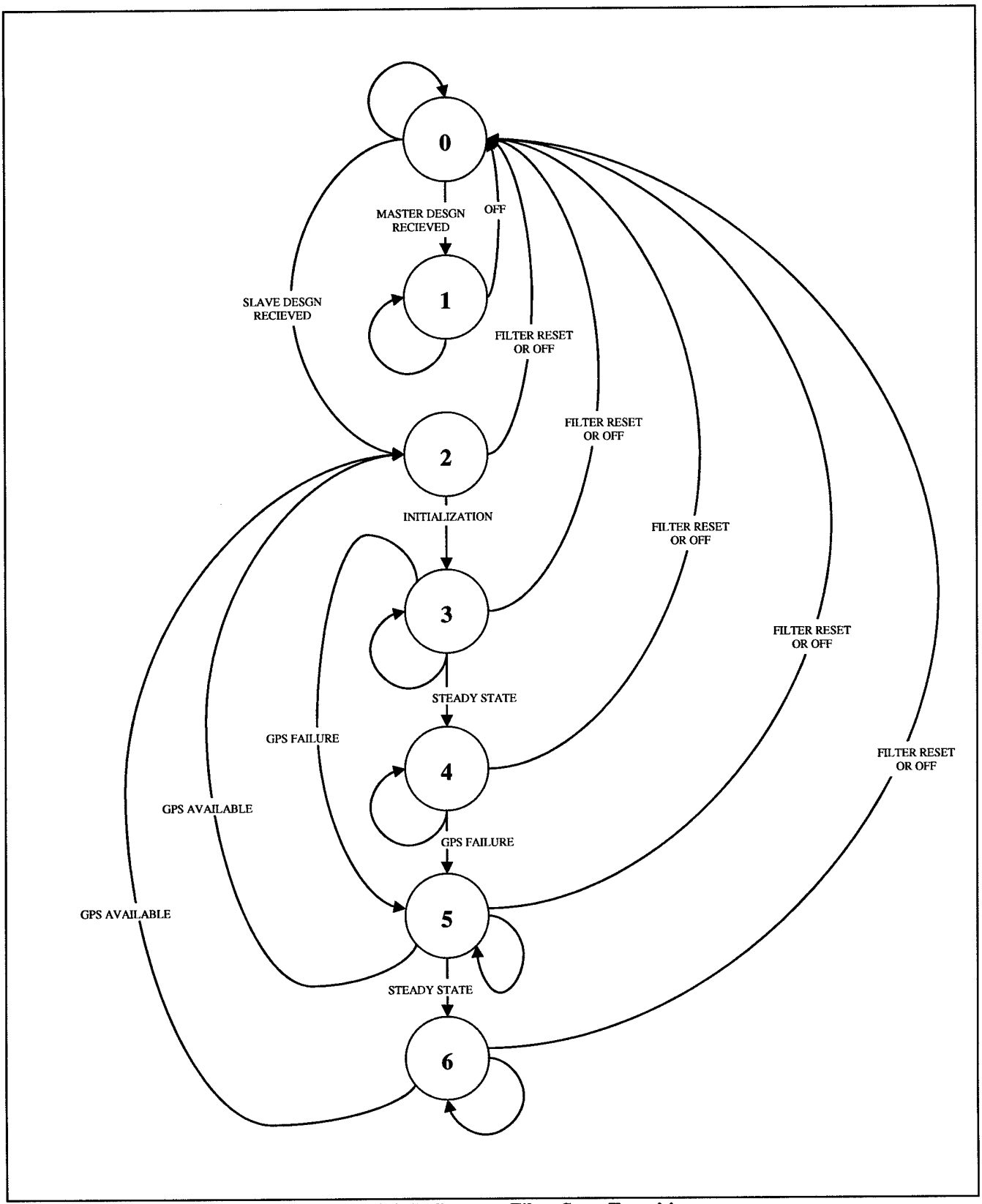


Figure 4.4-4: Refinement Filter State Transitions

CSC Inputs	Version	Definition	Description
CSCI_VERS_ID	1,2,3,4		Denotes operational version
GPS_HEALTH_IND	2,3,4		Indicates the health status of GPS signal. When this signal is too weak, backup mode takes over.
MMS	2,3,4		Master Message Indicator: Check if the master message is ready for transmission
MS_IND	2,3,4		Master/Slave Indicator: Checks for pilot designation for platform either master or slave aircraft.
RF_ENABLE	2,3,4		On/Off Switch
RF_KALMAN_RESULTS_PKG	1,2,3,4	Time (TC), RF_STATE, RSI, XA, PA	Refinement Filter results and covariance, XA and COV
RF_OPR_SW	1,2,3,4		Refinement filter operation switch: ON/OFF switch for system initialization/reset
GPS_VAL_IND	2,3,4		Master Navigation Validity Indicator
SNAV_VAL_IND	2,3,4		Self Navigation Validity Indicator

Table 4.4-5: RF_EXEC_CTRL Inputs

CSC Outputs	Version		Description
CSCI_VERS_ID	1,2,3,4		Denotes operational version
RF_HEALTH_STAT_MSG	1,2,3,4	RF_STATE, RF_STATUS_IND, XA, PA,SCOUNT, SATVIS,SNAVTOV	Indicates health of the refinement filter to the host to determine whether the filter should be reset, switch to backup mode, or system normal.
RF_STATE	3,4		Indicates the state value of the refinement filter (See below for definitions of states). [OUTPUTS TO OCP_INT_PROC]
RF_STATE	1,2,3,4		Indicates the state value of the refinement filter (See below for definitions of states). [OUTPUTS TO RF_Kalman_Proc Module]

Table 4.4-6: RF_EXEC_CTRL Outputs

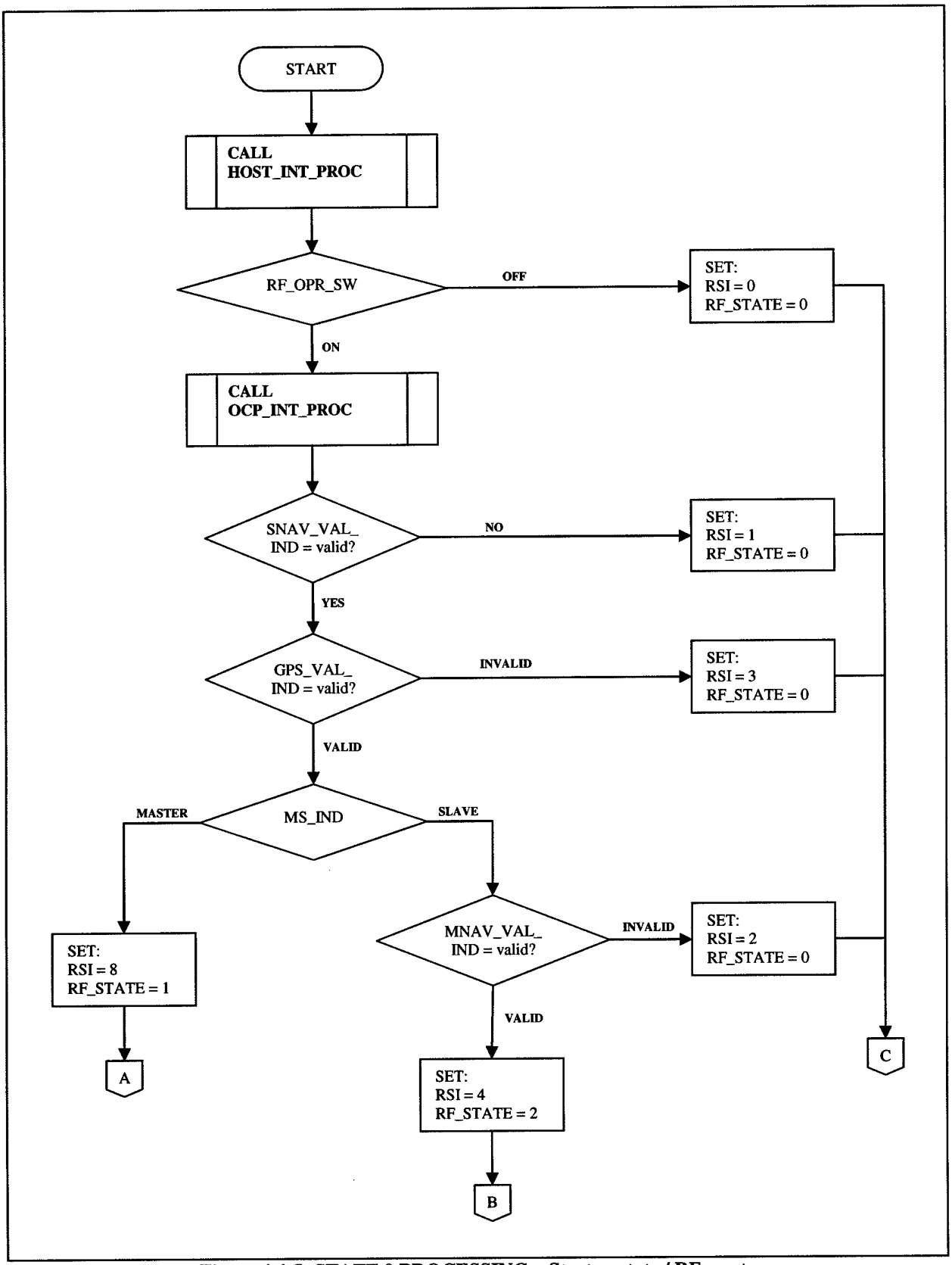


Figure 4.4-5: STATE 0 PROCESSING – Startup state / RF reset

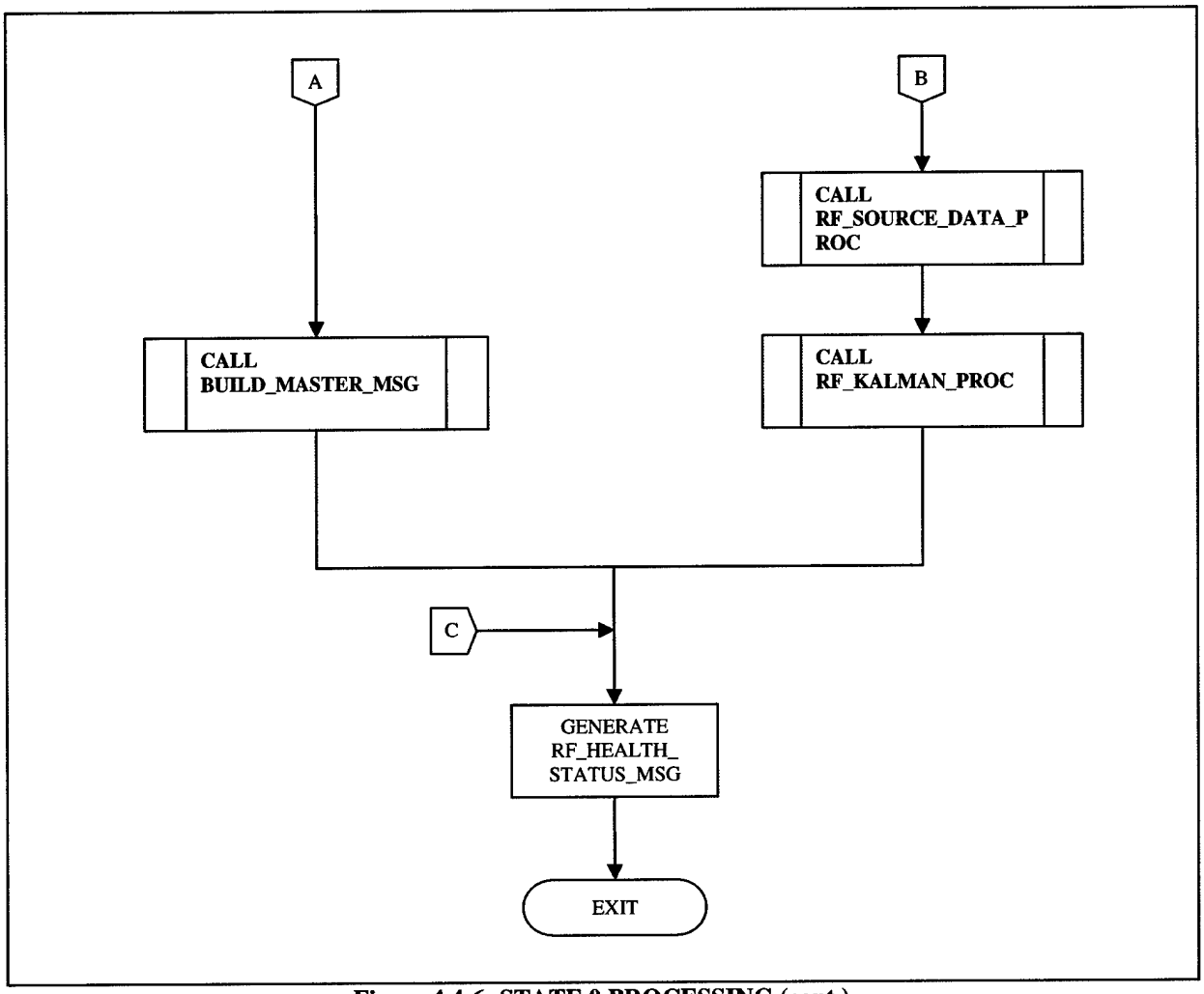


Figure 4.4-6: STATE 0 PROCESSING (cont.)

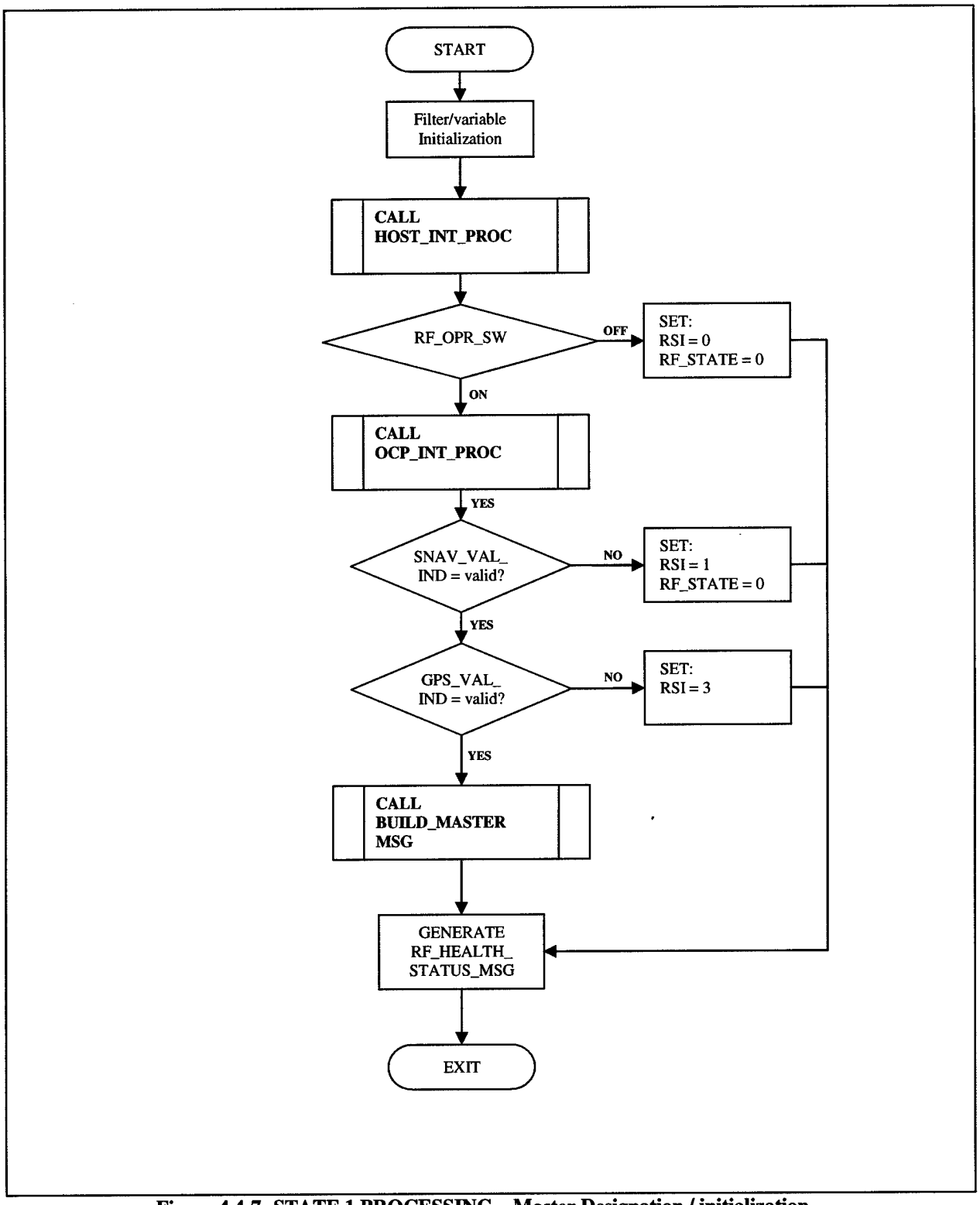


Figure 4.4-7: STATE 1 PROCESSING – Master Designation / initialization

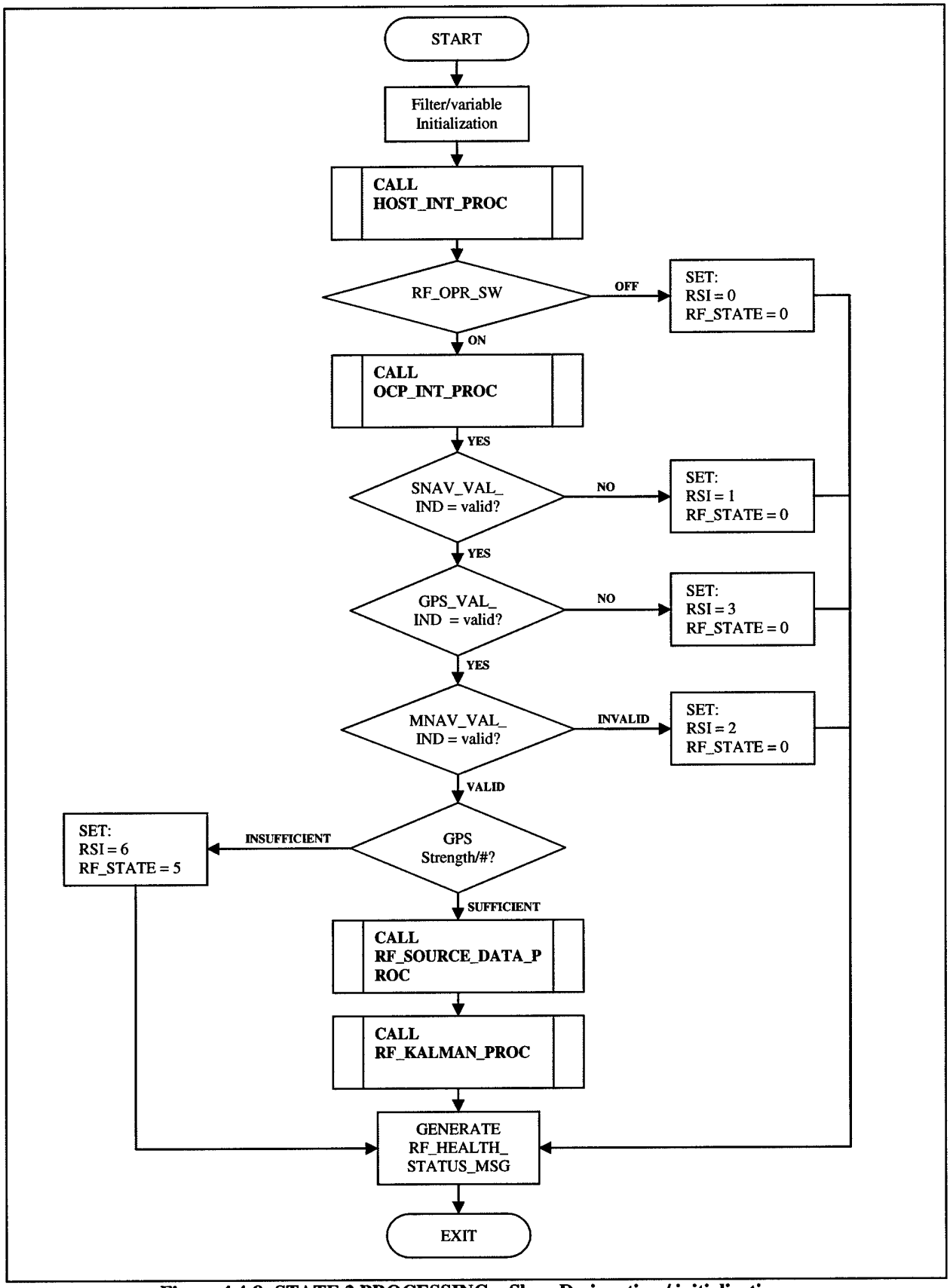


Figure 4.4-8: STATE 2 PROCESSING – Slave Designation / initialization

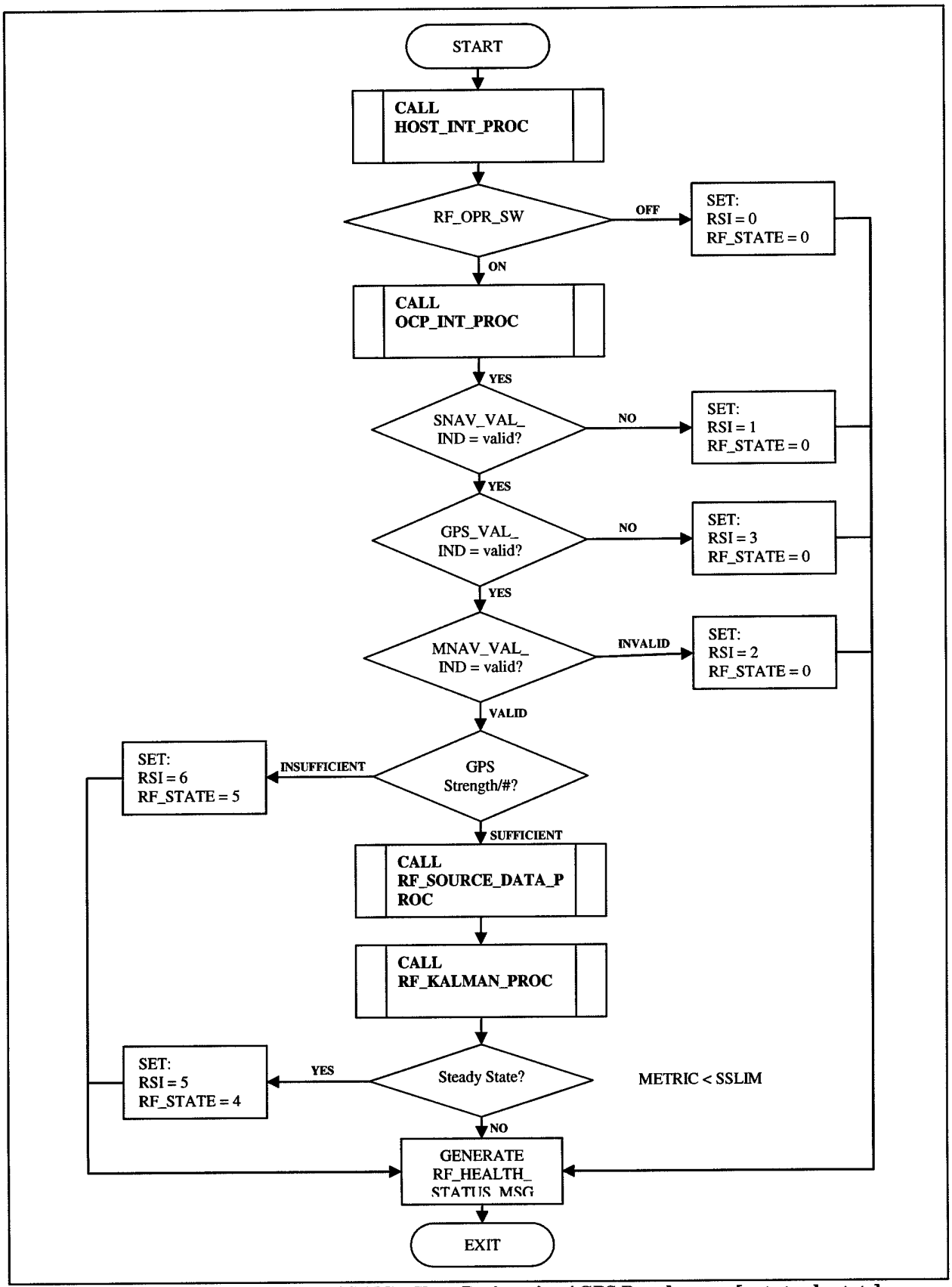


Figure 4.4-9: STATE 3 PROCESSING - Slave Designation / GPS Pseudorange [not steady state]

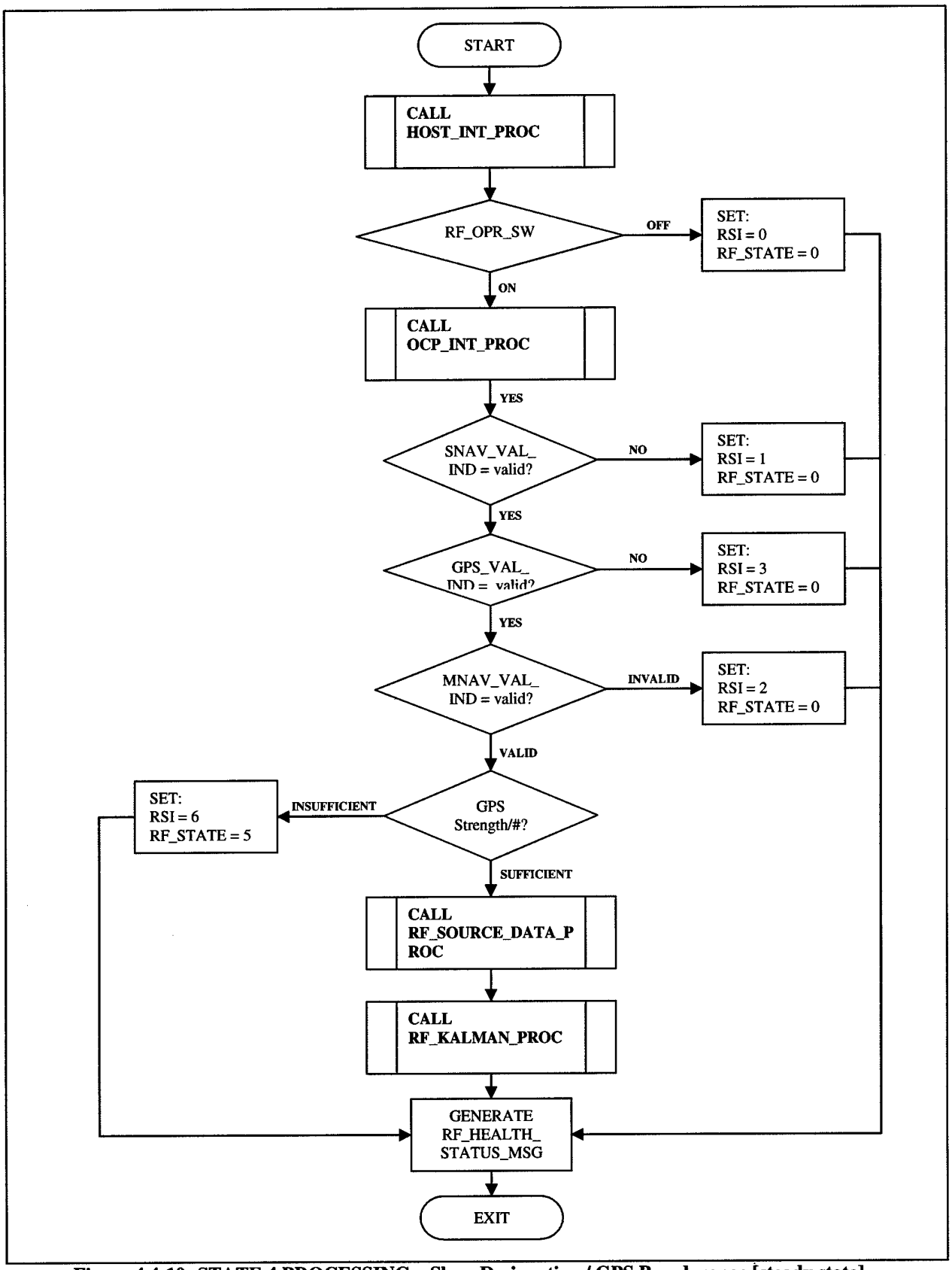


Figure 4.4-10: STATE 4 PROCESSING – Slave Designation / GPS Pseudorange [steady state]

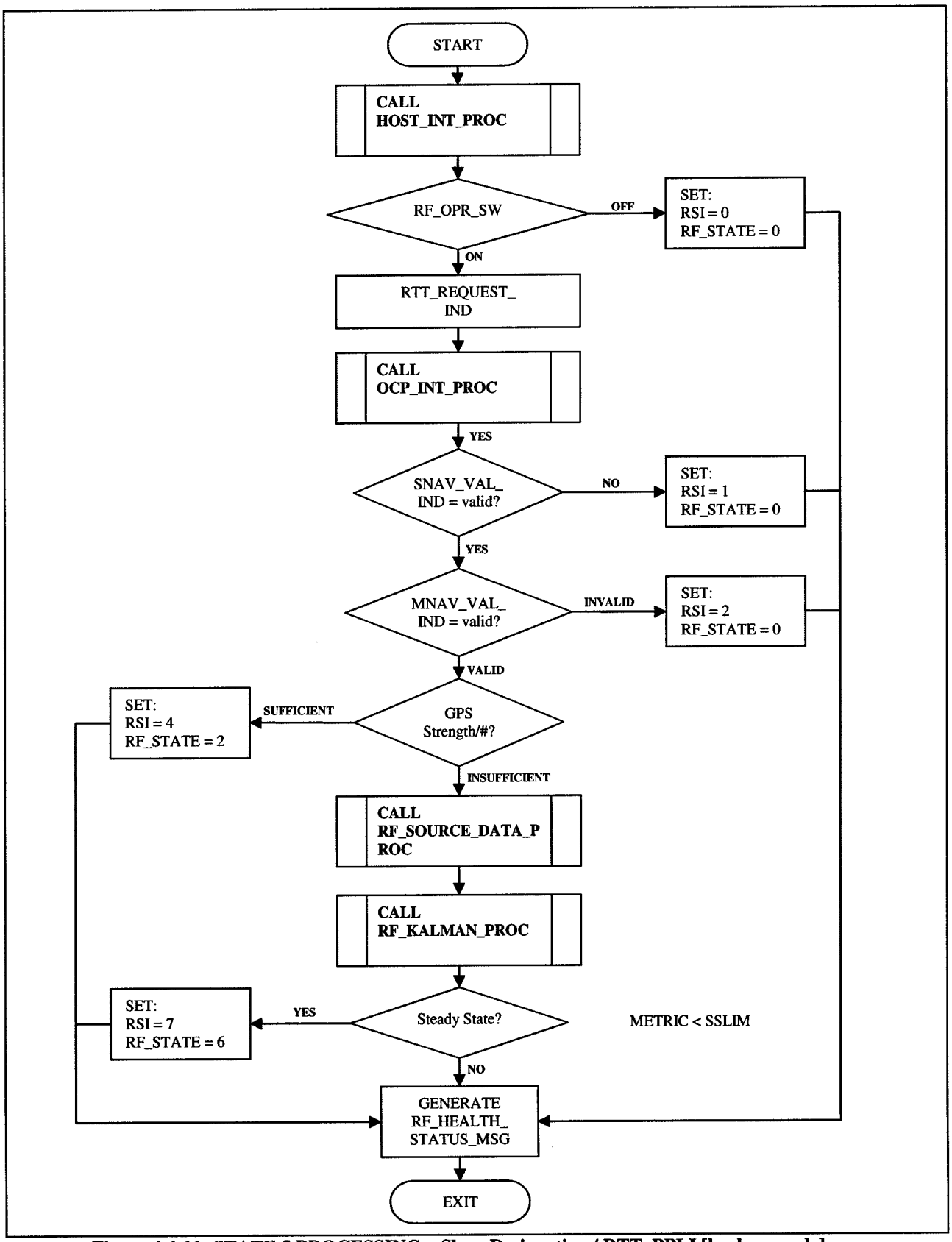


Figure 4.4-11: STATE 5 PROCESSING – Slave Designation / RTT, PPLI [backup mode]

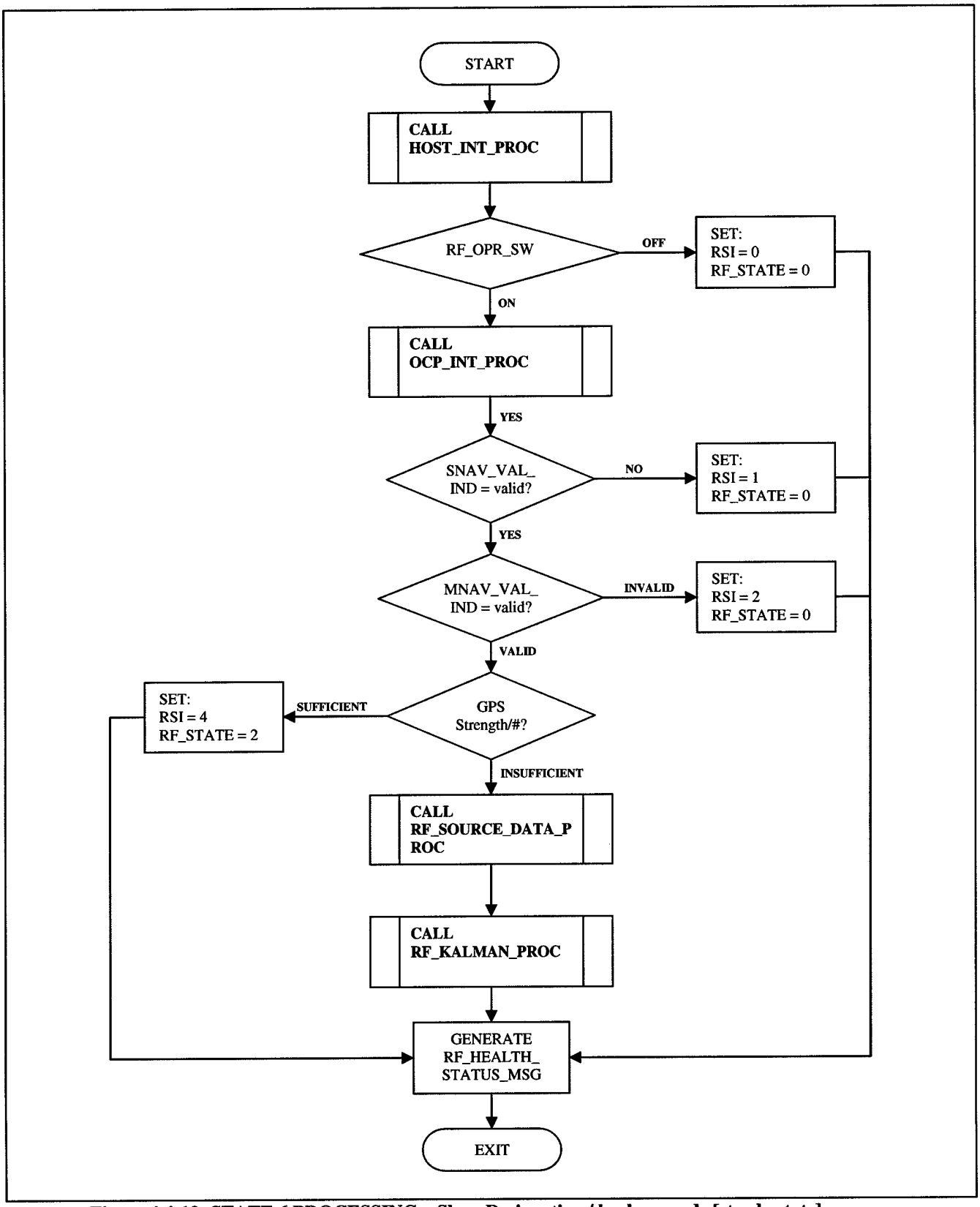


Figure 4.4-12: STATE 6 PROCESSING – Slave Designation / backup mode [steady state]

4.4.2 RF_SOURCE_DATA_PROC

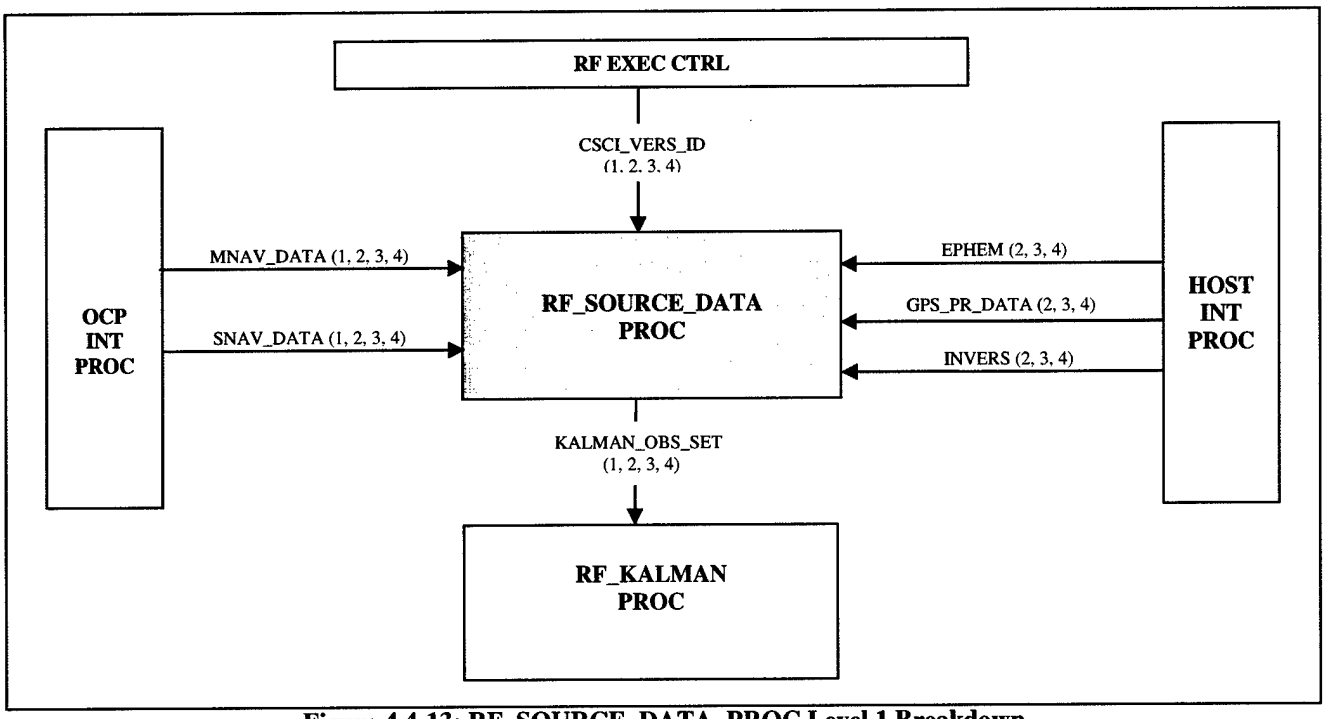


Figure 4.4-13: RF_SOURCE_DATA_PROC Level 1 Breakdown

The Refinement Filter Source Data Processing unit synchronizes and prepares data for Kalman processing. It is essential that all data is time synchronized before the Kalman filter can be applied.

The RF_SOURCE_DATA_PROC receives the true and estimated navigation and GPS data for both the master and slave. The master's GPS and navigation data is extracted from the Master message-derived navigation data package and then paired with platform's own navigation solution.

CSCI_VERS_ID value is passed from the RF_EXEC CTRL to determine which modes of system operation (i.e laboratory or field) should be executed. Figure 4.4-14 is a breakdown of the SLCSCs in the RF_SOURCE_PROC. Table 4.4-7 lists the SLCSCs and their corresponding functions. This table also indicates the software versions in which the SLCSC shall be utilized.

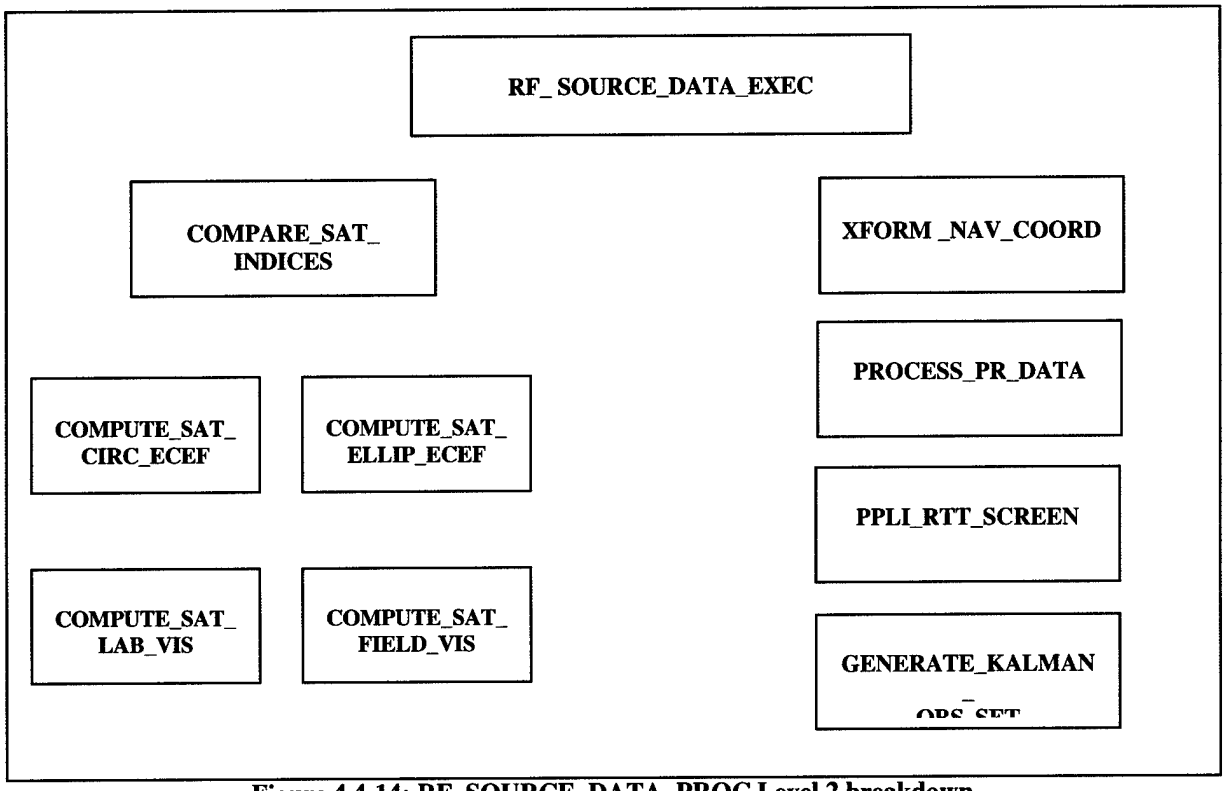


Figure 4.4-14: RF_SOURCE_DATA_PROC Level 2 breakdown

SLCSC	Description
RF_SOURCE_DATA_EXEC	Coordinates self navigation (HOST) with master navigation solution (OCP).
COMPARE_SAT_INDICES	Compares measured satellite indices for master/slave. Selects candidate observation group for common elements.
COMPUTE_SAT_CIRC_ECEF	Computes Satellite ECEF elements for common satellites when CSCI_VERS_ID = 2.0 (Laboratory Mode)
COMPUTE_SAT_ELLIP_ECEF	Computes Satellite ECEF elements for common satellites when CSCI_VERS_ID = 3.0 (Field Mode)
COMPUTE _SAT_LAB _VIS	Only satellites at least 5° above horizon are candidates when CSCI_VERS_ID = 2.0 (Laboratory Mode)
COMPUTE _SAT_FIELD _VIS	Only satellites at least 5° above horizon are candidates when CSCI_VERS_ID = 3.0 (Field Mode)
XFORM_NAV _COORD	Transforms self-navigation into Local Level and ECEF coordinates.
PPLI_RTT_SCREEN	Called only when CSCI_VERS_ID = 4.0 (Backup Mode) Takes PPLI/RTT Data when GPS is not valid.
PROCESS_PR_DATA	Converts satellite code phase data from master and self to generate arrays of psuedorange data from all common satellites in units of meter
GENERATE_KALMAN_OBS_SET	Generates Kalman Observation set from Satellite or PPLI/RTT Data

Table 4.4-7: RF_SOURCE_DATA_PROC SLCSC Descriptions

Additional processing is required for Version 3.0 in order to synchronizemaster and self data. For Version 2.0, a circular orbit model is used, since it is an excellent representation of the actual GPS satellite orbital dynamics. However, for Version 3, the GPS satellite pseudorange data requires the implementation of the true elliptical satellite model by using ephemeris data for each of the participating satellites. . The circular and elliptical satellite models are both used to locate satellites within the ECEF coordinate frame at specified times. It is also essential to convert the master and self coordinates into the ECEF coordinate frame. Both satellite and platform coordinates in ECEF are computed, and then both are transformed into Local Level coordinates and difference vector DIFF is generated. The DIFF vector is used to compute the elevation of the satellites to determine and eliminate the satellites that do not achieve at least five degrees of elevation from filter consideration. The direction cosine shall be generated as functions of member platform positions and the ECEF positions of the satellites, as computed via satellite ephemeris data.

The RF_SOURCE_DATA PROC generates a Kalman Observation Set Package from the satellites or from PPLI/RTT to output to the RF_KALMAN_PROC CSC.

The data included in this package is the following:

- HA (12, 8) – Observation Matrix
- YA (INDEX) – Observation Measurement Vector
- YAEXP (INDEX) – Observation Predicted Vector
- SATVIS (j) – Satellite Visibility Vector
- DXTRUE, DYTRUE, DZTRUE – True Navigation Error Composites
- BCTRUE – True Clock Bias
- FCTRUE – True Frequency Drift

The generation of the package is created by performing a defined set of computations and data transfers which is described in detail in the SRS. The Kalman Observation Set Package includes the same data for all versions however the computations of these values differ according to the version being utilized. Figures 4.4-15 and 4.4-16 represent a functional flow chart for RF_SOURCE_DATA_PROC.

INPUTS	DESCRIPTION
MNAV_DATA	
MNAVTOV	Time of Validity of MASTER Navigation solution
MLAT	ESTIMATED MASTER Latitude position [rad]
MLATR	ESTIMATED MASTER Latitude rate [rad/s]
MLON	ESTIMATED MASTER Longitude position [rad]
MLONR	ESTIMATED MASTER Longitude rate [rad/s]
MALT	ESTIMATED MASTER Altitude [m]
MALTR	ESTIMATED MASTER Altitude rate [m/s]
TMLAT*	TRUE MASTER Latitude position [rad]
TMLATR*	TRUE MASTER Latitude rate [rad/s]
TMLON*	TRUE MASTER Longitude position [rad]
TMLONR*	TRUE MASTER Longitude rate [rad/s]
TMALT*	TRUE MASTER Altitude [m]
TMALTR*	TRUE MASTER Altitude rate [m/s]
MSAT(12)**	MASTER Ordered Satellite List
MPRR(12)**	MASTER Ordered measured code phase array
MGPSTOV**	MASTER predicted GPS time of validity
MVIS**	Number of satellites visible to MASTER
BCTRUE	TRUE MASTER Clock Bias
FCTRUE	TRUE MASTER Frequency Drift
SNAV_DATA	
SNAVTOV	Time of Validity of SELF Navigation solution
SLAT	ESTIMATED SELF Latitude position [rad]
SLATR	ESTIMATED SELF Latitude rate [rad/s]
SLON	ESTIMATED SELF Longitude position [rad]
SLONR	ESTIMATED SELF Longitude rate [rad/s]
SALT	ESTIMATED SELF Altitude [m]
SALTR	ESTIMATED SELF Altitude rate [m/s]
TSLAT*	TRUE SELF Latitude position [rad]
TSLATR*	TRUE SELF Latitude rate [rad/s]
TSLON*	TRUE SELF Longitude position [rad]
TSLONR*	TRUE SELF Longitude rate [rad/s]
TSALT*	TRUE SELF Altitude [m]
TSALTR*	TRUE SELF Altitude rate [m/s]
SSAT(12)**	SELF Ordered Satellite List
SPRR(12)**	SELF Ordered measured code phase array
SGPSTOV**	SELF predicted GPS time of validity
SVIS**	Number of satellites visible to SELF
BCTRUE	TRUE SELF Clock Bias
FCTRUE	TRUE SELF Frequency Drift
INVERS(30)**	Satellite sorting index
EPHEM(30,23)**	Satellite Ephemeris Data
CSCI_VERS_IND	CSCI Version Indicator
NVS	Number of Visible Satellites
RF_STATE	State the Refinement Filter resides

Table 4.4-8: RF_SOURCE_DATA_PROC Inputs

* - Provided only if CSCI_VERS_IND = 2 (Version 2.0)
** - Provided only if CSCI_VERS_IND = 3 (Version 3.0)

OUTPUTS	DESCRIPTION
MNAV_DATA	
MNAVTOV	Time of Validity of MASTER Navigation solution
MLAT	ESTIMATED MASTER Latitude position [rad]
MLATR	ESTIMATED MASTER Latitude rate [rad/s]
MLON	ESTIMATED MASTER Longitude position [rad]
MLONR	ESTIMATED MASTER Longitude rate [rad/s]
MALT	ESTIMATED MASTER Altitude [m]
MALTR	ESTIMATED MASTER Altitude rate [m/s]
TMLAT*	TRUE MASTER Latitude position [rad]
TMLATR*	TRUE MASTER Latitude rate [rad/s]
TMLON*	TRUE MASTER Longitude position [rad]
TMLONR*	TRUE MASTER Longitude rate [rad/s]
TMALT*	TRUE MASTER Altitude [m]
TMALTR*	TRUE MASTER Altitude rate [m/s]
MSAT(12)**	MASTER Ordered Satellite List
MPRR(12)**	MASTER Ordered measured code phase array
MGPSTOV**	MASTER predicted GPS time of validity
MVIS**	Number of satellites visible to MASTER
BCTRUE	TRUE MASTER Clock Bias
FCTRUE	TRUE MASTER Frequency Drift
SNAV_DATA	
SNAVTOV	Time of Validity of SELF Navigation solution
SLAT	ESTIMATED SELF Latitude position [rad]
SLATR	ESTIMATED SELF Latitude rate [rad/s]
SLON	ESTIMATED SELF Longitude position [rad]
SLONR	ESTIMATED SELF Longitude rate [rad/s]
SALT	ESTIMATED SELF Altitude [m]
SALTR	ESTIMATED SELF Altitude rate [m/s]
TSLAT*	TRUE SELF Latitude position [rad]
TSLATR*	TRUE SELF Latitude rate [rad/s]
TSLON*	TRUE SELF Longitude position [rad]
TSLONR*	TRUE SELF Longitude rate [rad/s]
TSALT*	TRUE SELF Altitude [m]
TSALTR*	TRUE SELF Altitude rate [m/s]
SSAT(12)**	SELF Ordered Satellite List
SPRR(12)**	SELF Ordered measured code phase array
SGPSTOV**	SELF predicted GPS time of validity
SVIS**	Number of satellites visible to SELF
BCTRUE	TRUE SELF Clock Bias
FCTRUE	TRUE SELF Frequency Drift
INVERS(30)**	Satellite sorting index
EPHEM(30,23)**	Satellite Ephemeris Data
CSCI_VERS_IND	
NVS	
RF_STATE	

Table 4.4-9: RF_SOURCE_DATA_PROC Outputs

* - Provided only if CSCI_VERS_IND = 2 (Version 2.0)
** - Provided only if CSCI_VERS_IND = 3 (Version 3.0)

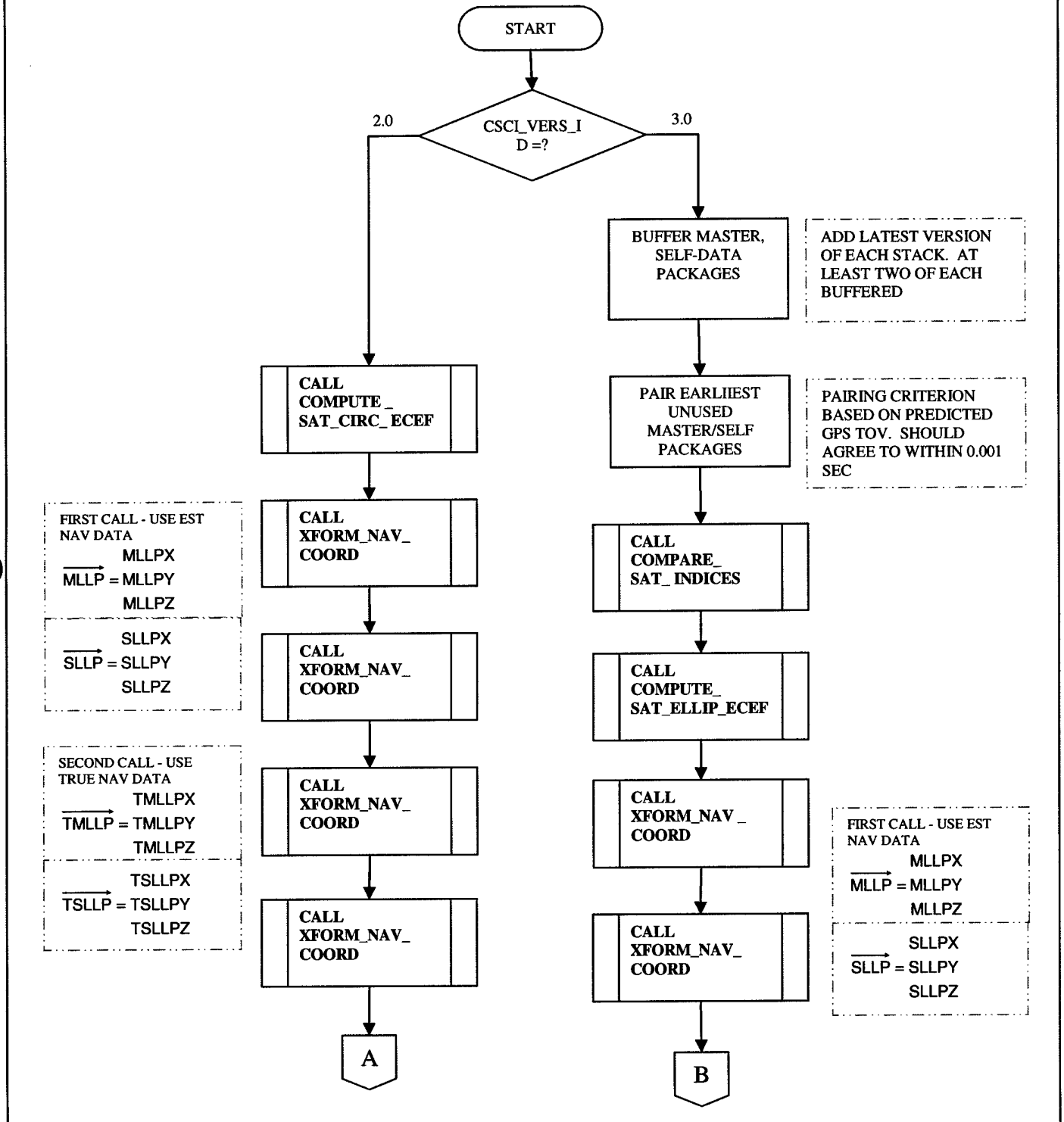


Figure 4.4-15: RF_SOURCE_DATA_EXEC Logical data flow diagram

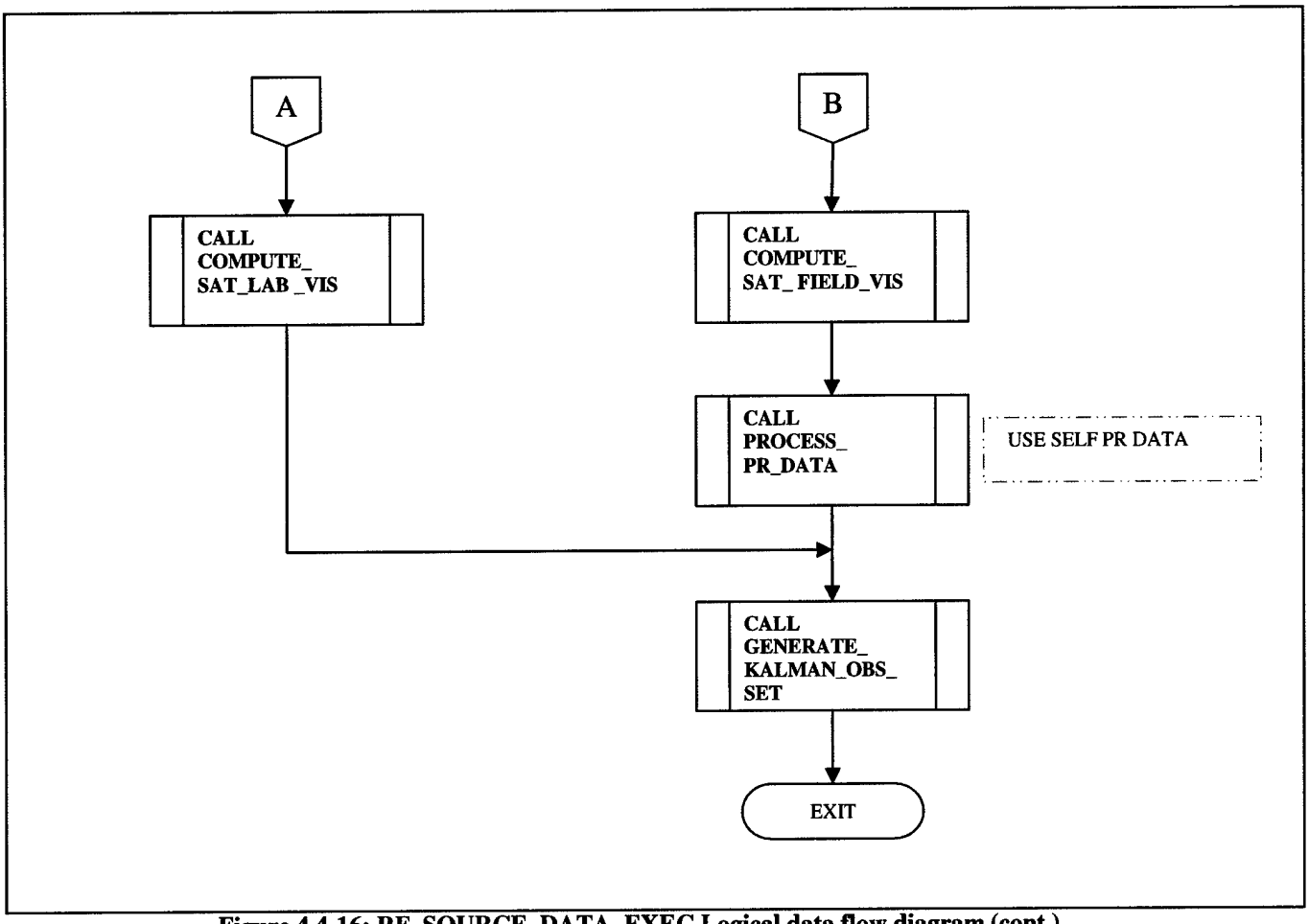


Figure 4.4-16: RF_SOURCE_DATA_EXEC Logical data flow diagram (cont.)

4.4.3 RF_KALMAN_PROC

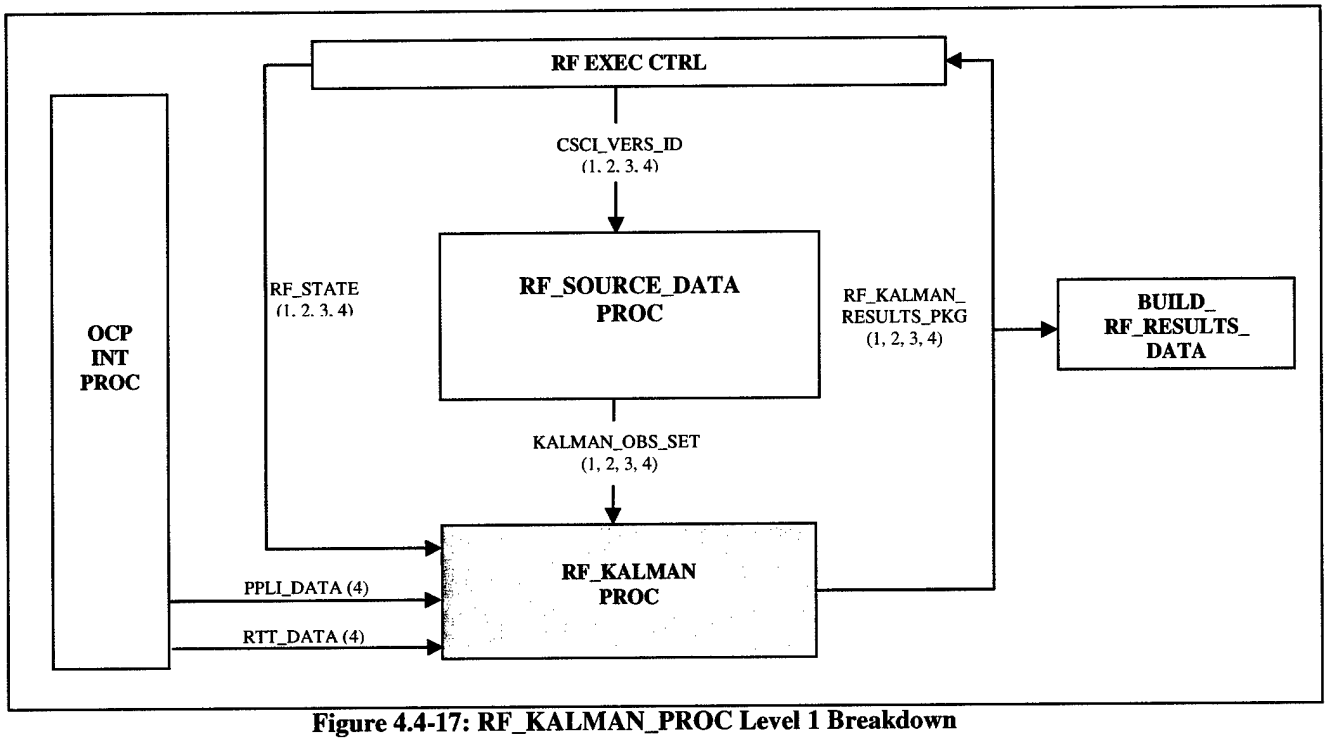


Figure 4.4-17: RF_KALMAN_PROC Level 1 Breakdown

The RF_KALMAN_PROC performs the Extended Kalman Filter (EKF) processing needed to estimate the numeric values of the eight Refinement Filter Error terms defined as follows:

- XA(1): LL x-coordinate relative position error [m]
- XA(2): LL x-coordinate relative position error rate [m/s]
- XA(3): LL y-coordinate relative position error [m]
- XA(4): LL y-coordinate relative position error rate [m/s]
- XA(5): LL z-coordinate relative position error [m]
- XA(6): LL z-coordinate relative position error rate [m/s]
- XA(7): Relative clock bias [ns]
- XA(8): Relative frequency drift [ns/s]

The observation set required for the refinement process is generated by RF_SOURCE_DATA_PROC. This package contains:

- Observation matrix
- Final number of paired satellites visible above 5° elevation
- Observation measurement vector
- Observation prediction vector
- Reference time at which measurements are valid since GPS time of week [s]
- Ordered array of common visible satellites above 5° elevation
- True error composites
- True clock bias
- True frequency drift.

The results from the Kalman filtering consists of a data package with the following information:

- Observation set time of validity
- RF State
- Refinement Filter Status Indicator
- Refinement Filter Solution Vector
- Error covariance
- Error

As part of the operation of the Kalman Filter, RF_KALMAN_PROC shall also perform the initialization of the EKF matrix elements.

CSC INPUTS	DESCRIPTION
KALMAN_OBS_SET	
HA(12,8)	Observation matrix
SCOUNT	Final number of paired satellites visible above ANGLIM
YA(I), I = 1, 14	Observation measurement vector
YAEXP(I), I = 1, 14	Observation prediction vector
TC, TOLD	Reference time at which measurements are valid since GPS time of week [s]
SATVIS(j), j = 1, SCOUNT	Ordered array of common visible satellites above ANGLIM
DXTRUE DYTRUE DZTRUE	True XYZ direction error composites. Only used when CSCI_VERS_ID = 2.0 (Laboratory Mode)
BCTRUE	True clock bias
FCTRUE	True frequency drift.
PPLI_DATA RTT DATA	Used only when the RF State is in Backup Mode (when GPS is either not on or valid). Contains pseudo range data
RF_STATE	Indicates the state value of the refinement filter
RSI	Refinement Status Indicator

Table 4.4-10: RF_KALMAN_PROC Inputs

CSC OUTPUTS	DESCRIPTION
RF_KALMAN_RESULTS_PKG	
TC	Observation set time of validity
RF_STATE	Indicates the state value of the refinement filter
RSI	Refinement Filter Status Indicator
XA(k), k = 1, 8	Refinement Filter Solution Vector
SIGX	COVARIANCE DIAGONAL TERMS
SIGY	
SIGZ	
SIGBC	
SIGFC	
ERRX	ERROR TERMS
ERRY	
ERRZ	
ERRBC	
ERRFC	
SCOUNT	Final number of paired satellites visible above ANGLIM
SATVIS(j), j = 1, SCOUNT	Ordered array of common visible satellites above ANGLIM

Table 4.4-11: RF_KALMAN_PROC Outputs

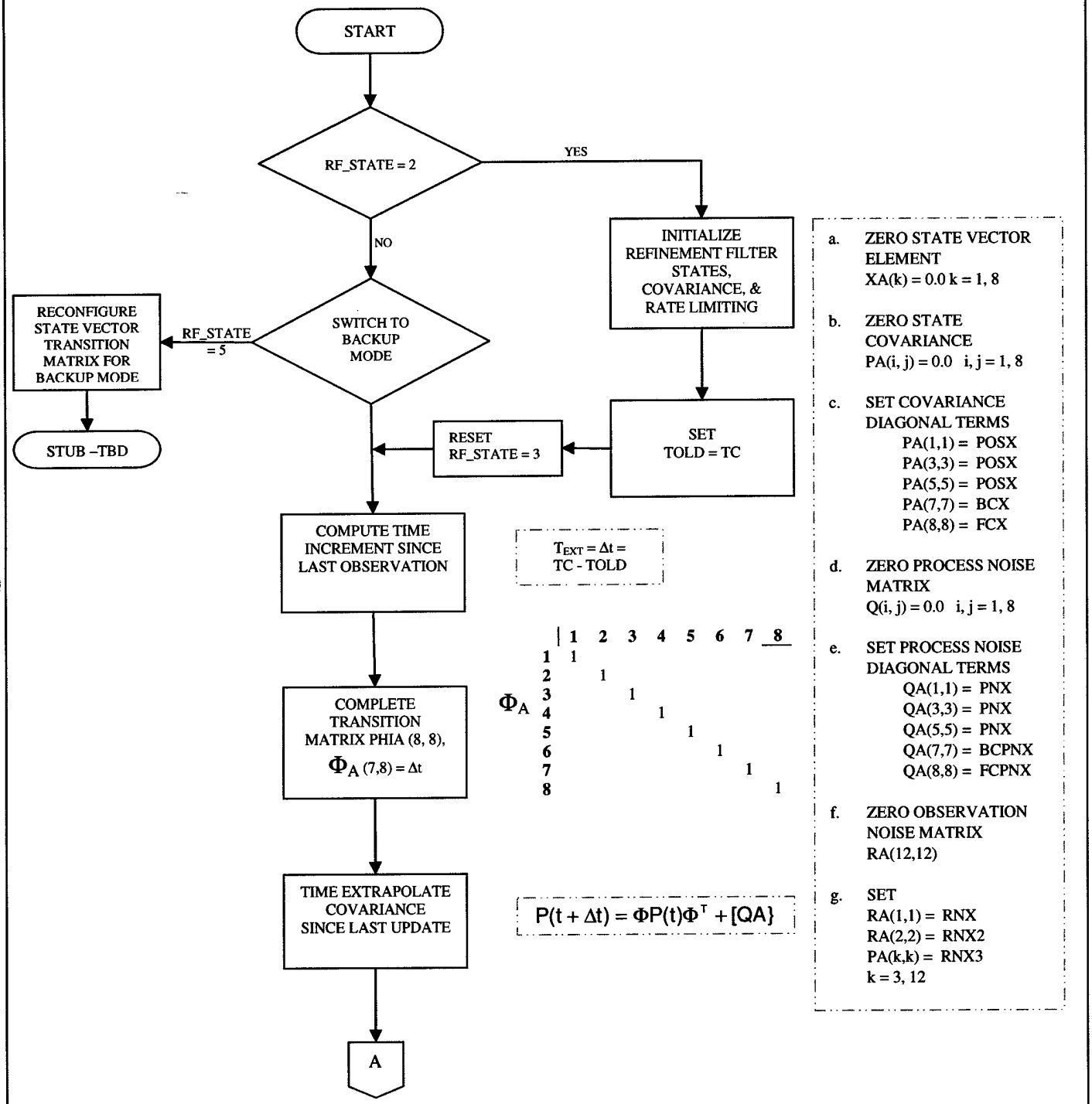


Figure 4.4-18: RF_KALMAN_PROC Logical data flow diagram

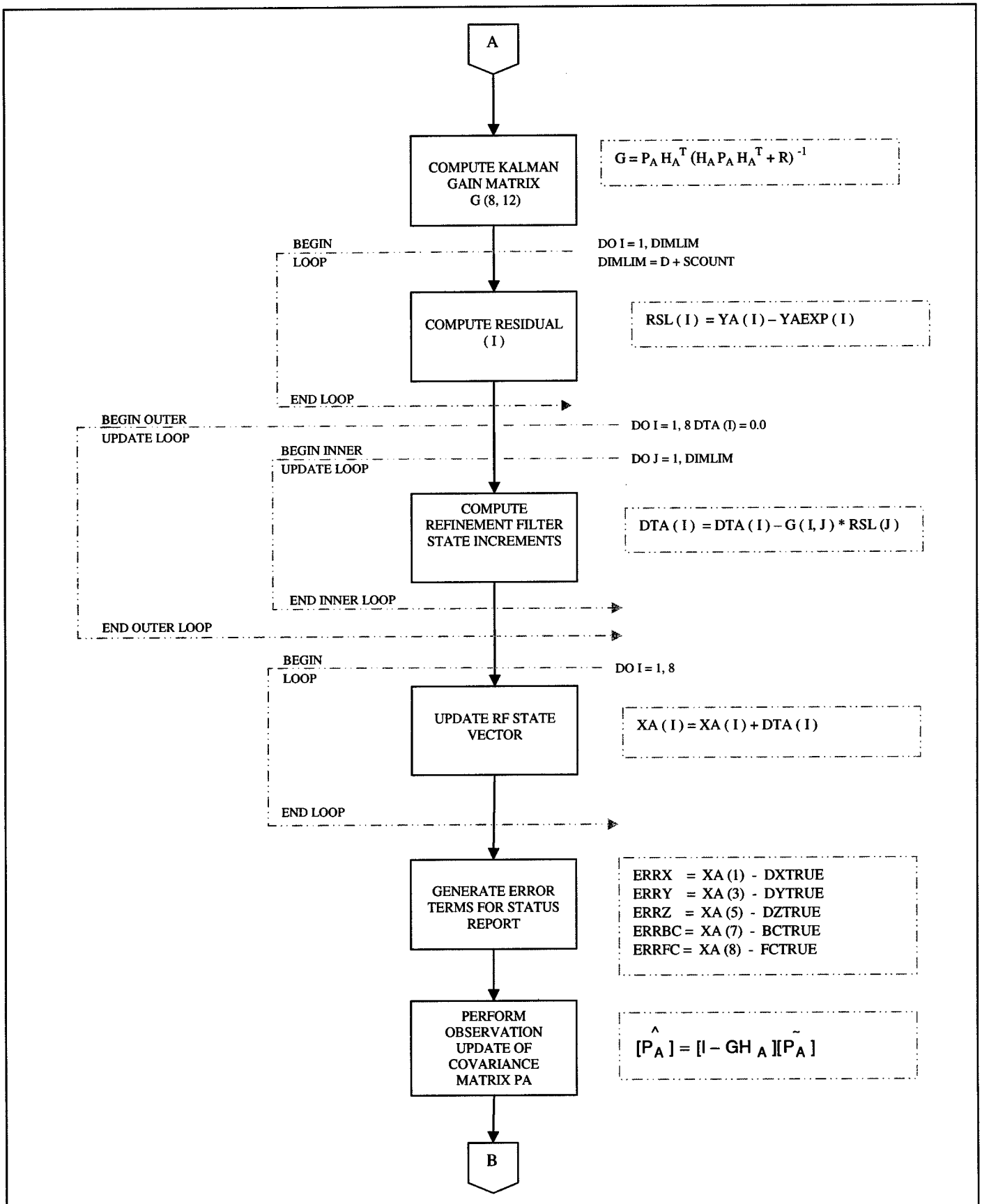


Figure 4.4-19: RF_KALMAN_PROC Logical data flow diagram (cont.)

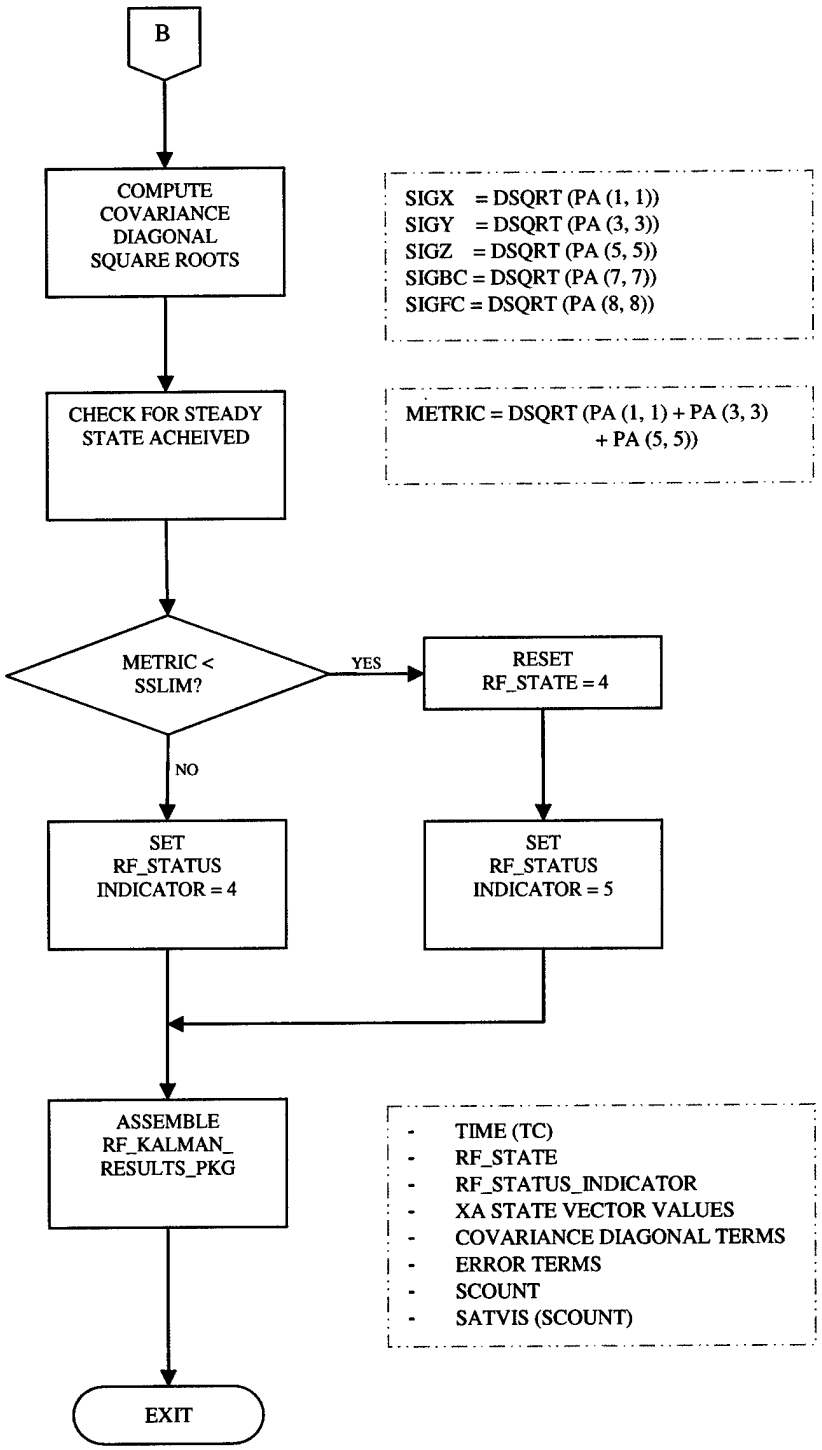


Figure 4.4-20: RF_KALMAN_PROC Logical data flow diagram (cont.)

4.4.4 OCP_INT_PROC

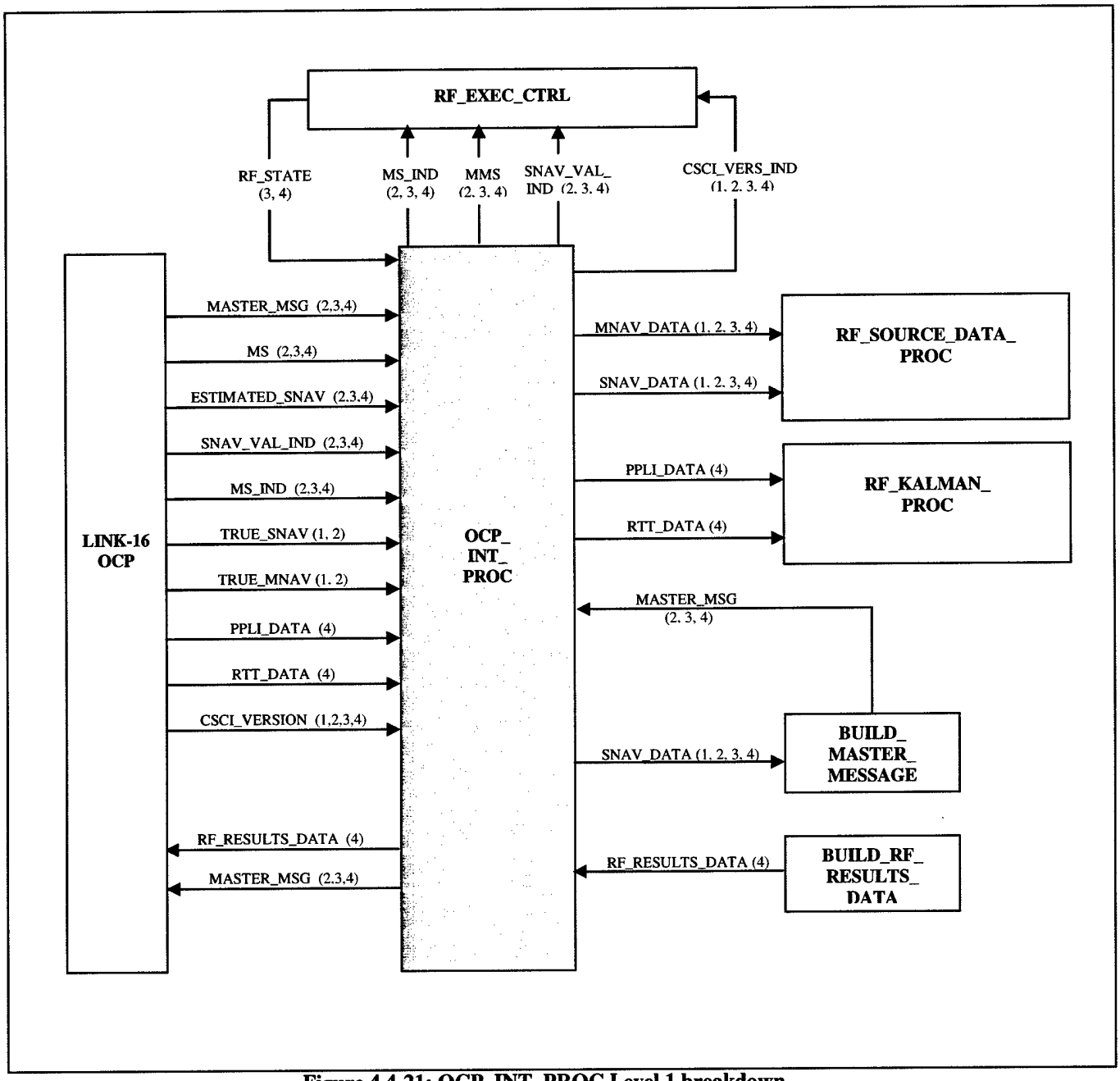


Figure 4.4-21: OCP_INT_PROC Level 1 breakdown

The OCP interface Processing provides a two-way interface between the Link-16 Operational Software and DGPS RF processing. This interface is responsible for the following general functions:

(a) Decomposition of received Master Message to extract and format master navigation data and measured GPS pseudorange information, both of which are required within the DGPS RF algorithm.

(b) Rescaling of our navigation data from English to Metric units for use within the DGPS RF algorithm.

The Master Message is a TADIL-J message received via normal L16 message processing. Our navigation data is derived from either L16 navigation solution (Versions 3.0, 4.0) or from LNS simulator data (Versions 1.0, 2.0).

If platform is designated as "Master", then the algorithm – generated message that is submitted for L16 transmitter by OCP_INT_PROC. In projected Version 4.0, RF results are provided to L16 OCP for inclusion in atmospheric delay compensation processing.

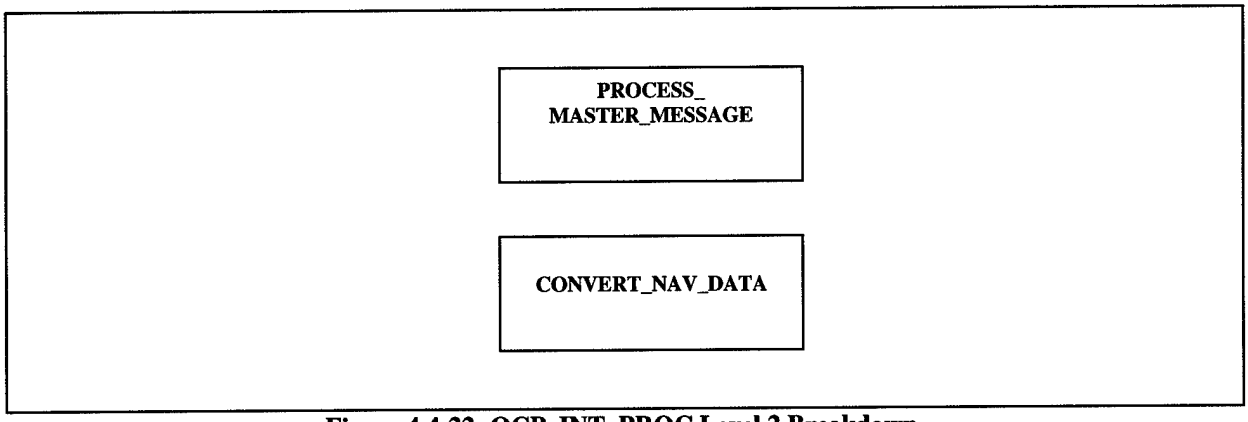


Figure 4.4-22: OCP_INT_PROC Level 2 Breakdown

SLCSC	Description
PROCESS_MASTER_MESSAGE	Decomposes the received Master Message Data into Master NAV data and GPS data
CONVERT_NAV_DATA	Converts L16 Nav data into metric units

Table 4.4-12: OCP_INT_PROC SLCSC Descriptions

CSC Inputs	Description
CSCI_VERS_IND	Denotes Operational Version
RF_STATE	Indicates the state value of the refinement filter
MASTER_MSG_	This message contains Master's nav solution master's GPS data
PPLI_DATA RTT_DATA	Used only when the RF State is in Backup Mode (when GPS is either not on or valid). Contains pseudo range data
EST_SNAV	ESTIMATED SELF Navigation Data
TRUE_SNAV_	TRUE SELF Navigation Data
TRUE_MNAV	TRUE MASTER Navigation Data
SNAV_VAL_IND	Self Navigation Validity Indicator
MNAV_VAL_IND	Master Navigation Validity Indicator
MS_IND	Master/Slave Indicator
MMS	Master Message Prepared Signal
RF_RESULTS_DATA	Refinement Filter corrections and covariance to send to AF (used when CSCI = 4, Backup Mode)

Table 4.4-13: OCP_INT_PROC Inputs

CSC Outputs	Description
PPLI_DATA RTT_DATA	Used only when the RF State is in Backup Mode (when GPS is either not on or valid). Contains pseudo range data
MNAV_DATA	MASTER Navigation Solution
SNAV_DATA	SELF Navigation Solution
RF_RESULTS_MSG	Refinement Filter corrections and covariance to send to AF
MASTER_MSG	This message contains Master's nav solution Master's GPS data
MS_IND	Master/Slave Indicator
MMS	Master Message Prepared Signal
SNAV_VAL_IND	Self Navigation Validity Indicator
MNAV_VAL_IND	Master Navigation Validity Indicator
CSCI_VERS_IND	Denotes Operational Version

Table 4.4-14: OCP_INT_PROC Outputs

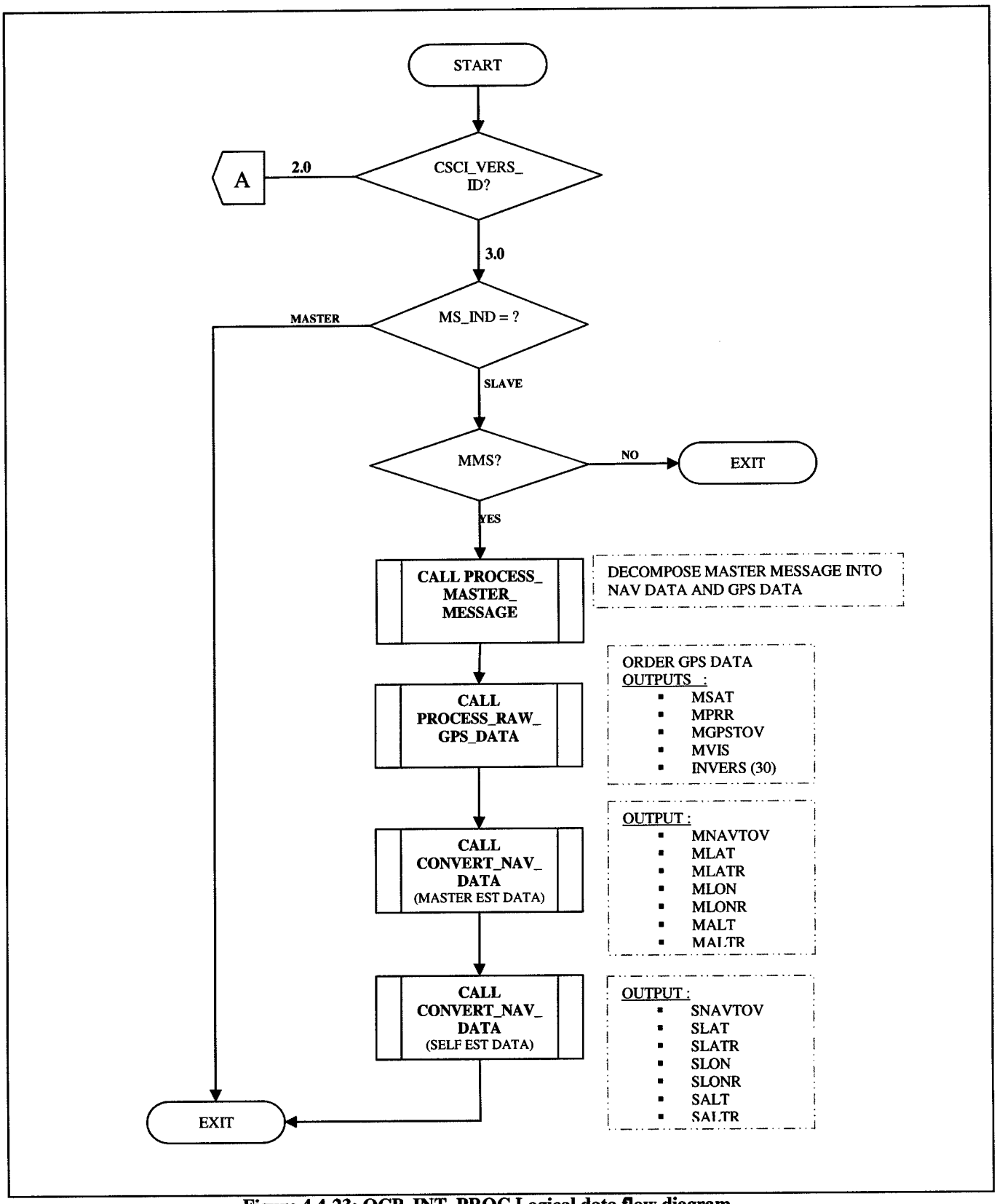


Figure 4.4-23: OCP_INT_PROC Logical data flow diagram

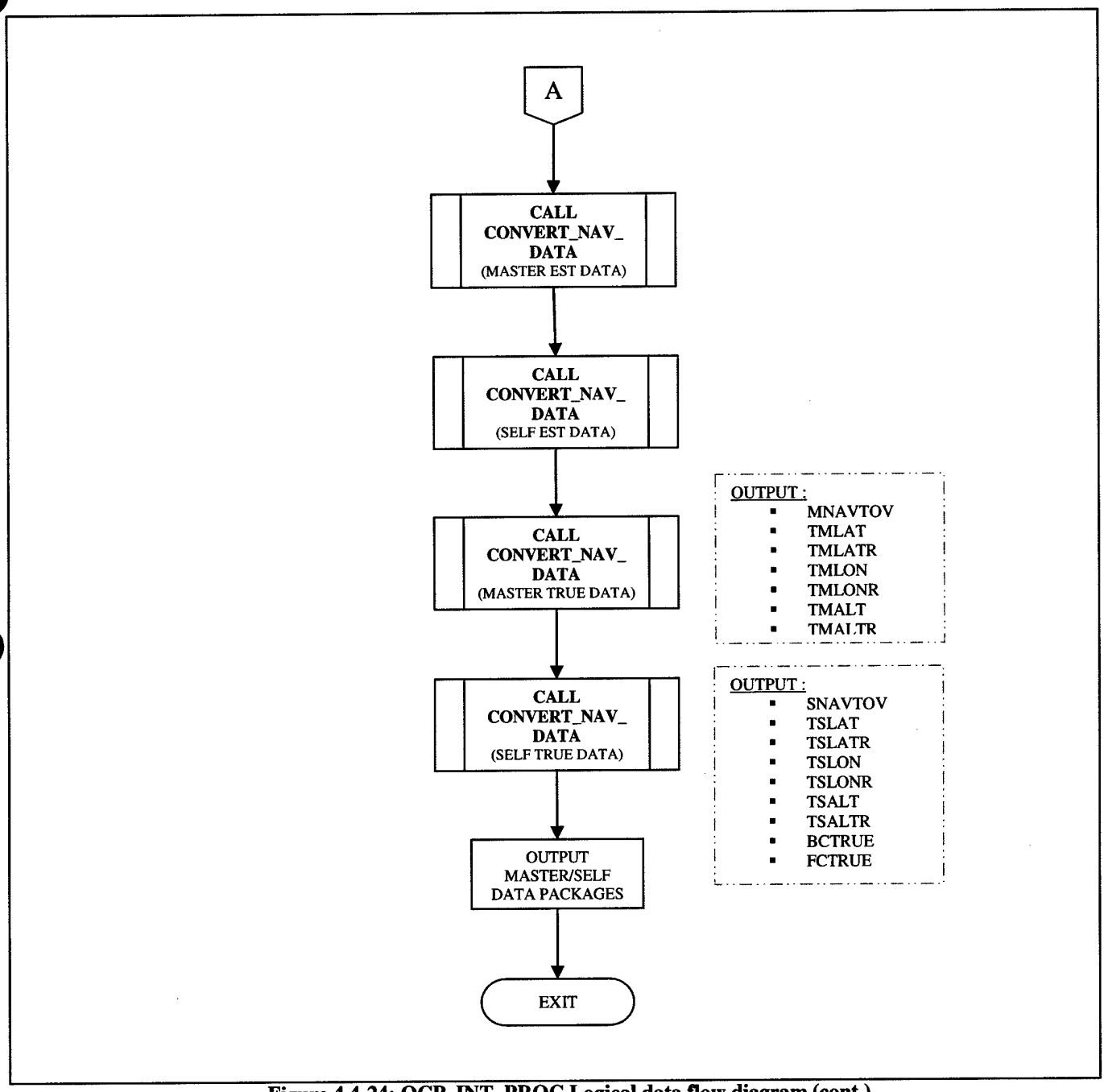


Figure 4.4-24: OCP_INT_PROC Logical data flow diagram (cont.)

4.4.5 HOST_INT_PROC

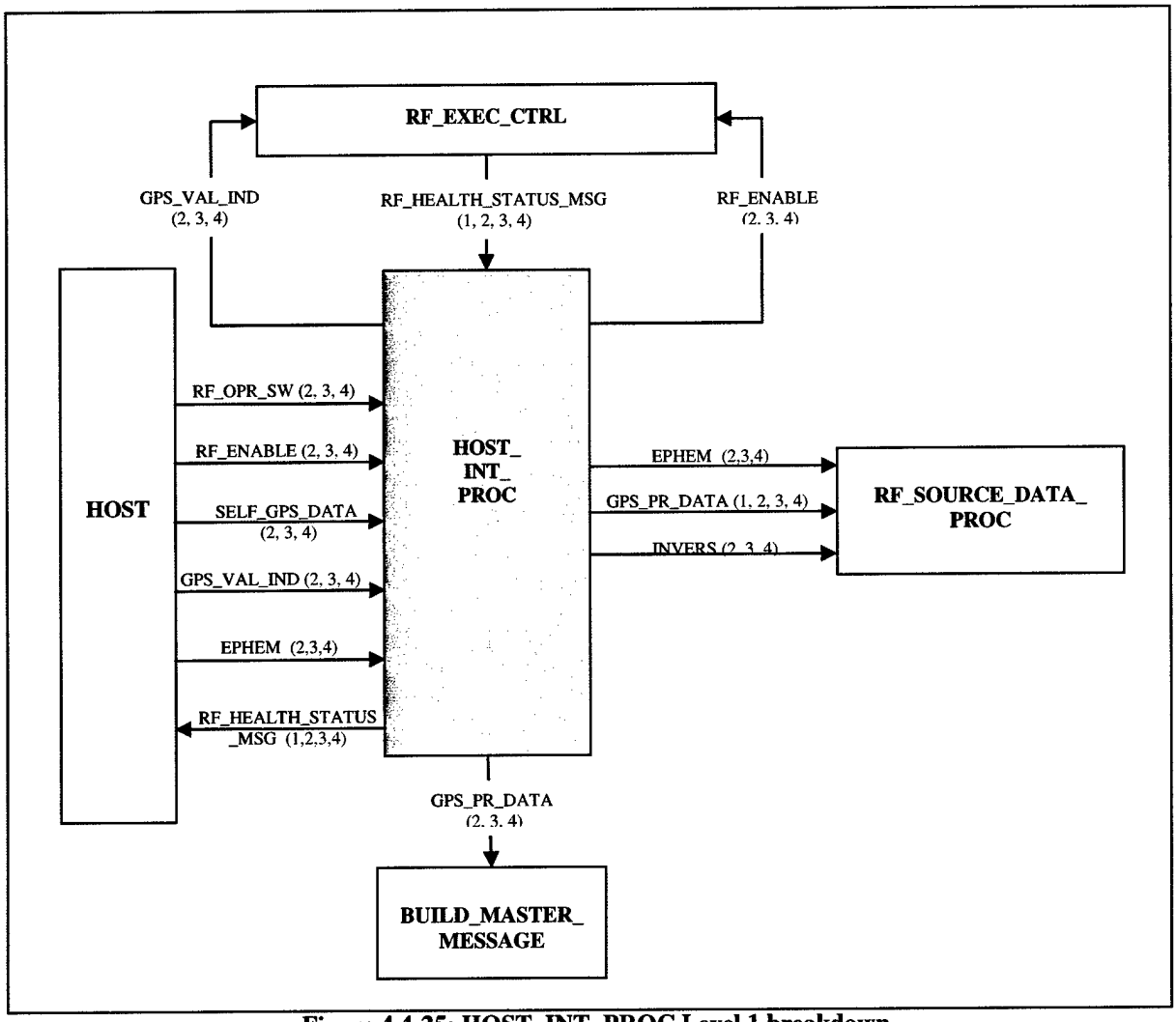


Figure 4.4-25: HOST_INT_PROC Level 1 breakdown

The host Interface Processing function serves as the primary two way data path between the DGPS RF algorithm and the host processor. With respect to input from the host, it performs the following functions:

(a) Interprets and reformats DGPS RF control inputs provided by the host, namely:
- RF_ENABLE (Operate/Non-operate switch)
- RF_OPR_SW (Operator command algorithm reset indicator)

(b) Interprets and reformats raw input provided by the GPS receiver, including GPS validity indicator, Self-GPS pseudorange measurements, and ephemeris data.

With respect to host output, HOST_INT_PROC is responsible for providing an RF Health and Status Message, which includes the full filter state estimates and co variances plus RF State and Status Indicators.

CSC inputs and outputs are summarized, respectively, in Tables 4.4-15 and 4.4-16. Two SLCSCs are utilized within HOST_INT_PROC. Their functions are described in Table 4.4-17.

CSC Inputs	Version	Description
RF_OPR_SW	1,2,3,4	Operator Reset Command
RF_ENABLE	2,3,4	RF Enable Command
SELF_GPS_DATA_	2,3,4	Pseudorange and satellite ID message data
GPS_VAL_IND	2,3,4	GPS Validity Indicator
EPHEM_RAW	2,3,4	Raw Ephemeris Data message
RF_HEALTH_STATUS_MSG	2,3,4	Summary of GPS results and status

Table 4.4-15: HOST_INT_PROC Input Variables

CSC Outputs	Version	Description
RF_HEALTH_STATUS_MSG	2,3,4	Summary of GPS results and status

Table 4.4-16: HOST_INT_PROC Output Variables

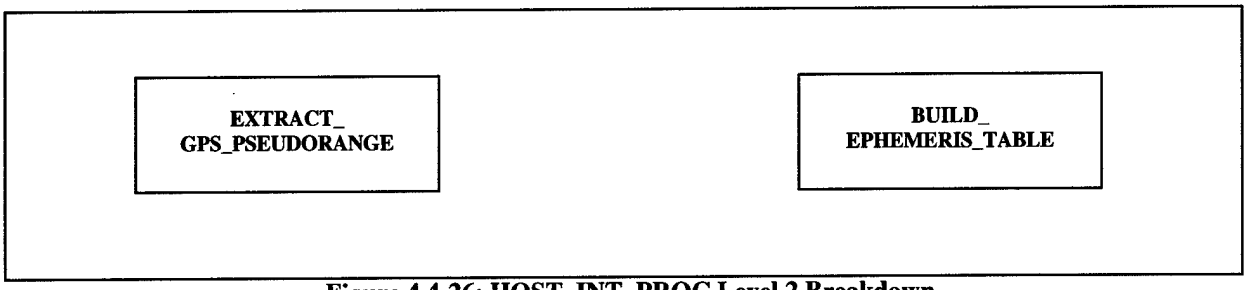


Figure 4.4-26: HOST_INT_PROC Level 2 Breakdown

SLCSC	Description
EXTRACT_GPS_PSEUDORANGE	Decomposes GPS receiver data to provide an ordered array of satellite ID's and code phase data.
BUILD_EPHEMERIS_TABLE	Reformats GPS receiver ephemeris message to provide standard ephemeris array.

Table 4.4-17: HOST_INT_PROC SLCSC Descriptions

4.4.6 Build_Master_Message

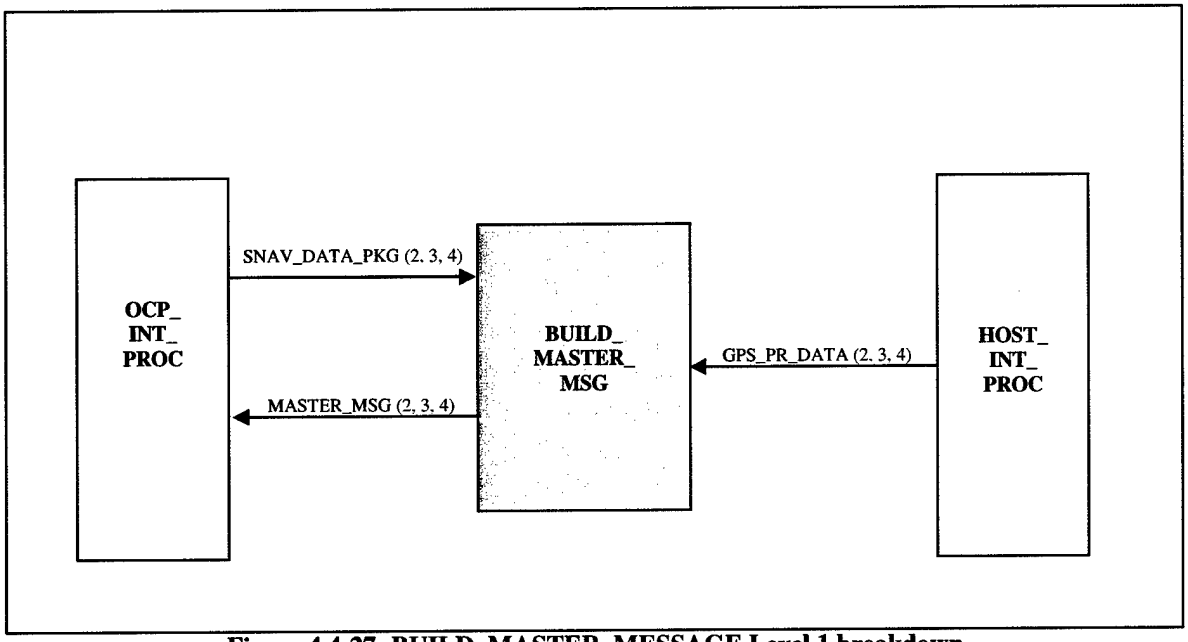


Figure 4.4-27: BUILD_MASTER_MESSAGE Level 1 breakdown

The purpose of this function is to construct outgoing Master Messages containing self navigation and measured pseudorange data collected at own location. This function shall be invoked only when own platform has been designated as Master.

The contents and format of the J28.7 Master Message are presented in paragraph 4.5.4.

4.4.7 BUILD_RF_RESULTS_DATA

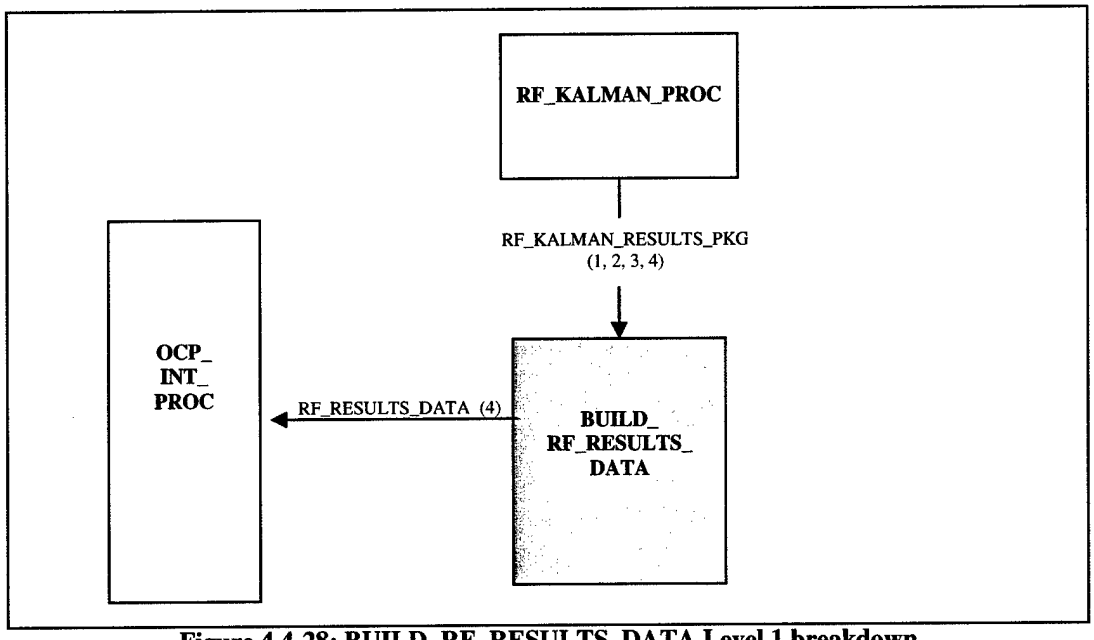


Figure 4.4-28: BUILD_RF_RESULTS_DATA Level 1 breakdown

This function is intended as a place holder for a potential enhanced Version 4.0 of the DGPS RF algorithm. This version of the software would combine the functionality of the DGPS RF with that of an Atmospheric Calibration Filter (AF), now being developed under separate contract. The basic concept of Version 4.0 is to improve basic Link-16 navigation performance via advanced atmospheric calibration of TOA range measurements. The Atmospheric Filter utilizes the position estimates of the transmitter and receiver of a PPLI message as basic input. The DGPS RF has demonstrated the ability to reduce errors in position estimates to extremely small levels. These precise DGPS RF position estimates can generate a corrective term that will reduce the range estimate error of the AF, if applied to the existing Link-16 navigation solution.

The purpose of Build_RF_Results_Data is to package the error estimates of the DGPS RF Kalman Filter for transfer to the Link-16 OCP. Note that this function exists only for the Enhanced Version 4.0 of the DGPS algorithm.

4.5 Laboratory Testing Requirements for DGPS RF Algorithm-TATS and Link16 OCP Design Modifications

The FY04 plans for the DGPS RF project provide for laboratory testing of a fully integrated version of the algorithm within the Link-16 Operational Computer Program within an actual terminal. The test bed which shall be utilized for this activity is BAE SYSTEMS Terminal ATP Test Set (TATS), which has served as a production test facility in several versions over the past twenty years. The FY04 schedule, beginning in October 2003 is nine months in duration, which is relatively short for the tasks at which require significant modifications to both the TATS and Link-16 operational software. In order to provide a level of risk reduction for this activity, advanced design and planning for both areas was scheduled as part of the FY03 activities. The purpose of this section is to summarize the work performed to date in preparation for the FY04 activities.

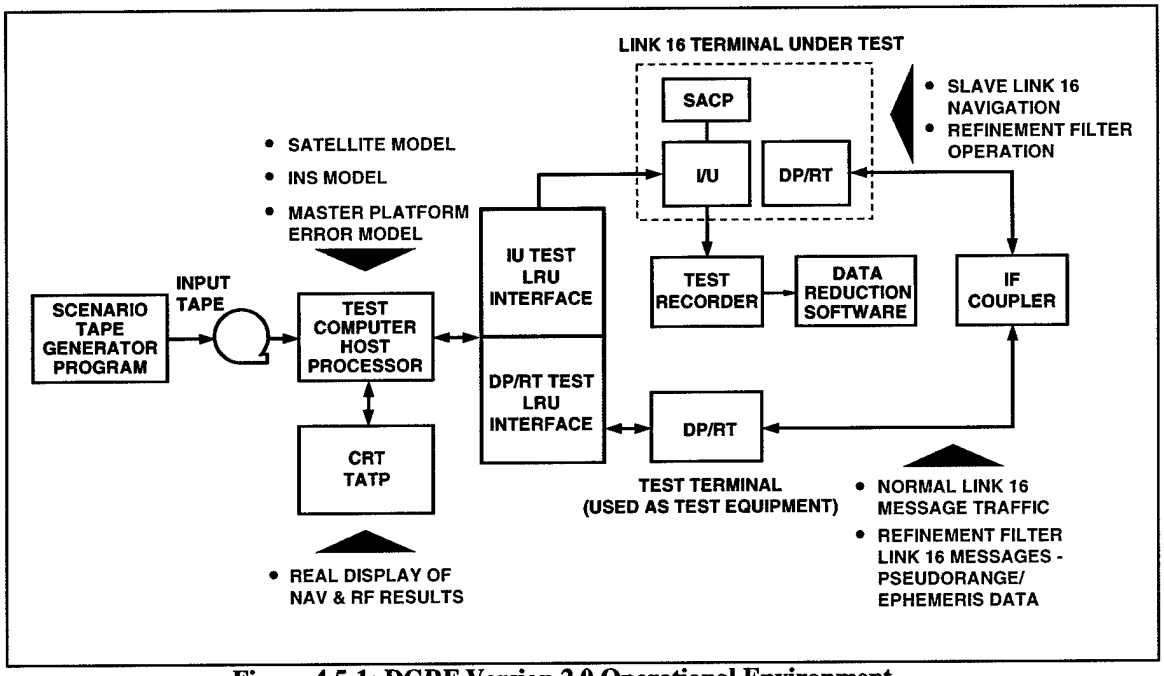


Figure 4.5-1: DGRF Version 2.0 Operational Environment

The operational environment for the planned laboratory testing is centered about the Terminal ATP Test Set – TATS (Figure 4.5 – 1), on which the modified terminal will be exercised. As always, the task of the TATS will be to replicate at both RF and host terminal interfaces, exactly the data transfer in both format and medium that the terminal requires for operation. In addition to the currently operational Link-16 host and RF interfaces, the TATS must be adapted to provide the following capabilities required for DGPS RF operation that do not currently exist:

(1) Replication of the output of a functional GPS receiver, including simulated pseudorange measurements and ephemeris data characteristic of a constellation of

GPS satellites in elliptical orbit, and provided to the terminal via the host 1553 interface.

(2) Generation of Master Message traffic provided via the terminal RF interface which incorporates master navigation data and simulated pseudorange data measured at the master's true position.

(3) Additional input data representing DGPS operator control commands, and new DGPS RF output data and status needed for data reduction and analysis

The new operational DGPS RF source code will be embedded within the Link-16 operational software, and will need to interface with specific portions of that code to obtain data required for its operation. The general requirements listed above may be broken down into subtasks for both the TATS Operational Software and the Link-16 OCP, which are presented below:

Test Set (TATS) Software Required Modifications:

(a) Inclusion of three new operator control inputs which invoke the operational modes of the DGPS algorithm: (1) RF_Enable , which enables operation of the DGPS RF Algorithm, (2) RF_OPR_SW, which represents an operator-derived input commanding algorithm reinitialization, (3) MS_IND, the Master/Slave Designator. All inputs to be provided via the host 1553 MUX input at times specified via initialization data.

(b) Inclusion of a test cycle parameter DGPS_KAL_CYCLE, which will specify the Kalman Cycle rate of the DGPS algorithm. This value may take on the values of 1,2, or 4 Hz, to be provided via the host 1553 MUX input. This input to be required at startup only.

(c) Inclusion of a (precomputed) real matrix of dimension 30 x 23 containing the satellite constellation ephemeris data (EPHEM). This data to be provided via the host 1553 MUX input at startup only.

(d) Inclusion of the GPS satellite pseudorange data (SELF_GPS_DATA) representative of the measurements taken at the position of the slave (self) at measurement times. This data to be provided via the host 1553 MUX input at the same rate specified in (b). The measured GPS pseudorange data shall be precomputed, and shall be provided as a data file to the TATS for application as above.

(e) Inclusion of the Master Reporting Message at the rate selected in (b) and transmitted by the Test Terminal to the TUT as a TADIL-J message. This message shall contain the master platform position and recorded GPS pseudorange block data. These messages shall be precomputed and stored within the TATS for transmission at the appropriate transmission times.

(f) Generation of true master and slave position solutions, which are transferred to the TUT via the 1553 MUX input. Note that this data will be utilized only by Version 2.0 of the DGPS RF software for laboratory usage only, and will be used for post-

test data reduction only. This data shall be provided at a rate of once per 250 milliseconds.

Link-16 OCP Required Modifications

The Link-16 OCP shall require the following additional functionality to support the implementation and testing of the DGPS RF operation:

(a) Transfer of the self-derived GPS pseudorange data and ephemeris data read via 1553 MUX to the Host Interface Processing function.

(b) Transfer of the true self and master navigation data blocks (defined in paragraph 4.4.2) to OCP Interface Processing function

(c) Transfer of DGPS RF Control Data, provided via 1553 MUX to OCP Interface Processing.

(d) Extraction of own self-navigation data, computed within Link-16 OCP to OCP Interface Processing.

(e) Acceptance of RF Performance Data from Host Interface Processing for transfer to host via 1553 MUX output.

(f) Acceptance of DGPS RF core-generated Master Message for transmission via Link-16. (Done only when designated as master platform).

Specific details of each of these requirements for both the TATS and Link-16 OCP are presented in subsequent paragraphs.

4.5.1 DGPS RF Control Inputs

The satisfaction of the requirements specified in (a) and (b) for the TATS –provided input data variables RF_ENABLE, RF_OPER_SW, MS_IND, and DGPS_KAL_CYCLE requires the provision of a DGPS_Control_Word within the 1553 MUX whose fields control the inputs values of these variables. This word shall be provided at each MUX cycle (50 ms). The word image of the DGPS_Control_Word shall be as depicted in Figure 4.5 – 2. Note that only the initial values of MS_IND and DGPS_KAL_CYCLE shall be utilized by the program. Any subsequent changes to these variables shall be ignored.

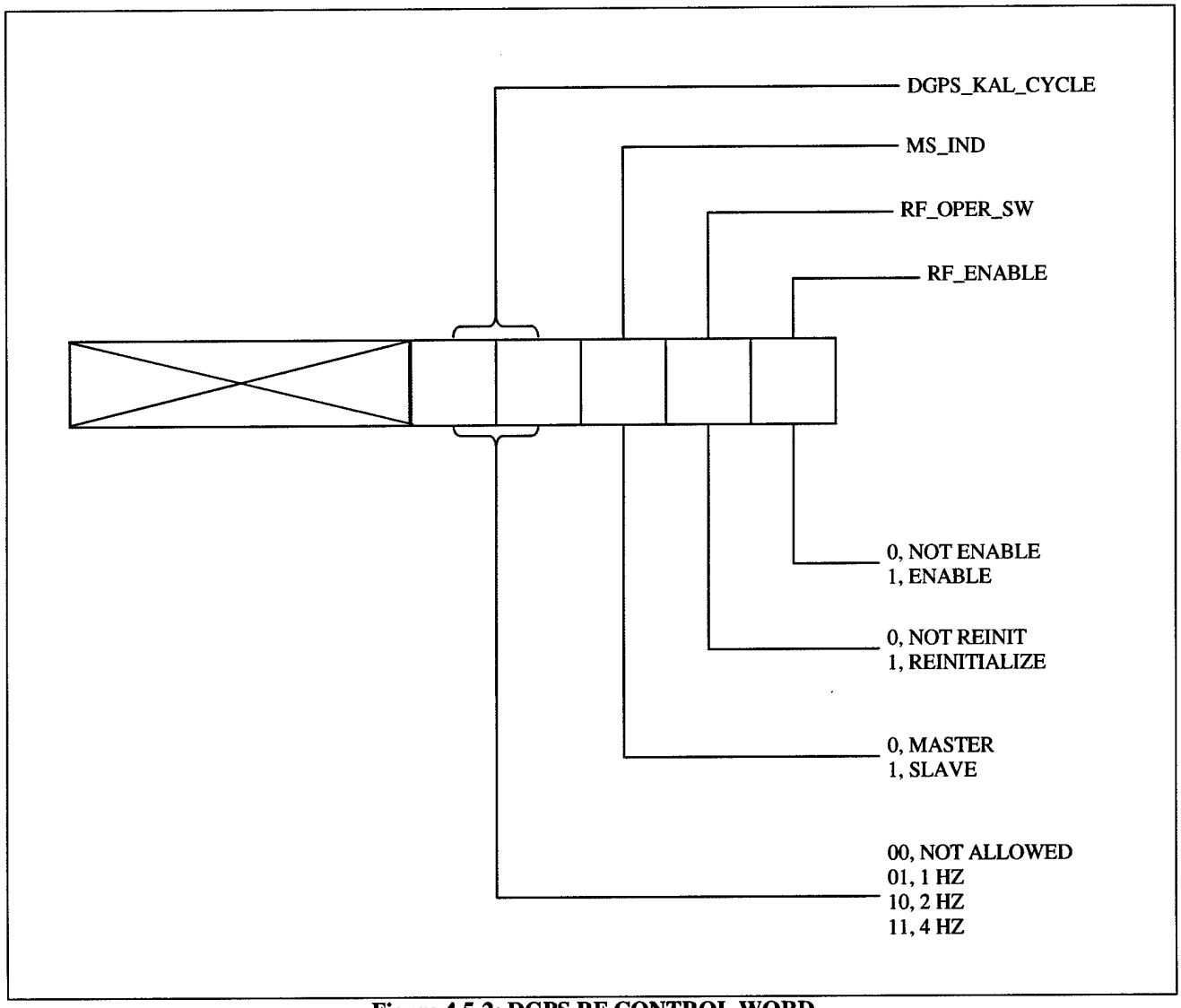


Figure 4.5-2: DGPS RF CONTROL WORD

4.5.2 GPS Constellation Ephemeris Data

There will be 24 operational GPS satellites at any given time, and thus a full expression of the total ephemeris data will require a matrix of dimension 24 x 23. For laboratory purposes, however, we will assume that the full number of orbiting satellites can equal 30. Therefore, the ephemeris data which will be required to be transferred from the TATS to the TUT via the 1553 MUX will be of dimension 30 x 23, where the data from each visible satellite, indexed by I, will be inputted in turn. The format for this data will be in double precision floating point, and will follow the following order (Table 4.5 – 1) for each satellite, designated by SV # I.

The values for this data shall be precomputed and shall be provided to the TATS as a data file. It is required that this data be provided only once at program initiation.

Ephemeris Variable No.	Data Description	LSB
1	Satellite Vehicle No.	--
2	Week No.	--
3	TGD(Group Delay)	2^{-31}
4	TOC(Clock Correction)	2^{4}
5	AF2 (Clock Correction)	2^{-55}
6	AF1(Clock Correction)	2^{-43}
7	AF0(Clock Correction)	2^{-31}
8	Crs(Sine Harmonic)	2^{-5}
9	Delta n(Mean Motion Diff.)	2^{-43}
10	Mo(Mean Anamaly)	2^{-31}
11	Cuc(Amp. Sine Harmonic)	2^{-29}
12	Es(Satellite Eccentricity)	2^{-33}
13	Cus(Sine Harmonic)	2^{-29}
14	Sqrt. Minor Axis	2^{-19}
15	Toc(Ref.Time Ephemeris)	2^{4}
16	Cic(Cosine Harmonic)	2^{-29}
17	Ogao(Long. Of Ascending Node)	2^{-31}
18	Cis (Amp. Of Sine Harmonic)	2^{-29}
19	io (Inclination Angle)	2^{-31}
20	Crc(Cos Harmonic Correction)	2^{-5}
21	Omega(Angle of Perigee)	2^{-31}
22	Odot(Rate of RA)	2^{-43}
23	idot(Rate of Inclination Ang.)	2^{-43}

Table 4.5-1: Ephemeris Data per Visible Satellite

4.5.3 GPS Pseudorange Data

The GPS pseudorange data which will be supplied to the TATS will be synthesized data not derived from an actual GPS pseudorange measurement from satellites in elliptical orbits. The data synthesis of the code phase measurements must take into account the variable transit time of the satellite signals. For this reason, this computation will require an iterative algorithm, to be described herein, to make the derived measurements compatible with the measurement time, referred to as the GPS Predicted Time (PGPSTM). This computation shall be performed offline from the TATS, and the results provided as a data file which shall be read at the rate specified (1,2,or 4Hz) in the initialization data. The GPS pseudorange data shall be supplied in blocks whose formats are derived from, but are not identical to Canadian Marconi NORTHSTAR Output Message 23. The primary difference between Message 23 and the format specified here is that in the former case, the pseudorange measurements provided by the GPS receiver are expressed in units of code phase counts, while in our case, the pseudorange measurements shall be converted directly to meters.

The algorithm for generating the synthetic pseudorange measurement is based upon the standard algorithm for computing satellite ECEF positions, which has been summarized in Figure 3.3.2 – 2 of this report, except that no measurements are required, Instead, we must compute the coarse time of transmission (TC) for each individual satellite measurement by compensating for the signal time of transit based upon the true ranges. The process, in summary is as follows:

(a) Begin the pseudorange estimation process by zeroing the variable PR which represents the estimated pseudorange variable.

(b) Compute the ECEF position of the affected satellite using the elliptical model.

(c) Compute the pseudorange estimate in meters using the computed range difference between satellite and platform as well as clock bias contributions, if necessary.

(d) Use the estimated pseudorange value to recompute TC, and iterate.

(e) Stop the computation when pseudorange value differences are below a preselected value.

Iteration #	TC(Seconds)	Satellite Number ISVNO	PR (meters)
1	238487.000000071	6	22148112.1031953
2	238486.926121919	6	22148141.8754309
3	238486.926121819	6	22148141.8754863
4	238486.926121819	6	22148141.8754863
5	238486.926121819	6	22148141.8754863
6	238486.926121819	6	22148141.8754863
7	238486.926121819	6	22148141.8754863
8	238486.926121819	6	22148141.8754863
9	238486.926121819	6	22148141.8754863
10	238486.926121819	6	22148141.8754863

Table 4.5-2: Synthetic Pseudorange Estimate Convergence

As noted in Figure 4.5 -2, above, the algorithm converges within three iterations to high accuracy. The above computation has been performed for Satellite No. 6, and indicates the rapid estimation of the coarse time of transmission and the generated value of pseudorange (PR) in meters. The algorithm pseudocode is presented in Figure 4.5 – 4, below.

```
For K = 1, 24

Compute mean motion
  DELN = EPHEM(K,9)                  mean motion difference [semicircle/s]
  SAS = EPHEM(K,14)                  Square root of semi-major axis [m^1/2]
  AS = SAS^2                         Semi-major axis
  N = DSQRT(MU/AS^3) + DELN          Computed mean motion

Compute mean anomaly
  M0 = EPHEM(K,10)                   Mean anomaly at reference time [smcir]
  MK = M0 + N*(TC - TOE)             Mean anomaly

Solve for eccentric anomaly          E + MK + ES*SIN(EOLD) [Newton-Raphson]
  ES = EPHEM(K,12)                   Eccentricity
  EKNEW = MK                         Kepler's equation for Eccen. anomaly
  FOR JX:1:100 DO
       EKOLD = EKNEW
       EKNEW = MK + ES*SIN(EKOLD)
  END
  E = EKNEW                          Eccentricity anomaly

Compute time correction term
  DEKTR = F*ES*SAS*SIN(E)            Relativistic correction term

  AF0 = EPHEM(K,7)                   7 -
  AF1 = EPHEM(K,6)                   6 -
  AF2 = EPHEM(K,5)                   5 -
  TGD = EPHEM(K,3)                   3 -
  TOC = EPHEM(K,4)                   4 -
```

```
DELT = AF0 + AF1*(TC - TOC) +
     AF2*(TC - TOC)^2 + DELTR - TGD      Transit time

T = TC - DELT                            GPS time of transmission correction
R = AS*(1.0 - ES*SIN(E))                 Satellite average radius R

Compute satellite true anomaly, NU
NUM = SQRT(1.0 - ES*ES)*SIN(E))/         Sin(true anomaly)
            (1.0 - ES*COS(E))
DENOM = (COS(E)-ES)/(1.0-ES*COS(E))      Cos(true anomaly)
NU = ATAN2(NUM, DENOM)                   Four quadrant inverse tangent
W = EPHEM(K,21)                          Argument of perigee [semicircle]
PHI = NU + W                             Argument of latitude

Compute correction terms from harmonics
CUS = EPHEM(K,13)                        Amplitude of sine harmonic correction
                                         term to the argument of latitude [rad]
CUC = EPHEM(K,11)                        Amplitude of cos harmonic correction
                                         term to the argument of latitude [rad]
CRS = EPHEM(K,8)                         Amplitude of sine harmonic correction
                                         term to orbit radius [m]
OMEGA0 = EPHEM(K,17)                     Longitude of ascending node of orbit
                                         plane at weekly epoch
CIS = EPHEM(K,18)                        Amplitude of sine harmonic correction
                                         term to the angle of inclination [rad]
CRC = EPHEM(K,20)                        Amplitude of cos harmonic correction
                                         term to the orbit radius [m]
CIC = EPHEM(K,16)                        Amplitude of cos harmonic correction
                                         term to the angle of inclination [rad]
IOTA = EPHEM(K,19)                       Inclination angle at reference time
                                         [semicircle]
IDOT = EPHEM(K,23)                       Rate of inclination angle [semicir/s]

C2PHI = COS(2*PHI)                       Double angle
S2PHI = SIN(2*PHI)

Second harmonic perturbations
DELPHI = CUS*S2PHI + CUC*C2PHI           Argument of latitude correction
DELR = CRS*S2PHI + CRC*C2PHI             Radius correction
DELI = CIS*S2PHI + CIC*C2PHI             Corrected inclination

PHI = PHI + DELPHI                       Corrected argument of latitude
R = R + DELR                             Corrected radius
IOTA = IOTA + DELI + IDOT*(T-TOE)        Corrected inclination

OMEGAD = EPHEM(K,22)                     Rate of right ascension [semicircle/s]
OER = OMEGA0 + OMEGAD*(T-TOE) -          Corrected longitude of ascending node
      OMEGAE*T

Compute satellite ECEF vector (meters)
```

```
  ECEFXS(K)  = R*COS(OER)* COS(PHI)-R*SIN(OER)*COS(IOTA)*SIN(PHI)
  ECEFYS(K)  = R*SIN(OER)* COS(PHI)+R*COS(OER)*COS(IOTA)*SIN(PHI)
  ECEFZS(K)  = R*SIN(IOTA)*SIN(PHI)

Compute Provisional Pseudorange Estimate
   PR = SQRT[(ECEFXS(K)-XP)**2 +(ECEFYS(K)-YP)**2+(ECEFZS(K)-ZP)**2 ]
   ECEFYS(K)  = R*SIN(OER)* COS(PHI)+R*COS(OER)*COS(IOTA)*SIN(PHI)
   IF(CHANGE_IN_PR > LIM) THEN
      GO TO 5000
   END IF
 END FOR
```

Figure 4.5-3: Generation of Synthetic Pseudorange

4.5.4 Master Reporting Message Characteristics

The Master Reporting message, transmitted via the TT to the TUT shall be a Link-16 TADIL-J message, designated as a **J28.7** message. As a TADIL-J message, the J28.7 shall consist of an initial word, extension word, and 0 to 4 continuation words, depending on the number of satellites observed by the master platform. A maximum of 10 sets of satellite IDs and corresponding pseudoranges , together with a time tag and master platform navigation data shall be packaged within each J28.7 message transmitted. The number of continuation words required as a function of number of observed satellites shall be as follows:

Number of Observed Satellites	Number of Continuation Words
1	0
2-3	1
4-5	2
6-8	3
9-10	4

Note that the minimum sustainable satellite count to assure DGPS operation is 5, so that there will nearly always be 2 or more continuation words provided in each J28.7 message. A provisional detailed format for the initial, extension, and continuation words of the J28.7 message has been prepared and is presented in Figure 4.5-4, over the next 15 pages.

J28.7 MESSAGE SUMMARY

PURPOSE: The J28.7 Master message is used to provide pseudorange data for up to 10 satellites and position data from the master platform to the slaves. The message contains the Master platforms latitude, longitude, altitude, time tag of Nav solution, up to 10 pseudoranges to satellites and the associated satellite number. The message will be transmitted at a one to four Hz rate (rate is TBD). The amount of data in the message requires that some of the data be coded to reduce word size.

The data will be coded as indicated below.

Data Element	# BITS	LSB	RANGE	Comments
Satellite Number	5	1	0 - 24	unsigned integer
Pseudorange by	21	11.6 ft	10k - 18k	Range is offset 10k - unsigned integer
Latitude	24	2.256 ft	+90 deg	signed integer
Longitude	24	4.512 ft	0-360 deg	unsigned integer
Altitude	14	6.1 ft	-20k - +80k	offset by 20k unsigned integer
Number of Satellites	4	1	0 - 10	max number of pseudorages in a message = 10
Time Tag	24	1	0-11059199	LSB is 1 slot

DATA ELEMENT SUMMARY

J28.7I

DATA ELEMENT	# BITS
Word Format	2
Label, TADIL J	5
Sub label, TADIL J	3
Message Length Indicator	3
Number of Satellites	4
Master Latitude	24
Master Longitude	24
Satellite 1's Number	5

J28.7E0

DATA ELEMENT	# BITS
Word Format	2
Master Altitude	14
Time Tag	24
Satellite 1's Pseudorange	21
Satellite 2's Number	5
Spare	4

J28.7C1

DATA ELEMENT	# BITS
Word Format	2
Continuation Word Label	5
Satellite 3's Number	5
Satellite 4's Number	5
Satellite 5's Number	5
Satellite 6's Number	5
Satellite 2's Pseudorange	21
Satellite 3's Pseudorange	21
Spare	1

J28.7C2

DATA ELEMENT	# BITS
Word Format	2
Continuation Word Label	3
Satellite 4's Pseudorange	21
Satellite 5's Pseudorange	21
Satellite 7's Number	5
Satellite 8's Number	5
Satellite 9's Number	5
Spare	6

J28.7C3

DATA ELEMENT	# BITS
Word Format	2
Continuation Word Label	5
Satellite 6's Pseudorange	21
Satellite 7's Pseudorange	21
Satellite 8's Pseudorange	21

J28.7C4

DATA ELEMENT	# BITS
Word Format	2
Continuation Word Label	5
Satellite 9's Pseudorange	21
Satellite 10's Pseudorange	21
Satellite 10's Number	5
Spare	

J2.0 TRANSMIT/RECEIVE RULES

TITLE: MASTER Message

TRANSMIT RULES

1.

RECEIVE RULES

WORD MAP

(U) WORD NUMBER: J28.7I

(U) WORD TITLE: MASTER MESSAGE

24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
<--- LATITUDE								NUMBER OF SATELLITES				MESSAGE LENGTH INDICATOR			SUBLABEL, TADIL J			LABEL, TADIL J					WORD FORMAT	
24								4				3			3			5					2	

49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25
<--- LONGITUDE																								--->
<--- 24																								

69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50
SATELLITE 1'S NUMBER																			--->
5																			

WORD DESCRIPTION

WORD NUMBER: J28.7I

WORD TITLE: MASTER MESSAGE INTIAL WORD

REFERENCE DFI/DUI	DATA FIELD DESCRIPTOR	BIT POSITION	# BITS	RESOLUTION, CODING, ETC
**** ***	WORD FORMAT	0- 1	2	00
*** ***	LABEL, TADIL J	2- 6	5	00010
*** ***	SUBLABEL, TADIL J	7- 9	3	000
*** ***	MESSAGE LENGTH INDICATOR	10- 12	3	0 NO ADDITIONAL WORDS. 1-4 NUMBER OF ADDITIONAL WORDS.
*** ***	NUMBER OF SATELLITES	13	4	0 TO 10
*** ***	MASTER'S LATITUDE	17	24	
*** ***	MASTER'S LONGITUDE	41	24	
*** ***	SATELLITE NUMBER FOR SAT 1'ST	65	5	

J28.7 WORD NOTES

FIELD CODING FOR J28.7I (SHEET 1)

WORD MAP

WORD NUMBER: J28.7E0

WORD TITLE: MASTER MESSAGE EXTENSION WORD

24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
<--- TIME TAG									MASTER ALTITUDE														WORD FORMAT	
24									14														2	

49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25
<--- SATELLITE 1'S PSEUDORANGE																							--->	
21																								

69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50
SPARE				SATELLITE'S 2'S NUMBER															--->
4				5															

WORD DESCRIPTION

WORD NUMBER: J28.7E0

WORD TITLE: MASTER MESSAGE EXTENTION WORD

REFERENCE DFI/DUI	DATA FIELD DESCRIPTOR	BIT POSITION	# BITS	RESOLUTION, CODING, ETC
**** ***	WORD FORMAT	0- 1	2	1
*** ***	MASTER ALTITUDE	2- 15	14	
*** ***	TIME TAG	16- 39	24	
*** ***	SATELLITE 1'S PSEUDORANGE	40- 60	21	
*** ***	SATELLITE 2'2 NUMBER	61- 65	21	
*** ***	SPARE (SP)	66- 69	5	

FIELD CODING FOR J28.7E0 (SHEET 1)

DFI DUI DUI/DI NAME DI BIT CODE DUI/DI EXPLANATION

WORD MAP

WORD NUMBER: J28.7C1

WORD TITLE: MASTER MESSAGE CONTINUATION WORD 1

24	23	22:	21	20	19	18	17:	16	15	14	13	12:	11	10	09	08	07:	06	05	04	03	02:	01	00:
SATELLITE 6'S NUMBER:			SATELLITE 5'S NUMBER					SATELLITE 4'S NUMBER					SATELLITE 3'S NUMBER					CONTINUATION WORD LABEL					WORD FORMAT	
<---																								
			5					5					5					5					2		

49	48:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27:	26	25
		SATELLITE 2'S PSEUDORANGE																						
<---																							--->	
<---									21														--->	

69	68:	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50
SP-AR-E		SATELLITE 3'S PSEUDORANGE																	--->
1							21												

WORD DESCRIPTION

WORD NUMBER: J28.7C1

WORD TITLE: MASTER MESSAGE CONTINUATION WORD 1

REFERENCE DFI/DUI	DATA FIELD DESCRIPTOR	BIT POSITION	# BITS	RESOLUTION, CODING, ETC
**** ***	WORD FORMAT	0- 1	2	01
**** ***	CONTINUATION WORD LABEL	2- 6	5	00001
*** ***	SATELLITE 3'S NUMBER	7- 11	5	
*** ***	SATELLITE 4'S NUMBER	12- 16	5	
*** ***	SATELLITE 5'S NUMBER	17- 21	5	
*** ***	SATELLITE 6'S NUMBER	22- 26	5	
*** ***	SATELLITE 2'S PSEUDORANGE	27- 47	21	
*** ***	SATELLITE 3'S PSEUDORANGE	48- 68	21	
*** ***	SPARE	69	1	

WORD MAP

WORD NUMBER: J28.7C2

WORD TITLE: MASTER MESSAGE CONTINUATION WORD 2

24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SATELLITE 4'S PSEUDORANGE																	CONTINUATION WORD LABEL					WORD FORMAT		
<---																								
21																	5					2		

49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25
<---		SATELLITE 5' PSEUDORANGE																					--->	
<---		21																						

69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50
SPARE						SATELLITE 9'S NUMBER					SATELLITE 8'S NUMBER					SATELLITE 7'S NUMBER --->			
6						5					5					5			

WORD DESCRIPTION

WORD NUMBER: J28.7C2

WORD TITLE: MASTER MESSAGE CONTINUATION WORD 2

REFERENCE DFI/DUI	DATA FIELD DESCRIPTOR	BIT POSITION	# BITS	RESOLUTION, CODING, ETC
**** ***	WORD FORMAT	0- 1	2	01
**** ***	CONTINUATION WORD LABEL	2- 6	5	00001
*** ***	SATELLITE 4'S PSEUDORANGE	7- 27	21	
*** ***	SATELLITE 5'S PSEUDORANGE	28- 48	21	
*** ***	SATELLITE 7'S NUMBER	49- 53	5	
*** ***	SATELLITE 8'S NUMBER	54- 58	5	
*** ***	SATELLITE 9'S NUMBER	59- 63	5	
*** ***	SPARE	64- 69	6	

FIELD CODING FOR J28.7C1 (SHEET 1)

DFI DUI DUI/DI NAME DI BIT CODE DUI/DI EXPLANATION

WORD MAP

WORD NUMBER: J28.7C3

WORD TITLE: MASTER MESSAGE CONTINUATION WORD 3

24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
<--- SATELLITE 6'S PSEUDORANGE																		CONTINUATION WORD LABEL				WORD FORMAT		
21																		5				2		

49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25
<---	SATELLITE 7'S PSEUDORANGE																					--->		
	21																							

69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50

```
:                    SATELLITE 8'S PSEUDORANGE
:
:
-------------------------------------------------------------------------------
:                              21                                          --->
-------------------------------------------------------------------------------
```

WORD DESCRIPTION

WORD NUMBER: J28.7C3

WORD TITLE: MASTER MESSAGE CONTINUATION WORD 3

REFERENCE DFI/DUI	DATA FIELD DESCRIPTOR	BIT POSITION	# BITS	RESOLUTION, CODING, ETC
**** ***	WORD FORMAT	0- 1	2	01
**** ***	CONTINUATION WORD LABEL	2- 6	5	00001
*** ***	SATELLITE 6'S PSEUDORANGE	7- 11	5	
*** ***	SATELLITE 7'S PSEUDORANGE	12- 16	5	
*** ***	SATELLITE 7'S NUMBER	17- 21	5	
*** ***	SATELLITE 8'S NUMBER	22- 26	5	
*** ***	SATELLITE 9'S NUMBER	27- 47	21	
*** ***	SPARE	69	1	

WORD MAP

WORD NUMBER: J28.7C4

WORD TITLE: MASTER MESSAGE CONTINUATION WORD 4

24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SATELLITE 96'S PSEUDORANGE																		CONTINUATION WORD LABEL				WORD FORMAT		
<---																								
21																		5				2		

49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25
	SATELLITE 10'S PSEUDORANGE																							
<---																						--->		
<---	21																							

69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50
SPARE																SATELLITE 10'S NUMBER		--->	
16																			

WORD DESCRIPTION

WORD NUMBER: J28.7C4
WORD TITLE: MASTER MESSAGE CONTINUATION WORD 4

REFERENCE DFI/DUI	DATA FIELD DESCRIPTOR	BIT POSITION	# BITS	RESOLUTION, CODING, ETC
**** ***	WORD FORMAT	0- 1	2	01
**** ***	CONTINUATION WORD LABEL	2- 6	5	00001
*** ***	SATELLITE 9'S PSEUDORANGE	7- 27	21	
*** ***	SATELLITE 10'S PSEUDORANGE	28- 48	21	
*** ***	SATELLITE 10'S NUMBER	49- 53	5	
*** ***	SPARE	54- 69		

Figure 4.5-4: J28.7 Structure and Format

5. Results and Discussion – Phase 1 Activities October 2002 – September 2003

BAE SYSTEMS believes that the principal goals of this phase of the project were successfully achieved. These goals centered around the development of a robust, reliable algorithm capable of achieving extremely precise relative positioning and time synchronization. During the course of this work, a detailed real-time design for the algorithm operational software was developed and documented. Finally, extensive long-lead design for support of the planned laboratory demonstration was accomplished, which will assure a smooth transition to the next project phase. The studies performed in this project phase resulted in many lessons learned, which contributed to a high degree of confidence in the integrity of the basic design. A brief overview of the project highlights and the most important insights that were achieved follows.

5.1 Validity of Fundamental Error Model

The fundamental error modeling of the Link-16 position errors was predicated on the assumption that these terms are slowly varying as compared with the amount of state information provided via the GPS pseudorange data. The simulations performed as part of Test Bed 1 and Test Bed 2 studies (Paragraphs 4.2 and 4.3) verified these assumptions in all cases studied. In particular, the application of the Link-16 Navigation Simulator for Test Bed 2 to represent the dynamics of the navigation error model proved the success of the DGPS RF algorithm under realistic tactical scenarios.

5.2 Open vs. Closed Loop Operation – Solution Stability Issues

The studies performed during Test Bed 1 operation uncovered the potential of solution unstability when operating in closed loop mode, that is applying the corrections derived by the DGPS algorithm to the Link-16 navigation solution and zeroing the state elements thereafter. (Paragraphs 4.2.1 – 4.2.2) In all cases studied, this resulted in solution divergence, sometimes after hundreds of seconds of apparently stable operation. When DGPS RF errors were kept separate from the Link-16 navigation, the instability disappeared and reliable operation was maintained indefinitely in all scenarios.

The instability noted in closed loop operation does not mean that a fully integrated closed loop solution cannot be achieved. However, such a solution would require extensive modifications to the Link-16 algorithm to maintain the assumptions of Kalman optimality that were violated by a simple reset of error terms used in these studies. There appears to be no reason why the full benefits of the algorithm cannot be realized in open loop operation, so there was no urgency to refine the closed loop solution as described above. Nevertheless, we expect to revisit this issue in the future to determine if still further improvements can be achieved by tighter integration of the two algorithms.

5.3 Sensitivity to Update Rate

A critical lesson learned during these studies is that, within the range of 0.25 and 4 Hz rate of pseudorange observations, the steady state errors remain virtually constant. (Paragraph 4.2.6) Of course, more rapid convergence to the steady state values is noted at the higher observation rates. At 4 Hz, for example, steady state is achieved within 5 to 10 seconds of algorithm initiation while at the 1 Hz rate, such convergence may require 50 to 60 seconds. Operationally, we would prefer to use the lowest rate possible consistent with solution steady state accuracy, since the higher rates impact the usage of Master Messages and expend Link-16 bandwidth at a higher than desired level. A compromise value would seem to be the 1 Hz rate, which is the planned rate which we hope to use during theTATS laboratory demonstration.

5.4 Sensitivity to Link-16 Navigation Errors

One of the gratifying observations made during these studies was that there was virtually no sensitivity to the magnitude of the Link-16 navigation errors except at levels which are far beyond those expected in Link-16 hybrid navigation. (Paragraph 4.2.5) It was not until a level of more than 750 feet per axis that unstable operation was noted. This was not very surprising, since it is well known that Extended Kalman Filters usually have convergence problems when errors exceed some critical level. The fact that the DGPS algorithm would likely never be exposed to errors even a quarter as large as these critical levels indicates that it will function well in nearly all practical environments.

5.5 Effect of Velocity States on Algorithm Performance

At the inception of this study, provision was made for the inclusion of three velocity error states within the DGPS RF Kalman Filter to investigate their effectiveness in cases where significant unmodelled velocity existed in the navigation solution. (Paragraph 4.2.4). Two important conclusions were reached when testing the effectiveness of these state variables: (a) When used in closed loop operation, their presence within the filter proved to be destabilizing, and resulted in highly oscillatory performance prior to solution divergence, and (b) In open loop operation resulting in stable performance, there was virtually no effect on accuracy whether or not the velocity states were used. The clear conclusion is that these terms are best deleted from the filter design since they provide no positive effect in stable solutions, and are harmful in solutions tending to instability.

5.6 Clock Bias/Frequency Estimation Performance

The precision navigation goals inherent in the DGPS RF algorithm make it mandatory to estimate residual GPS time and frequency errors at the same time that we estimate the position errors. Although a typical GPS time offset will normally be relatively small, on the order of tens of nanoseconds, at most, this term must be calibrated out by the filter. Likewise, the relative

frequency drift of the GPS oscillators must be estimated to assure good drift characteristics in periods of GPS observation interruption. The tests conducted using Test Bed 2 assumed both time and frequency errors at least one order of magnitude more than would be normally expected, with excellent results (Paragraph 4.3.5). For these tests, a bias term of 200 nanoseconds and frequency drift of 10 ns/seconds was assumed. Filter results calibrated bias to fractions of a nanosecond, and frequency drift to well under one nanosecond per second. No attempt to exceed these initial errors was made, but there is no reason to suspect that any reasonable limit applies, as long as initial covariance estimates exceed these errors.

5.7 Solution Sensitivity to Kalman Filter Process Noise

The addition of process noise is a vital part of the filter design to account for unmodelled error sources which, if not compensated for, could cause error terms to exceed covariance values. For a design of this kind, where many satellites are processed in each filter update cycle, one would expect that higher levels of process noise would tend to be helpful in keeping the filter gains from dropping to undesirably low levels. This indeed turns out to be the case. (Paragraph 4.2.9). When very low levels of process noise (i.e. less than 1 foot per axis per cycle) are added, the filter still converges, but tends to do so sluggishly. By comparison, addition of 5-10 foot process noise/cycle results in rapid and consistent convergence of error terms. The final value selected for position process noise was, in fact 10 feet per axis,

5.8 Solution Sensitivity to Low Pass Filtering

The error terms estimated by the DGPS RF filter are, in fact, the residual errors for the Link-16 hybrid inertial navigation. In normal Link-16 navigation, the application of position updates caused by receipt of PPLI messages will result in step changes in estimated position error . When comparing DGPS RF estimates for these errors, error spikes coinciding with the Link-16 navigation filter are noted. (Paragraph 4.3.2) The actual position errors of the two platforms are, in fact, smoothly varying terms. While each spike is eliminated within a single update cycle, these discontinuities, it is desirable to low-pass process these errors with a separate LP filter having time constant on the order of a few seconds. These error graphs are improved representatives of the true state of the error estimation process, and a low pass filtering process was therefore included within the final software design for this algorithm.

5.9 Sensitivity to Number of Satellites Processed

Normal operation of the DGPS RF algorithm will process pseudoranges from as many satellites as are received which lie at least five degrees above the horizon. Typical satellite counts in these cases range from 8 – 11. However, it is possible to conceive of partial GPS jamming circumstances where the estimator may be forced to operate with a number of satellites smaller than these numbers. (Paragraph 4.3.4) Hence, a special test was performed which steadily reduced the number of satellites to investigate the performance deterioration, if any. The result of this test was that there was virtually no degradation of performance when satellites processed were reduced to six. Reducing the number to five resulted in brief spikes up to 10

feet, which were quickly reduced by further processing. It was only when the number of satellites was reduced to four that the solution was observed to collapse. It should be noted that there was no specific selection criteria for satellites used within this test, with the deleted satellites simply taken from the end of the satellite processing list. It is possible that the four satellite case failed simply because the satellites remaining failed to provide observational data in orthogonal directions. Whatever the case was for this test, the algorithm itself demonstrated remarkable robustness as to the number of satellites required for acceptable levels of operation.

5.10 Real-Time Software Design for the DGPS RF Algorithm

The generation of a modular, robust real time implementation of the DGPS RF algorithm was a prime goal of this phase of the project. The resulting design (Paragraph 4.4) is the result of these efforts. The differing environments for testing of this algorithm (PC, Laboratory, and Airborne Testing) required the development of no less than three different variants (Versions 1, 2, and 3) which were specifically designed for operation in their respective venues. (Note: A fourth version has been proposed, as well, which would add the capabilities of the Atmospheric Refinement Filter, as discussed in Paragraph 6.0)

The key to minimum redesign of the DGPS RF operational software was to concentrate the key processing requirements within five "core" elements, namely:

(a) RF Executive Control
(b) RF Source Data Processing
(c) RF Kalman Processing
(d) Build Master Message
(e) Build RF Results Data

These elements require virtually no modification for all program versions. The primary differences in the operational software reside within the two interface modules, namely:

(f) OCP Interface Processing
(g) Host Interface Processing

The differences in these two versions are due primarily to the change from a simulator representation (LNS) of the Link-16 navigation utilized in Version 1, to the interface with a working Link-16 terminal OCP for Version 2. Little modification will be required to move to Version 3, the airborne test software, and these changes will depend primarily on the specific GPS receiver which will be selected for these test activities.

The success of this software design may be judged by the fact that all Test Bed 2 simulation results presented in this report were generated with the software build to Version 1 specifications.

5.11 Preparation for Laboratory Testing – Modification of TATS and Link-16 OCP.

A significant amount of time was expended during phase 1 of this project in initiating design work for the Link-16 OCP as well as the Terminal ATP Set to prepare for the phase 2 laboratory testing. In the case of the TATS, specific modifications were identified covering new input data streams required to represent the GPS pseudorange measurements, ephemeris data, and the J28.7 master message formats. None of these inputs are currently in use for the TATS unit. However, the time sequencing of each of these streams is essentially predetermined, since they will be functions of aircraft trajectories and the satellite ephemeris data. Therefore, it will be possible to precompute approximately 90% of all of this data. The effect of this is to eliminate a great deal of expensive redesign for the TATS software. We need simply to assure that the appropriate message data and measurements are "metered out" within the TATS according to the scenario selected. This will have the result of greatly reducing cost and risk for modifying the TATS software to reconfigure it for the laboratory testing.

The majority of the labor hours scheduled to be expended within phase two deal with the reconfiguration of the Link-16 OCP software to accommodate the DGPS RF algorithm. This will involve a degree of refinement and repackaging of the core and interface modules described in Paragraph 4.4, and listed in Appendix A in order to assure compatibility of data interfaces with the existing OCP design. We have taken great pains to simplify the OCP and Host interface requirements. In particular, the data transfer of navigation data from the Link-16 hybrid navigation is effectively "one way". That is to say that no direct feedback into the complex navigation software of DGPS RF results is required, or even allowed. The incorporation of the DGPS RF results will be performed within the host software, with no disruption of the hybrid navigation within the terminal. Once again, a precise and well defined interface design will pay large dividends in minimizing risk and cost for the upcoming software laboratory demonstration.

6. Recommendations

The results of this study to date indicate that the expected performance levels of the DGPS RF algorithm are indeed achievable. The goals of better than 1 meter relative performance in position, 0.1 ft/sec in velocity, and time synchronization relative to master of 1 nanosecond or better have been achieved in Version 1.0 in a laptop/PC environment. It is expected that the laboratory demonstration of the DGPS RF algorithm using a Link-16 terminal in conjunction with the Terminal ATP Test Set during October 2003 through May 2004 will confirm these performance levels.

6.1 Proposed Airborne Flight Testing of DGPS RF Version 3.0

Because of the modular structure of the DGPS RF operational code, few modifications will need to be made to achieve Version 3.0 capability, that is, to generate a code version capable of airborne flight testing using actual aircraft and GPS receiver assets. The only code changes required will be to modify Host Interface Processing to accommodate the precise format of the pseudorange and ephemeris data provided by the GPS receiver selected for testing. Because of the extremely accurate positioning provided by the algorithm, an equally accurate position reference must be provided to measure true system performance where two or more aircraft are equipped with DGPS RF modified terminal assets. It is clear that only an external Differential GPS navigation reference can provide these levels of accuracy.

BAE has had experience using Differential GPS references in tests performed at Wallops Island in conjunction with Naval Weapons Center, Dahlgren Va personnel for Link-16 Sensor Registration tests performed in support of the Single Integrated Air Picture (SIAP) In these tests, performed during 2001 (Figure 6-1), three test aircraft were flown in the vicinity of the Wallops Island SPY-1 radar for the purpose of testing sensor registration algorithms. A key component of the test was a special aircraft equipped with Differential GPS navigation which served as a reference for the estimated aircraft positions computed by the SPY-1 radar when corrected for bias errors by the sensor registration algorithm. The equipment carried by this test aircraft has an estimated position error in the vicinity of 1 meter, which is within the range that we would be seeking for any flight test of the DGPS RF algorithm.

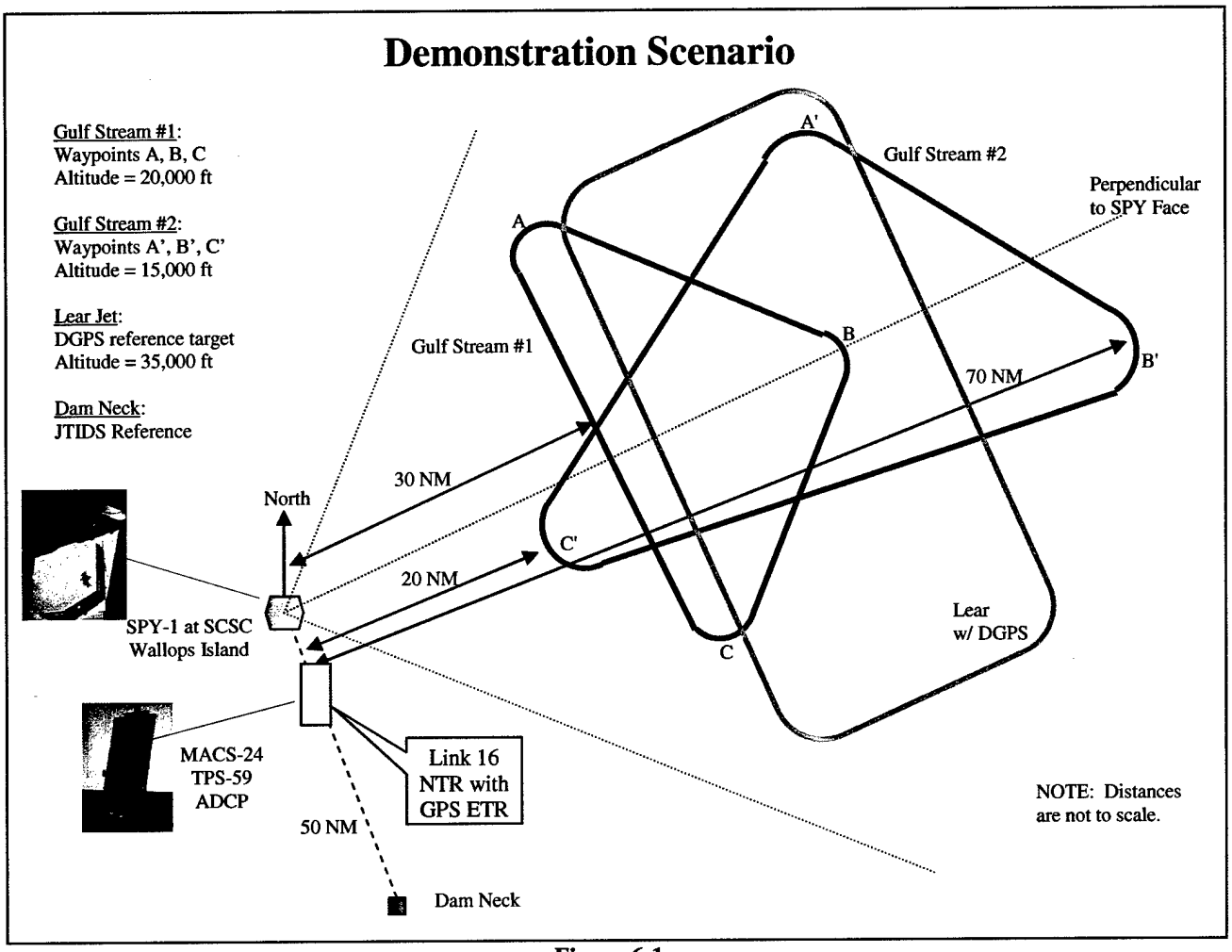


Figure 6-1

A flight test community suitable for our proposed test could use two aircraft playing the roles of master and slave platforms. Alternatively, and less expensive to conduct, the master could be played by a ground station located at a surveyed position on the earth. (Figure 6-2), while the slave would still require a fully integrated pallet containing INS, Link-16, GPS receiver, and an independent differential GPS position reference and a flight recorder to capture test data. The major expenses for such a test would be the construction of the test pallet and, of course the costs incidental to the aircraft itself (fuel, crew, etc.). One possibility to minimize these costs could be to use an aircraft with Link-16 (MIDS) already integrated, such as the F/A-18, but such an aircraft would need to be equipped with a flight recorder and have the GPS interface modified to allow pseudorange measurements to be passed via the 1553 MUX input. Detailed tradeoffs of cost and logistics will be required to select the optimum test asset for this exercise.

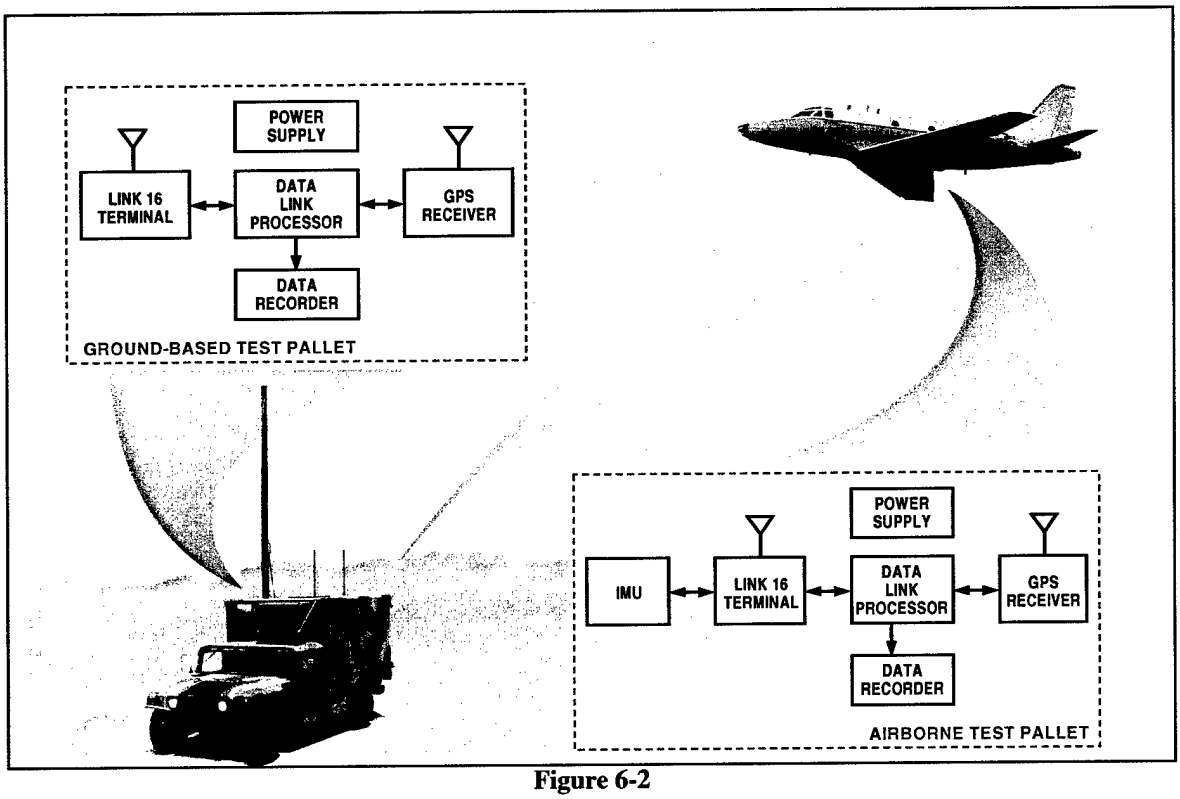


Figure 6-2

6.2 Integration of DGPS RF With Atmospheric Refinement Filter

It is a well known fact that uncertainties in atmospheric refraction leading to unmodelled PPLI Time of Arrival errors represent the largest single component of Link-16 navigation errors. For this reason, BAE SYSTEMS, under ONR sponsorship, has initiated a study to apply sensor registration methodology to produce real time estimates of refraction model parameters. The resulting algorithm, designated as the **Atmospheric Filter (AF)**, has been shown to produce extremely accurate estimates of true range between Link-16 transmitter and receiver, in the range of one to two meters. The existing constant two-parameter refraction model now implemented in Link-16 averages 10 or more meter errors in range estimates for short range, and can suffer errors of more than 100 meters for ranges in excess of 150 kilometers due to atmospheric variation.

Since both the Atmospheric Filter and the DGPS RF have as their ultimate purpose the improvement of community navigation, these algorithms should be viewed as complementary in nature. In particular, the DGPS RF can provide the extremely accurate true range estimate needed for the AF, while the AF range quality can help to maintain precision navigation accuracy even during periods of GPS jamming or interruption, once the atmospheric calibration has been accomplished. The DGPS RF algorithm is scheduled for laboratory integration during phase 2 of the project commencing in October 2003. This may be viewed as an opportunity to also perform laboratory validation of the AF algorithm as well. Paragraph 4.5 of this report introduced the modifications required to exercise the DGPS algorithm on the TATS equipment,

including provision of the GPS satellite observations. Compared to these efforts, the addition of a true atmospheric model needed to exercise the AF would be a relatively minor addition to the tasks already scheduled for the TATS. Similar modifications to the Link-16 OCP to support the AF algorithm are likewise simpler than their DGPS counterparts, since no external observations are required for the AF algorithm. This algorithm requires only the availability of the existing PPLI message traffic, which is now supported by the TATS architecture.

7. Conclusions

The results presented in detail in paragraph 4.3 indicate that the goals assumed at the beginning of this project have indeed been realized. The clear result of this study is that we have succeeded in demonstrating the practicality of a software algorithm capable of refining mobile community navigation and time synchronization to levels that, to the author's knowledge, have not previously been demonstrated without the prior establishment of a surveyed reference point, as in conventional differential GPS operation. The DGPS RF algorithm has been proven to provide a relative positioning accuracy between a platform and a designated mobile master reference of well under one meter, as well as a relative time synchronization of one nanosecond or better. This capability has been achieved using existing Link-16 capabilities combined with processing of pseudorange measurements from the GPS satellite community. Aside from the necessity of providing these GPS measurements to the Link-16 terminal, all of the necessary modifications reside exclusively in the software algorithm designed and developed during the course of this project. The fundamental basis of this technique is to exploit the robust and reliable hybrid navigation processing that currently exists in each and every Link-16 terminal, and remove the slowly varying errors that result through GPS psuedorange measurements.

The implications of this capability are significant to US warfighting ability in at least two areas, namely (a) Cooperative Tactics for Hostile Emitter Location, and (b) Remote Time Transfer among mobile participants. We consider each of these areas in turn, below.

7.1 Cooperative Tactics for Hostile Emitter Location

Rapid and effective localization of hostile emitters remains a critical operational requirement for many tactical missions. The difficulty of geolocation of hostile emitters with a single search platform is well known. A single aircraft must fly a lengthy base leg to develop significant target observability, and the time required to geolocate a target at a range of, say, fifty miles, can often be measured in hundreds of seconds. By comparison, the use of multiple aircraft well correlated in position and time which are sharing measurement data can reduce the time for geolocation to less than ten seconds or better. (Figure 7-1). The application of the DGPS RF algorithm technology to this mission is self-evident. Residual geolocation errors of emitters will always be measured in multiples of the relative positioning accuracy of the measurement platforms. The benefits of mobile members whose relative positioning errors are less than a meter means that targets can be located to comparable accuracy through a variety of sensor measurements. Phase interferometry is one such method. Delta Frequency and Time

algorithms provide still another option for precise geolocation. All require the utmost in position correlation between measurement platforms. As part of the architecture studies performed by BAE SYSTEMS on the DARPA-sponsored AT-3 emitter geolocation study in 2001, the DGPS RF algorithm was embedded within the sensor package to provide necessary navigation and time synchronization precision (Figure 7-2)

The benefits of the DGPS RF Algorithm to the emitter location problem are not restricted to geolocation. Many sophisticated target identification algorithms also exist which depends on exquisitely accurate time synchronization between multiple platforms. This synch is necessary to classify the precise radar type, and indeed provide a signature of the individual emitter via "pulse fingerprints" which include, for example, pulse rise time. Relative synchronization of collector platforms to nanosecond level is a vital component of this processing. DGPS RF is designed to provide exactly this capability.

7.2 Remote Time Transfer Applications

The US Navy recognizes the importance of maintenance of precise timing within individual ships and task forces for the purpose of *synchronization* of various communication, navigation, and fire control units. A system engineering team (SET) designated as Common Time Reference (CTR) was, in fact established in 1998, to specifically explore this capability. The CTR SET was absorbed by the SIAP organization in 2001. The CTR SET established requirements for both intraplatform and interplatform transfer of precision time references. A key requirement identified by the CTR SET is the ability to achieve extremely precise time transfer from a remote standard time reference to, say, a fleet at sea.

The CTR Draft Requirements Document states that a passive (i.e. non transmitting platform) is capable of inheriting UTC time to within 50 nanoseconds via GPS reception. *However, by permitting transfer of data among platforms together with common viewing of multiple GPS satellites, significantly higher accuracy may be achieved.* This, in the words of the draft requirements document, is a precise restatement of the principles of the DGPS RF algorithm. We have already demonstrated the capability of sub-nanosecond calibration via the algorithm. The application of this capability to CTR would be straightforward and effective using Link-16 as the data transfer medium as explained within this document. The potential time transfer applications are not limited to mobile seaborne members. A reduced DGPS RF state estimator retaining only clock bias and frequency between a given transmitter and receiver located at surveyed positions could easily bring this time transfer capability to land-based remote stations

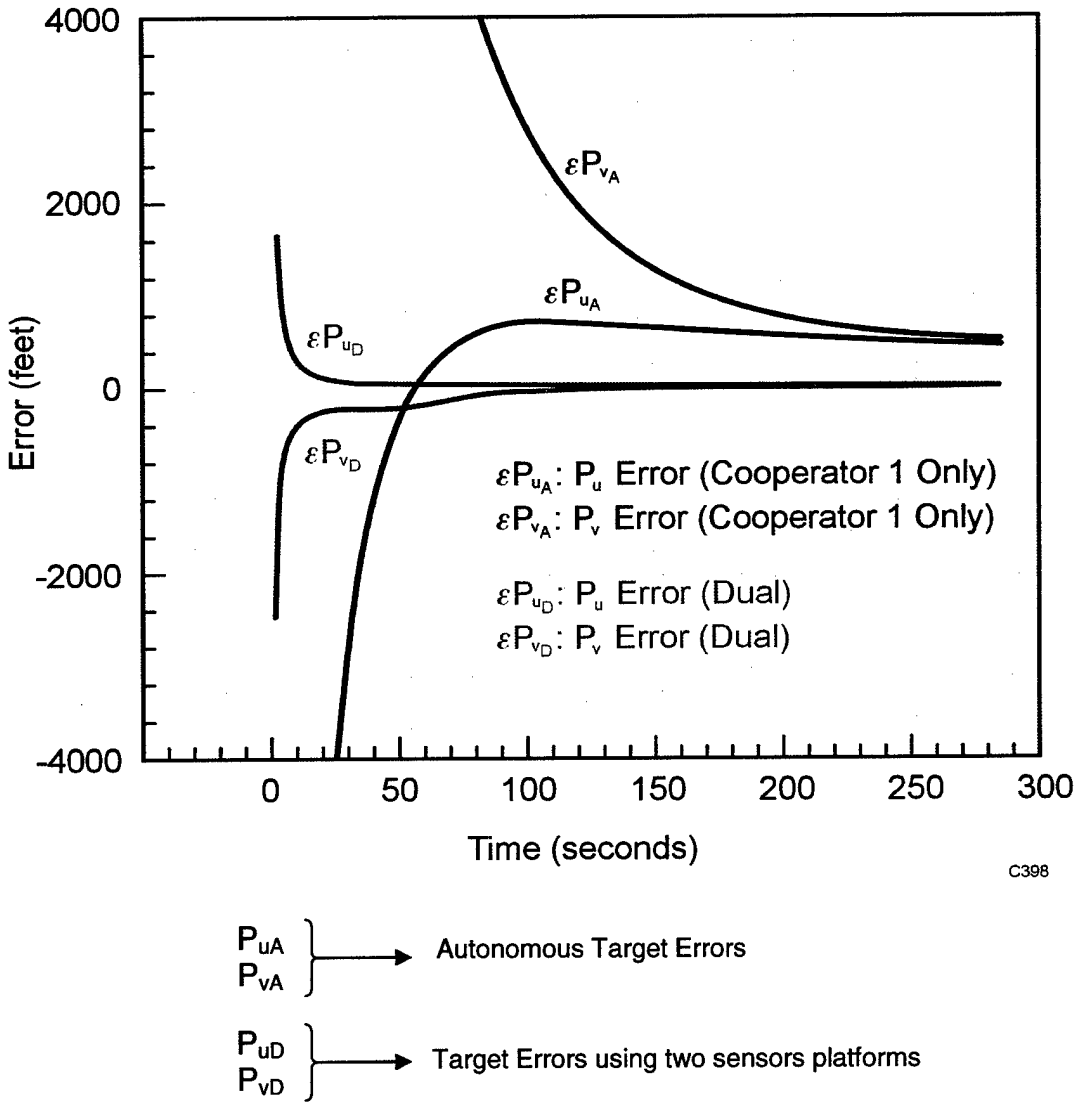


Figure 7-1: Corporative Link-16 Emitter Location Tactics Aided By Precise Community Nav and Time Synch

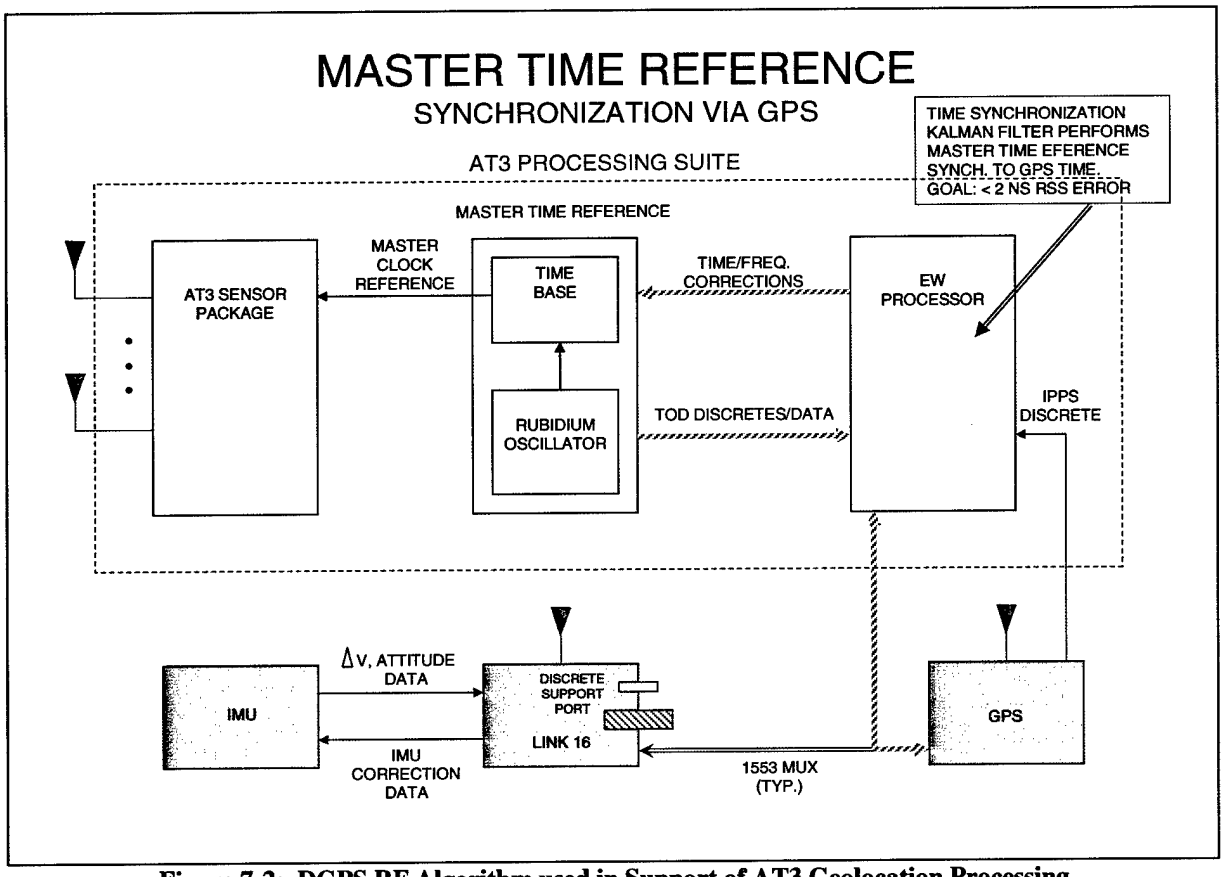


Figure 7-2: DGPS RF Algorithm used in Support of AT3 Geolocation Processing

8. References

Differential Operation of NAVSTAR GPS. R. Kalafus, J.V. Icons, N.Knable. ION Navigation Journal. March 1983. (Reprinted in ION GPS Reprints, Vol. II, June 1984).

A Kalman Filter Approach to Prediction GPS Geodesy. R. Brown, P.Y.C Hwang. ION Navigation Journal, December 1983. (Reprinted in ION GPS Reprints, Vol II, June 1984).

Global Position Systems, Inertial Navigation, and Integration. M.S. Grewal. L. Weill, A. Andrews. Wiley & Sons, 2001

Canadian Marconi Northstar User's Manual. Canadian Marconi Company, Montreal, Canada. April 2000.

Applied Optimal Estimation. Edited by A. Gelb. MIT Press, 1974

Appendix A:
Software Requirements Specifications

Software Requirements Specification

Application of Differential GPS Processing to Precision Link-16 Navigation and Time Synchronization

Version 1.0

Prepared by
BAE SYSTEMS

March 27, 2003

Revision History

Name	Date	Reason For Changes	Version

Table of Contents

Table of Figures

Table of Tables

1. Scope

1.1 Identification

This Software Design Document (SDD) establishes the architecture and detailed algorithms description for the Computer Software Configuration Item (CSCI) identified as the Differential GPS Refinement Filter (DGRF).

1.2 System overview

MIDS is an advanced information distribution system that integrates communication, navigation, and identification capabilities for application to airborne, land-based, and maritime tactical operations. The DGRF represents a developmental enhancement to the organic navigation capabilities of the MIDS terminal. By the processing of differential GPS pseudorange measurements received at a designated platform (master) and any other platform, the DGRF permits the computation of relative position error components (Δx, Δy, Δz), as well as relative measures of Clock Bias (BC) and Frequency Drift (fC) between master and slave platforms

1.3 Document Overview

The purpose of this SDD is to specify in detail the architecture of the DGRF algorithm, and to provide a sample tool (FORTRAN) implementation of this algorithm. As is specified in more detail in Paragraph 2.0, there will be four variants of the DGRF, which differ in both functional capability and operating environment. This document supports the design and algorithm requirements of all four DGPS variants.

2. Overall Description

2.1 Functional Operation

The DGRF relies on the existing organic navigation capability of a L16-equipped platform as a foundation, a reference for navigation and synchronization enhancements. The master platform provides a report to all other platforms, designated as Master Message. This message contains data relating to the position and velocity of the master at a designated time, plus all received GPS pseudorange data at another designated time. When received by another platform, this master data will be combined with similar navigation and GPS data received at own position within an Extended Kalman Filter (EKF). The result of these calculations is an optimal estimate of three positions and two time differences as corrections to the reference navigation solution. Performance goals are one meter or less relative positioning and one nanosecond relative time between all platforms.

2.2 CSCI Versions and Operating Environment

The DGRF shall be produced in four versions, as described below, which differ slightly in operative capability and operating environment.

- Version 1.0 – Simulator Version
- Version 2.0 – Laboratory Version
- Version 3.0 – Basic Operator Version
- Version 4.0 – Extended Operator Version (with backup mode)

2.2.1 Version 1.0 – PC Simulation Environment

Version 1.0 represents the FORTRAN simulation version of the DGRF, implementing the basic GPS-driven capability of the fundamental estimation algorithm. The true and estimated navigational solutions of a mobile community are produced by BAE SYSTEMS Link 16 Navigation Simulator (LNS). A simplified circular constellation model of GPS satellite motion has been developed and appended. The LNS, GPS Model, and V1.0 DGRF algorithm are listed within BAE's NETSIM PC Operating environment. This configuration will be utilized for project Phase 1 and demonstration of DGRF capabilities (Figure 2.2-1).

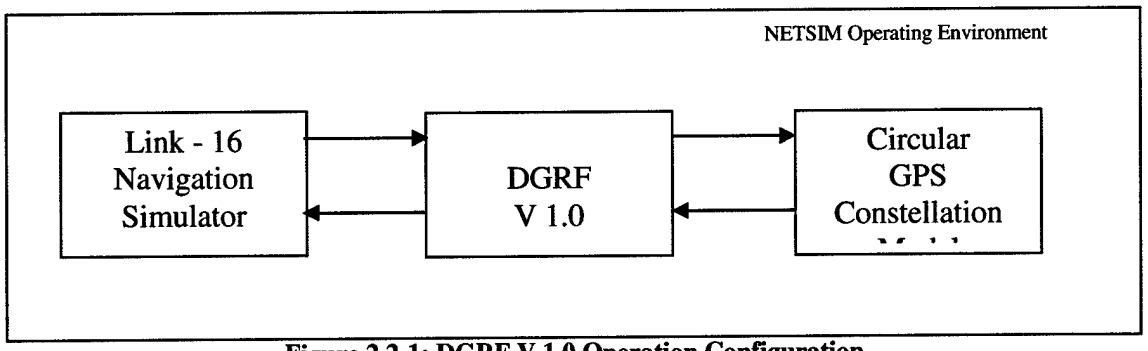


Figure 2.2-1: DGRF V 1.0 Operation Configuration

2.2.2 Version 2.0 Laboratory Test Environment

Version 2.0 of the DGRF represents a free-integration of the DGRF algorithm within the L-16 terminal Operational Computer Program (OCP). It is designed for operation on BAE SYSTEMS Terminal ATP Test Set (TATS), which provides a real-time testing environment for L-16 terminal assets. V 2.0 shall use a full elliptical GPS model, driving the 1553 host inputs to the L16 terminal with GPS pseudorange measurements calculated from that model (Figure 2.2-2). This version of the DGRF is completely equivalent to the flight-operation Version 3.0, with the exception that the GPS unit inputs are simulated with the TATS unit, and will be representative of the Canadian-Marconi North Star GPS Receiver. Version 2.0 will be utilized for project Phase 2 demonstration in March 2004.

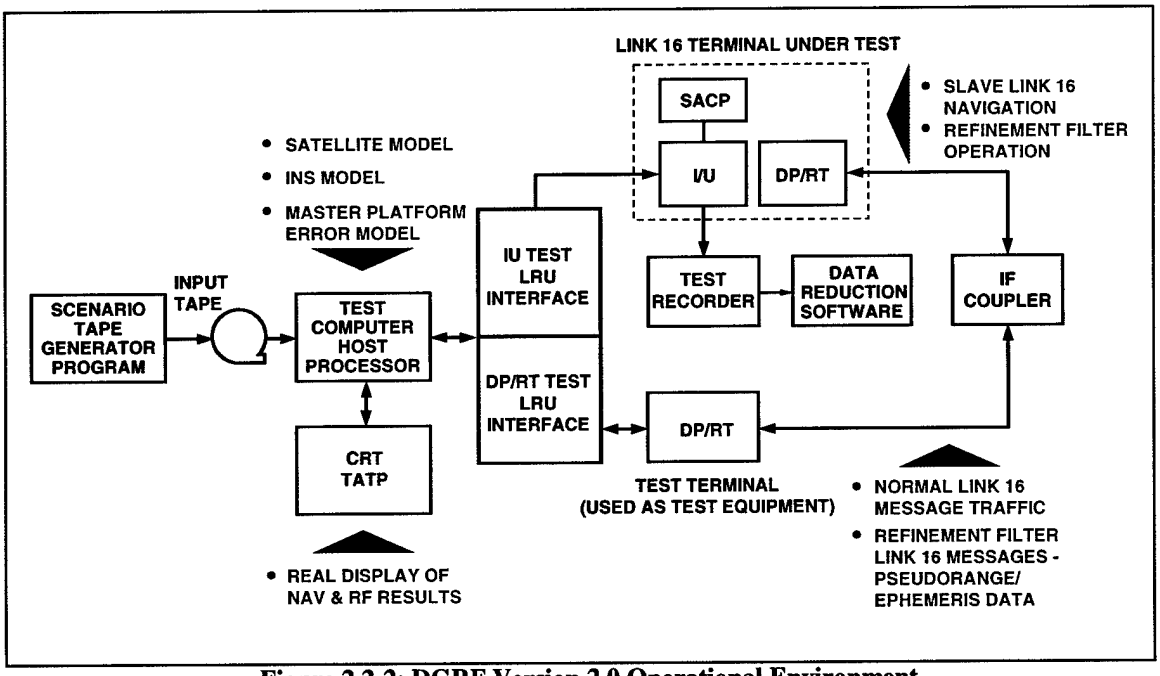


Figure 2.2-2: DGRF Version 2.0 Operational Environment

2.2.3 Version 3.0 Baseline Flight Configuration

Version 3.0 of the DGRF shall differ only slightly from Version 2.0, primarily in that it will be tailored to accept GPS pseudorange data from an actual GPS unit, type TBD. It is thus suitable for adaptation to flight testing utilizing arbitrary GPS reference units (Figure 2.2-3).

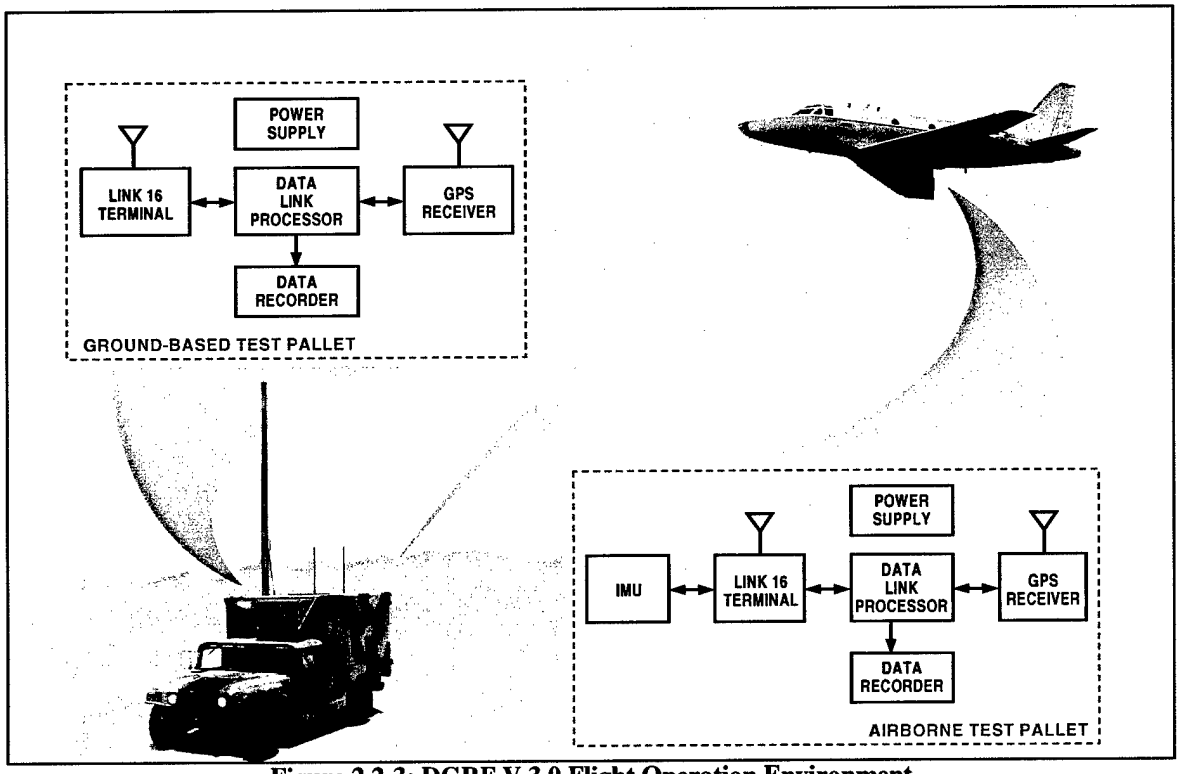


Figure 2.2-3: DGRF V 3.0 Flight Operation Environment

2.2.4 Version 4.0 Enhanced Operational Capability

Version 4.0 represents a flight qualified GPRF design incorporating two potential enhancements to baseline version:

(a) Incorporation of a Backup Mode to guarantee graceful degradation of the precision navigation solution upon jamming and/or failure of GPS aiding.

(b) Integration of the DGRF with the Atmospheric Filter by providing precision community navigation data to the AF algorithm.

Version 4.0 should be considered a "place-holder" for desirable additional capabilities for precision navigation

2.2.5 Document Data Nomenclature

Because for the differing data requirements of the four software versions, certain operation of data paths will be represented as a function of the CSCI Version indicator. These paths are labeled as to version usage via a subset of the integers (1, 2, 3, 4), where each number designates the version(s) that data element will be required.

3. Requirements

3.1 RF External Interface Requirements

This section specifies external software interfaces for the Refinement Filter. The Refinement Filter will interface with the Host and the Link-16 Operational Computer Program.

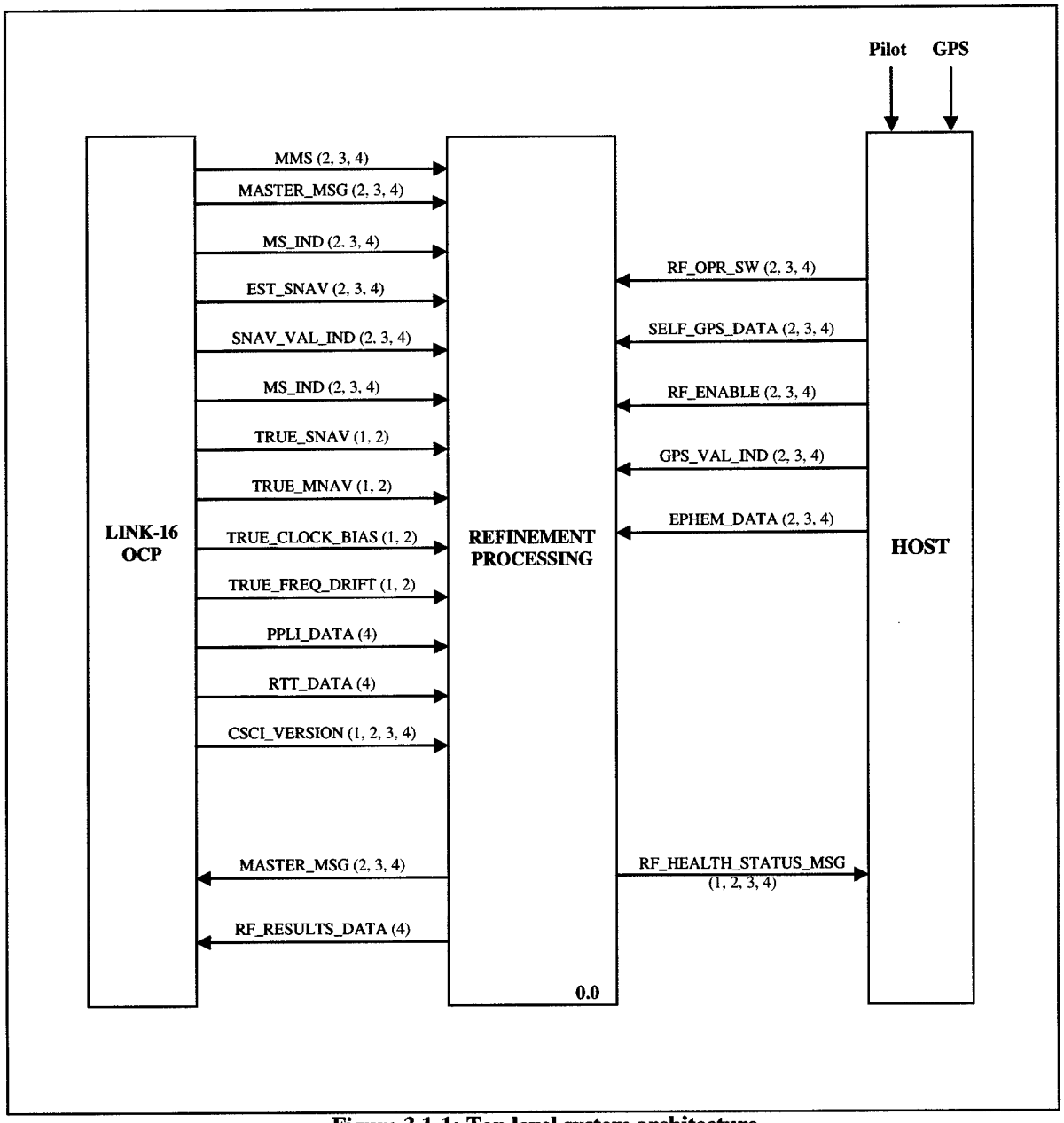


Figure 3.1-1: Top level system architecture

Inputs	Version	Definition	Description
MMS	2,3,4		Master Message Signal - indicates master message is present and valid
MASTER_MSG	2,3,4	MNAVTOV, MLAT, MLON, MALT, MEVEL, MNVEL, MALTR, GPS_HEALTH_IND, MGPSTOV, UMSAT, UMPRR	Master's Navigation solution, along with master GPS data
SNAV_VAL_IND	2,3,4		Self Navigation Validity Indicator
EST_SNAV	1,2,3,4	SLAT, SLON, SALT, SEVEL, SNVEL, SALTR, SNAVTOV	Navigation solution from L16 OCP
MS_IND	2,3,4		Master/Slave indicator
TRUE_SNAV	1,2	TSLAT, TSLON, TSALT, TLATR, TLONR, TSALTR, TSNAVTOV	TRUE SELF navigation solution
TRUE_MNAV	1,2	TMLAT, TMLON, TMALT, TMLATR, TMLONR, TMALTR, TMNAVTOV	TRUE MASTER navigation solution
TRUE_CLOCK_BIAS	1,2	BCTRUE	TRUE clock bias
TRUE_FREQ_DRIFT	1,2	FCTRUE	TRUE frequency drift
PPLI_DATA	4		Requested PPLI
RTT_DATA	4		Requested RTT
CSCI_VERSION	1,2,3,4	CSCI_VERS_ID	Denotes operational version
RF_OPR_SW	2,3,4		Reset/Initiate signal from Host pilot
SELF_GPS_DATA	2,3,4	USSAT, USPRR, SGPSTOV, GPS_HEALTH_IND	Received GPS data from HOST
RF_ENABLE	2,3,4		ON/OFF switch
GPS_VAL_IND	2,3,4		GPS validity Indicator
EPHEM_DATA	2,3,4	EPHEM(30,23)	Ephemeris satellite data received from the host for pseudorange calculations

Table 3.1-1: REFINEMENT_PROC External Interfaces - Inputs

Outputs	Version	Definition	Description
MASTER_MSG	2,3,4	MNAVTOV,MLAT, MLON, MALT, MEVEL, MNVEL, MALTR, GPS_HEALTH_IND, MGPSTOV, UMSAT, UMPRR	Master's Navigation solution, along with master GPS data
RF_RESULTS_DATA	4	XA, PA	Refinement Filter corrections and covariance message sent to L16 OCP for AF
RF_HEALTH_ STATUS_MSG	1,2,3,4	RF_STATE, RF_STATUS_IND, XA, PA, SCOUNT, SATVIS, SNAVTOV	Indicates health of the refinement filter to the host to determine whether the filter should be reset, switch to backup mode, or system normal.

Table 3.1-2: REFINEMENT_PROC External Interfaces - Outputs

3.1.1 RF Architecture

The Refinement Processing is organized into six operational CSC's depicted below.

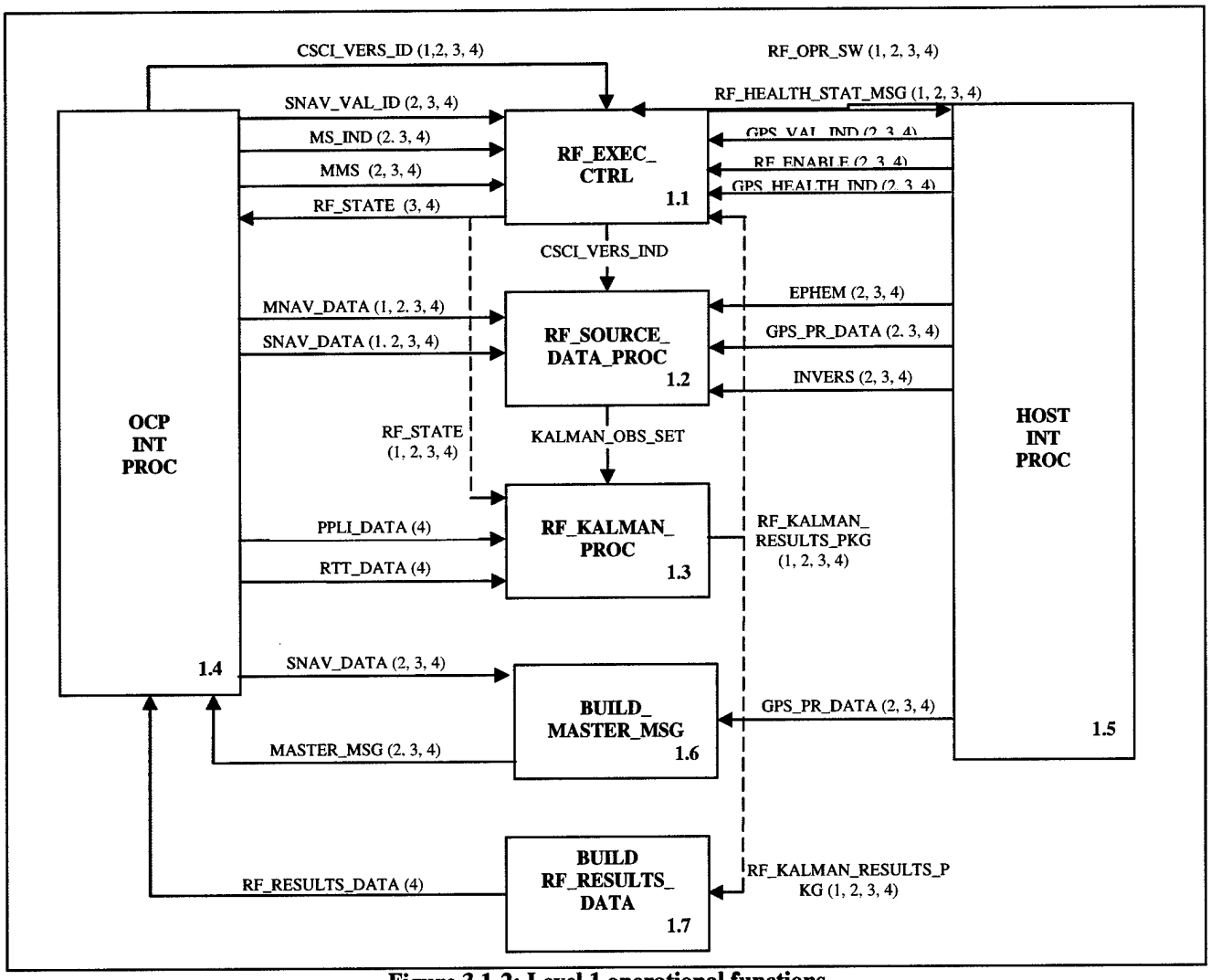


Figure 3.1-2: Level 1 operational functions

CSC	Description
RF_EXEC_CTRL	Controls the data flow within Refinement Processing depending on inputs to the system.
RF_SOURCE_DATA_PROC	Prepares and synchronizes all data for Kalman filtering.
RF_KALMAN_PROC	Applies Kalman filtering to the data passed from the RF_SOURCE_DATA_PROCESSING unit.
HOST_INT_PROC	Interfaces with host and processes data that enters this system from the host, mainly control signals and GPS data.
OCP_INT_PROC	L16 OCP interface from which all OCP data enters the system, mainly navigation solution including position, velocity, and time.
BUILD_MASTER_MSG	Called if the platform is designated the master of the network from the host. The master message is sent at a 2 Hz rate to all the slaves in the network.
BUILD_RF_RESULTS_DATA	Called at the end of each time step if the platform is designated a slave at the host. This block composes a Packed 4 message to be used in the Atmospheric Filter contained within the Link-16 Operational Computer Program.

Table 3.1-3: REFINEMENT_PROC CSC Descriptions

3.1.1.1 RF_EXEC_CTRL

The Refinement Filter Executive Control (RF_EXEC_CTRL) serves as a conductor for the entire Refinement Process. It conducts the flow of data between the interface processing CSCs and the data processing CSCs. The refinement filter can be represented by a finite state machine with 7 states. The RF_EXEC_CTRL then determines the state of the filter depending on the health of incoming data, validity checks, and control inputs.

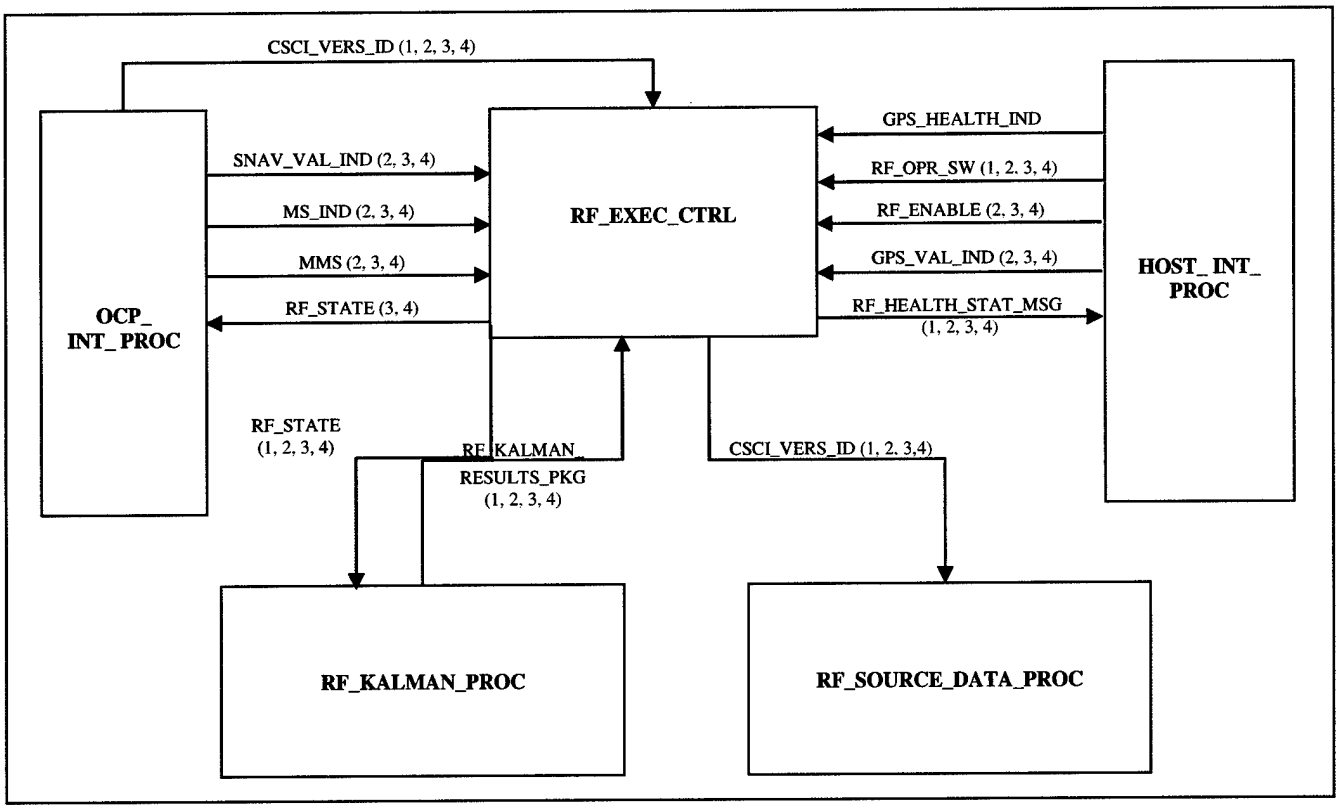


Figure 3.1-3: RF_EXEC_CTRL Level 1 Breakdown

CSC Inputs	Version	Definition	Description
CSCI_VERS_ID	1,2,3,4		Denotes operational version
GPS_HEALTH_IND	2,3,4		Indicates the health status of GPS signal. When this signal is too weak, backup mode takes over.
MMS	2,3,4		Master Message Indicator: Check if the master message is ready for transmission
MS_IND	2,3,4		Master/Slave Indicator: Checks for pilot designation for platform either master or slave aircraft.
RF_ENABLE	2,3,4		On/Off Switch
RF_KALMAN_RESULTS_PKG	1,2,3,4	Time (TC), RF_STATE, RSI, XA, PA	Refinement Filter results and covariance, XA and COV
RF_OPR_SW	1,2,3,4		Refinement filter operation switch: ON/OFF switch for system initialization/reset
GPS_VAL_IND	2,3,4		Master Navigation Validity Indicator
SNAV_VAL_IND	2,3,4		Self Navigation Validity Indicator

CSC Outputs	Version		Description
CSCI_VERS_ID	1,2,3,4		Denotes operational version
RF_HEALTH_STAT_MSG	1,2,3,4	RF_STATE, RF_STATUS_IND, XA, PA,SCOUNT, SATVIS,SNAVTOV	Indicates health of the refinement filter to the host to determine whether the filter should be reset, switch to backup mode, or system normal.
RF_STATE	3,4		Indicates the state value of the refinement filter (See below for definitions of states). [OUTPUTS TO OCP_INT_PROC]
RF_STATE	1,2,3,4		Indicates the state value of the refinement filter (See below for definitions of states). [OUTPUTS TO RF_Kalman_Proc Module]

RF_STATUS_INDICATOR (RSI)

RSI	Status	Conditions
0	RF Inoperative	Off Selected
1	RF Inoperative	SNAV_VAL_IND = invalid
2	RF Inoperative	MNAV_VAL_IND = invalid
3	RF Inoperative	GPS failure
4	RF Operative	Normal Mode / Non Steady State
5	RF Operative	Normal Mode / Steady State
6	RF Operative	Backup / Non Steady State
7	RF Operative	Backup / Steady State
8	RF Master Mode	
9	RF Observation Failure Reset Command	

RF_STATE

The Refinement Filter can be represented as a finite state machine with seven states.

State 0 – Initial startup state, RF reset
- If the filter should fail any validity check, become instable, or obtain deficient observations, it returns to this state

State 1 – Master designation
- Requires no Kalman Processing

State 2 – Slave designation / not yet initialized
- Initializes filter variables
  - PA, PX, QA, RA
- RTT_REQUEST = OFF

State 3 – Slave Designation / GPS Pseudorange - steady state not achieved
- Normal mode
- Filter working, not yet steady state

State 4 – Slave Designation / GPS Pseudorange - steady state achieved
- state at which RF will most likely exist

State 5 – Slave Designation / RTT, PPLI – backup mode initialization (not steady state)
- GPS problem
  - Insufficient signal strength
  - Number of satellites in view fall under minimum requirement

- RTT_REQUEST = ON
- Using PPLI/RTT in Kalman Processing
- Reconfigure RF: 5 → 8 states
- Steady state not yet achieved

State 6 – Slave Designation / backup mode – steady state achieved

- Backup until GPS becomes available

The state of the refinement filter determines the output of the RF_EXEC_CTRL

The following diagrams depict the transition conditions between states of the refinement filter and the logical dataflow and processes called within each state.

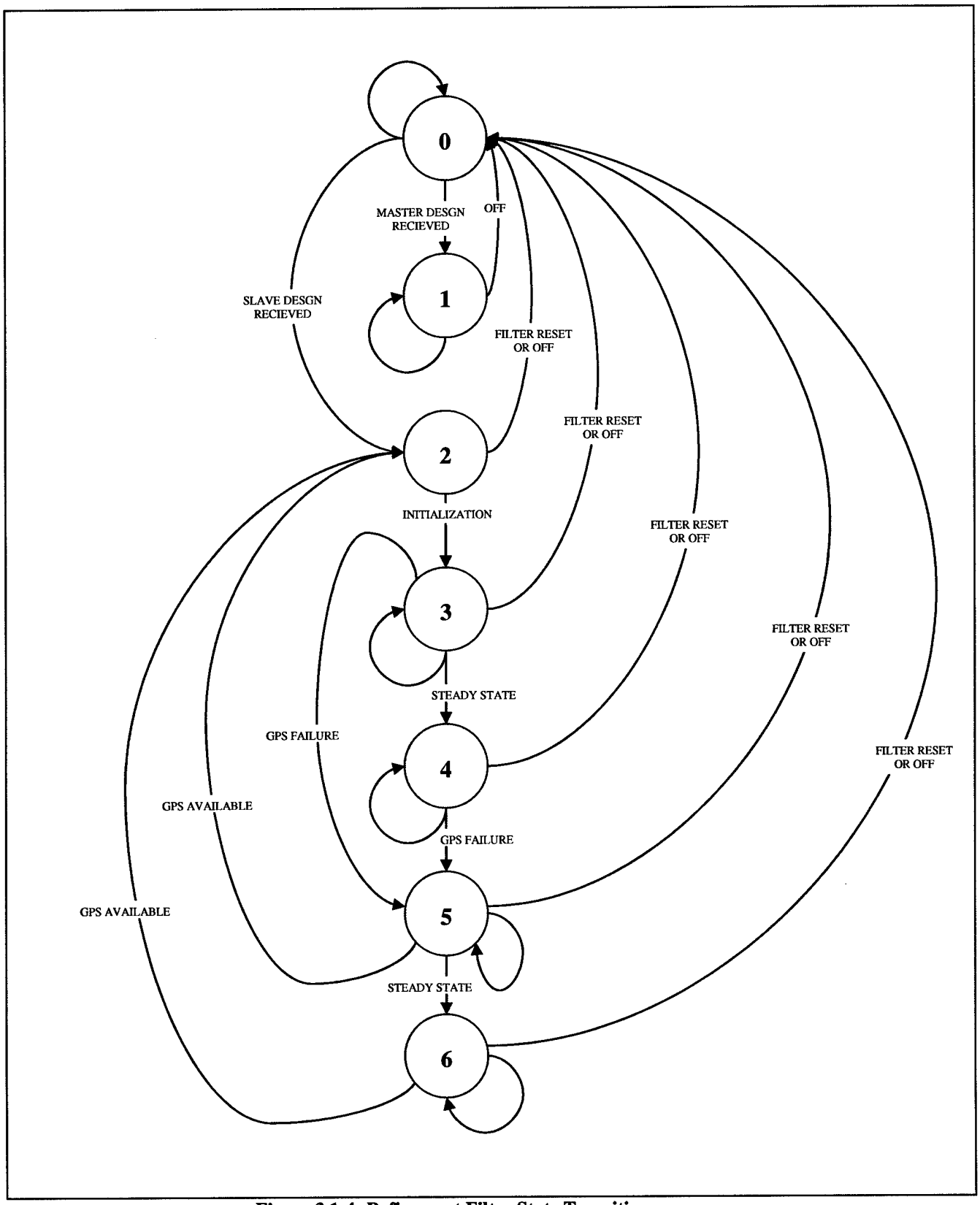


Figure 3.1-4: Refinement Filter State Transitions

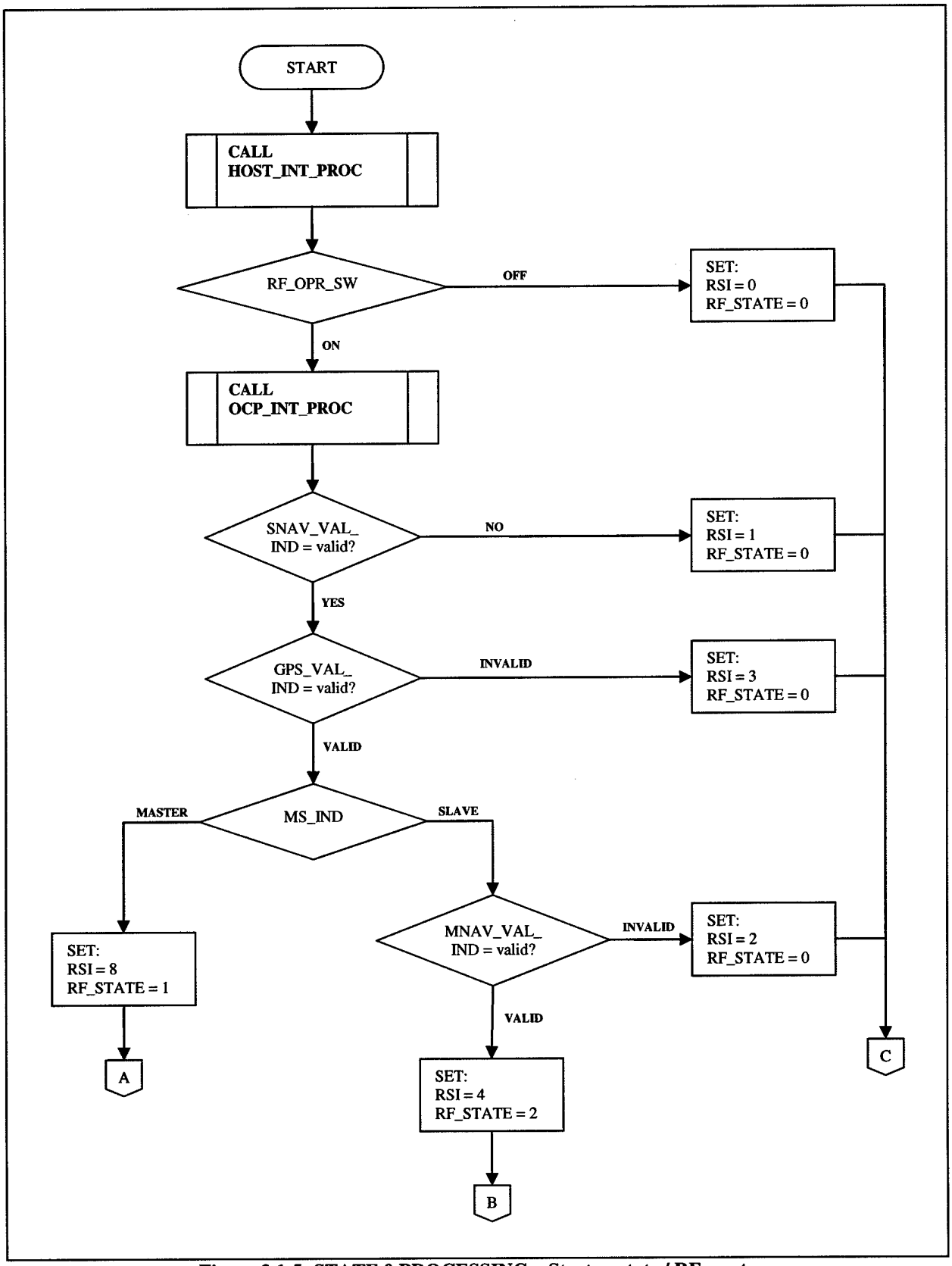


Figure 3.1-5: STATE 0 PROCESSING – Startup state / RF reset

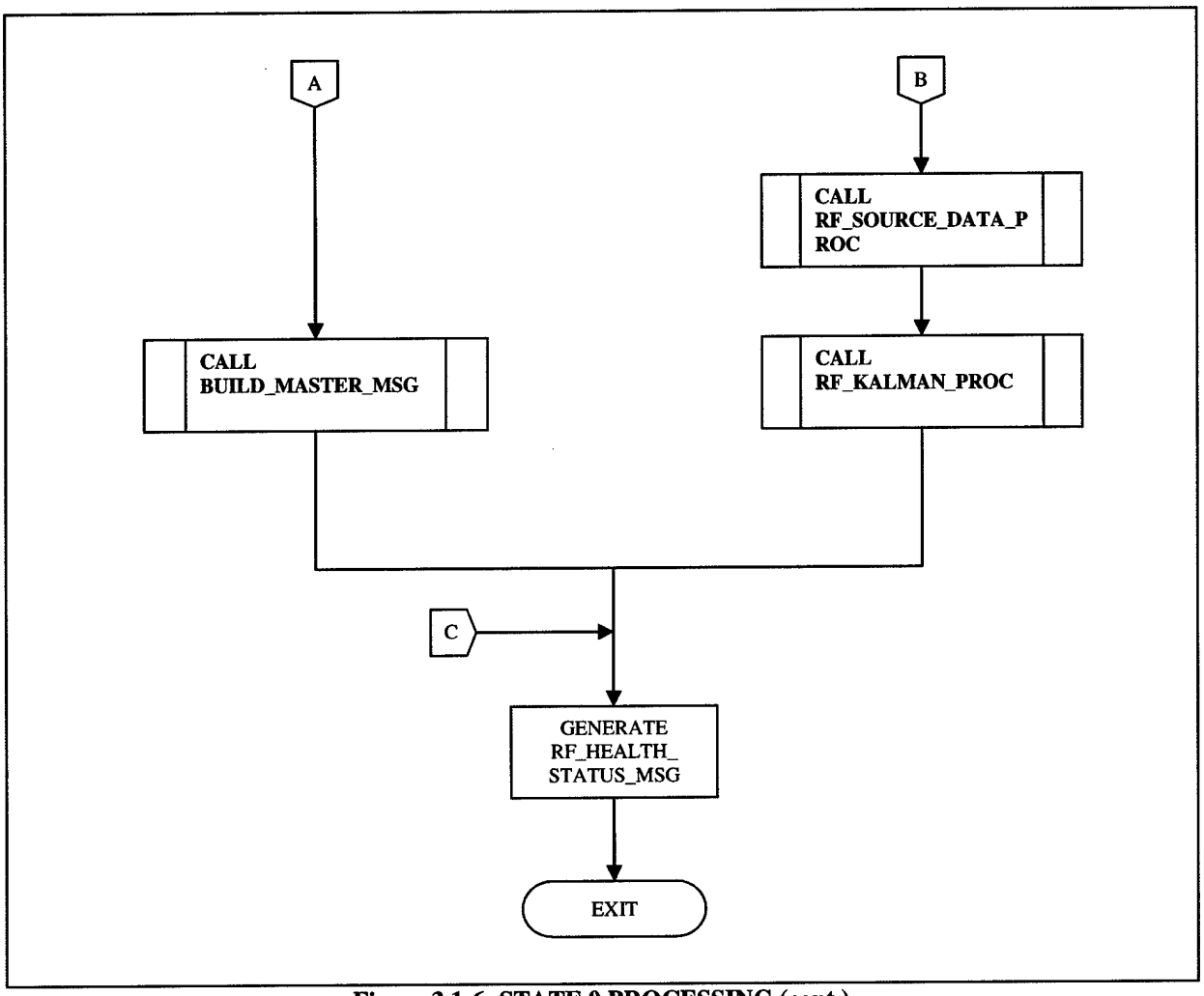


Figure 3.1-6: STATE 0 PROCESSING (cont.)

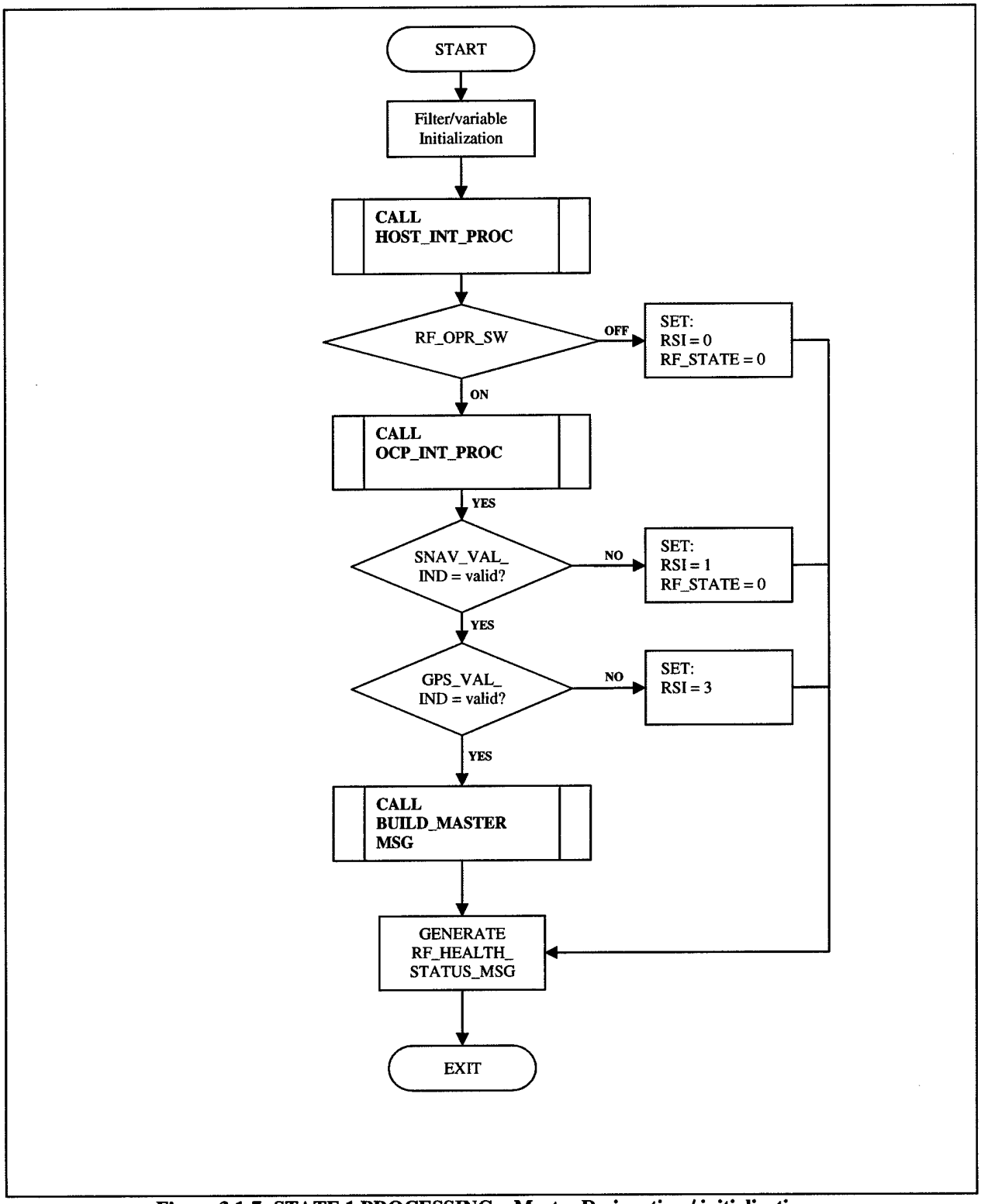


Figure 3.1-7: STATE 1 PROCESSING – Master Designation / initialization

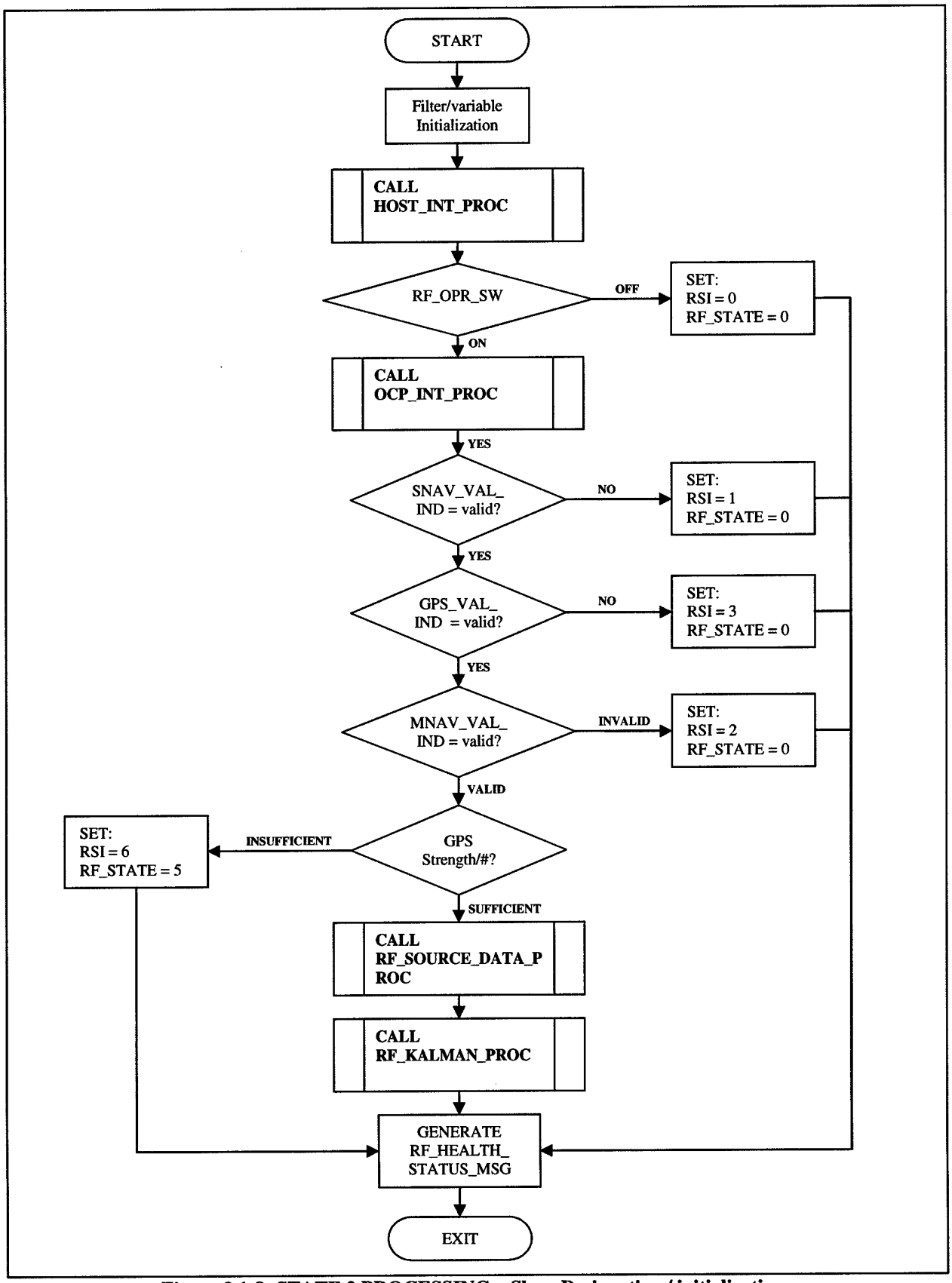


Figure 3.1-8: STATE 2 PROCESSING – Slave Designation / initialization

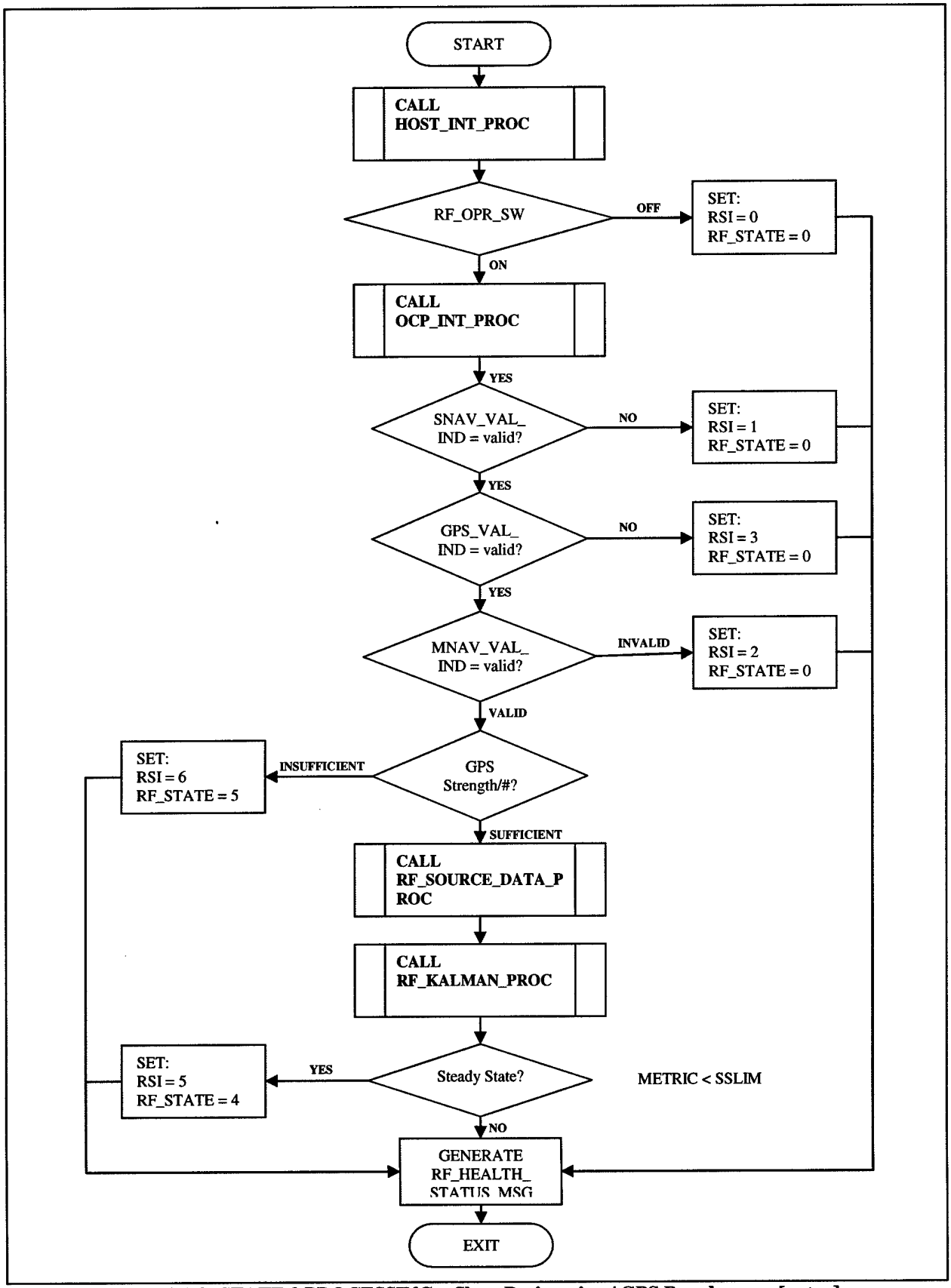


Figure 3.1-9: STATE 3 PROCESSING – Slave Designation / GPS Pseudorange [not ss]

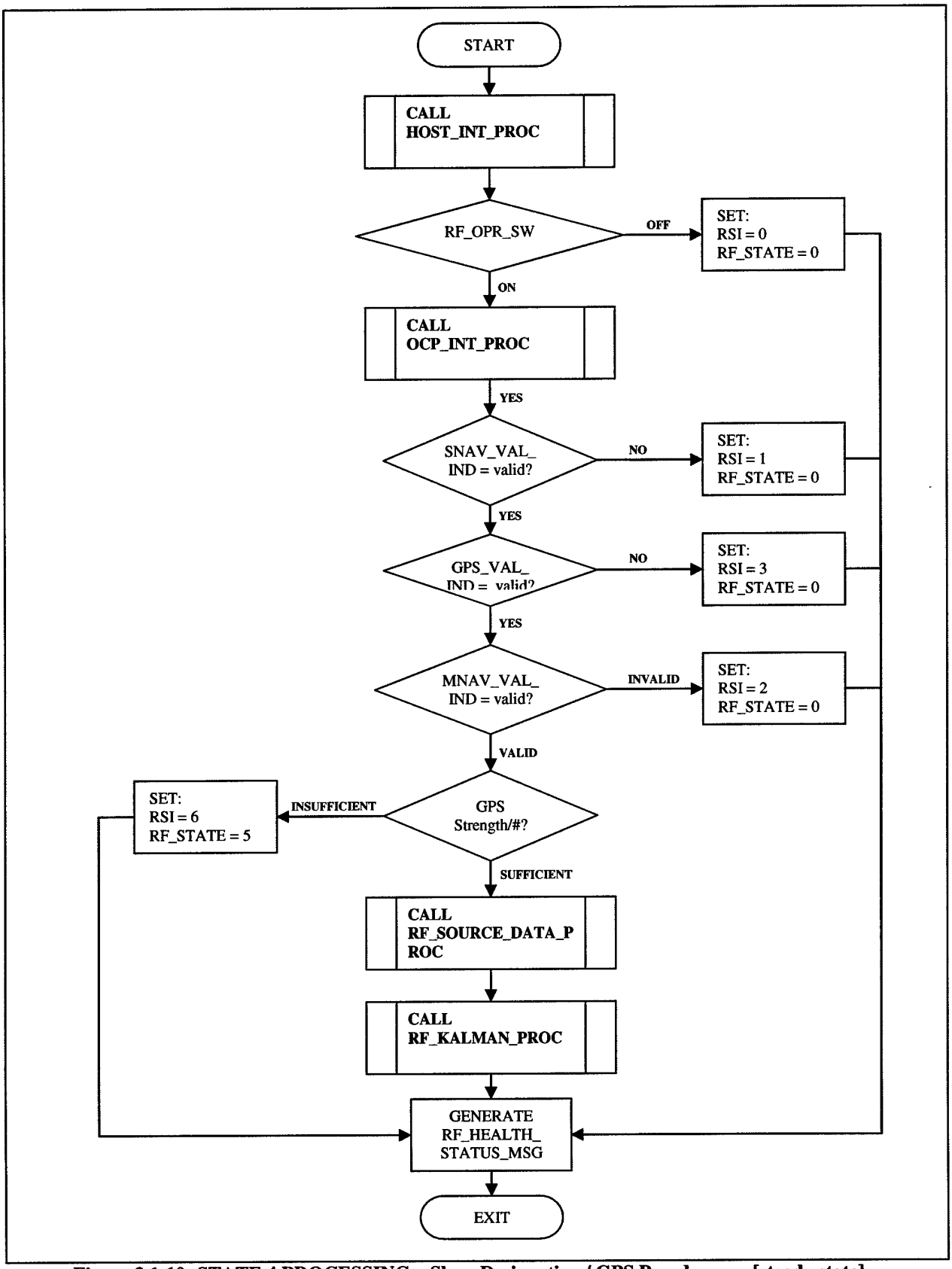


Figure 3.1-10: STATE 4 PROCESSING – Slave Designation / GPS Pseudorange [steady state]

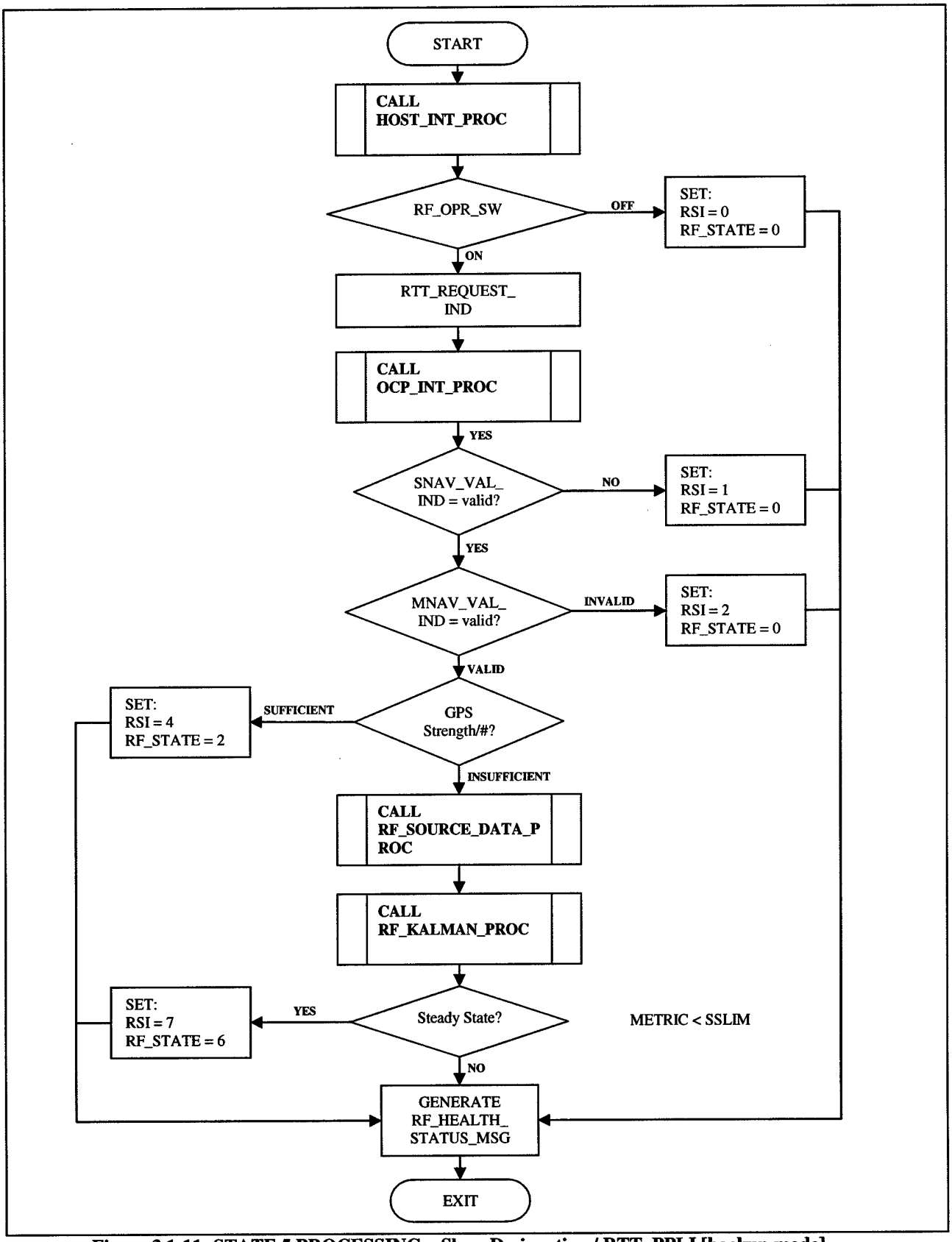


Figure 3.1-11: STATE 5 PROCESSING – Slave Designation / RTT, PPLI [backup mode]

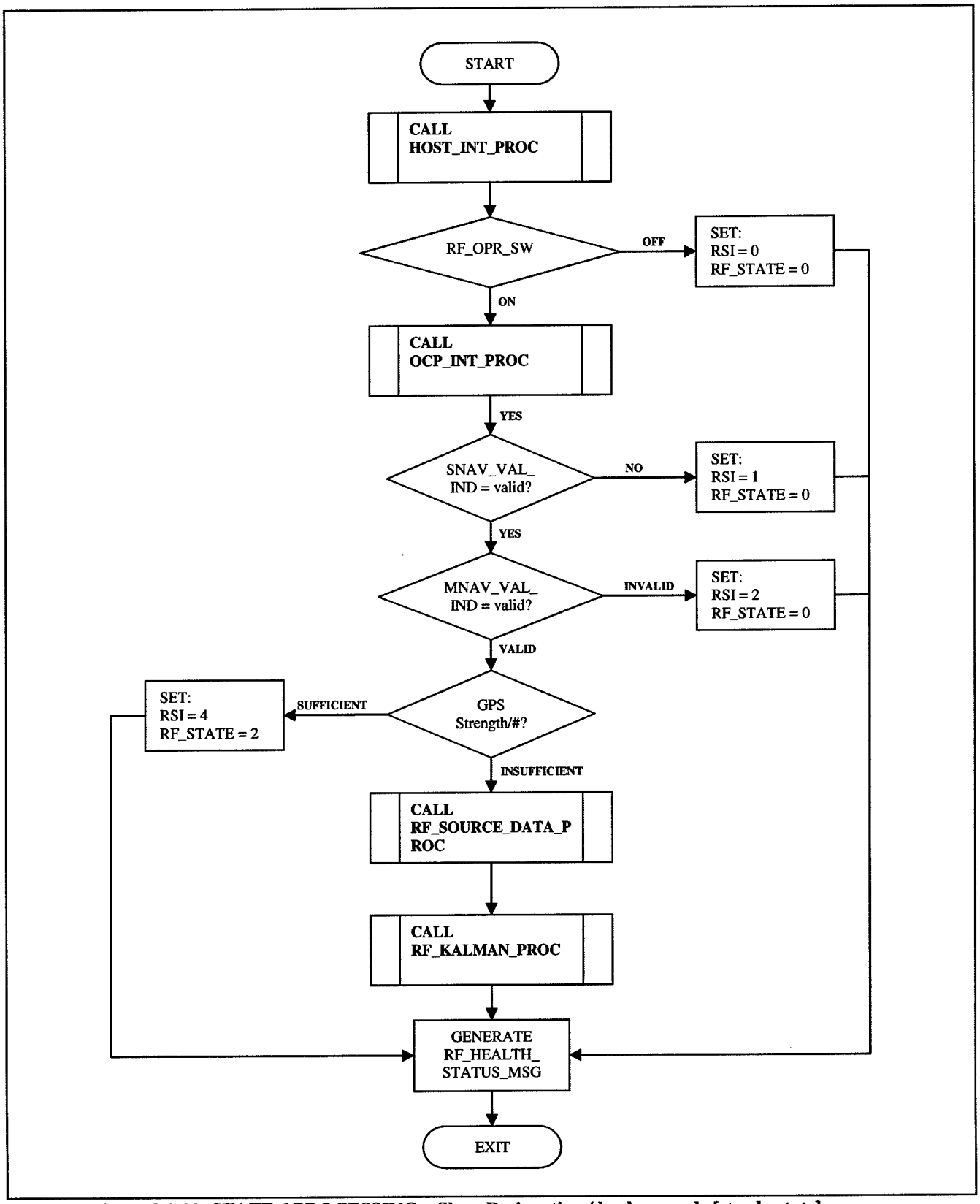


Figure 3.1-12: STATE 6 PROCESSING – Slave Designation / backup mode [steady state]

3.1.1.2 RF_SOURCE_DATA_PROC

The Refinement Filter Source Data Processing unit synchronizes and prepares data for Kalman processes. Master GPS and Navigation data are extracted from the master message. This data is then paired with the platform's own navigation solution. It is essential that all data is time synchronized before the Kalman filter can be applied.

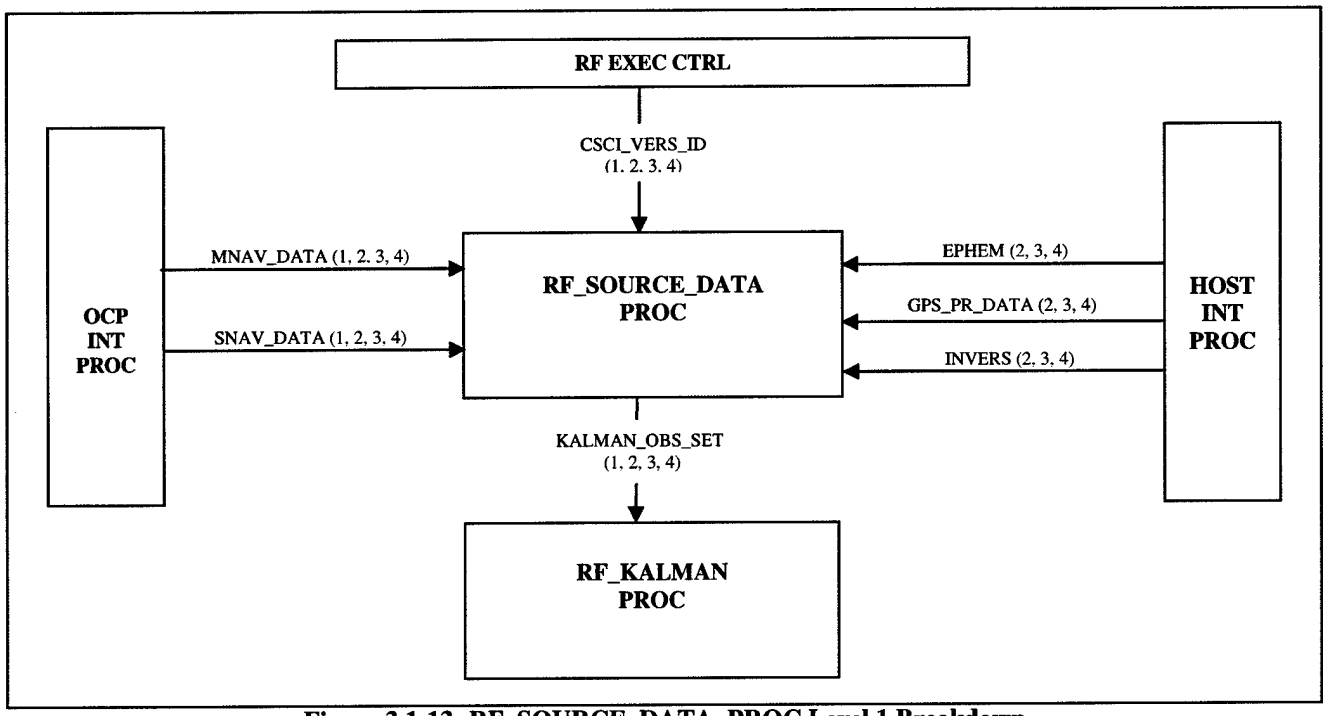


Figure 3.1-13: RF_SOURCE_DATA_PROC Level 1 Breakdown

CSC Inputs	Version	Description
CSCI_VERS_ID	1,2,3,4	Denotes operational version
GPS_PR_DATA	2,3,4	Received GPS data from HOST
INVERS (30)	2,3,4	Satellite sorting index
MNAV_DATA	1,2,3,4	MASTER navigation solution
SNAV_DATA	1,2,3,4	SELF navigation solution
EPHEM(23, 30)	2,3,4	Ephemeris Data

CSC Outputs	Description
KALMAN_OBS_SET	Data contains from Satellite or PPLI/RTT Data

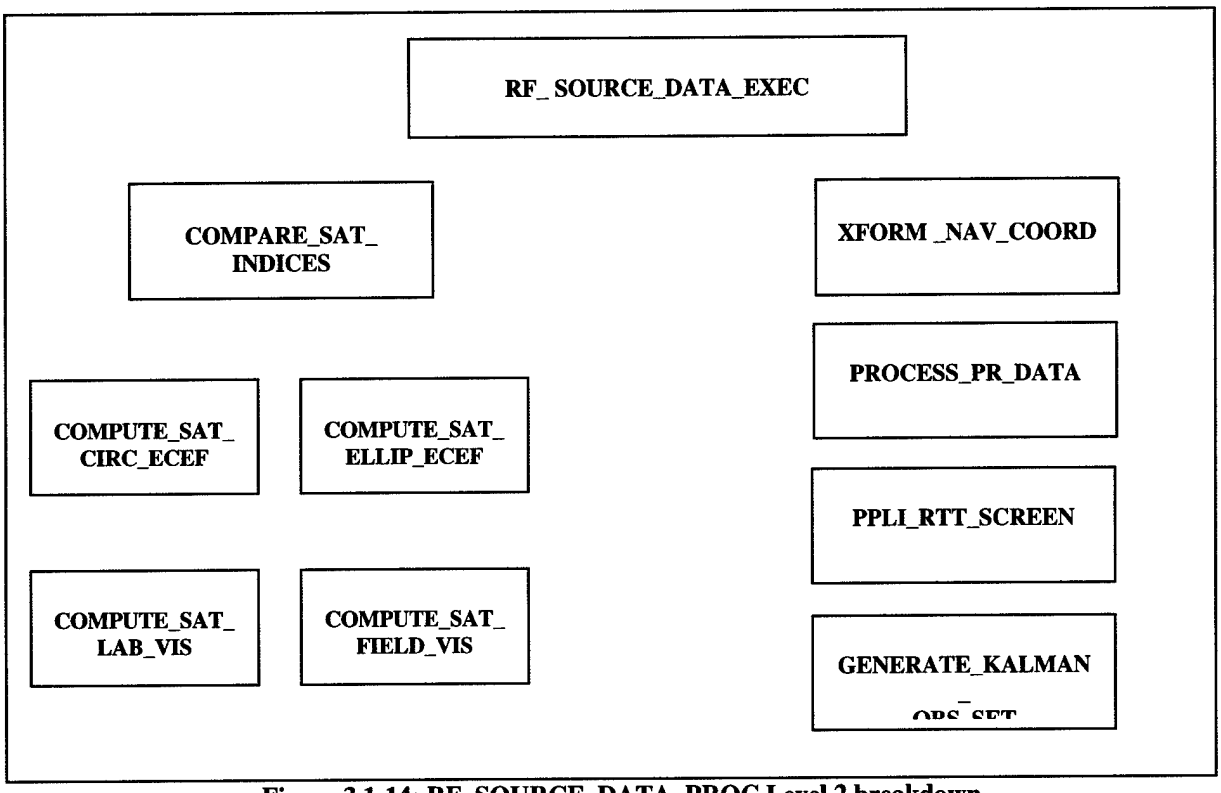


Figure 3.1-14: RF_SOURCE_DATA_PROC Level 2 breakdown

SLCSC	Description
RF_SOURCE_DATA_EXEC	Coordinates self navigation (HOST) with master navigation solution (OCP). Coordinates execution of subroutines within RF_SOURCE_DATA_PROC
COMPARE_SAT_INDICES	Compares measured satellite indices for master/slave. Selects candidate observation group for common elements.
COMPUTE_SAT_CIRC_ECEF	Computes Satellite ECEF elements for common satellites when CSCI_VERS_ID = 2.0 (Laboratory Mode)
COMPUTE_SAT_ELLIP_ECEF	Computes Satellite ECEF elements for common satellites when CSCI_VERS_ID = 3.0 (Field Mode)
COMPUTE _SAT_LAB _VIS	Only satellites at least 5o above horizon are candidates when CSCI_VERS_ID = 2.0 (Laboratory Mode)
COMPUTE _SAT_FIELD _VIS	Only satellites at least 5o above horizon are candidates when CSCI_VERS_ID = 3.0 (Field Mode)
XFORM_NAV _COORD	Transforms self-navigation into Local Level and ECEF coordinates.
PPLI_RTT_SCREEN	Called only when CSCI_VERS_ID = 4.0 (Backup Mode) Takes PPLI/RTT Data when GPS is not valid.
PROCESS_PR_DATA	Converts satellite code phase data from master and self to generate arrays of psuedorange data from all common satellites in units of meter
GENERATE_KALMAN_OBS_SET	Generates Kalman Observation set from Satellite *or* PPLI/RTT Data

Table 3.1-4: RF_SOURCE_DATA_PROC SLCSC Descriptions

3.1.1.2.1 RF_SOURCE_DATA_EXEC

This SLCSC coordinates the (asynchronous) host NAV data with master message-derived navigation data. The RF_SOURCE_DATA_EXEC controls all execution modes of system operation (i.e laboratory or field). This SLCSC also coordinates the execution of the various subordinate routines with RF_Source_Data_Processing module.

INPUTS	DESCRIPTION
MNAV_DATA	
MNAVTOV	Time of Validity of MASTER Navigation solution
MLAT	ESTIMATED MASTER Latitude position [rad]
MLATR	ESTIMATED MASTER Latitude rate [rad/s]
MLON	ESTIMATED MASTER Longitude position [rad]
MLONR	ESTIMATED MASTER Longitude rate [rad/s]
MALT	ESTIMATED MASTER Altitude [m]
MALTR	ESTIMATED MASTER Altitude rate [m/s]
TMLAT*	TRUE MASTER Latitude position [rad]
TMLATR*	TRUE MASTER Latitude rate [rad/s]
TMLON*	TRUE MASTER Longitude position [rad]
TMLONR*	TRUE MASTER Longitude rate [rad/s]
TMALT*	TRUE MASTER Altitude [m]
TMALTR*	TRUE MASTER Altitude rate [m/s]
MSAT(12)**	MASTER Ordered Satellite List
MPRR(12)**	MASTER Ordered measured code phase array
MGPSTOV**	MASTER predicted GPS time of validity
MVIS**	Number of satellites visible to MASTER
BCTRUE	TRUE MASTER Clock Bias
FCTRUE	TRUE MASTER Frequency Drift
SNAV_DATA	
SNAVTOV	Time of Validity of SELF Navigation solution
SLAT	ESTIMATED SELF Latitude position [rad]
SLATR	ESTIMATED SELF Latitude rate [rad/s]
SLON	ESTIMATED SELF Longitude position [rad]
SLONR	ESTIMATED SELF Longitude rate [rad/s]
SALT	ESTIMATED SELF Altitude [m]
SALTR	ESTIMATED SELF Altitude rate [m/s]
TSLAT*	TRUE SELF Latitude position [rad]
TSLATR*	TRUE SELF Latitude rate [rad/s]
TSLON*	TRUE SELF Longitude position [rad]
TSLONR*	TRUE SELF Longitude rate [rad/s]
TSALT*	TRUE SELF Altitude [m]
TSALTR*	TRUE SELF Altitude rate [m/s]
SSAT(12)**	SELF Ordered Satellite List
SPRR(12)**	SELF Ordered measured code phase array
SGPSTOV**	SELF predicted GPS time of validity
SVIS**	Number of satellites visible to SELF
BCTRUE	TRUE SELF Clock Bias
FCTRUE	TRUE SELF Frequency Drift
INVERS(30)**	Satellite sorting index
EPHEM(30,23)**	Satellite Ephemeris Data
CSCI_VERS_IND	
NVS	
RF_STATE	

Table 3.1-5: RF_SOURCE_DATA_EXEC Inputs

OUTPUTS	DESCRIPTION
MNAV_DATA	
MNAVTOV	Time of Validity of MASTER Navigation solution
MLAT	ESTIMATED MASTER Latitude position [rad]
MLATR	ESTIMATED MASTER Latitude rate [rad/s]
MLON	ESTIMATED MASTER Longitude position [rad]
MLONR	ESTIMATED MASTER Longitude rate [rad/s]
MALT	ESTIMATED MASTER Altitude [m]
MALTR	ESTIMATED MASTER Altitude rate [m/s]
TMLAT*	TRUE MASTER Latitude position [rad]
TMLATR*	TRUE MASTER Latitude rate [rad/s]
TMLON*	TRUE MASTER Longitude position [rad]
TMLONR*	TRUE MASTER Longitude rate [rad/s]
TMALT*	TRUE MASTER Altitude [m]
TMALTR*	TRUE MASTER Altitude rate [m/s]
MSAT(12)**	MASTER Ordered Satellite List
MPRR(12)**	MASTER Ordered measured code phase array
MGPSTOV**	MASTER predicted GPS time of validity
MVIS**	Number of satellites visible to MASTER
BCTRUE	TRUE MASTER Clock Bias
FCTRUE	TRUE MASTER Frequency Drift
SNAV_DATA	
SNAVTOV	Time of Validity of SELF Navigation solution
SLAT	ESTIMATED SELF Latitude position [rad]
SLATR	ESTIMATED SELF Latitude rate [rad/s]
SLON	ESTIMATED SELF Longitude position [rad]
SLONR	ESTIMATED SELF Longitude rate [rad/s]
SALT	ESTIMATED SELF Altitude [m]
SALTR	ESTIMATED SELF Altitude rate [m/s]
TSLAT*	TRUE SELF Latitude position [rad]
TSLATR*	TRUE SELF Latitude rate [rad/s]
TSLON*	TRUE SELF Longitude position [rad]
TSLONR*	TRUE SELF Longitude rate [rad/s]
TSALT*	TRUE SELF Altitude [m]
TSALTR*	TRUE SELF Altitude rate [m/s]
SSAT(12)**	SELF Ordered Satellite List
SPRR(12)**	SELF Ordered measured code phase array
SGPSTOV**	SELF predicted GPS time of validity
SVIS**	Number of satellites visible to SELF
BCTRUE	TRUE SELF Clock Bias
FCTRUE	TRUE SELF Frequency Drift
INVERS(30)**	Satellite sorting index
EPHEM(30,23)**	Satellite Ephemeris Data
CSCI_VERS_IND	
NVS	
RF_STATE	

Table 3.1-6: RF_SOURCE_DATA_EXEC Outputs

* - Provided only if CSCI_VERS_IND = 2 (Laboratory Mode)
** - Provided only if CSCI_VERS_IND = 3 (Field Mode)

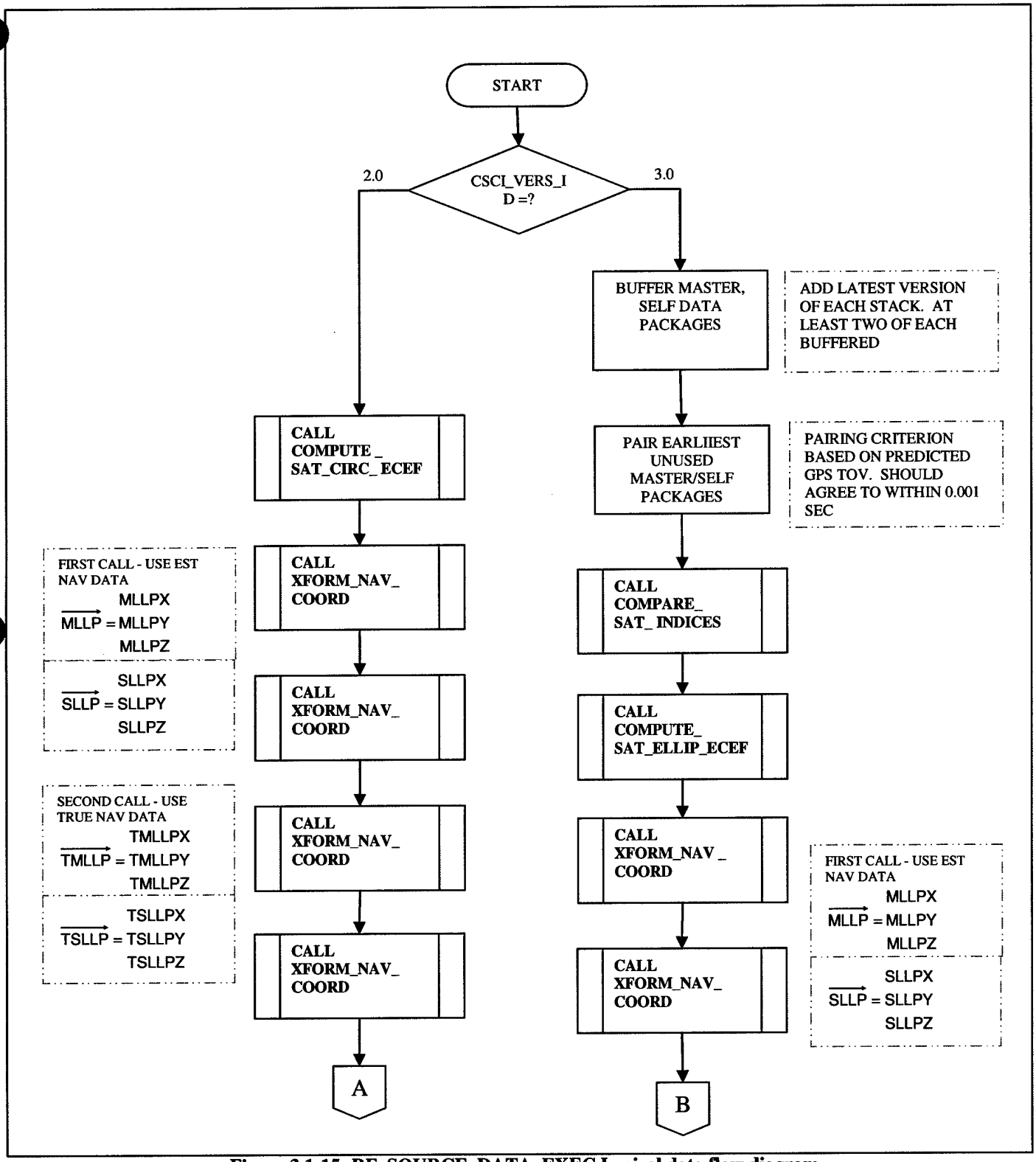


Figure 3.1-15: RF_SOURCE_DATA_EXEC Logical data flow diagram

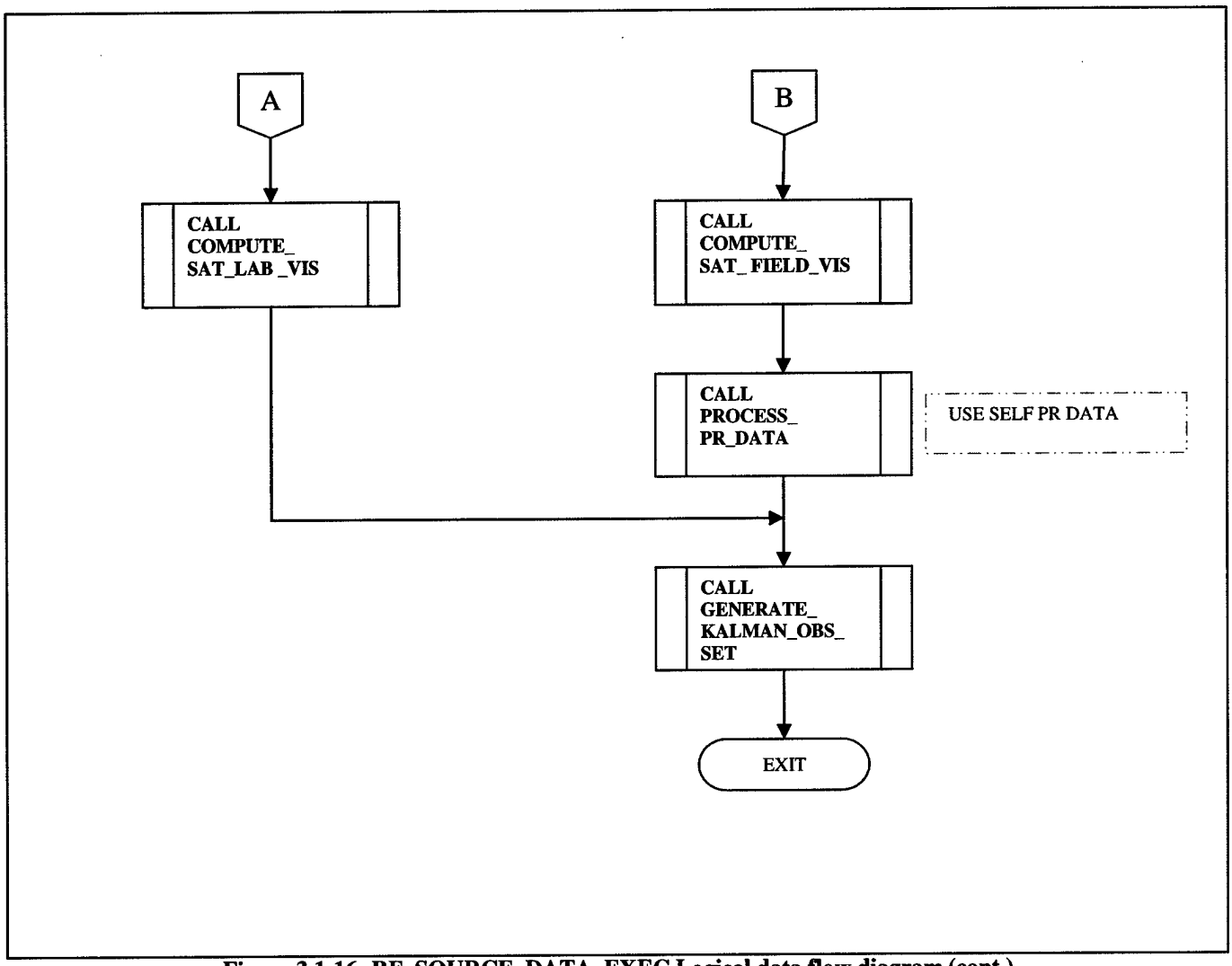


Figure 3.1-16: RF_SOURCE_DATA_EXEC Logical data flow diagram (cont.)

3.1.1.2.2 COMPARE_SAT_INDICES

COMPARE_SAT_INDICES compares ordered satellite indices for master/slave and selects the common satellites. These satellites are aligned in a single array containing data required for pseudorange comparison.

INPUTS	DESCRIPTION
MSAT(12)	Ordered array of MASTER visible satellites
SSAT(12)	Ordered array of SELF visible satellites
MPRR(12)	Ordered array of MASTER code phase
SPRR(12)	Ordered array of SELF code phase
OUTPUTS	**DESCRIPTION**
NVS	Number visible satellites - common to MASTER and SELF
SAT(NVS)	Ordered array of common visible satellites
PRR1(NVS)	MASTER array of code phase for common satellites
PRR2(NVS)	SELF array of code phase for common satellites
MVIS	Number of satellites visible to MASTER
SVIS	Number of satellites visible to SELF

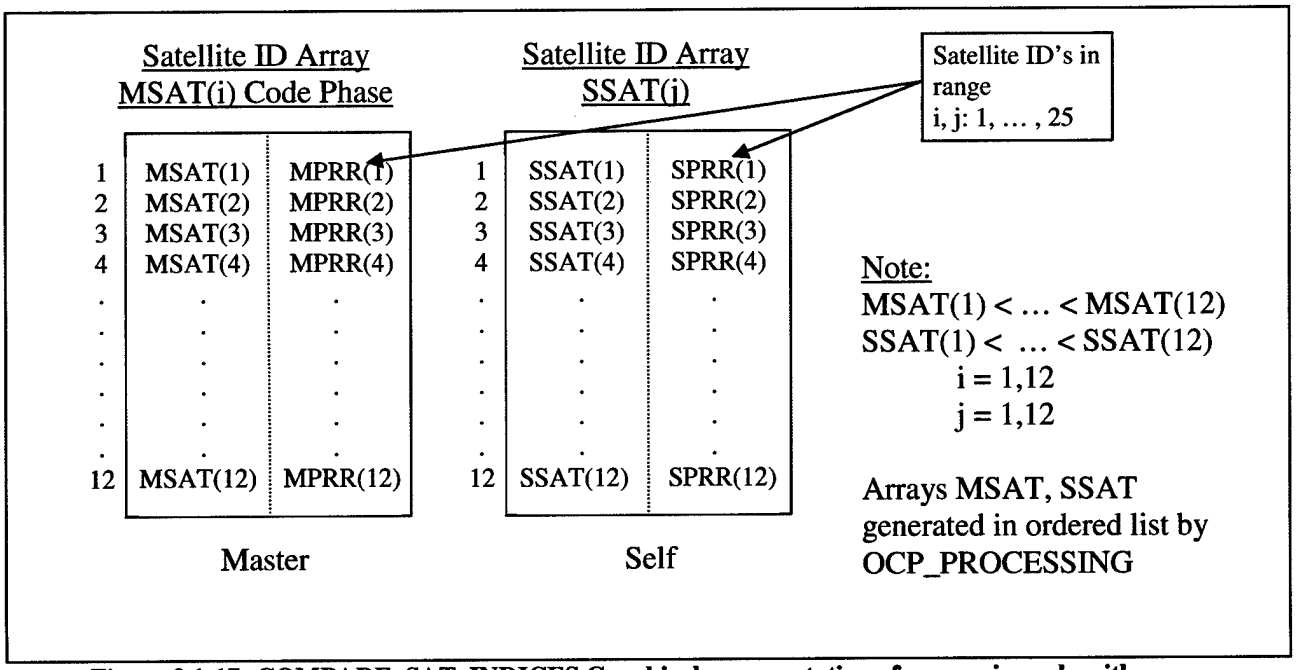


Figure 3.1-17: COMPARE_SAT_INDICES Graphical representation of comparison algorithm

Comparison Algorithm:

```
FOR i = 1, MVIS
   FOR j = 1, SVIS
      IF MSAT(i) = SSAT(j)
            THEN store satellite ID in array SAT(NVS) and pseudoranges in
            corresponding PRR1(NVS) and PRR2(NVS) arrays
      IF SSAT(j) > MSAT(i) end inner loop
```

$$\begin{bmatrix} SAT(1) \\ SAT(2) \\ \cdot \\ \cdot \\ \cdot \\ SAT(NVS) \end{bmatrix}$$ =Ordered list of matched satellite ID's

Where:
NVS = # of Matched Satellites
SAT(j) - Satellite indices satisfying match

Code phase pseudorange data for satellites contained in ordered array SAT(NVS) is transferred to corresponding processed raw pseudorange arrays containing only data for matched satellites:
MPRR(12) → PRR1(NVS)
SPRR(12) → PRR2(NVS)

Algorithm Pseudocode:

NVS = 0, SAT(i) = 0, i = 1, 12

```
DO X I = 1, MVIS
DO Y J = 1, SVIS
   CASE: MSAT(I) = SSAT(J)
         NVS = NVS + 1
         SAT(NS) = MSAT(I)
         PRR1(NS) = MPRR(I)
         PRR2(NS) = SPRR(J)
         EXIT Y LOOP

   CASE: MSAT(I) < SSAT(J)
         EXIT Y LOOP

   CASE: MSAT(I) > SSAT(J)
         CONTINUE Y LOOP
Y CONTINUE
X CONTINUE
```

3.1.1.2.3 COMPUTE_ SAT_CIRC_ECEF

This SLCSC performs the conversion of the circular satellite constellation coordinates into Earth Centered Earth Fixed coordinates for the laboratory model.

Conditions:
CSCI_VERS_IND = 2.0 (Laboratory Mode)
GPS_VAL_IND = valid

INPUTS	DESCRIPTION
SGPSTOV	GPS system time
NVS	Number visible satellites - common to MASTER and SELF
SAT(NVS)	Ordered array of common visible satellites
OUTPUTS	**DESCRIPTION**
ECEFXS(K)	ECEF x-coordinate for satellite K
ECEFYS(K)	ECEF y-coordinate for satellite K
ECEFZS(K)	ECEF z-coordinate for satellite K

Stored Constants:

```
OMEGA0(K), K = 1, 30                Lat coordinates of satellites
ETA0(I), K = 1, 30                  Lon coordinates of satellites
OMEGAE = 7.292115467 E-5 [RAD/S]    Earth's rotational rate
OMEGAS = 1.45858522 E-4 [RAD/S]     Satellite orbital rate
RK = 87051108 [meters]              Radius of satellite orbit
IOTA = 55.0/2*pi [RAD]              Satellite Inclination
```

Circular constellation propagation algorithm:

```
FOR K = 1, NVS
      OMEGAK = OMEGA0 (SAT (K)) - TC * OMEGE
      ETAK   = ETA (SAT (K))    + TC * OMEGS

      ECEF (1, 1) = XKP*DCOS (OMEGAK) - YKP*DSIN (OMEGAK)*DCOS (IOTA)
      ECEF (2, 1) = XKP*DSIN (OMEGAK) + YKP*DCOS (OMEGAK)*DCOS (IOTA)
      ECEF (3, 1) = YKP*DSIN (IOTA)
```

3.1.1.2.4 COMPUTE_SAT_ELLIP_ECEF.

This SLCSC performs the conversion of Elliptical satellite constellation coordinates into ECEF coordinates for the real flight model.

INPUTS	DESCRIPTION
EPHEM(23,30)	Satellite Ephemeris Data
NVS	Number visible satellites - common to MASTER and SELF
SAT(NVS)	Ordered array of common visible satellites
INVERS(30)	Satellite sorting index
SGPSTOV	Predicted GPS time
PRR2(NVS)	SELF array of code phase for common satellites
OUTPUTS	**DESCRIPTION**
ECEFXS(L)	ECEF x-coordinate for satellite L
ECEFYS(L)	ECEF y-coordinate for satellite L
ECEFZS(L)	ECEF z-coordinate for satellite L

Satellite Ephermis Data for all satellites (visible or not). Ephermis Data Indexed in Following Order:

SVNO = EPHEM (I, 1)
WEEKNO = EPHEM (I, 2)
TGD = EPHEM (I, 3)
TOC = EPHEM (I, 4)
AF2 = EPHEM (I, 5)
AF1 = EPHEM (I, 6)
AF0 = EPHEM (I, 7)
CRS = EPHEM (I, 8)
DELTAN = EPHEM (I, 9)
M0 = EPHEM (I, 10)
CUC = EPHEM (I, 11)
ES = EPHEM (I, 12)
CUS = EPHEM (I, 13)
SQRTAS = EPHEM (I, 14)
TOE = EPHEM (I, 15)
CIC = EPHEM (I, 16)
OMEGA0 = EPHEM (I, 17)
CIS = EPHEM (I, 18)
IOTA0 = EPHEM (I, 19)
CRC = EPHEM (I, 20)
OMEGA = EPHEM (I, 21)
OMEGADT = EPHEM (I, 22)
IOTADT = EPHEM (I, 23)

NOTE: I is EPHEM array index, not satellite #, We will later use INVERS array to reference desired satellite index.

Dimension of EPHEM

I index is determined by order that GPS reports PR data. Host interface processing will provide INVERS array to allow correct indexing

Conditions:
CSCI_VERS_ID = 3.0 (Field Mode)
GPS_VAL_IND = valid

Stored Constants:

```
RK = 87051108 [meters]          Radius of satellite orbit
C =  2.99792498 E8[m/s]         Speed of light in vacuum
CR                              Code phase conversion factor
OMEGAE = 7.292118467 E-5        Earth's rotational rate
OMEGAS = 1.45858522 E-4         Satellite orbital rate
MU = 3.986008 E14
F = -4.442807633 E-10           ?
```

Algorithm Pseudocode:

```
Compute psuedorange
  IPGPS = PGPST                 Integer value of PGPST
  FPGPS = PGPST - FLOAT(IPGPS)  Fractional GPS
  PRSEC = FPGPS - (CDPHSE/CR)   Pseudorange in seconds from code phase
  IF (PRSEC < 0.0) THEN         Numerical validity check
       PRSEC = 1.0 + PRSEC

  PR = PRSEC*C                  Multiply by C for true pseudorange
  TC = PGPSTM - PR/C            Coarse GPS system time
  TOE = EPHEM(INVERS(ISVNO),15) Ephemeris reference time (from Epoch)
  TK = TC - TOE                 Difference between TC and TOE

  IF (TK > 302400.0) THEN
       TC = TC - 604800.0
  IF (TK < 302400.0) THEN
       TC = TC + 604800.0

Compute mean motion
  DELN = EPHEM(INVERS(ISVNO),9)   mean motion difference [semicircle/s]
  SAS = EPHEM(INVERS(ISVNO),14)   Square root of semi-major axis [m^1/2]
  AS = SAS^2                      Semi-major axis
  N = DSQRT(MU/AS^3) + DELN       Computed mean motion

Compute mean anomaly
  M0 = EPHEM(INVERS(ISVNO),10)    Mean anomaly at reference time [smcir]
  MK = M0 + N*(TC - TOE)          Mean anomaly

Solve for eccentric anomaly       E + MK + ES*SIN(EOLD) [Newton-Raphson]
  ES = EPHEM(INVERS(ISVNO),12)    Eccentricity
  EKNEW = MK                      Kepler's equation for Eccen. anomaly
  FOR JX:1:100 DO
       EKOLD = EKNEW
       EKNEW = MK + ES*SIN(EKOLD)
  END
```

```
  E = EKNEW                                  Eccentricity anomaly

Compute time correction term
  DEKTR = F*ES*SAS*SIN(E)                    Relativistic correction term

  AF0 = EPHEM(INVERS(ISVNO),7)               7 -
  AF1 = EPHEM(INVERS(ISVNO),6)               6 -
  AF2 = EPHEM(INVERS(ISVNO),5)               5 -
  TGD = EPHEM(INVERS(ISVNO),3)               3 -
  TOC = EPHEM(INVERS(ISVNO),4)               4 -
  DELT = AF0 + AF1*(TC - TOC) +              Transit time
       AF2*(TC - TOC)^2 + DELTR - TGD

  T = TC - DELT                              GPS time of transmission correction
  R = AS*(1.0 - ES*SIN(E))                   Satellite average radius R

Compute satellite true anomaly, NU
  NUM = SQRT(1.0 - ES*ES)*SIN(E))/           Sin(true anomaly)
              (1.0 - ES*COS(E))
  DENOM = (COS(E)-ES)/(1.0-ES*COS(E))        Cos(true anomaly)
  NU = ATAN2(NUM, DENOM)                     Four quadrant inverse tangent
  W = EPHEM(INVERS(ISVNO),21)                Argument of perigee [semicircle]
  PHI = NU + W                               Argument of latitude

Compute correction terms from harmonics
  CUS = EPHEM(INVERS(ISVNO),13)              Amplitude of sine harmonic correction
                                             term to the argument of latitude [rad]

  CUC = EPHEM(INVERS(ISVNO),11)              Amplitude of cos harmonic correction
                                             term to the argument of latitude [rad]

  CRS = EPHEM(INVERS(ISVNO),8)               Amplitude of sine harmonic correction
                                             term to orbit radius [m]

  OMEGA0 = EPHEM(INVERS(ISVNO),17)           Longitude of ascending node of orbit
                                             plane at weekly epoch

  CIS = EPHEM(INVERS(ISVNO),18)              Amplitude of sine harmonic correction
                                             term to the angle of inclination [rad]

  CRC = EPHEM(INVERS(ISVNO),20)              Amplitude of cos harmonic correction
                                             term to the orbit radius [m]

  CIC = EPHEM(INVERS(ISVNO),16)              Amplitude of cos harmonic correction
                                             term to the angle of inclination [rad]

  IOTA = EPHEM(INVERS(ISVNO),19)             Inclination angle at reference time
                                             [semicircle]

  IDOT = EPHEM(INVERS(ISVNO),23)             Rate of inclination angle [semicir/s]

  C2PHI = COS(2*PHI)                         Double angle
  S2PHI = SIN(2*PHI)

  Second harmonic perturbations
  DELPHI = CUS*S2PHI + CUC*C2PHI             Argument of latitude correction
  DELR = CRS*S2PHI + CRC*C2PHI               Radius correction
  DELI = CIS*S2PHI + CIC*C2PHI               Corrected inclination
```

```
PHI = PHI + DELPHI                      Corrected argument of latitude
R = R + DELR                            Corrected radius
IOTA = IOTA + DELI + IDOT*(T-TOE)       Corrected inclination

OMEGAD = EPHEM(INVERS(ISVNO),22)        Rate of right ascension [semicircle/s]
OER = OMEGA0 + OMEGAD*(T-TOE) -         Corrected longitude of ascending node
      OMEGAE*T
```

Compute satellite ECEF vector (meters)

```
ECEFXS(L) = R*COS(OER)* COS(PHI)-R*SIN(OER)*COS(IOTA)*SIN(PHI)
ECEFYS(L) = R*SIN(OER)* COS(PHI)+R*COS(OER)*COS(IOTA)*SIN(PHI)
ECEFZS(L) = R*SIN(IOTA)*SIN(PHI)
```

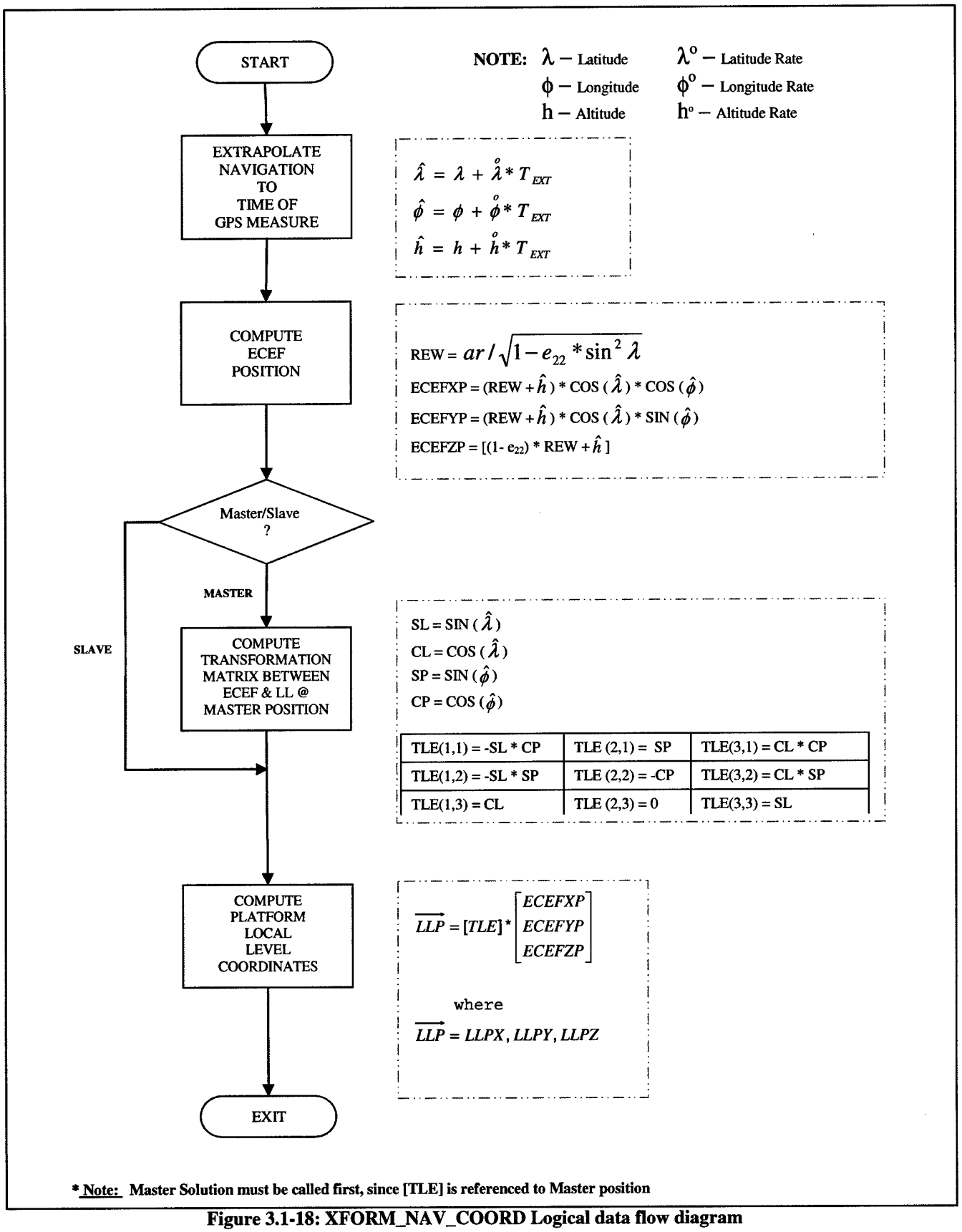
3.1.1.2.5 XFORM_NAV_COORD

The responsibility of XFORM_NAV_COORD is to extrapolate both self and master navigation solution into ECEF coordinates and local level coordinate values. This process will be involved for both self and master navigation solutions.

INPUTS	DESCRIPTION
MS_IND	MASTER/SLAVE indicator
NAVTOV	Time of Validity of MASTER Navigation solution
LAT	Latitude position [rad]
LATR	Latitude rate [rad/s]
LON	Longitude position [rad]
LONR	Longitude rate [rad/s]
ALT	Altitude [m]
ALTR	Altitude rate [m/s]
OUTPUTS	**DESCRIPTION**
LLPX	Local level x-coordinate value
LLPY	Local level y-coordinate value LLP
LLPZ	Local level z-coordinate value
TLE (3,3)	Local Level transformation matrix

Stored Constants:

ar = 2.09257382 E7 Earth semi-minor radius [ft]
e22 = 6.74499984 E-3 Earth eccentricity squared



* Note: Master Solution must be called first, since [TLE] is referenced to Master position

Figure 3.1-18: XFORM_NAV_COORD Logical data flow diagram

3.1.1.2.6 COMPUTE_LAB_SAT_VIS

COMPUTE_LAB_SAT_VIS restricts DGPS satellite usage to set of satellites at least 5 degrees above the horizon. This SLCSC performs the computation of satellite direction cosines CXS, CYS, CZS for all candidate satellites. This block also computes predicted and true pseudorange values between master and self to all selected satellites

INPUTS	DESCRIPTION
NVS	Number visible satellites - common to MASTER and SELF
SAT(NVS)	Ordered array of common visible satellites
ECEFXS(K)	ECEF x-coordinate for satellite K
ECEFYS(K)	ECEF y-coordinate for satellite K
ECEFZS(K)	ECEF z-coordinate for satellite K
MLLPX	ESTIMATED MASTER local level x-coordinate value
MLLPY	ESTIMATED MASTER local level y-coordinate value
MLLPZ	ESTIMATED MASTER local level z-coordinate value
SLLPX	ESTIMATED SLAVE local level x-coordinate value
SLLPY	ESTIMATED SLAVE local level y-coordinate value
SLLPZ	ESTIMATED SLAVE local level z-coordinate value
TMLLPX	TRUE MASTER local level x-coordinate value
TMLLPY	TRUE MASTER local level y-coordinate value
TMLLPZ	TRUE MASTER local level z-coordinate value
TSLLPX	TRUE SLAVE local level x-coordinate value
TSLLPY	TRUE SLAVE local level y-coordinate value
TSLLPZ	TRUE SLAVE local level z-coordinate value
TLE()	Local Level transform matrix
XA(k)	Refinement Filter solution vector, k = 1,7

OUTPUTS	DESCRIPTION
SCOUNT	Final number of paired satellites visible above ANGLIM
SATVIS(SCOUNT)	Ordered array of common visible satellites above ANGLIM
EMPR(SCOUNT)	PREDICTED Pseudorange Vector to satellites from MASTER
ESPR(SCOUNT)	PREDICTED Pseudorange Vector to satellites from SELF
TMPR(SCOUNT)	TRUE Pseudorange Vector to satellites from MASTER
TSPR(SCOUNT)	TRUE Pseudorange Vector to satellites from SELF
CXS(SCOUNT)	Direction cosine x-component
CYS(SCOUNT)	Direction cosine y-component
CZS(SCOUNT)	Direction cosine z-component

Stored Constants:

ANGLIM = 5 degrees	Angle of inclination that marks the boundary of visibility [rad]

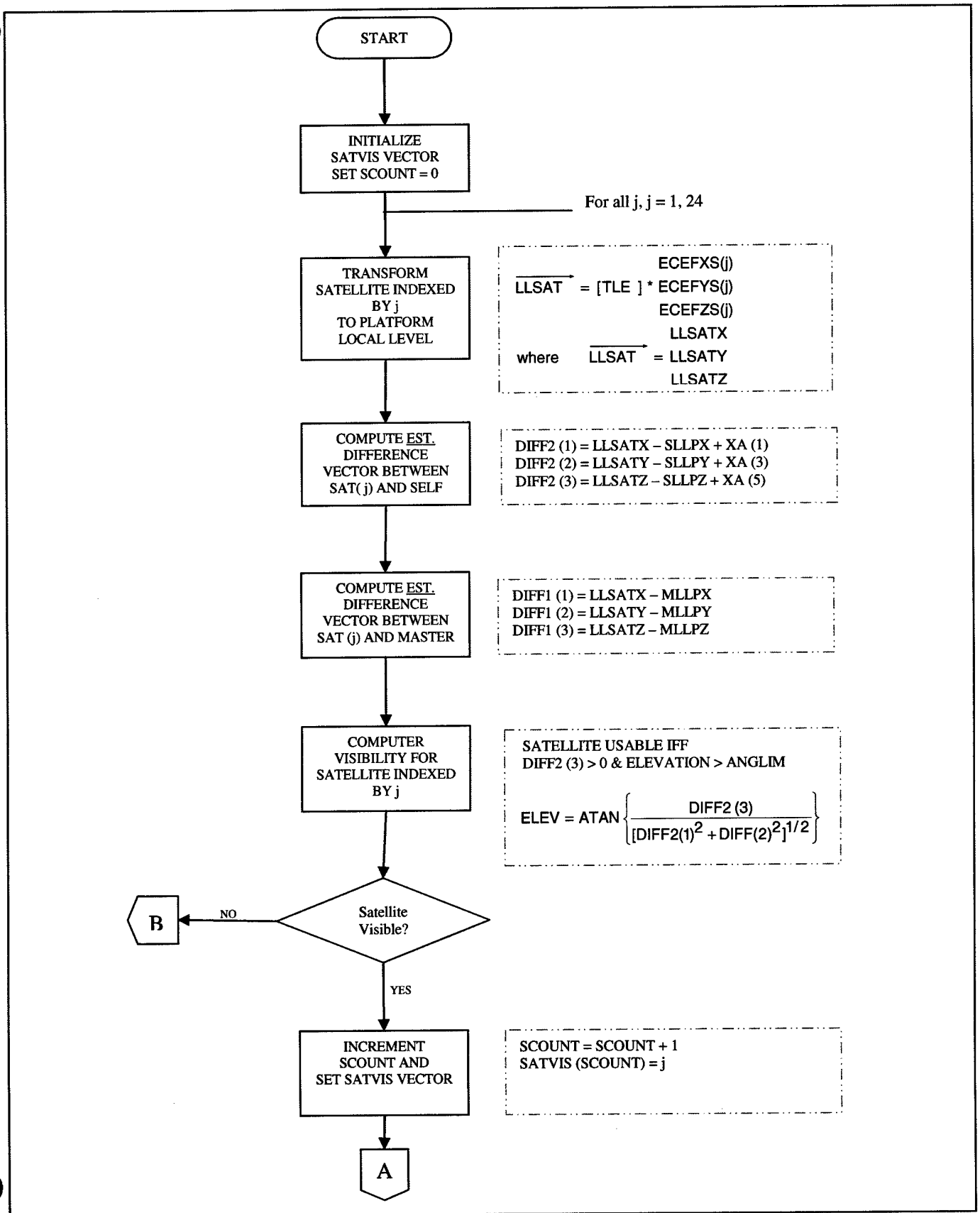


Figure 3.1-19: COMPUTE_SAT_LAB_VIS Logical data flow diagram

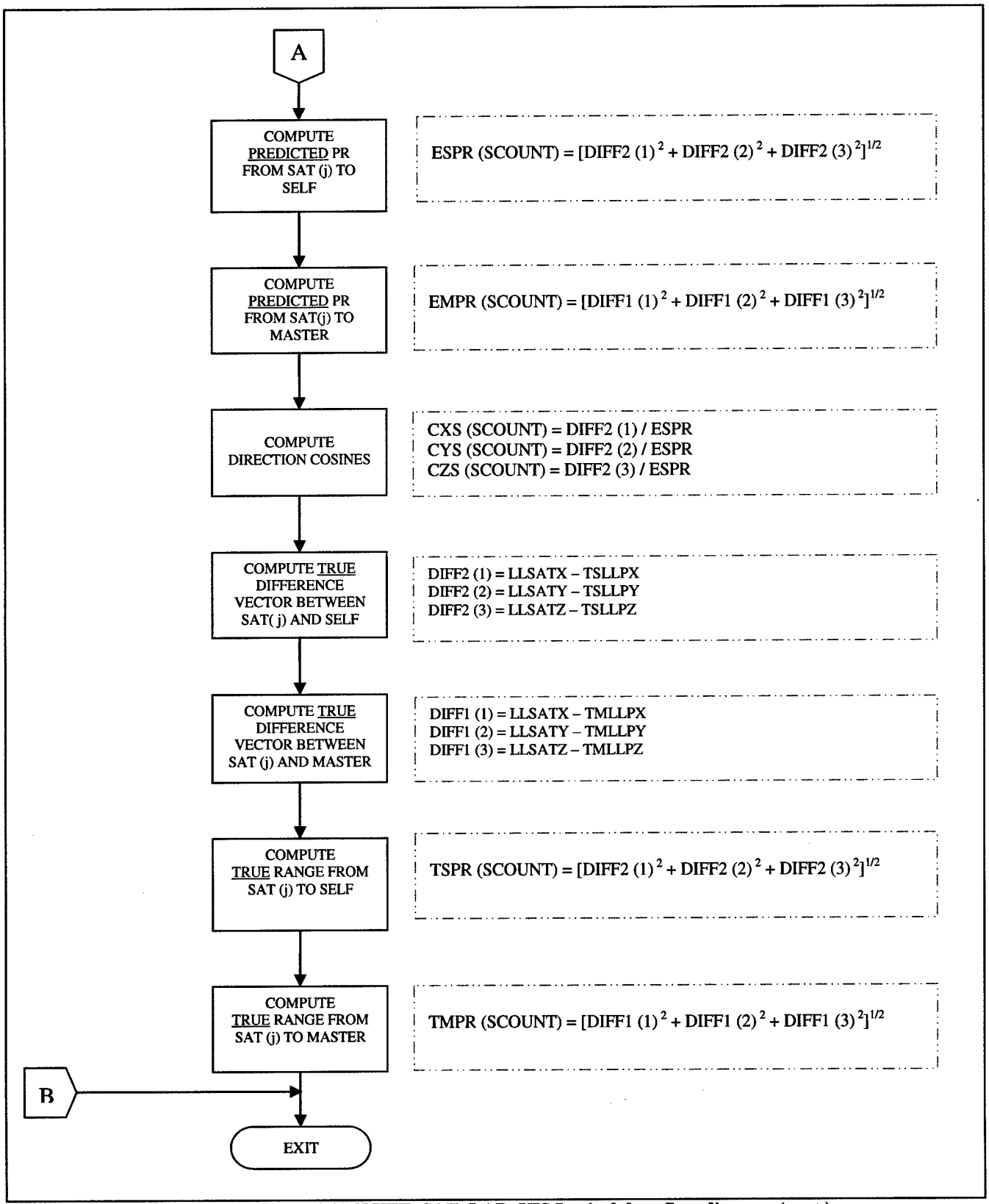


Figure 3.1-20: COMPUTE_SAT_LAB_VIS Logical data flow diagram (cont.)

3.1.1.2.7 PROCESS_PR_DATA

PROCESS_PR_DATA performs a conversion of satellite codephase data to generate an array of pseudorange data between the master or self and common visible satellites.

INPUTS	DESCRIPTION
SCOUNT	Final paired satellite count satisfying elevation criteria
SATVIS(K), K = 1,SCOUNT	Satellite Visibility vector
NVS	Number of satellites paired, but not yet checked for elevation
MS_IND	Master/Slave data indicator
PRR1(j), j=1,NVS	MASTER sorted code phase data
PRR2(j), j=1,NVS	SELF sorted code phase data
MPGPSTM(j), j=1,NVS	MASTER sorted predicted GPS time vector
SPGPSTM(j), j=1,NVS	SELF sorted predicted GPS time vector

OUTPUTS	DESCRIPTION
EMPR(K)	MASTER predicted pseudorange vector [m]
ESPR(K)	SELF predicted pseudorange vector [m]

Stored Constants:

C = 2.99792498 E8 [m/s]	Speed of light
CR = TBD	Code Phase conversion constant

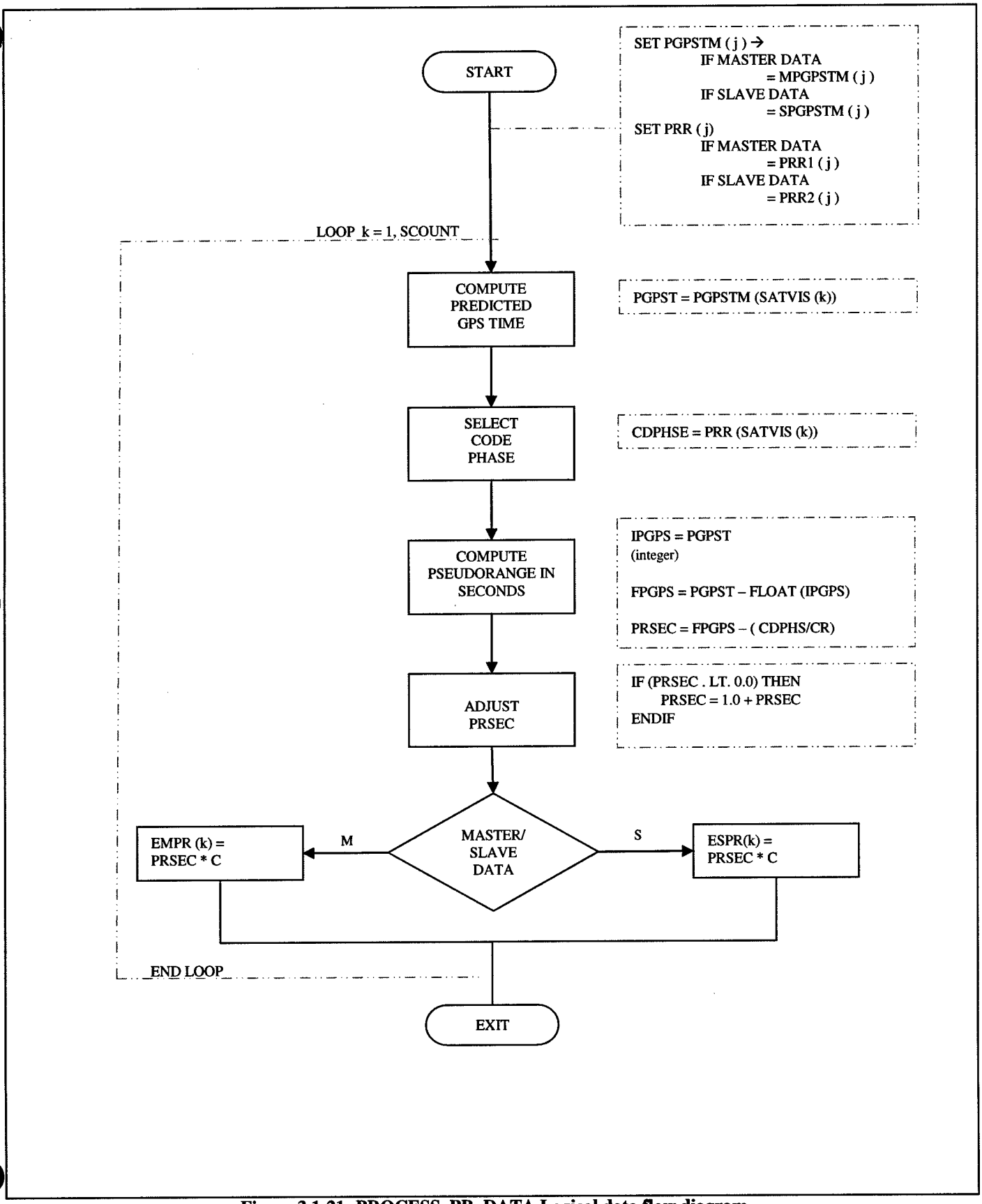


Figure 3.1-21: PROCESS_PR_DATA Logical data flow diagram

3.1.1.2.8 GENERATE_KALMAN_OBS_SET

GENERATE_KALMAN_OBS_SET generates the Refinement Filter measurement array, Y, and the expected measurement array, YEXP. The SLCSC also generates the appropriate observation matrix, [H] from the array of visible satellites. This routine shall be capable of supporting both primary mode (DGPS) observations, as well as the backup (PPLI/RTT) mode.

INPUTS	DESCRIPTION
SCOUNT*	Number of selected observations
EMPR(SCOUNT)*	EST(predicted) pseudorange array from MASTER
ESPR(SCOUNT)*	EST(predicted) pseudorange array from SELF
TMPR(SCOUNT)*	TRUE(measured) pseudorange array from MASTER
TSPR(SCOUNT)*	TRUE(measured) pseudorange array from SELF
SATVIS(K)*, K = 1, NS	Satellite Visibility vector
MLLPX	ESTIMATED MASTER local level x-coordinate value
MLLPY	ESTIMATED MASTER local level y-coordinate value
MLLPZ	ESTIMATED MASTER local level z-coordinate value
SLLPX	ESTIMATED SLAVE local level x-coordinate value
SLLPY	ESTIMATED SLAVE local level y-coordinate value
SLLPZ	ESTIMATED SLAVE local level z-coordinate value
TMLLPX	TRUE MASTER local level x-coordinate value
TMLLPY	TRUE MASTER local level y-coordinate value
TMLLPZ	TRUE MASTER local level z-coordinate value
TSLLPX	TRUE SLAVE local level x-coordinate value
TSLLPY	TRUE SLAVE local level y-coordinate value
TSLLPZ	TRUE SLAVE local level z-coordinate value
BPR2M**	Backup measured range to master
BBC2M**	Backup measured clock bias to master
TC	Observation set time of validity
RF_STATE	State of the refinement filter
X(A), A = 1, 8	Current RF state vector
CXS(SCOUNT)	Direction cosine x-component
CYS(SCOUNT)	Direction cosine y-component
CZS(SCOUNT)	Direction cosine z-component

OUTPUTS	DESCRIPTION
HA(12,8)	Observation matrix for Kalman Filter
YA(14)	Inputs to the Kalman Filter
YAEXP(14)	Expected Kalman solutions
SCOUNT	Number of selected observations
TC	Observation set time of validity
SATVIS(K)*, K = 1, NS	Satellite Visibility vector
DXTRUE	TRUE Error composite x-coordinate value
DYTRUE	TRUE Error composite y-coordinate value
DZTRUE	TRUE Error composite z-coordinate value

* Required when CSCI_VERS_ID =1,2, or 3 (primary mode only)
** Required when CSCI_VERS_ID = 4 (Backup Mode)

Stored Constants:

C = 0.299792498 [m/ns] Speed of light

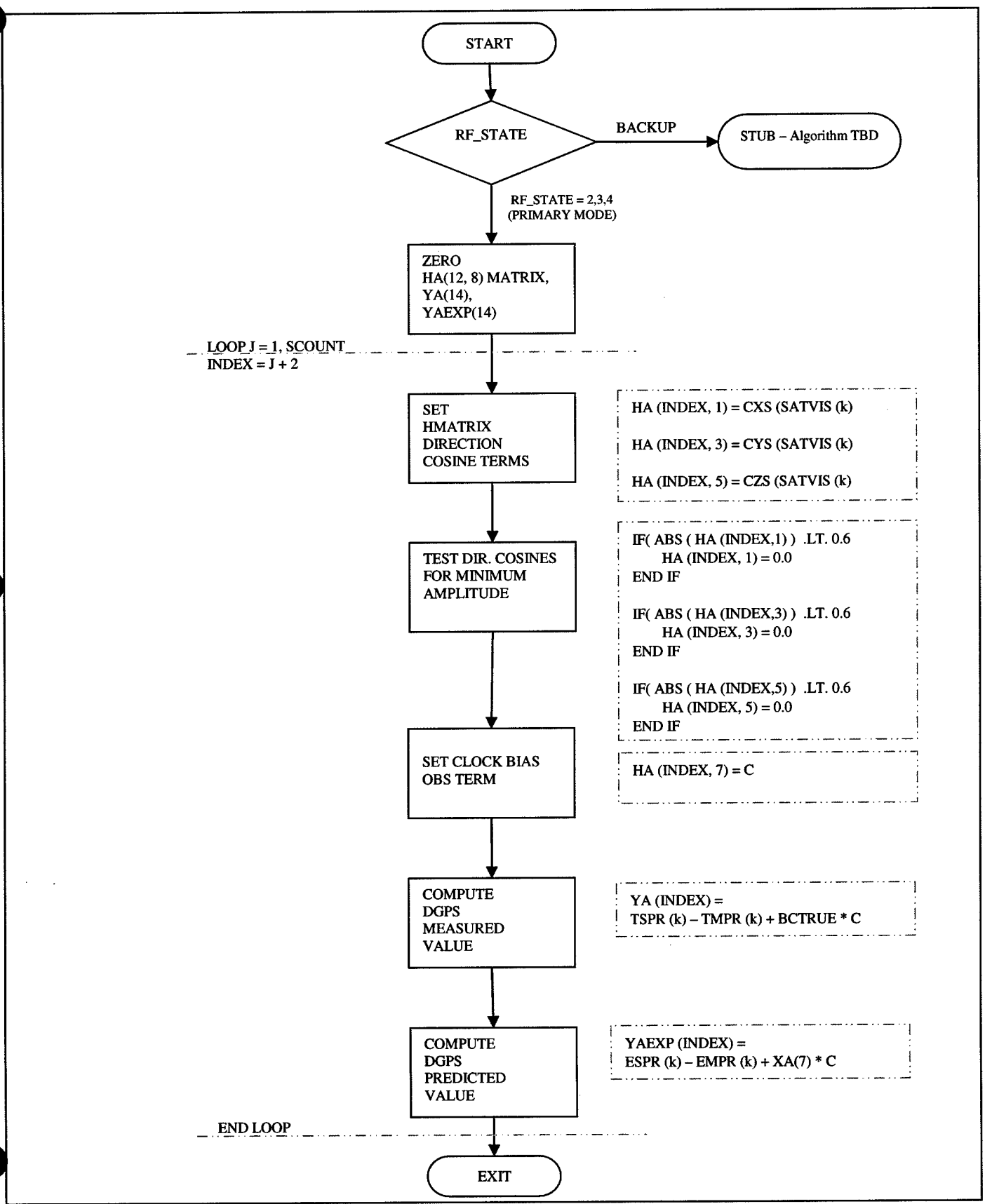


Figure 3.1-22: GENERATE_KALMAN_OBS_SET Logical data flow diagram

3.1.1.3 RF_KALMAN_PROC

This CSC performs the extended Kalman Filter (EKF) processing needed to estimate the numeric values of the eight Refinement Filter Error terms as follows:

XA(1): LL x-coordinate relative position error [m]
XA(2): LL x-coordinate relative position error rate [m/s]
XA(3): LL y-coordinate relative position error [m]
XA(4): LL y-coordinate relative position error rate [m/s]
XA(5): LL z-coordinate relative position error [m]
XA(6): LL z-coordinate relative position error rate [m/s]
XA(7): Relative clock bias [ns]
XA(8): Relative frequency drift [ns/s]

As part of the operation of the Kalman Filter, RF_KALMAN_PROC shall also perform the initialization of the EKF matrix elements.

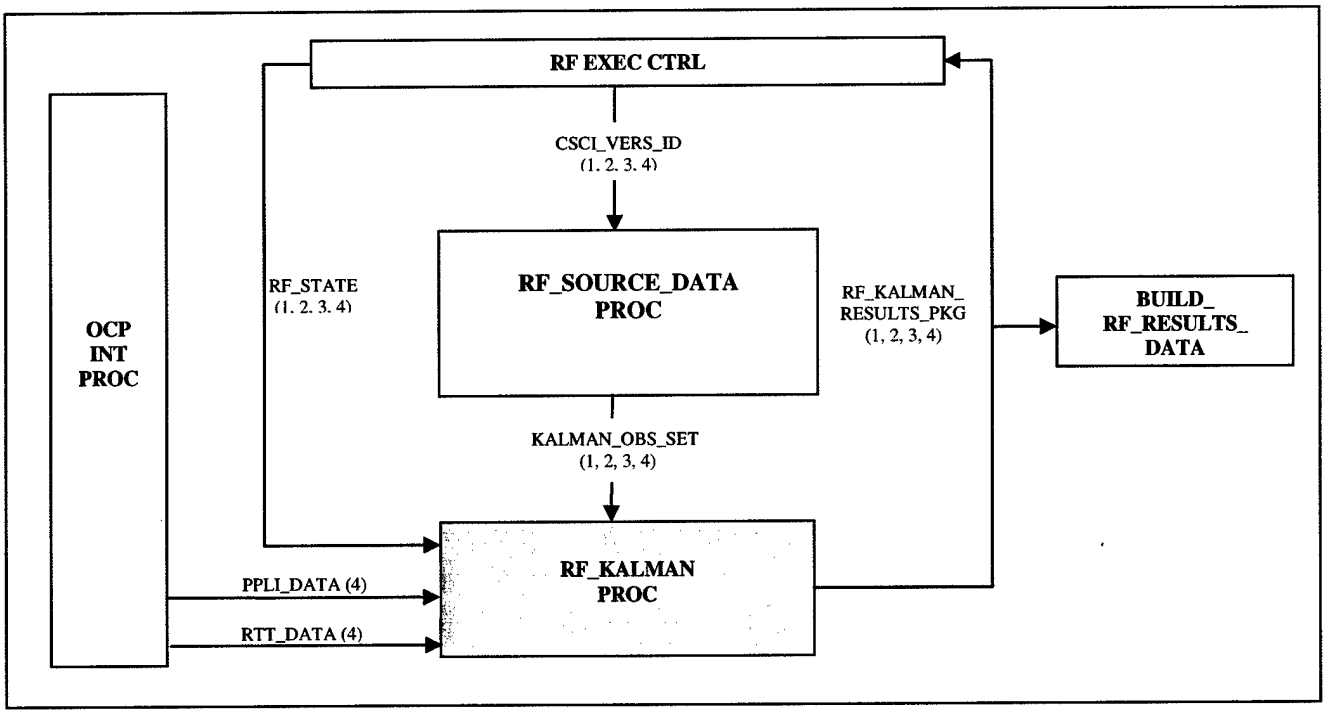


Figure 3.1-23: RF_KALMAN_PROC Level 1 Breakdown

CSC INPUTS	DESCRIPTION
KALMAN_OBS_SET	
HA(12,8)	Observation matrix
SCOUNT	Final number of paired satellites visible above ANGLIM
YA(I), I = 1, 14	Observation measurement vector
YAEXP(I), I = 1, 14	Observation prediction vector
TC, TOLD	Reference time at which measurements are valid since GPS time of week [s]
SATVIS(j), j = 1, SCOUNT	Ordered array of common visible satellites above ANGLIM
DXTRUE	True XYZ direction error composites. Only used when CSCI_VERS_ID = 2.0 (Laboratory Mode)
DYTRUE	
DZTRUE	
BCTRUE	True clock bias
FCTRUE	True frequency drift.
PPLI_DATA RTT DATA	Used only when the RF State is in Backup Mode (when GPS is either not on or valid). Contains pseudo range data
RF_STATE	Indicates the state value of the refinement filter
RSI	Refinement Status Indicator

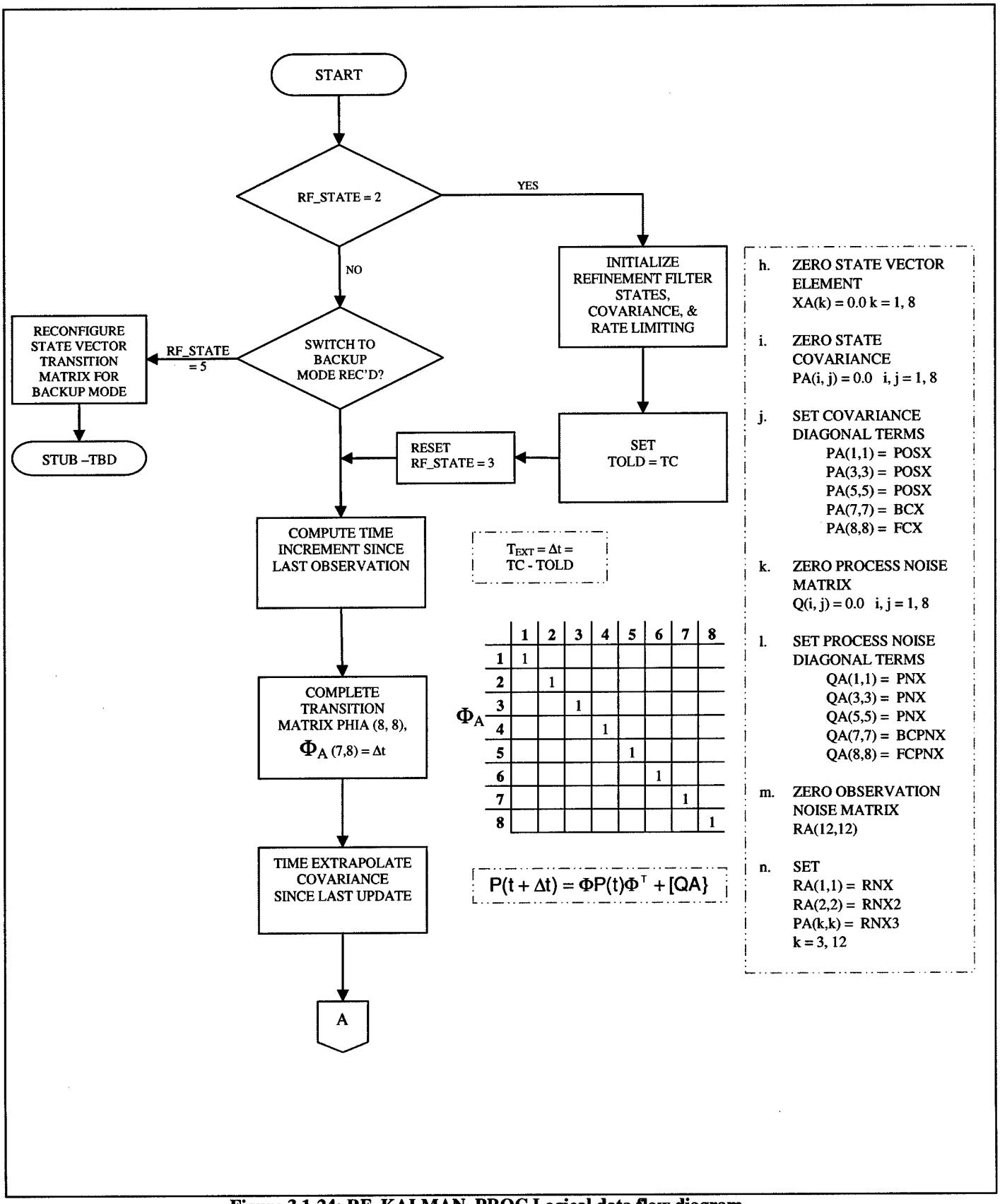
Table 3.1-7: RF_KALMAN_PROC Inputs

CSC OUTPUTS	DESCRIPTION
RF_KALMAN_RESULTS_PKG	
TC	Observation set time of validity
RF_STATE	Indicates the state value of the refinement filter
RSI	Refinement Filter Status Indicator
XA(k), k = 1, 8	Refinement Filter Solution Vector
SIGX	COVARIANCE DIAGONAL TERMS
SIGY	
SIGZ	
SIGBC	
SIGFC	
ERRX	ERROR TERMS
ERRY	
ERRZ	
ERRBC	
ERRFC	
SCOUNT	Final number of paired satellites visible above ANGLIM
SATVIS(j), j = 1, SCOUNT	Ordered array of common visible satellites above ANGLIM

Table 3.1-8: RF_KALMAN_PROC Outputs

Stored Constants:

POS = (1000.0)**2	Initial Position Covariance Constant [m2]
BCX = (100.0)**2	Initial Clock Bias Covariance Constant [ns2]
FCX = (10.0)**2	Initial Freq Drift Covariance Constant [ns2/sec2]
PNX = (1000.0)**2	Process Noise Position Constant [m2]
BCPNX = (10.0)**2	Process Noise Clock Bias Constant [ns2]
FCPNX = (1.0)**2	Process Noise Freq Drift Constant [ns2/sec2]
RNX = (21.0)**2	Range Measurement Variance Constant [m2]
RNX2 = (12.5)**2	RTT Clock Bias Measurement Variance Constant [ns2]
RNX3 = (4.0)**2	DGPS Measurement Variance Constant [m2]
SSLIM = TBD	Steady State Indication [m]



	1	2	3	4	5	6	7	8
1	1							
2		1						
3			1					
4				1				
5					1			
6						1		
7							1	
8								1

Figure 3.1-24: RF_KALMAN_PROC Logical data flow diagram

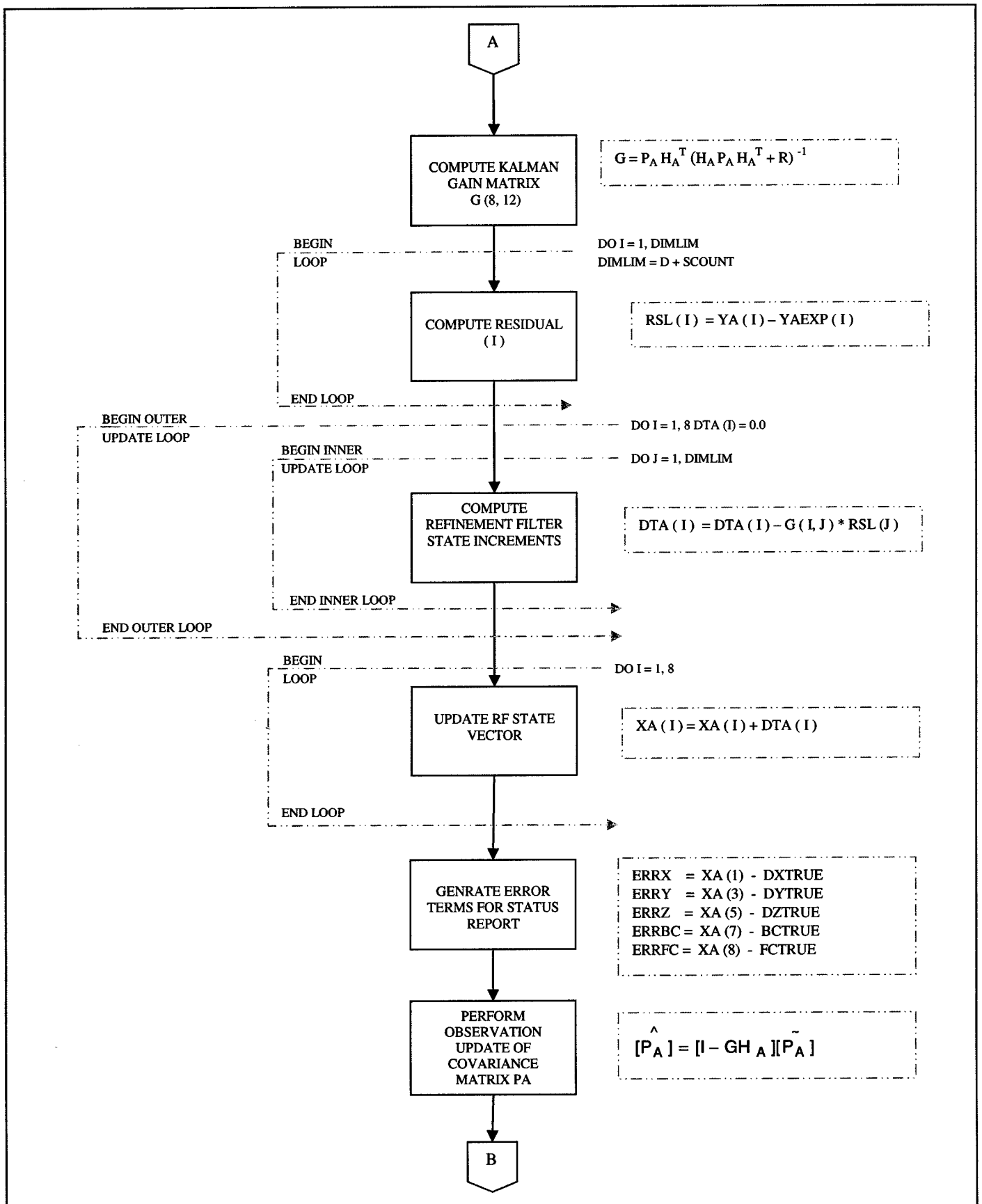


Figure 3.1-25: RF_KALMAN_PROC Logical data flow diagram (cont.)

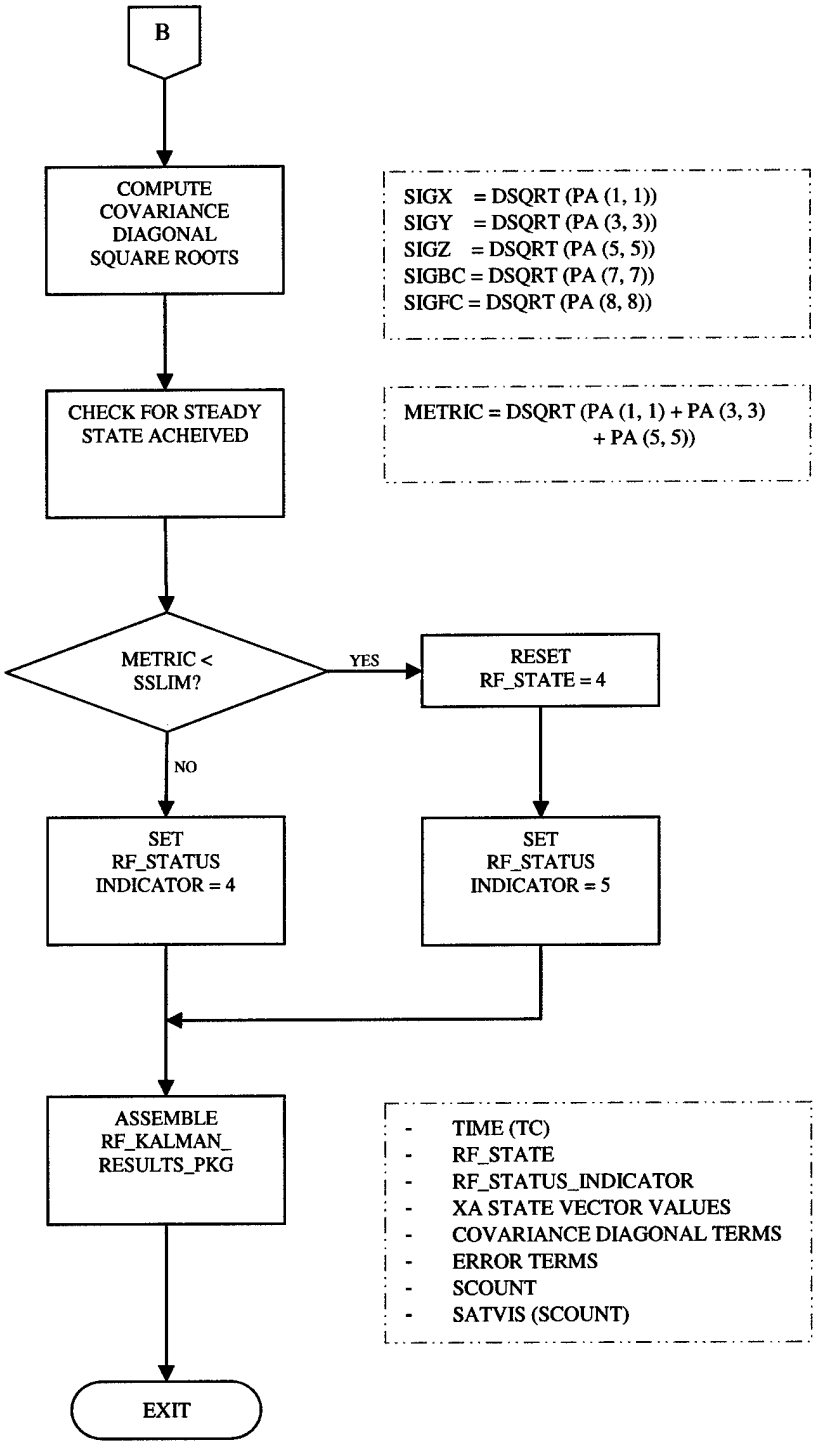


Figure 3.1-26: RF_KALMAN_PROC Logical data flow diagram (cont.)

3.1.1.4 OCP_INT_PROC

The OCP Interface Processing provides a two-way interface between the Link 16 Operational Software and Refinement Filter Processing. This SLCSC processes received Master Message Data and produces appropriate Master Data Package for further processing within the refinement filter. It also processes the self-navigation solution to produce an appropriate self-navigation data package to further processing within the Refinement Filter.

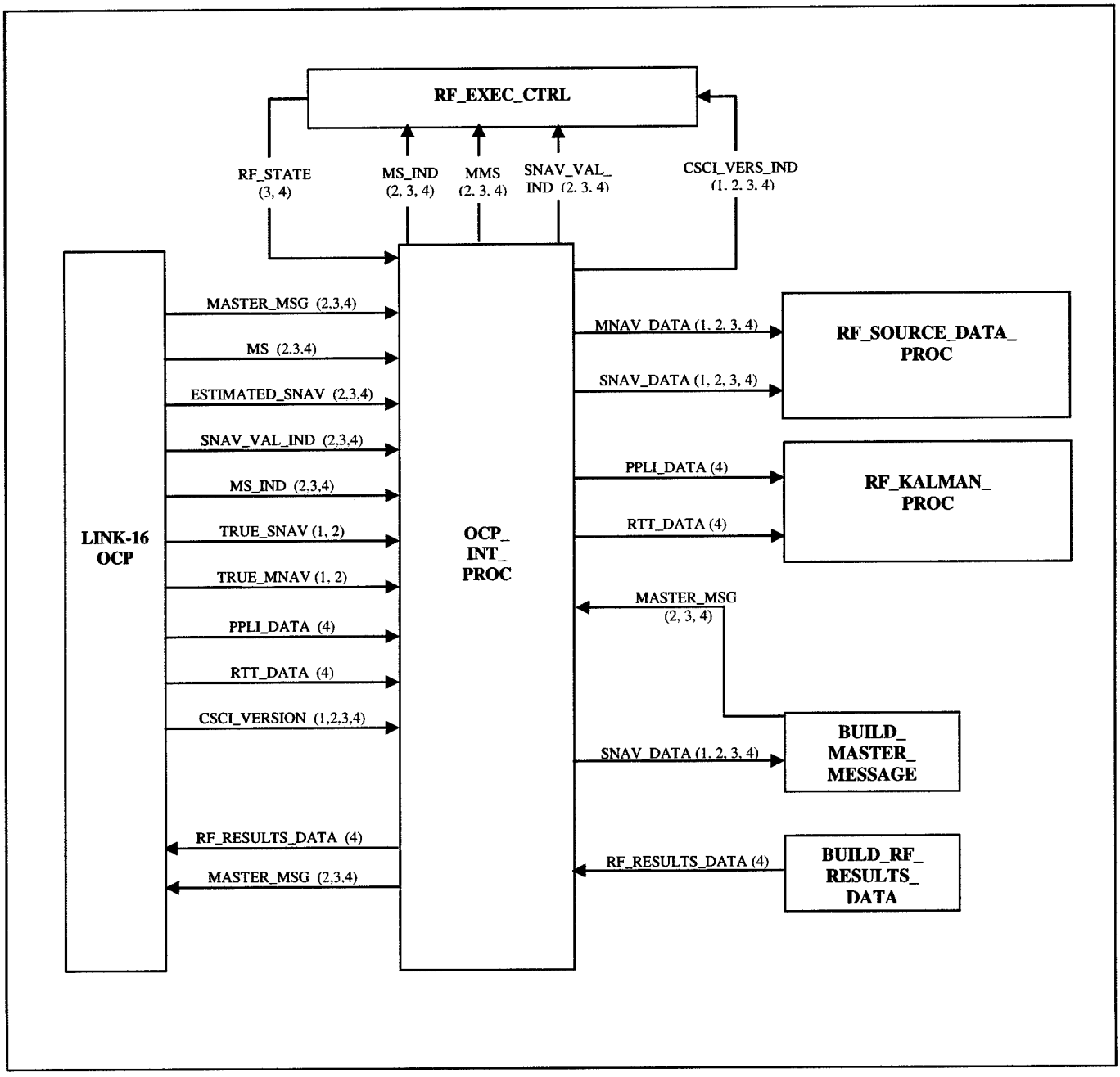


Figure 3.1-27: OCP_INT_PROC Level 1 Breakdown

CSC Inputs	Description
CSCI_VERS_IND	Denotes Operational Version
RF_STATE	Indicates the state value of the refinement filter
MASTER_MSG_	This message contains - Master's nav solution - master's GPS data
PPLI_DATA RTT_DATA	Used only when the RF State is in Backup Mode (when GPS is either not on or valid). Contains pseudo range data
EST_SNAV	ESTIMATED SELF Navigation Data
TRUE_SNAV_	TRUE SELF Navigation Data
TRUE_MNAV	TRUE MASTER Navigation Data
SNAV_VAL_IND	Self Navigation Validity Indicator
MNAV_VAL_IND	Master Navigation Validity Indicator
MS_IND	Master/Slave Indicator
MMS	Master Message Prepared Signal
RF_RESULTS_DATA	Refinement Filter corrections and covariance to send to AF (used when CSCI = 4, Backup Mode)

CSC Outputs	Description
PPLI_DATA RTT_DATA	Used only when the RF State is in Backup Mode (when GPS is either not on or valid). Contains pseudo range data
MNAV_DATA	MASTER Navigation Solution
SNAV_DATA	SELF Navigation Solution
RF_RESULTS_MSG	Refinement Filter corrections and covariance to send to AF
MASTER_MSG	This message contains - Master's nav solution - Master's GPS data
MS_IND	Master/Slave Indicator
MMS	Master Message Prepared Signal
SNAV_VAL_IND	Self Navigation Validity Indicator
MNAV_VAL_IND	Master Navigation Validity Indicator
CSCI_VERS_IND	Denotes Operational Version

INPUTS	DESCRIPTION
MS_IND	MASTER/SLAVE Indicator
LF_IND	LABORATORY/FIELD Indicator
MASTER_MSG	Received MASTER Message
MMS	MASTER Message Signal
TRUE MASTER NAV DATA	
TMNAVTOV	TRUE Time of Validity of MASTER Navigation solution
TMLAT	TRUE MASTER Latitude position [rad]
TMNVEL	TRUE MASTER North Velocity [ft/s]
TMLON	TRUE MASTER Longitude position [rad]
TMEVEL	TRUE MASTER East Velocity [ft/s]
TMALT	TRUE MASTER Altitude [ft]
TMALTR	TRUE MASTER Altitude rate [ft/s]
ESTIMATED SELF NAV DATA	
SNAVTOV	Time of Validity of SELF Navigation solution
SLAT	ESTIMATED SELF Latitude position [rad]
SNVEL	ESTIMATED SELF North Velocity [ft/s]
SLON	ESTIMATED SELF Longitude position [rad]
SEVEL	ESTIMATED SELF East Velocity [ft/s]
SALT	ESTIMATED SELF Altitude [ft]
SALTR	ESTIMATED SELF Altitude rate [ft/s]
TRUE SELF NAV DATA	
TSNAVTOV	TRUE Time of Validity of SELF Navigation Solution
TSLAT	TRUE SELF Latitude position [rad]
TSNVEL	TRUE SELF North Velocity [ft/s]
TSLON	TRUE SELF Longitude position [rad]
TSEVEL	TRUE SELF East Velocity [ft/s]
TSALT	TRUE SELF Altitude [ft]
TSALTR	TRUE SELF Altitude rate [ft/s]
BCTRUE	TRUE Clock Bias
FCTRUE	TRUE Frequency Drift

OUTPUTS	DESCRIPTION
MNAV DATA PACKAGE	
MNAVTOV	Time of Validity of MASTER Navigation solution
MLAT	ESTIMATED MASTER Latitude position [rad]
MLATR	ESTIMATED MASTER Latitude rate [rad/s]
MLON	ESTIMATED MASTER Longitude position [rad]
MLONR	ESTIMATED MASTER Longitude rate [rad/s]
MALT	ESTIMATED MASTER Altitude [m]
MALTR	ESTIMATED MASTER Altitude rate [m/s]
TMLAT	TRUE MASTER Latitude position [rad]
TMLATR	TRUE MASTER Longitude rate [m/s]
TMLON	TRUE MASTER Longitude position [rad]
TMLONR	TRUE MASTER Longitude rate [m/s]
TMALT	TRUE MASTER Altitude [m]
TMALTR	TRUE MASTER Altitude rate [m/s]
MSAT(12)	Master Ordered Satellite List
MPRR(12)	Master Ordered Measured Code Phase Array
MGPSTOV	MASTER predicted GPS time of validity
MVIS	Number of satellites visible to MASTER
INVERS(30)	Satellite sorting index
SNAV DATA PACKAGE	
SNAVTOV	Time of Validity of SELF Navigation solution
SLAT	ESTIMATED SELF Latitude position [rad]
SLATR	ESTIMATED SELF Latitude rate [rad/s]
SLON	ESTIMATED SELF Longitude position [rad]
SLONR	ESTIMATED SELF Longitude rate [rad/s]
SALT	ESTIMATED SELF Altitude [m]
SALTR	ESTIMATED SELF Altitude rate [m/s]
TSLAT	TRUE SELF Latitude position [rad]
TSLATR	TRUE SELF Latitude rate [rad/s]
TSLON	TRUE SELF Longitude position [rad]
TSLONR	TRUE SELF Longitude rate [rad/s]
TSALT	TRUE SELF Altitude [m]
TSALTR	TRUE SELF Altitude rate [m/s]
SNAV_VAL_IND	Self Navigation Validity Indicator
MNAV_VAL_IND	Master Navigation Validity Indicator

Stored Constants:

ar = 2.09257382 E7	Earth semi-minor radius [ft]
rad = 57.295779513	radians [rad]

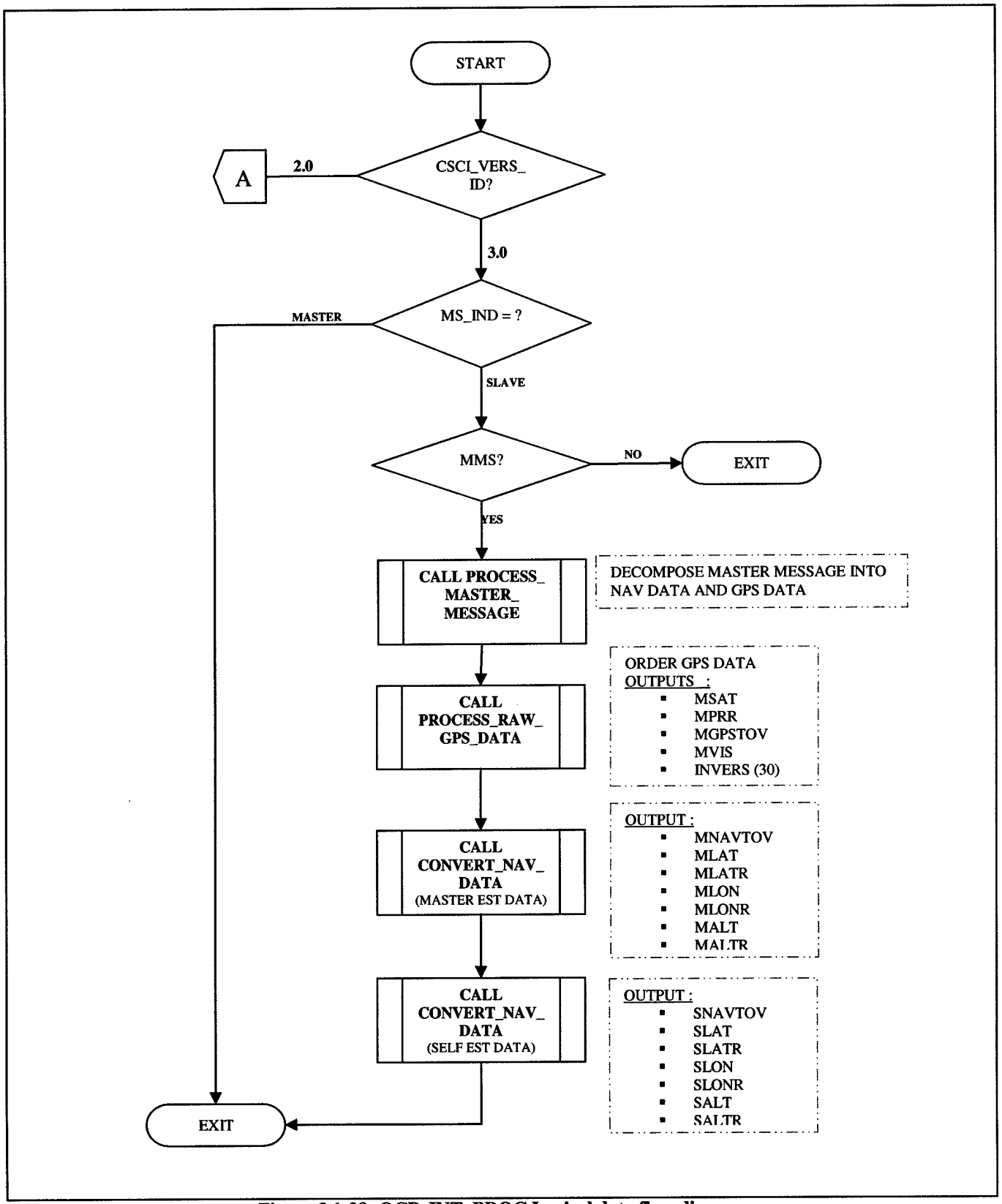


Figure 3.1-28: OCP_INT_PROC Logical data flow diagram

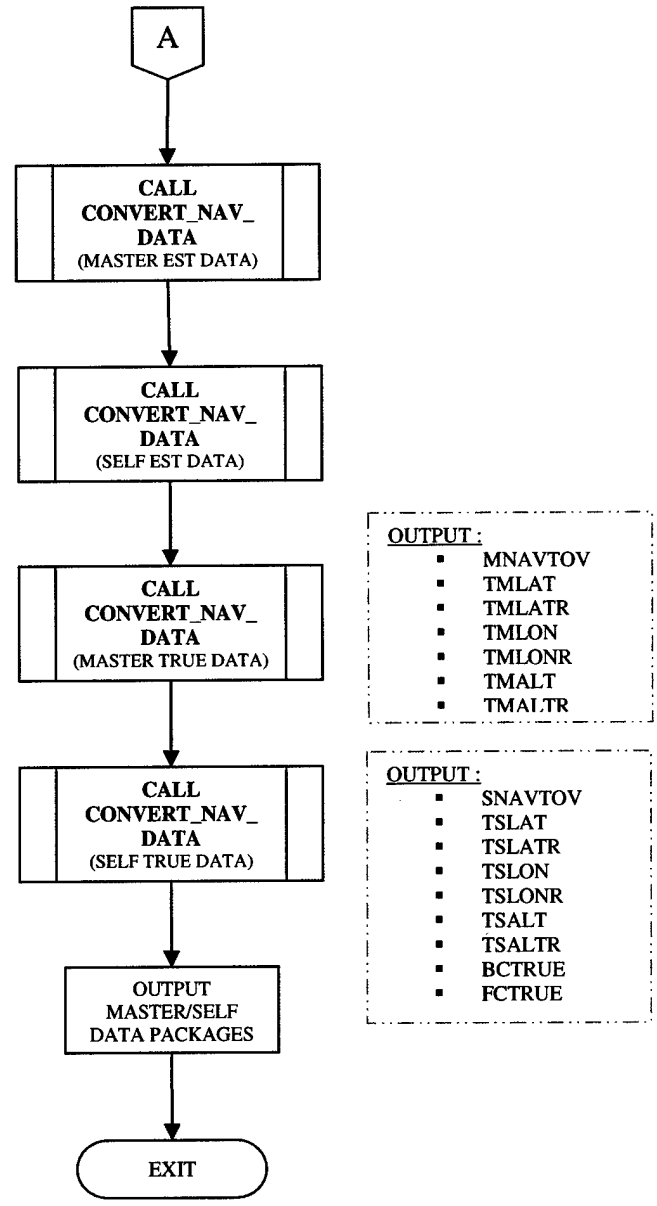


Figure 3.1-29: OCP_INT_PROC Logical data flow diagram (cont.)

3.1.1.4.1 PROCESS_MASTER_ MSG

PROCESS_MASTER_MSG decomposes the received Master Message Data into Master NAV data and GPS data

INPUTS	**DESCRIPTION**
RECEIVED MASTER MESSAGE	This message contains - Master's NAV solution - Master's GPS Data
OUPUTS	**DESCRIPTION**
MASTER NAV SOLUTION	
MNAVTOV	Time of Validity of MASTER Navigation solution
MLAT	ESTIMATED MASTER Latitude position [rad]
MNVEL	ESTIMATED MASTER North Velocity [ft/s]
MLON	ESTIMATED MASTER Longitude position [rad]
MEVEL	ESTIMATED MASTER East Velocity [ft/s]
MALT	ESTIMATED MASTER Altitude [ft]
MALTR	ESTIMATED MASTER Altitude rate [ft/s]
MASTER GPS DATA	
GPS_HEALTH_IND	Indicates the health status of GPS signal(Binary)
MGPSTOV	Time of Validity of MASTER GPS Solution
UMSAT(12)	Unordered array of MASTER visible satellites
UMPRR(12)	Unordered array of MASTER Code Phase

3.1.1.4.2 PROCESS_RAW_GPS_DATA

This SLCSC processes the GPS data decomposed from the MASTER_MSG. This function produces ordered arrays of satellite ID's and code phase data. The INVERS array is also generated for SOURCE_DATA_PROC.

INPUTS	DESCRIPTION
USSAT(12)	Unordered array of MASTER visible satellites
UMPRR(12)	Unordered array of MASTER code phase
OUTPUTS	**DESCRIPTION**
MSAT(12)	Ordered array of MASTER visible satellites
MPRR(12)	Ordered array of MASTER code phase
INVERS(30)	Satellite sorting index
MVIS	Number of Satellites visible to the MASTER

Pseudocode:

INVERS array generation

```
DO X j = 1, 12
     ID = MSAT[j]
     INVERS[ID] = j

X CONTINUE
```

Insertion sort algorithm

```
DO Y j = 1, SAT.length                  Loop through target array & put each
                                        item in the proper slot, shifting other
                                        items out of the way as you go.

     ID = SAT[j];
     i = j - 1;
     WHILE (i >= 0 && SAT[i] > ID)
        SAT[i+1] = SAT[i];
        i = i-1;

SAT[i+1] = ID;                          SAT[i] is the item that should precede
                                        value.  Put value after it.

Y CONTINUE
```

This process is repeated for each of the SELF and MASTER satellite ID and code phase data.

3.1.1.4.3 CONVERT_NAV_DATA

This SLCSC converts L16 Nav data into metric units. CONVERT_NAV_DATA also generates Latitude and Longitude Rate terms.

INPUTS	DESCRIPTION
NVEL	North Velocity [ft/sec]
EVEL	East Velocity Rate [ft/sec]
ALT	Altitude [ft]
ALTR	Altitude rate [ft/s]
OUPUTS	**DESCRIPTION**
LATR	Latitude rate [rad/s]
LONR	Longitude rate [rad/s]
ALT	Altitude [m]
ALTR	Altitude rate [m/s]

Stored Constants:

ar = 2.09257382 E7	Earth semi-minor radius [ft]
rad = 57.295779513	radians [rad]
FT_M = 3.28083991666 [ft/m]	Feet → meters conversion

The following equations is used to convert the required Navigation data

$$\frac{NVEL\ [ft/s]}{ar\,[ft]} \times rad = LATR\ [rad/s]$$

$$\frac{EVEL[ft/s]}{ar[ft] \times \cos(LAT)} \times rad = LONR[rad/s]$$

$$\frac{ALT\ [ft]}{FT\_M\ [ft/m]} = ALTR\ [m]$$

$$\frac{ALTR\ [ft/s]}{FT\_M\ [ft/m]} = ALTR\ [m/s]$$

3.1.1.5 HOST_INT_PROC

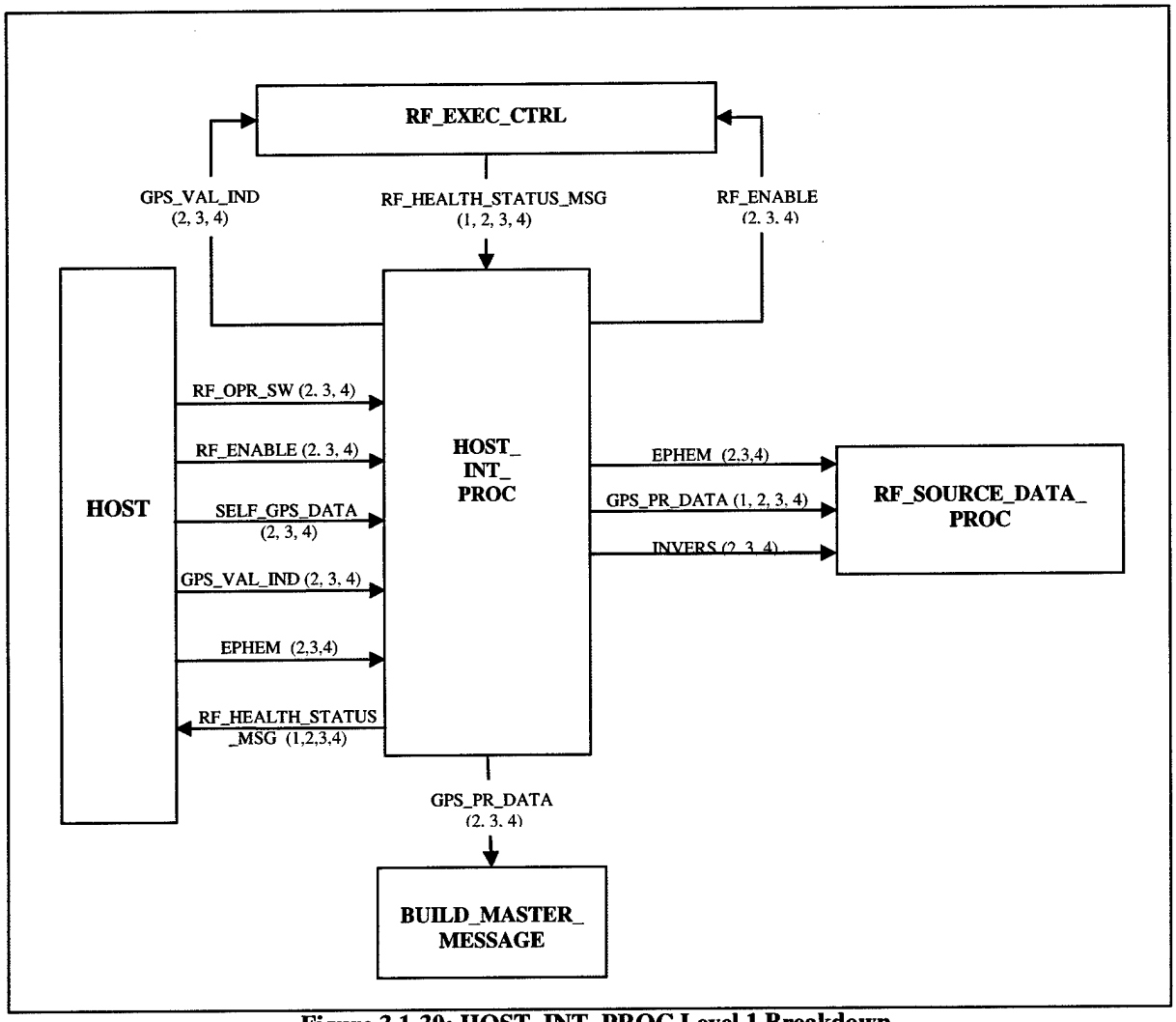


Figure 3.1-30: HOST_INT_PROC Level 1 Breakdown

The host Interface Processing function serves as the primary two way data path between the DGPS RF algorithm and the host processor. With respect to input from the host, it performs the following functions:

(a) Interprets and reformats DGPS RF control inputs provided by the host, namely:
- RF_ENABLE (Operate/Non-operate switch)
- RF_OPR_SW (Operator command algorithm reset indicator)

(b) Interprets and reformats raw input provided by the GPS receiver, including GPS validity indicator, Self-GPS pseudorange measurements, and ephemeris data.

With respect to host output, HOST_INT_PROC is responsible for providing an RF Health and Status Message, which includes the full filter state estimates and co variances plus RF State and Status Indicators.

CSC inputs and outputs are summarized, respectively, in Tables 4.4.5-1 and 4.4.5-2. Two SLCSCs are utilized within HOST_INT_PROC. Their functions are described in Table 4.4.5-3.

CSC Inputs	Version	Description
RF_OPR_SW	1,2,3,4	Operator Reset Command
RF_ENABLE	2,3,4	RF Enable Command
SELF_GPS_DATA_	2,3,4	Pseudorange and satellite ID message data
GPS_VAL_IND	2,3,4	GPS Validity Indicator
EPHEM_RAW	2,3,4	Raw Ephemeris Data message
RF_HEALTH_STATUS_MSG	2,3,4	Summary of GPS results and status

Table 3.1-9: HOST_INT_PROC Input Variables

CSC Outputs	Version	Description
RF_HEALTH_STATUS_MSG	2,3,4	Summary of GPS results and status

Table 3.1-10: HOST_INT_PROC Output Variables

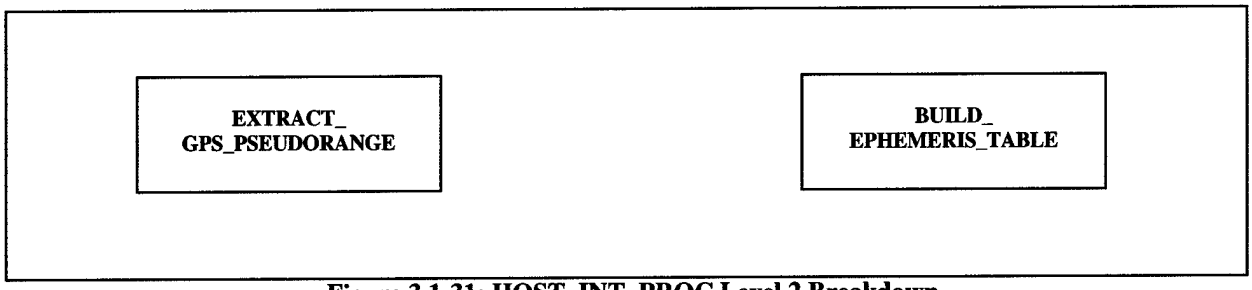


Figure 3.1-31: HOST_INT_PROC Level 2 Breakdown

SLCSC	Description
EXTRACT_GPS_PSEUDORANGE	Decomposes GPS receiver data to provide an ordered array of satellite ID's and code phase data.
BUILD_EPHEMERIS_TABLE	Reformats GPS receiver ephemeris message to provide standard ephemeris array.

Table 3.1-11: HOST_INT_PROC SLCSC Descriptions

3.1.1.6 BUILD_MASTER_MSG

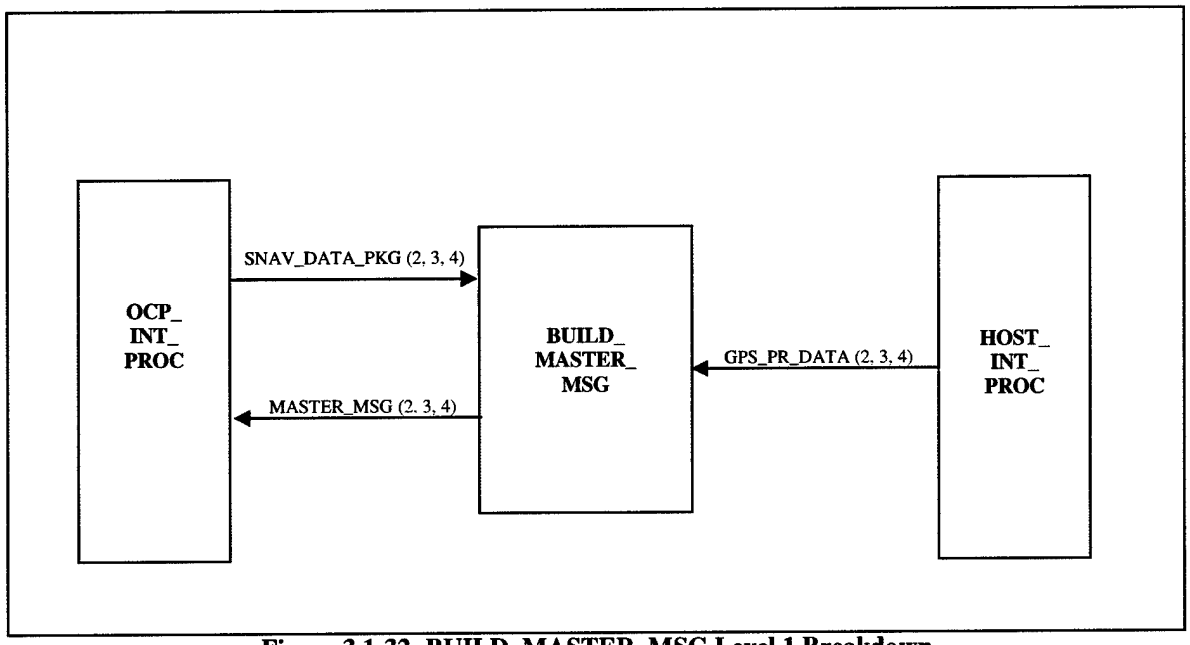


Figure 3.1-32: BUILD_MASTER_MSG Level 1 Breakdown

The purpose of this function is to construct outgoing Master Messages containing self navigation and measured pseudorange data collected at own location. This function shall be invoked only when own platform has been designated as Master.

The contents and format of the J28.7 Master Message are presented in paragraph 4.5.4.

3.1.1.7 BUILD_RF_RESULTS_DATA

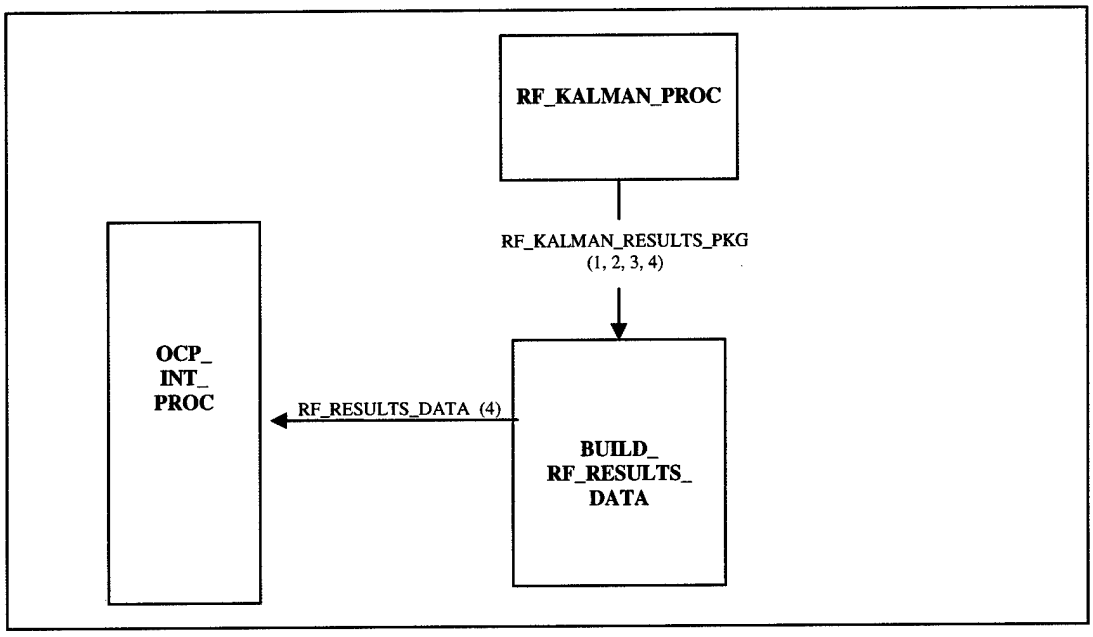


Figure 3.1-33: BUILD_RF_RESULTS_DATA Level 1 Breakdown

This function is intended as a place holder for a potential enhanced Version 4.0 of the DGPS RF algorithm. This version of the software would combine the functionality of the DGPS RF with that of an Atmospheric Calibration Filter (AF), now being developed under separate contract. The basic concept of Version 4.0 is to improve basic Link-16 navigation performance via advanced atmospheric calibration of TOA range measurements. The Atmospheric Filter utilizes the position estimates of the transmitter and receiver of a PPLI message as basic input. The DGPS RF has demonstrated the ability to reduce errors in position estimates to extremely small levels. These precise DGPS RF position estimates can generate a corrective term that will reduce the range estimate error of the AF, if applied to the existing Link-16 navigation solution.

The purpose of Build_RF_Results_Data is to package the error estimates of the DGPS RF Kalman Filter for transfer to the Link-16 OCP. Note that this function exists only for the Enhanced Version 4.0 of the DGPS algorithm.

4. GLOSSARY

VARIABLE	DESCRIPTION
ALT	Altitude
ALTR	Altitude rate
ANGLIM	5 degree limit of visibility [rad]
ar = 2.09257382 E7	Earth semi-minor radius [ft]
BBC2M	Backup measured clock bias to master
BCTRUE	True clock bias
BLCK	Number of PR reports (?)
BPR2M	Backup measured range to master
C = 0.983569 [ft/ns]	Speed of light in vacuum 2.99792498 E8 [m/s]
CDPHASE	GPS Code phase value
CR	Code phase conversion factor
CXS(SCOUNT)	Direction cosine x-component
CYS(SCOUNT)	Direction cosine y-component
CZS(SCOUNT)	Direction cosine z-component
DXTRUE, DYTRUE, DZTRUE	True error composites.
e22 = 6.74499984 E-3	Earth eccentricity squared
ECEFXS(K)	ECEF x-coordinate for satellite K
ECEFYS(K)	ECEF y-coordinate for satellite K
ECEFZS(K)	ECEF z-coordinate for satellite K
EMPR(SCOUNT)	PREDICTED Pseudorange Vector to satellites from MASTER
EPHEM(30,23)	Satellite Ephemeris Data
ESPR(SCOUNT)	PREDICTED Pseudorange Vector to satellites from SELF
ETA0(I), K = 1, 30	Lon coordinates of satellites
EVEL	East Velocity Rate [ft/sec]
FCTRUE	TRUE Frequency Drift
GPS_HEALTH_IND	Indicates the health status of GPS signal. (Binary)
HA(12,8)	Observation matrix for Kalman Filter
INVERS(30)	Satellite sorting index
IOTA = 55.0/2*pi	Satellite Inclination [rad]
ISVN0	Satellite vehicle number
LAT	Latitude position [rad]
LATR	Latitude rate [rad/s]
LLPX	Local level x-coordinate value
LLPY	Local level y-coordinate value
LLPZ	Local level z-coordinate value
LON	Longitude position [rad]
LONR	Longitude rate [rad/s]
MALT	ESTIMATED MASTER Altitude
MALTR	ESTIMATED MASTER Altitude rate
MEVEL	ESTIMATED MASTER East Velocity [ft/s]
MGPSTOV	MASTER predicted GPS time of validity

MGPSTOV	MASTER predicted GPS time of validity
MLAT	ESTIMATED MASTER Latitude position [rad]
MLATR	ESTIMATED MASTER Latitude rate [rad/s]
MLLPX	ESTIMATED MASTER local level x-coordinate value
MLLPY	ESTIMATED MASTER local level y-coordinate value
MLLPZ	ESTIMATED MASTER local level z-coordinate value
MLON	ESTIMATED MASTER Longitude position [rad]
MLONR	ESTIMATED MASTER Longitude rate [rad/s]
MNAVTOV	Time of Validity of MASTER Navigation solution
MNVEL	ESTIMATED MASTER North Velocity [ft/s]
MPGPSTM(NVS)	MASTER sorted predicted GPS time vector
MPRR(12)	MASTER Ordered measured code phase array
MS_IND	MASTER/SLAVE indicator
MSAT(12)	MASTER Ordered array of visible satellites
MVIS	Number of satellites visible to MASTER
NAVTOV	Time of Validity of MASTER Navigation solution
NVEL	North Velocity [ft/sec]
NVS	Number visible satellites - common to MASTER and SELF
OMEGA0(K), K = 1, 30	Lat coordinates of satellites
OMEGAS = 1.45858522 E-4	Satellite orbital rate [rad/s]
OMEGE = 7.292115467 E-5	Earth's rotational rate [rad/s]
PGPSTM	Predicted GPS time
PRR1(NVS)	MASTER array of code phase for common satellites
PRR2(NVS)	SELF array of code phase for common satellites
RF_ENABLE	ON/OFF switch
RF_OPR_SW	Reset/Initiate signal from Host pilot
RF_STATE	State of the refinement filter
RF_STATUS_IND	Refinement filter status indicator
RK = 87051108	Radius of satellite orbit [m]
SALT	ESTIMATED SELF Altitude
SALTR	ESTIMATED SELF Altitude rate
SAT(NVS)	Ordered array of common visible satellites
SATVIS(SCOUNT)	Ordered array of common visible satellites above ANGLIM
SCOUNT	Final number of paired satellites visible above ANGLIM
SELF_GPS_DATA	Received GPS data from HOST
SEVEL	ESTIMATED SELF East Velocity [ft/s]
SGPSTOV	SELF Predicted GPS time of validity
SLAT	ESTIMATED SELF Latitude position [rad]
SLATR	ESTIMATED SELF Latitude rate [rad/s]
SLLPX	ESTIMATED SLAVE local level x-coordinate value
SLLPY	ESTIMATED SLAVE local level y-coordinate value
SLLPZ	ESTIMATED SLAVE local level z-coordinate value
SLON	ESTIMATED SELF Longitude position [rad]
SLONR	ESTIMATED SELF Longitude rate [rad/s]
SNAVTOV	Time of Validity of SELF Navigation solution
SNVEL	ESTIMATED SELF North Velocity [ft/s]
SPGPSTM(NVS)	SELF sorted predicted GPS time vector

SPRR(12)	SELF Ordered measured code phase array
SSAT(12)	SELF Ordered array of visible satellites
SVIS	Number of satellites visible to SELF
TC	Observation set time of validity (GPS system time [s])
TLE()	Local Level transform matrix
TMALT	TRUE MASTER Altitude
TMALTR	TRUE MASTER Altitude rate
TMEVEL	TRUE MASTER East Velocity [ft/s]
TMLAT	TRUE MASTER Latitude position [rad]
TMLATR	TRUE MASTER Latitude rate [rad/s]
TMLLPX	TRUE MASTER local level x-coordinate value
TMLLPY	TRUE MASTER local level y-coordinate value
TMLLPZ	TRUE MASTER local level z-coordinate value
TMLON	TRUE MASTER Longitude position [rad]
TMLONR	TRUE MASTER Longitude rate [rad/s]
TMNVEL	TRUE MASTER North Velocity [ft/s]
TMPR(SCOUNT)	TRUE Pseudorange Vector to satellites from MASTER
TSALT	TRUE SELF Altitude
TSALTR	TRUE SELF Altitude rate
TSEVEL	TRUE SELF East Velocity [ft/s]
TSLAT	TRUE SELF Latitude position [rad]
TSLATR	TRUE SELF Latitude rate [rad/s]
TSLLPX	TRUE SLAVE local level x-coordinate value
TSLLPY	TRUE SLAVE local level y-coordinate value
TSLLPZ	TRUE SLAVE local level z-coordinate value
TSLON	TRUE SELF Longitude position [rad]
TSLONR	TRUE SELF Longitude rate [rad/s]
TSNVEL	TRUE SELF North Velocity [ft/s]
TSPR(SCOUNT)	TRUE Pseudorange Vector to satellites from SELF
UMPRR(12)	Unordered array of MASTER code phase
UMSAT(12)	Unordered array of SELF visible satellites
USPRR(12)	Unordered array of SELF code phase
USSAT(12)	Unordered array of MASTER visible satellites
XA(k)	Refinement Filter Solution Vector, k = 1, 8
YA(14)	Inputs to the Kalman Filter
YAEXP(14)	Expected Kalman solutions

Appendix B:
Simulated Code Listing

```
C-----------------------------------------------------------------------
      SUBROUTINE RF_EXEC_CTRL (RF_OPR_SW, MS_IND, GPS_HEALTH_IND,
     C    RF_OUTPUT_DATA, LF_IND, SNAV_VAL_IND, MNAV_VAL_IND, DATA_STATUS_ MSG,
     C    CLK_BIAS_FREQ_DRIFT, RTT_REQUEST_IND, RF_STATE, RF_HEALTH_STATUS_MSG)
     C
     C
     C    VERSION NO. 1.0  DATE: 03 APRIL 2003
     C
     C    PURPOSE:  TO SERVE AS AN EXECUTIVE CONTROL OF THE REFINEMENT PROCESS.  THIS
     C           CSC IS THE ONLY INTERFACE TO THE HOST_INTERFACE_PROC AND OCP_INT_PROC
     C
     C
     C
     C  INPUTS:
     C
     C  CRF_OPR_SW          - ON/OFF switch for system initialization/reset
     C  MS_IND           - Master/Slave Indicator
     C  GPS_HEALTH_IND      - Indicates the health status of GPS signal
     C  RF_OUTPUT_DATA      - Refinement Filter results and covariance, XA and COV
     C  LF_IND           - Laboratory/Field Indicator
     C  SNAV_VAL_IND     - Self Navigation Validity Indicator
     C  MNAV_VAL_IND     - Master Navigation Validity Indicator
     C  DATA_STATUS_MSG     - Indicates status of data alignment for processing.
     C
     C
     C  OUTPUTS:
     C  CLK_BIAS_FREQ_DRIFT - Send clock bias and frequency drift data if requested
     C  RTT_REQUEST_IND     - Send request for RTT and PPLI when system resides in
     C                  backup mode.
     C  RF_STATE         - Indicates the state value of the refinement filter
     C  RF_HEALTH_STATUS_MSG   - Determines whether the filter should be reset,
     C                  switch to backup mode, or system normal.
     C
     C
     C
     C  VERSION 1
     C  -------------
     C  INPUTS: TC, T, TLAT, TLON, TALT, ELAT, ELON, EALT, TLATR,
     C        TLONR, TALTR, ELATR, ELONR, EALTR, CBIAS, FDRIFT
     C
     C
     C  LOCAL VARIABLES:
     C  RSI              - RF Status Indicator
     C
     C
     C
     C  GLOBAL VARIABLES:
     C
     C
     C
     C  CALLED BY:
     C     RNS
     C
     C
     C
     C  ROUTINES CALLED:
     C     HOST_INT_PROC
     C     OCP_INT_PROC
     C     BUILD_MASTER_MSG
     C     RF_SOURCE_DATA_PROC
     C     RF_KALMAN_PROC
     C
     C
     C  MODICATION HISTORY:
```

```
C      V1.0   ORIGINAL VERSION DATE       RESPONSIBLE
C
C             FOR VERSION 1.0      4/03/03       JYH
C
C
C
C
C-----------------------------------------------------------------------

C   SUBROUTINE RF_EXEC_CTRL (RF_OPR_SW, MS_IND, GPS_HEALTH_IND,
C   1   RF_OUTPUT_DATA, LF_IND, SNAV_VAL_IND, MNAV_VAL_IND, DATA_STATUS_ MSG,
C   2   CLK_BIAS_FREQ_DRIFT, RTT_REQUEST_IND, RF_STATE, RF_HEALTH_STATUS_MSG)E

    SUBROUTINE RF_EXEC_CTRL(TC, TLAT, TLON, TALT, ELAT, ELON, EALT, TLATR, TLONR,
    1   TALTR, ELATR, ELONR, EALTR, CBIAS, FDRIFT)


C*********************************************************************
C
C       DECLARATION OF VARIABLES
C
C*********************************************************************

    IMPLICIT NONE
    !Inputs
    INTEGER*4 RSI, RF_STATE, CSCI_VERS_IND
    INTEGER*4 RF_OPR_SW, SNAV_VAL_IND, MNAV_VAL_IND, GPS_STRENGTH, SS

    INTEGER*4 SATVIS(12)

    INTEGER*4 MS_IND, GPS_HEALTH_IND, RTT_REQUEST_IND, MVIS, SVIS, SCOUNT

    REAL*8    TC, TOLD, TLAT(10), TLON(10), TALT(10), ELAT(10), ELON(10), EALT(10)
    REAL*8    TLATR(10), TLONR(10), TALTR(10), ELATR(10), ELONR(10), EALTR(10),CBIAS(10), FDRIFT(10)


C*********************************************************************
C
C        SOURCE DATA VARIABLES
C
C*********************************************************************

    REAL*8    MSAT(12), SSAT(12), MPRR(12), SPRR(12), SAT(24), PRR1(24), PRR2(24)
    REAL*8    EPHEM(23,30), INVERS(30), ECEFXS(30), ECEFYS(30), ECEFZS(30)
    REAL*8    OMEGA0(24), ETA0(24)
    REAL*8    TMGPSTOV, TMNAVTOV, TMLAT, TMLATR, TMLON, TMLONR, TMALT, TMALTR, TMLLPX,TMLLPY, TMLLPZ
    REAL*8    TSGPSTOV, TSNAVTOV, TSLAT, TSLATR, TSLON, TSLONR, TSALT, TSALTR, TSLLPX,TSLLPY, TSLLPZ
    REAL*8    MGPSTOV, MNAVTOV, MLAT, MLATR, MLON, MLONR, MALT, MALTR, MLLPX, MLLPY, MLLPZ
    REAL*8    SGPSTOV, SNAVTOV, SLAT, SLATR, SLON, SLONR, SALT, SALTR, SLLPX, SLLPY, SLLPZ

    REAL*8    MPGPSTM(24), SPGPSTM(24)
    REAL*8    TLE(3,3), BCTRUE, FCTRUE
    REAL*8    ECEFS(3,1), LLSAT(3,1), LLSATX(24), LLSATY(24), LLSATZ(24)
    REAL*8    SDIFF(3), MDIFF(3), TSDIFF(3), TMDIFF(3), VIS(24)
    REAL*8    EMPR(12), ESPR(12), TMPR(12), TSPR(12),CXS(24), CYS(24), CZS(24)

    REAL*8    BPR2M, BBC2M, DXTRUE, DYTRUE, DZTRUE
    REAL*8    MNVEL, MEVEL, TMNVEL, TMEVEL, SNVEL, SEVEL, TSNVEL, TSEVEL

C*********************************************************************
C
C        KALMAN FILTER VARIABLES

C*********************************************************************
    REAL*8 PHIA(8,8), YA(12),YAEXP(12),XA(8),HA(12,8)
    REAL*8 PA(8,8),RA(12,12),QA(8,8)
```

```
      INTEGER*4   POSX, BCX, FCX, PNX, BCPNX, FCPNX, RNX, RNX2, RNX3, NVS, I, J
      REAL*8    IOTA, OMEGE, OMEGS, RAD, ANGLIM


      DATA POSX/1000000.0/
      DATA BCX/10000.0/
      DATA FCX/100.0/

      DATA PNX/1000000.0/
      DATA BCPNX/100.0/
      DATA FCPNX/1.0/

      DATA RNX/441.0/
      DATA RNX2/156.25/
      DATA RNX3/16.0/

C     DATA SSLIM/TBD/


      DATA IOTA/55.0/
      DATA OMEGE,OMEGS/7.292115467E-5,1.45858522E-4/
      DATA RAD/57.2957795131/
      DATA ANGLIM/5.0/


      DATA CSCI_VERS_IND/1/
      DATA RF_OPR_SW/1/
      DATA MS_IND/2/
      DATA GPS_HEALTH_IND/1/
      DATA GPS_STRENGTH/1/
      DATA SNAV_VAL_IND/1/
      DATA MNAV_VAL_IND/1/
      DATA RTT_REQUEST_IND/0/
      DATA NVS/24/



      DATA OMEGA0/30.0,30.0,30.0,30.0,
     1              90.0,90.0,90.0,90.0,
     2              150.0,150.0,150.0,150.0,
     3              210.0,210.0,210.0,210.0,
     4              270.0,270.0,270.0,270.0,
     5              330.0,330.0,330.0,330.0/

      DATA ETA0/137.0,257.0,17.0,57.0,
     1            177.0,297.0,57.0,97.0,
     2            217.0,337.0,97.0,137.0,
     3            257.0,17.0,137.0,177.0,
     4            297.0,57.0,177.0,217.0,
     5            337.0,97.0,217.0,257.0/

C     RF_OUTPUT_DATA, DATA_STATUS_MSG, RF_HEALTH_STATUS_MSG

C*********************************************************************
C
C         CONVERT ALL SATELLITE ANGULAR VARIABLES FROM DEGREES TO RADIANS
C
C*********************************************************************


      OPEN(UNIT = 41, FILE = 'DXYZtrue.txt')
      OPEN(UNIT = 42, FILE = 'TIME.TXT')
      OPEN(UNIT = 44, FILE = 'VAL.TXT')
      OPEN(UNIT = 99, FILE = 'CXS_HA_ECEF.TXT')
```

```
      OPEN(UNIT = 61, FILE = 'EX.DAT')
      OPEN(UNIT = 71, FILE = 'EXCOV.DAT')
      OPEN(UNIT = 63, FILE = 'EY.DAT')
C     OPEN(UNIT = 73, FILE = 'EYCOV.DAT')
      OPEN(UNIT = 65, FILE = 'EZ.DAT')
C     OPEN(UNIT = 75, FILE = 'EZCOV.DAT')
C     OPEN(UNIT = 67, FILE = 'BC.DAT')
C     OPEN(UNIT = 77, FILE = 'BCCOV.DAT')

      WRITE(42, *) TC
      WRITE(42, *) RF_STATE


      IF (TC.LT.0.01) THEN
         RF_STATE = 2
      ENDIF



C**********************************************************************
C
C        VERSION 1.0 VARIABLE DECLARATIONS
C
C**********************************************************************

      MNAVTOV = TC
      MLAT = ELAT(1)
      MLON = ELON(1)
      MALT = EALT(1)
      MNVEL = ELATR(1)
      MEVEL = ELONR(1)
      MALTR = EALTR(1)
      TMNAVTOV = TC
      TMLAT = TLAT(1)
      TMLON = TLON(1)
      TMALT = TALT(1)
      TMNVEL = TLATR(1)
      TMEVEL = TLONR(1)
      TMALTR = TALTR(1)

      SNAVTOV = TC
      SLAT = ELAT(2)
      SLON = ELON(2)
      SALT = EALT(2)
      SNVEL = ELATR(2)
      SEVEL = ELONR(2)
      SALTR = EALTR(2)
      TSNAVTOV = TC
      TSLAT = TLAT(2)
      TSLON = TLON(2)
      TSALT = TALT(2)
      TSNVEL = TLATR(2)
      TSEVEL = TLONR(2)
      TSALTR = TALTR(2)
      BCTRUE = CBIAS(2)
      FCTRUE = FDRIFT(2)

C     WRITE(44, *) MNAVTOV, SNAVTOV, SLAT, MLON


C**********************************************************************
C
C        REFINEMENT FILTER REPRESENTED AS A FINITE STATE MACHINE WITH 7 STATES.
C      DEPENDING ON THE STATE, THE CORRESPONDING ACTIONS ARE PERFORMED
```

```
C
C*****************************************************************

   SELECT CASE (RF_STATE)

C*****************************************************************
C
C        CASE OF STATE 0:  STARTUP STATE, RF RESET
C
C*****************************************************************

      CASE (0)
C         CALL HOST_INT_PROC()
          IF (RF_OPR_SW.EQ.0) THEN
             RSI = 0
             RF_STATE = 0

          ELSE
             CALL OCP_INT_PROC (CSCI_VERS_IND, MS_IND, MNAVTOV, MLAT, MNVEL, MLON,
    1           MEVEL, MALT, MALTR, TMNAVTOV, TMLAT, TMNVEL, TMLON, TMEVEL, TMALT, TMALTR,
    2           SNAVTOV, SLAT, SNVEL, SLON, SEVEL, SALT, SALTR, TSNAVTOV, TSLAT, TSNVEL,
    3           TSLON, TSEVEL, TSALT, TSALTR, BCTRUE, FCTRUE)

             IF (SNAV_VAL_IND.EQ.0) THEN
                RSI = 1
                RF_STATE = 0

             ELSEIF (GPS_HEALTH_IND.EQ.0) THEN
                RSI = 3
                RF_STATE = 0

             ELSEIF (MS_IND.EQ.1) THEN
                RSI = 8
                RF_STATE = 1
C               CALL BUILD_MASTER_MSG()

             ELSEIF (MNAV_VAL_IND.EQ.0) THEN
                RSI = 2
                RF_STATE = 0

             ELSE
                RSI = 4
                RF_STATE = 2

                CALL RF_SOURCE_DATA_PROC(CSCI_VERS_IND, NVS,RF_STATE,MNAVTOV,MLAT,MLATR,
    1              MLON,MLONR,MALT,MALTR,TMNAVTOV,TMLAT,TMLATR,TMLON,TMLONR,TMALT,TMALTR,MSAT,
    2              MPRR,MGPSTOV,MVIS,SNAVTOV,SLAT,SLATR,SLON,SLONR,SALT,SALTR,TSNAVTOV,TSLAT,
    3              TSLATR,TSLON,TSLONR,TSALT,TSALTR,SSAT,SPRR,SGPSTOV,SVIS,INVERS,EPHEM,HA,XA,
    4              SCOUNT, YA, YAEXP, TC, SATVIS, DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE,
    5              OMEGA0,ETA0,IOTA)

                CALL RF_KALMAN_PROC(PA, PHIA, RA, QA, HA, SCOUNT, YA, YAEXP, TC, TOLD,
    1              DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE, XA)

             ENDIF
          ENDIF

C         CALL GENERATE_RF_HEALTH_STATUS_MSG()

C*****************************************************************
C
C        CASE OF STATE 1: MASTER DESIGNATION
C
C*****************************************************************
```

```
      CASE (1)
C        CALL HOST_INT_PROC()
         IF (RF_OPR_SW.EQ.0) THEN
            RSI = 0
            RF_STATE = 0

         ELSE
            CALL OCP_INT_PROC (CSCI_VERS_IND, MS_IND, MNAVTOV, MLAT, MNVEL, MLON,
     1         MEVEL, MALT, MALTR, TMNAVTOV, TMLAT, TMNVEL, TMLON, TMEVEL, TMALT, TMALTR,
     2         SNAVTOV, SLAT, SNVEL, SLON, SEVEL, SALT, SALTR, TSNAVTOV, TSLAT, TSNVEL,
     3         TSLON, TSEVEL, TSALT, TSALTR, BCTRUE, FCTRUE)

            IF (SNAV_VAL_IND.EQ.0) THEN
               RSI = 1
               RF_STATE = 0

            ELSEIF (GPS_HEALTH_IND.EQ.0) THEN
               RSI = 3
               RF_STATE = 0

            ELSE
C              CALL BUILD_MASTER_MSG()

            ENDIF

         ENDIF

C        CALL GENERATE_RF_HEALTH_STATUS_MSG()


C*********************************************************************
C
C        CASE OF STATE 2: SLAVE DESIGNATION, INITIALIZATION
C
C*********************************************************************

      CASE (2)

      IOTA = IOTA/RAD
      ANGLIM = ANGLIM/RAD

      WRITE(42, *) RF_STATE

      DO 100 I = 1,24
         OMEGA0(I) = OMEGA0(I)/RAD
         ETA0(I) = ETA0(I)/RAD
 100     CONTINUE

C*********************************************************************
C
C       COMPUTE AUXILIARY FILTER TRANSITION MATRIX PHIA(8,8)
C
C*********************************************************************

      DO 10 I = 1,8
      DO 10 J = 1,8

         PHIA(I,J) = 0.0
  10      CONTINUE

      DO I = 1,8
         PHIA(I,I) = 1.0
      END DO

      PHIA(1,2) = 0.25
      PHIA(3,2) = 0.25
```

```
      PHIA(5,6) = 0.25
      PHIA(7,8) = 0.25

C*********************************************************************
C
C       INITIALIZE AUXILIARY FILTER STATES,COVARIANCE,RATE LIMITING
C       AND OBSERVATION MATRICES
C
C*********************************************************************

      DO I = 1,8
         XA(I) = 0.0
      END DO

      DO 15 I = 1,8
      DO 15 J = 1,8
      PA(I,J) = 0.0
      QA(I,J) = 0.0
 15    CONTINUE

      DO I = 1,5,2
         PA(I,I) = POSX
      END DO

      DO I = 2,6,2
         PA(I,I) = (0.0)**2
C        PA(I,I) = (1.0)**2
      END DO

      PA(7,7) = BCX
      PA(8,8) = FCX

      DO I = 1,5,2
         QA(I,I) = PNX
      END DO


      DO I = 2,6,2
         QA(I,I) = (0.0)**2
C        QA(I,I) = (0.2)**2
      END DO


      QA(7,7) = BCPNX
      QA(8,8) = FCPNX


      DO 20 I = 1,12
      DO 20 J = 1,12
         RA(I,J) = 0.0
 20    CONTINUE

      RA(1,1) = RNX
      RA(2,2) = RNX2

      DO I = 3,12
         RA(I,I) = RNX3
      END DO

      TOLD = TC

      RF_STATE = 3


C---------------------------------------------------------------------
C
```

```
C       RF INITIALIZATION
C-------------------------------------------------------------------
C         CALL HOST_INT_PROC()
          IF (RF_OPR_SW.EQ.0) THEN
             RSI = 0
             RF_STATE = 0

          ELSE
             CALL OCP_INT_PROC (CSCI_VERS_IND, MS_IND, MNAVTOV, MLAT, MNVEL, MLON,
     1          MEVEL, MALT, MALTR, TMNAVTOV, TMLAT, TMNVEL, TMLON, TMEVEL, TMALT, TMALTR,
     2          SNAVTOV, SLAT, SNVEL, SLON, SEVEL, SALT, SALTR, TSNAVTOV, TSLAT, TSNVEL,
     3          TSLON, TSEVEL, TSALT, TSALTR, BCTRUE, FCTRUE)


             IF (SNAV_VAL_IND.EQ.0) THEN
                RSI = 1
                RF_STATE = 0

             ELSEIF (GPS_HEALTH_IND.EQ.0) THEN
                RSI = 3
                RF_STATE = 0

             ELSEIF (MNAV_VAL_IND.EQ.0) THEN
                RSI = 2
                RF_STATE = 0

             ELSE IF (GPS_STRENGTH.EQ.0)THEN
                RSI = 6
                RF_STATE = 5

             ELSE
                CALL RF_SOURCE_DATA_PROC(CSCI_VERS_IND, NVS,RF_STATE,MNAVTOV,MLAT,MLATR,
     1             MLON,MLONR,MALT,MALTR,TMNAVTOV,TMLAT,TMLATR,TMLON,TMLONR,TMALT,TMALTR,MSAT,
     2             MPRR,MGPSTOV,MVIS,SNAVTOV,SLAT,SLATR,SLON,SLONR,SALT,SALTR,TSNAVTOV,TSLAT,
     3             TSLATR,TSLON,TSLONR,TSALT,TSALTR,SSAT,SPRR,SGPSTOV,SVIS,INVERS,EPHEM,HA,XA,
     4             SCOUNT, YA, YAEXP, TC, SATVIS, DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE,
     5             OMEGA0,ETA0,IOTA)

                CALL RF_KALMAN_PROC(PA, PHIA, RA, QA, HA, SCOUNT, YA, YAEXP, TC, TOLD,
     1             DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE, XA)

             ENDIF
          ENDIF

C         CALL GENERATE_RF_HEALTH_STATUS_MSG()


C*****************************************************************
C
C        CASE OF STATE 3: SLAVE DESIGNATION, GPS PR, NOT STEADY STATE
C
C*****************************************************************

       CASE (3)
      WRITE(42, *) RF_STATE, "a"
C         CALL HOST_INT_PROC()
          IF (RF_OPR_SW.EQ.0) THEN
             RSI = 0
             RF_STATE = 0

          ELSE
             CALL OCP_INT_PROC (CSCI_VERS_IND, MS_IND, MNAVTOV, MLAT, MNVEL, MLON,
     1          MEVEL, MALT, MALTR, TMNAVTOV, TMLAT, TMNVEL, TMLON, TMEVEL, TMALT, TMALTR,
     2          SNAVTOV, SLAT, SNVEL, SLON, SEVEL, SALT, SALTR, TSNAVTOV, TSLAT, TSNVEL,
     3          TSLON, TSEVEL, TSALT, TSALTR, BCTRUE, FCTRUE)
```

```
C            WRITE(44,*)MLATR,MLONR,MALTR
C            WRITE(44,*)MNVEL,MEVEL,SALTR


             IF (SNAV_VAL_IND.EQ.0) THEN
                RSI = 1
                RF_STATE = 0

             ELSEIF (GPS_HEALTH_IND.EQ.0) THEN
                RSI = 3
                RF_STATE = 0

             ELSEIF (MNAV_VAL_IND.EQ.0) THEN
                RSI = 2
                RF_STATE = 0

             ELSEIF (GPS_STRENGTH.EQ.0) THEN
                RSI = 6
                RF_STATE = 5

             ELSE
                CALL RF_SOURCE_DATA_PROC(CSCI_VERS_IND, NVS,RF_STATE,MNAVTOV,MLAT,MLATR,
     1             MLON,MLONR,MALT,MALTR,TMNAVTOV,TMLAT,TMLATR,TMLON,TMLONR,TMALT,TMALTR,MSAT,
     2             MPRR,MGPSTOV,MVIS,SNAVTOV,SLAT,SLATR,SLON,SLONR,SALT,SALTR,TSNAVTOV,TSLAT,
     3             TSLATR,TSLON,TSLONR,TSALT,TSALTR,SSAT,SPRR,SGPSTOV,SVIS,INVERS,EPHEM,HA,XA,
     4             SCOUNT, YA, YAEXP, TC, SATVIS, DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE,
     5             OMEGA0,ETA0,IOTA)

                CALL RF_KALMAN_PROC(PA, PHIA, RA, QA, HA, SCOUNT, YA, YAEXP, TC, TOLD,
     1             DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE, XA)

C               IF (SS.EQ.1) THEN
C                  RSI = 5
C                  RF_STATE = 4
C               ENDIF

             ENDIF

          ENDIF

C         CALL GENERATE_RF_HEALTH_STATUS_MSG()


C*****************************************************************
C
C        CASE OF STATE 4: SLAVE DESIGNATION, GPS PR, STEADY STATE
C
C*****************************************************************

       CASE (4)
C         CALL HOST_INT_PROC()
          IF (RF_OPR_SW.EQ.0) THEN
             RSI = 0
             RF_STATE = 0
          WRITE(42, *) RF_STATE
          ELSE
             CALL OCP_INT_PROC (CSCI_VERS_IND, MS_IND, MNAVTOV, MLAT, MNVEL, MLON,
     1          MEVEL, MALT, MALTR, TMNAVTOV, TMLAT, TMNVEL, TMLON, TMEVEL, TMALT, TMALTR,
     2          SNAVTOV, SLAT, SNVEL, SLON, SEVEL, SALT, SALTR, TSNAVTOV, TSLAT, TSNVEL,
     3          TSLON, TSEVEL, TSALT, TSALTR, BCTRUE, FCTRUE)

             IF (SNAV_VAL_IND.EQ.0) THEN
                RSI = 1
                RF_STATE = 0
```

```
            ELSEIF (GPS_HEALTH_IND.EQ.0) THEN
               RSI = 3
               RF_STATE = 0

            ELSEIF (MNAV_VAL_IND.EQ.0) THEN
               RSI = 2
               RF_STATE = 0

            ELSEIF (GPS_STRENGTH.EQ.0) THEN
               RSI = 6
               RF_STATE = 5

            ELSE
               CALL RF_SOURCE_DATA_PROC(CSCI_VERS_IND, NVS,RF_STATE,MNAVTOV,MLAT,MLATR,
     1            MLON,MLONR,MALT,MALTR,TMNAVTOV,TMLAT,TMLATR,TMLON,TMLONR,TMALT,TMALTR,MSAT,
     2            MPRR,MGPSTOV,MVIS,SNAVTOV,SLAT,SLATR,SLON,SLONR,SALT,SALTR,TSNAVTOV,TSLAT,
     3            TSLATR,TSLON,TSLONR,TSALT,TSALTR,SSAT,SPRR,SGPSTOV,SVIS,INVERS,EPHEM,HA,XA,
     4            SCOUNT, YA, YAEXP, TC, SATVIS, DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE,
     5            OMEGA0,ETA0,IOTA)


               CALL RF_KALMAN_PROC(PA, PHIA, RA, QA, HA, SCOUNT, YA, YAEXP, TC, TOLD,
     1            DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE, XA)

            ENDIF

         ENDIF

C        CALL GENERATE_RF_HEALTH_STATUS_MSG()


C*****************************************************************
C
C        CASE OF STATE 5: SLAVE DESIGNATION, RTT&PPLI, NOT STEADY STATE
C
C*****************************************************************

      CASE (5)
C        CALL HOST_INT_PROC()
         IF (RF_OPR_SW.EQ.0) THEN
            RSI = 0
            RF_STATE = 0

         ELSE
C           CALL RTT_REQUEST()

            CALL OCP_INT_PROC (CSCI_VERS_IND, MS_IND, MNAVTOV, MLAT, MNVEL, MLON,
     1            MEVEL, MALT, MALTR, TMNAVTOV, TMLAT, TMNVEL, TMLON, TMEVEL, TMALT, TMALTR,
     2            SNAVTOV, SLAT, SNVEL, SLON, SEVEL, SALT, SALTR, TSNAVTOV, TSLAT, TSNVEL,
     3            TSLON, TSEVEL, TSALT, TSALTR, BCTRUE, FCTRUE)

            IF (SNAV_VAL_IND.EQ.0) THEN
               RSI = 1
               RF_STATE = 0

            ELSEIF (MNAV_VAL_IND.EQ.0) THEN
               RSI = 2
               RF_STATE = 0

            ELSEIF (GPS_STRENGTH.EQ.1) THEN
               RSI = 4
               RF_STATE = 2

            ELSE
```

```
                CALL RF_SOURCE_DATA_PROC(CSCI_VERS_IND, NVS,RF_STATE,MNAVTOV,MLAT,MLATR,
     1             MLON,MLONR,MALT,MALTR,TMNAVTOV,TMLAT,TMLATR,TMLON,TMLONR,TMALT,TMALTR,MSAT,
     2             MPRR,MGPSTOV,MVIS,SNAVTOV,SLAT,SLATR,SLON,SLONR,SALT,SALTR,TSNAVTOV,TSLAT,
     3             TSLATR,TSLON,TSLONR,TSALT,TSALTR,SSAT,SPRR,SGPSTOV,SVIS,INVERS,EPHEM,HA,XA,
     4             SCOUNT, YA, YAEXP, TC, SATVIS, DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE,
     5             OMEGA0,ETA0,IOTA)


                CALL RF_KALMAN_PROC(PA, PHIA, RA, QA, HA, SCOUNT, YA, YAEXP, TC, TOLD,
     1             DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE, XA)

C               IF (SS.EQ.1) THEN
C                  RSI = 7
C                  RF_STATE = 6
C               ENDIF

            ENDIF

         ENDIF

C        CALL GENERATE_RF_HEALTH_STATUS_MSG()



C*********************************************************************
C
C       CASE OF STATE 6: SLAVE DESIGNATION, RTT&PPLI, STEADY STATE
C
C*********************************************************************

      CASE (6)
         CALL HOST_INT_PROC()
         IF (RF_OPR_SW.EQ.0) THEN
            RSI = 0
            RF_STATE = 0

         ELSE
C           CALL RTT_REQUEST()

            CALL OCP_INT_PROC (CSCI_VERS_IND, MS_IND, MNAVTOV, MLAT, MNVEL, MLON,
     1          MEVEL, MALT, MALTR, TMNAVTOV, TMLAT, TMNVEL, TMLON, TMEVEL, TMALT, TMALTR,
     2          SNAVTOV, SLAT, SNVEL, SLON, SEVEL, SALT, SALTR, TSNAVTOV, TSLAT, TSNVEL,
     3          TSLON, TSEVEL, TSALT, TSALTR, BCTRUE, FCTRUE)

            IF (SNAV_VAL_IND.EQ.0) THEN
               RSI = 1
               RF_STATE = 0

            ELSEIF (MNAV_VAL_IND.EQ.0) THEN
               RSI = 2
               RF_STATE = 0

            ELSEIF (GPS_STRENGTH.EQ.1) THEN
               RSI = 4
               RF_STATE = 2

            ELSE
               CALL RF_SOURCE_DATA_PROC(CSCI_VERS_IND, NVS,RF_STATE,MNAVTOV,MLAT,MLATR,
     1             MLON,MLONR,MALT,MALTR,TMNAVTOV,TMLAT,TMLATR,TMLON,TMLONR,TMALT,TMALTR,MSAT,
     2             MPRR,MGPSTOV,MVIS,SNAVTOV,SLAT,SLATR,SLON,SLONR,SALT,SALTR,TSNAVTOV,TSLAT,
     3             TSLATR,TSLON,TSLONR,TSALT,TSALTR,SSAT,SPRR,SGPSTOV,SVIS,INVERS,EPHEM,HA,XA,
     4             SCOUNT, YA, YAEXP, TC, SATVIS, DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE,
     5             OMEGA0,ETA0,IOTA)

               CALL RF_KALMAN_PROC(PA, PHIA, RA, QA, HA, SCOUNT, YA, YAEXP, TC, TOLD,
```

```
     1               DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE, XA)

                ENDIF

             ENDIF

C            CALL GENERATE_RF_HEALTH_STATUS_MSG()


      END SELECT


      RETURN
      END

C#####################################################################

C     SUBROUTINE GENERATE_RF_HEALTH_MSG()




C     RETURN
C     END

C#####################################################################

C     SUBROUTINE RTT_REQUEST()




C     RETURN
C     END

C=====================================================================

      SUBROUTINE DTMAML( A, B, C, IRA, ICA, IRB, ICB, IFTRA, IFTRB )
C This subroutine multiplies two double precision matrices (or their transposes) of arbitrary size.
C Matrices A,B,C must be physically distinct.

C     ! History
C     !  Ver 1.00  JUL-01-1993  JEC  Initial Release

C     ! Input argument declarations.

      REAL*8     A(*), B(*)
      INTEGER*4  IRA, ICA, IRB, ICB
      INTEGER*4  IFTRA
      INTEGER*4  IFTRB

C     ! Output argument declarations.
      REAL*8     C(*)

      ! Local declarations.
      REAL*8     CSUM
      INTEGER*4  NRA, NCA, NRB, NCB
      INTEGER*4  NA, NB, NC
```

```
      INTEGER*4  NXA, NXB, NYB
      INTEGER*4  I, J, K
      INTEGER*4  IOP,IAA,IA,KAA,KAO,JBB,JBO,KBB,KBO,JCC,JCO,KC,JB,KA,
     $           KB,JC

C     ! Get the location (address) of C and see if its the same as that of A or B.
C       IF ( %LOC(C) .EQ. %LOC(A) .OR. %LOC(C) .EQ. %LOC(B) ) THEN
C         WRITE( *, '('' Error in MAT_MULT - C is A or B'')' )
C         RETURN
C       END IF

C       ! Determine option.
        NRA = IRA
        NCA = ICA
        NRB = IRB
        NCB = ICB
        IOP = IFTRA + IFTRB + IFTRB + 1
        GO TO ( 2, 5, 8, 11), IOP

C       ! Conventional multiply  A * B.
 2      CONTINUE

        IAA = 1
        IA  = 0
        KAA = NRA
        KAO = -NRA
        JBB = NCA
        JBO = -NCA
        KBB = 1
        KBO = 0
        JCC = NRA
        JCO = -NRA
        NXA = NRA
        NXB = NCB
        NYB = NCA
        GO  TO 15

C       ! Transpose A only.
 5      CONTINUE

        IAA = NRA
        IA  = -NRA
        KAA = 1
        KAO = 0
        JBB = NRB
        JBO = -NRB
        KBB = 1
        KBO = 0
        JCC = NCA
        JCO = -NCA
        NXA = NCA
        NXB = NCB
        NYB = NRA
        GO  TO 15

C       ! Transpose B only.
 8      CONTINUE

        IAA = 1
        IA  = 0
        KAA = NRA
        KAO = -NRA
        JBB = 1
        JBO = 0
        KBB = NRB
        KBO = -NRB
```

```
      JCC = NRA
      JCO = -NRA
      NXA = NRA
      NXB = NRB
      NYB = NCA
      GO  TO 15

C     ! Transpose A and B.
  11  CONTINUE

      IAA = NRA
      IA  = -NRA
      KAA = 1
      KAO = 0
      JBB = 1
      JBO = 0
      KBB = NRB
      KBO = -NRB
      JCC = NCA
      JCO = -NCA
      NXA = NCA
      NXB = NRB
      NYB = NRA

C     ! Multiply the matrices.
  15  CONTINUE

      DO I = 1, NXA
    IA = IA + IAA
    JC = JCO
    JB = JBO
    DO J = 1, NXB
       JC = JC + JCC
       JB = JB + JBB
       KA = KAO
       KB = KBO
       CSUM = 0.
       DO K = 1, NYB
          KA = KA + KAA
          KB = KB + KBB
          NA = KA + IA
          NB = KB + JB
          CSUM = CSUM + A(NA) * B(NB)
       END DO
       NC = JC + I
       C(NC) = CSUM
    END DO
      END DO

      RETURN
      END

C=======================================================================

      SUBROUTINE DMINV1(A,N,D)

C     ! This routine inverts a matrix, in place. This is an IBM subroutine, I beleive. I changed
C     ! the arguments, slightly, reformated code for clarity, and declared all variables. Since the
C     ! code was probably written in fortran 66 or fortran IV one has to be careful with do loops
C     ! if compiling under fortran 77.  Everything looks and tested ok.

C     ! History:
      !  Ver 1.01  OCT-13-1994  JEC  Brought over to RNS, reformated comments slightly.
C     !  Ver 1.00  FEB-02-1990  JEC  Initial creation.

C     ! Limitations:
```

```
C      !  Limited to 20x20 matrices - to expand change L and M working vectors.

     ! Argument declarations.
      REAL*8 A(*)
      REAL*8 D
      INTEGER*4 N

C     ! Local declarations.
      REAL*8     BIGA, HOLD
      INTEGER*4  L(20),M(20),NK,KK,K,J,KJ,IJ,IZ,I,KI,JI,JP,JK,JR,JQ,IK
      LOGICAL    FLAG

C     ! Search for largest element.
      FLAG = .FALSE.
      D = 1.0
      NK = -N
      DO 80  K = 1, N

     NK = NK + N
     L(K) = K
     M(K) = K
     KK = NK + K
     BIGA = A(KK)
     DO 20 J = K, N
         IZ = N * (J-1)
         DO 20 I = K, N
        IJ = IZ + I
 10              IF ( ABS(BIGA) - ABS(A(IJ)) ) 15,20,20
 15              BIGA = A(IJ)
        L(K) = I
        M(K) = J
 20      CONTINUE

C          ! Interchange rows.
   J = L(K)
    IF ( J-K ) 35,35,25
 25      KI = K - N
    DO  I = 1, N
        KI = KI + N
        HOLD = -A(KI)
        JI = KI - K + J
        A(KI) = A(JI)
        A(JI) = HOLD
    END DO

C          ! Interchange columns
 35      I = M(K)
    IF ( I - K ) 45,45,38
 38      JP = N * (I-1)
    DO J = 1, N
        JK = NK + J
        JI = JP + J
        HOLD = -A(JK)
        A(JK) = A(JI)
        A(JI) = HOLD
    END DO

C      ! Divide column by minus pivot (value of pivot element is contained in biga)
 45      IF(BIGA) 48,46,48
 46      FLAG = .TRUE.
    WRITE (6,200)
 200   FORMAT(/1x, "THE MATRIX IS SINGULAR")
         write(42,*)time, "singular"
    RETURN
 48     DO 55 I = 1, N
         IF ( I - K ) 50,55,50
```

```
 50         IK = NK + I
           A(IK) = A(IK) / (-BIGA)
 55       CONTINUE

C           ! Reduce matrix
       DO 65 I = 1, N
           IK = NK + I
           IJ = I - N
           DO 65 J = 1, N
          IJ = IJ + N
          IF ( I - K ) 60,65,60
 60              IF ( J - K ) 62,65,62
 62            KJ = IJ - I + K
          A(IJ) = A(IK) * A(KJ) + A(IJ)
 65       CONTINUE
C          ! Divide row by pivot
       KJ = K - N
       DO 75 J = 1, N
           KJ = KJ + N
           IF ( J-K ) 70,75,70
 70          A(KJ) = A(KJ) / BIGA
 75       CONTINUE

C           ! Product of pivots
       D = D * BIGA

C           ! Replace pivot by reciprocal
       A(KK) = 1.0D00 / BIGA

 80     CONTINUE

C       ! Final row and column interchange
       K = N
 100   K = K-1
       IF ( K ) 150,150,105
 105   I = L(K)
       IF ( I-K ) 120,120,108
 108   JQ = N * (K-1)
       JR = N * (I-1)
       DO 110 J = 1, N
      JK = JQ + J
      HOLD = A(JK)
      JI = JR + J
      A(JK) = -A(JI)
 110    A(JI) =HOLD
 120    J = M(K)
       IF ( J-K ) 100,100,125
 125   KI = K - N
       DO 130 I = 1, N
     KI = KI + N
     HOLD = A(KI)
     JI = KI - K + J
     A(KI) = -A(JI)
 130    A(JI) = HOLD
        GO TO 100

 150   RETURN
       END
```

```
C-------------------------------------------------------------------------
C      VERSION NO. 1.0   DATE: 24 FEBRUARY 2003
```

```
C
C     PURPOSE:  SYNCHRONIZES AND PREPARES DATA FOR KALMAN PROCESSING.  MASTER GPS
C           DATA ARE EXTRACTED FROM THE MASTER MESSAGE AND PAIRED WITH OWN NAV
C           SOLUTION.
C
C   INPUTS:
C
C   CSCI_VERS_IND - Laboratory/Field Indicator
C   EPHEM(30,23)  - Satellite Ephemeris Data
C
C
C   MNAV_DATA (MASTER NAV DATA PACKAGE)
C      MNAVTOV          - Time of Validity of MASTER Navigation solution
C      MLAT          - ESTIMATED MASTER Latitude position [rad]
C      MLATR         - ESTIMATED MASTER Latitude rate [rad/s]
C      MLON          - ESTIMATED MASTER Longitude position [rad]
C      MLONR         - ESTIMATED MASTER Longitude rate [rad/s]
C      MALT          - ESTIMATED MASTER Altitude [m]
C      MALTR         - ESTIMATED MASTER Altitude rate [m/s]
C      TMLAT         - TRUE MASTER Latitude position [rad]
C      TMLATR        - TRUE MASTER Latitude rate [rad/s]
C      TMLON         - TRUE MASTER Longitude position [rad]
C      TMLONR        - TRUE MASTER Longitude rate [rad/s]
C      TMALT         - TRUE MASTER Altitude [m]
C      TMALTR        - TRUE MASTER Altitude rate [m/s]
C      MNAV_VAL_IND  - MASTER navigation validity indicator
C
C   MGPS_DATA
C      MVIS          - Number of satellites visible to MASTER
C      MSAT(MVIS)       - MASTER Ordered Satellite List
C      MPRR(MVIS)       - MASTER Ordered measured code phase array
C      MGPSTOV          - MASTER predicted GPS time of validity
C      MINVERS(30)      - Satellite sorting index
C
C   SNAV_DATA (SELF NAV DATA PACKAGE)
C      SNAVTOV          - Time of Validity of SELF Navigation solution
C      SLAT          - ESTIMATED SELF Latitude position [rad]
C      SLATR         - ESTIMATED SELF Latitude rate [rad/s]
C      SLON          - ESTIMATED SELF Longitude position [rad]
C      SLONR         - ESTIMATED SELF Longitude  rate [rad/s]
C      SALT          - ESTIMATED SELF Altitude [m]
C      SALTR         - ESTIMATED SELF Altitude rate [m/s]
C      TSLAT         - TRUE SELF Latitude position [rad]
C      TSLATR        - TRUE SELF Latitude rate [rad/s]
C      TSLON         - TRUE SELF Longitude position [rad]
C      TSLONR        - TRUE SELF Longitude  rate [rad/s]
C      TSALT         - TRUE SELF Altitude [m]
C      TSALTR        - TRUE SELF Altitude rate [m/s]
C      BCTRUE        - TRUE clock bias
C      FCTRUE        - TRUE frequency drift.
C      SNAV_VAL_IND  - SELF navigation validity indicator
C
C   SGPS_DATA
C      SVIS          - Number of satellites visible to SELF
C      SSAT(SVIS)       - SELF Ordered Visible Satellite List
C      SPRR(SVIS)       - SELF Ordered measured code phase array
C      SGPSTOV          - SELF predicted GPS time of validity
C      SINVERS(30)      - Satellite sorting index
C
C   OMEGA0(24)          - Lat coordinates of satellites
C   ETA0(24)         - Lon coordinates of satellites
C   IOTA             - Satellite Inclination
C
C
C   OUTPUTS:
C
```

```
C   PA(8,8)              - Covariance Matrix
C   PHIA(8,8)          - State transition Matrix
C   RA(12,12)          - Measurement Error
C   QA(8,8)              - Process Noise
C   HA(12,8)           - Observation matrix
C   SCOUNT             - Final number of paired satellites visible above ANGLIM
C   YA(INDEX)            - INDEX = 1, 14          Observation measurement vector
C   YAEXP(INDEX)       - INDEX = 1, 14           Observation prediction vector
C   TC                 - Reference time at which measurements are valid since GPS time of week [s]
C   SATVIS(j)          - j = 1, SCOUNT           Ordered array of common visible satellites above
ANGLIM
C   DXTRUE
C   DYTRUE
C   DZTRUE             - TRUE error composites.
C   BCTRUE             - TRUE clock bias
C   FCTRUE             - TRUE frequency drift.
C
C
C   STORED CONSTANTS:
C
C   IX                 - 12478       Seed for random number generator
C   RNG1SG             - 21.0     Standard Dev
C   ZEROH              - 0.0D0       Mean for random number
C
C   OMEGAE             - 7.292115467 E-5 [RAD/S]  Earth's rotational rate
C   OMEGAS             - 1.45858522 E-4 [RAD/S]   Satellite orbital rate
C   RK                 - 87051108 [ft]                          Radius of satellite orbit
C   RAD                - 57.2957795131                          Radians/Degree
C   QUANT              - 12.5
C
C
C
C   CALLED BY:
C      RF_EXEC_CTRL
C
C
C
C   SUBROUTINES:
C      COMPARE_SAT_INDICES
C      COMPUTE_CIRC_SAT_ECEF
C      COMPUTE_ELLIP_SAT_ECEF
C      XFORM_NAV_COORDINATES
C      COMPUTE_LAB_SAT_VIS
C      COMPUTE_FIELD_SAT_VIS
C      PROCESS_PR_DATA
C      GENERATE_KALMAN_OBS_SET
C
C
C   MODICATION HISTORY:
C      V1.0   ORIGINAL VERSION DATE      RESPONSIBLE
C
C             FOR VERSION 1.0     4/03/03       JYH
C
C
C
C
C----------------------------------------------------------------------
      SUBROUTINE RF_SOURCE_DATA_PROC(CSCI_VERS_IND, NVS,RF_STATE,MNAVTOV,MLAT,MLATR,
     1   MLON,MLONR,MALT,MALTR,TMNAVTOV,TMLAT,TMLATR,TMLON,TMLONR,TMALT,TMALTR,MSAT,
     2   MPRR,MGPSTOV,MVIS,SNAVTOV,SLAT,SLATR,SLON,SLONR,SALT,SALTR,TSNAVTOV,TSLAT,TSLATR,
     3 TSLON,TSLONR,TSALT,TSALTR,SSAT,SPRR,SGPSTOV,SVIS,INVERS,EPHEM,HA,XA,
     4 SCOUNT, YA, YAEXP, TC, SATVIS, DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE,
     5 OMEGA0,ETA0,IOTA)
```

```
C*****************************************************************
C
C       DECLARATION OF VARIABLES
C
C*****************************************************************

C   COMMON    MVIS, SVIS, NVS, CSCI_VERS_IND, SCOUNT, TC

    IMPLICIT NONE

    INTEGER*4 MVIS, SVIS, NVS, SCOUNT, RF_STATE, CSCI_VERS_IND, MS, TE
    REAL*8    MSAT(MVIS), SSAT(SVIS), MPRR(MVIS), SPRR(MVIS), SAT(NVS), PRR1(NVS), PRR2(NVS)
    REAL*8    EPHEM(23,30), INVERS(30), ECEFXS(30), ECEFYS(30), ECEFZS(30)
    REAL*8    OMEGA0(24), ETA0(24), IOTA
    REAL*8    TMGPSTOV, TMNAVTOV, TMLAT, TMLATR, TMLON, TMLONR, TMALT, TMALTR, TMLLPX,TMLLPY, TMLLPZ
    REAL*8    TSGPSTOV, TSNAVTOV, TSLAT, TSLATR, TSLON, TSLONR, TSALT, TSALTR, TSLLPX,TSLLPY, TSLLPZ
    REAL*8    MGPSTOV, MNAVTOV, MLAT, MLATR, MLON, MLONR, MALT, MALTR, MLLPX, MLLPY, MLLPZ
    REAL*8    SGPSTOV, SNAVTOV, SLAT, SLATR, SLON, SLONR, SALT, SALTR, SLLPX, SLLPY, SLLPZ

    REAL*8    MPGPSTM(NVS), SPGPSTM(NVS)
    REAL*8    TLE(3,3),XA(8), BCTRUE, FCTRUE
    REAL*8    ECEFS(3,1), LLSAT(3,1), LLSATX(NVS), LLSATY(NVS), LLSATZ(NVS)
    REAL*8    SDIFF(3), MDIFF(3), TSDIFF(3), TMDIFF(3), VIS(NVS)
    REAL*8    SATVIS(NVS), EMPR(NVS), ESPR(NVS), TMPR(NVS), TSPR(NVS),CXS(24), CYS(24), CZS(24)

    REAL*8    BPR2M, BBC2M, TC, HA(12,8), YA(12),YAEXP(12), DXTRUE, DYTRUE, DZTRUE

    REAL*8 OMEGAE, OMEGAS, RK, RAD

    REAL*8 QUANT, RANGEM, RTRAB, RNG1SG, ZEROH, RAND, EHRNGE, BIASM

    INTEGER*4 IBIAS,IRANGE,INDEX,IX

    DATA   IX/12478/
    DATA   RNG1SG/21.0/
    DATA   ZEROH/0.0D0/

    DATA OMEGAE,OMEGAS/7.292115467E-5,1.45858522E-4/
    DATA RK/87051108./
    DATA RAD/57.2957795131/
    DATA QUANT/12.5/


C*****************************************************************
C
C       BEGIN
C
C*****************************************************************

    OPEN(UNIT = 41, FILE = 'DXYZtrue.txt')
    OPEN(UNIT = 43, FILE = 'DEBUG.TXT')




    IF (CSCI_VERS_IND.EQ.1) THEN
       CALL COMPUTE_SAT_CIRC_ECEF (TC, NVS, OMEGA0, ETA0, IOTA, ECEFXS, ECEFYS, ECEFZS)


       MS = 1
       TE = 1

       CALL XFORM_NAV_COORD(TMGPSTOV, TMNAVTOV, TMLAT, TMLATR, TMLON, TMLONR, TMALT, TMALTR,
    1      TMLLPX, TMLLPY, TMLLPZ, TLE, MS, TE)
```

```
          WRITE(99,912) MS,TMLLPX,TMLLPY,TMLLPZ
  912         FORMAT(1H ,'K/TLLP',1X,I1,3(1X,E12.6))
        MS = 2

        CALL XFORM_NAV_COORD(TSGPSTOV, TSNAVTOV, TSLAT, TSLATR, TSLON, TSLONR, TSALT, TSALTR,
     1     TSLLPX, TSLLPY, TSLLPZ, TLE, MS, TE)

          WRITE(99,913) MS,TSLLPX,TSLLPY,TSLLPZ
  913         FORMAT(1H ,'K/TLLP',1X,I1,3(1X,E12.6))

        MS = 1
        TE = 2

        CALL XFORM_NAV_COORD (MGPSTOV, MNAVTOV, MLAT, MLATR, MLON, MLONR, MALT, MALTR,
     1     MLLPX, MLLPY, MLLPZ, TLE, MS, TE)

          WRITE(99,910) MS,MLLPX,MLLPY,MLLPZ
  910         FORMAT(1H ,'K/LLP',1X,I1,3(1X,E12.6))
        MS = 2

        CALL XFORM_NAV_COORD(SGPSTOV, SNAVTOV, SLAT, SLATR, SLON, SLONR, SALT, SALTR,
     1     SLLPX, SLLPY, SLLPZ, TLE, MS, TE)

          WRITE(99,911) MS,SLLPX,SLLPY,SLLPZ
  911         FORMAT(1H ,'K/LLP',1X,I1,3(1X,E12.6))


        CALL COMPUTE_LAB_SAT_VIS(NVS, SAT, ECEFXS, ECEFYS, ECEFZS, MLLPX, MLLPY,
     1     MLLPZ, SLLPX, SLLPY, SLLPZ, TMLLPX, TMLLPY, TMLLPZ, TSLLPX, TSLLPY, TSLLPZ,
     2     TLE, XA, SCOUNT, SATVIS, EMPR, ESPR, TMPR, TSPR, CXS, CYS, CZS)


        CALL GENERATE_KALMAN_OBS_SET( SCOUNT, EMPR, ESPR, TMPR, TSPR, SATVIS, MLLPX,
     1     MLLPY, MLLPZ, SLLPX, SLLPY, SLLPZ, TMLLPX, TMLLPY, TMLLPZ, TSLLPX, TSLLPY, TSLLPZ,
     2     BPR2M, BBC2M, TC, RF_STATE, XA, HA, YA, YAEXP, DXTRUE, DYTRUE, DZTRUE,BCTRUE,CXS,CYS,CZS)
        RETURN

        ELSE
C-------------------------------------------------------------------------
C
C   INSERT CODE FOR BUFFERING OF MASTER/SELF DATA PACKAGES
C   PAIR EARLIEST UNUSED MASTER/SELF PACKAGES
C
C-------------------------------------------------------------------------
        CALL COMPARE_SAT_INDICES (MVIS, SVIS, MSAT, SSAT, MPRR, SPRR, NVS,
     1     SAT, PRR1, PRR2)

        CALL COMPUTE_SAT_ELLIP_ECEF(PRR2, EPHEM, NVS, SAT, INVERS, SGPSTOV,
     1     ECEFXS, ECEFYS, ECEFZS)

        MS = 1

        CALL XFORM_NAV_COORD (MGPSTOV, MNAVTOV, MLAT, MLATR, MLON, MLONR, MALT, MALTR,
     1     MLLPX, MLLPY, MLLPZ, TLE, MS, TE)

        MS = 2

        CALL XFORM_NAV_COORD(SGPSTOV, SNAVTOV, SLAT, SLATR, SLON, SLONR, SALT, SALTR,
     1     SLLPX, SLLPY, SLLPZ, TLE, MS, TE)

        IF (CSCI_VERS_IND-2) 210, 210, 220
  210       CALL XFORM_NAV_COORD(TMGPSTOV, TMNAVTOV, TMLAT, TMLATR, TMLON, TMLONR, TMALT, TMALTR,
     1       TMLLPX, TMLLPY, TMLLPZ, TLE, MS, TE)

          MS = 1
```

```
            CALL XFORM_NAV_COORD(TSGPSTOV, TSNAVTOV, TSLAT, TSLATR, TSLON, TSLONR, TSALT, TSALTR,
     1         TSLLPX, TSLLPY, TSLLPZ, TLE, MS, TE)

            MS = 2

C 220       CALL COMPUTE_SAT_VIS()

220     CALL PROCESS_PR_DATA( SCOUNT, SATVIS, NVS, PRR1, PRR2, MGPSTOV, SGPSTOV, EMPR, ESPR )

        CALL GENERATE_KALMAN_OBS_SET( SCOUNT, EMPR, ESPR, TMPR, TSPR, SATVIS, MLLPX,
     1      MLLPY, MLLPZ, SLLPX, SLLPY, SLLPZ, TMLLPX, TMLLPY, TMLLPZ, TSLLPX, TSLLPY, TSLLPZ,
     2      BPR2M, BBC2M, TC, RF_STATE, XA, HA, YA, YAEXP, DXTRUE, DYTRUE,DZTRUE,BCTRUE,CXS,CYS,CZS)

    ENDIF


    RETURN
    END



C#########################################################################
C   SUBROUTINE COMPARE_SAT_INDICES ( MVIS, SVIS, MSAT(MVIS), SSAT(SVIS), MPRR(MVIS),
C          SPRR(SVIS), NVS,SAT(NVS), PRR1(NVS), PRR2(NVS) )
C
C   PURPOSE:  COMPARES ORDERED SATELLITE INDICES AND ALIGNS SATELLITES IN A SINGLE
C             ARRAY CONTAINING DATA FOR PSEUDORANGE COMPARISON
C
C
C   INPUTS:
C
C   MVIS           - Number of satellites visible to MASTER
C   SVIS           - Number of satellites visible to SELF
C   MSAT(MVIS)       - Ordered array of MASTER visible satellites
C   SSAT(SVIS)       - Ordered array of SELF visible satellites
C   MPRR(MVIS)       - Ordered array of MASTER code phase
C   SPRR(SVIS)       - Ordered array of SELF code phase
C
C
C   OUTPUTS:
C   NVS            - Number visible satellites - common to MASTER and SELF
C   SAT(NVS)       - Ordered array of common visible satellites
C   PRR1(NVS)      - MASTER array of code phase for common satellites
C   PRR2(NVS)      - SELF array of code phase for common satellites
C
C
C
C#########################################################################

    SUBROUTINE COMPARE_SAT_INDICES ( MVIS, SVIS, MSAT, SSAT, MPRR, SPRR, NVS,
    1   SAT, PRR1, PRR2)

    IMPLICIT NONE

    INTEGER*4 MVIS, SVIS, NVS, I, J
    REAL*8    MSAT(MVIS), SSAT(SVIS), MPRR(MVIS), SPRR(SVIS), SAT(NVS)
    REAL*8    PRR1(NVS), PRR2(NVS)


    NVS = 0

    DO 100 I = 1, MVIS
       DO WHILE ((MSAT(I).LE.SSAT(J)) .AND. (J.LE.SVIS))
          IF (MSAT(I).EQ.SSAT(J)) THEN
             NVS = NVS + 1
```

```
            SAT(I) = MSAT(I)
            PRR1(I) = MPRR(I)
            PRR2(I) = SPRR(J)
            J = J+1
         ENDIF
      END DO

100   CONTINUE

      RETURN
      END

C#########################################################################
C   SUBROUTINE COMPUTE_SAT_CIRC_ECEF (SNAVTOV, NVS, SAT(NVS),
C          ECEFXS(K), ECEFYS(K), ECEFZS(K))
C
C   PURPOSE:  PERFORMS THE CONVERSION OF THE CIRCULAR SATELLITE CONSTELLATION
C             INTO ECEF COORDINATES FOR VERSION 1.0
C
C   INPUTS:
C   SGPSTOV          - GPS
C   NVS           - Number visible satellites common to MASTER and SELF
C   SAT(NVS)      - Ordered array of common visible satellites
C
C
C   OUTPUTS:
C   ECEFXS(K)     - ECEF x-coordinate for satellite K
C   ECEFYS(K)     - ECEF y-coordinate for satellite K
C   ECEFZS(K)     - ECEF z-coordinate for satellite K
C
C
C   STORED CONSTANTS
C   OMEGA0(K)     - K = 1, 30      Lat coordinates of satellites
C   ETA0(I)          - K = 1, 30   Lon coordinates of satellites
C   OMEGAE        - 7.292115467 E-5 [RAD/S]      Earth's rotational rate
C   OMEGAS        - 1.45858522 E-4 [RAD/S]       Satellite orbital rate
C   RK            - 87051108 [meters]     Radius of satellite orbit
C   IOTA          - 55.0/2*pi [RAD]       Satellite Inclination
C
C
C
C#########################################################################


      SUBROUTINE COMPUTE_SAT_CIRC_ECEF (TC, NVS, OMEGA0, ETA0, IOTA, ECEFXS, ECEFYS, ECEFZS)

      IMPLICIT NONE

      INTEGER*4   NVS, K
      REAL*8    SGPSTOV, SAT(NVS), ECEFXS(30), ECEFYS(30), ECEFZS(30)
      REAL*8    ETAK, OMEGAK, IOTA, OMEGA0(24), ETA0(24), TC
      REAL*8    OMEGAE, OMEGAS, RK, XKP,YKP

C     DATA IOTA/0.959931088596585/
      DATA OMEGAE,OMEGAS/7.292115467E-5,1.45858522E-4/
      DATA RK/87051108./
C     DATA RAD/57.2957795131/

C**********************************************************************
C
C        CIRCULAR CONSTELLATION PROPAGATION ALGORITHM
C
C**********************************************************************

C     IOTA = IOTA/RAD
C        SCOUNT = 0
```

```
      DO 220 K = 1, NVS
C*********************************************************************
C
C        INCREMENT ANGLE VARIABLES FOR SATELLITE I
C
C*********************************************************************

      OMEGAK = OMEGA0(K) - TC * OMEGAE
      ETAK   = ETA0(K)   + TC * OMEGAS



C*********************************************************************
C
C        COMPUTE ECEF POSITIONS OF EACH SATELLITE
C
C*********************************************************************

      XKP = RK*DCOS(ETAK)
      YKP = RK*DSIN(ETAK)

      ECEFXS(K) = XKP*DCOS(OMEGAK) - YKP*DSIN(OMEGAK)*DCOS(IOTA)
      ECEFYS(K) = XKP*DSIN(OMEGAK) + YKP*DCOS(OMEGAK)*DCOS(IOTA)
      ECEFZS(K) = YKP*DSIN(IOTA)

  220 CONTINUE

      RETURN
      END

C##########################################################################
C   SUBROUTINE COMPUTE_SAT_ELLIP_ECEF (PRR2, EPHEM(23,30), NVS, SAT(NVS), INVERS(30), SGPSTOV,
C          ECEFXS(K), ECEFYS(K), ECEFZS(K))
C
C   PURPOSE:  PERFORMS THE CONVERSION OF THE ELLIPTICAL SATELLITE CONSTELLATION
C             INTO ECEF COORDINATES FOR VERSION 2.0 AND ABOVE
C
C   INPUTS:
C   NVS           - Number visible satellites common to MASTER and SELF
C   SAT(NVS)      - Ordered array of common visible satellites
C   EPHEM(23,30)  - Satellite Ephemeris Data
C   INVERS(30)      - Satellite sorting index
C   SGPSTOV         - Predicted GPS time
C   BLCK          - Number of PR reports (?)
C   CDPHASE         - GPS Code phase value (?)
C   ISVN0         - Satellite vehicle number (?)
C
C
C
C   OUTPUTS:
C   ECEFXS(L)     - ECEF x-coordinate for satellite L
C   ECEFYS(L)     - ECEF y-coordinate for satellite L
C   ECEFZS(L)     - ECEF z-coordinate for satellite L
C
C
C   STORED CONSTANTS
C   OMEGA0(K)     - K = 1, 30      Lat coordinates of satellites
C   ETA0(I)         - K = 1, 30  Lon coordinates of satellites
C   OMEGAE        - 7.292115467 E-5 [RAD/S]      Earth's rotational rate
C   OMEGAS        - 1.45858522 E-4 [RAD/S]       Satellite orbital rate
C   RK            - 87051108 [meters]     Radius of satellite orbit
C   MU            - 3.986008 E14 [m^3/s^2]       Earth's uni gravitational param
C   F             - -4.442807633E-10
```

```
C
C
C  SVNO   = EPHEM(K,1)  Satellite vehicle number
C  WEEKNO = EPHEM(K,2)  Time of week *** difference from toe
C  TGD    = EPHEM(K,3)
C  TOC    = EPHEM(K,4)
C  AF2    = EPHEM(K,5)
C  AF1    = EPHEM(K,6)
C  AF0    = EPHEM(K,7)
C  CRS    = EPHEM(K,8)  Amplitude of sine harmonic correction term to orbit radius [m]
C  DELN   = EPHEM(K,9)  mean motion difference [semicircle/s]
C  M0     = EPHEM(K,10) Mean anomaly at reference time [smcir]
C  CUC    = EPHEM(K,11) Amplitude of cos harmonic correction term to the argument of latitude [rad]
C  ES     = EPHEM(K,12) Eccentricity
C  CUS    = EPHEM(K,13) Amplitude of sine harmonic correction term to the argument of latitude [rad]
C  SAS    = EPHEM(K,14) Square root of semi-major axis [m^1/2]
C  TOE    = EPHEM(K,15) Ephemeris reference time (from Epoch)
C  CIC    = EPHEM(K,16) Amplitude of cos harmonic correction term to the angle of inclination [rad]
C  OMEGA0 = EPHEM(K,17) Longitude of ascending node of orbit plane at weekly epoch
C  CIS    = EPHEM(K,18) Amplitude of sine harmonic correction term to the angle of inclination [rad]
C  IOTA   = EPHEM(K,19) Inclination angle at reference time [semicircle]
C  CRC    = EPHEM(K,20) Amplitude of cos harmonic correction term to the orbit radius [m]
C  W      = EPHEM(K,21) Argument of perigee [semicircle]
C  OMEGAD = EPHEM(K,22) Rate of right ascension [semicircle/s]
C  IDOT   = EPHEM(K,23) Rate of inclination angle [semicir/s]
C
C
C
C#########################################################################

      SUBROUTINE COMPUTE_SAT_ELLIP_ECEF (PRR2, EPHEM, NVS, SAT, INVERS, SGPSTOV,
     1      ECEFXS, ECEFYS, ECEFZS)


      INTEGER*4   NVS, ISVNO
      REAL*8     PRR2(NVS), EPHEM(23,30), SGPSTOV, SAT(NVS), INVERS(30), BLCK, CDPHASE
      REAL*8     ECEFXS(30), ECEFYS(30), ECEFZS(30)


      INTEGER*4 SVNO, IPGPS
      REAL*8     CDPHSE, CR
      REAL*8     FPGPS, PRSEC, TC, TOE, TK, DELN, SAS, AS, N, MU, M0, MK
      REAL*8     ES, EKNEW, JX, EKOLD, E, DEKTR, AF0, AF1, AF2, TGD, TOC, DELT, T
      REAL*8     R, NUM, DENOM, W, PHI, CUS, CUC, CRS, OMEGA0, CIS, CRC, CIC, IOTA, IDOT
      REAL*8     C2PHI, S2PHI, DELPHI, DELR, DELI, OMEGAD, OER


      DATA RAD/57.2957795131/
      DATA C/2.99792498E8/
      DATA RK/87051108./
      DATA PI/3.1415926/
      DATA OMEGAE,OMEGAS/7.292115467E-5,1.45858522E-4/
      DATA MU/3.986008E14/
      DATA F/-4.442807633E-10/
      DATA CR/1/

C*************************************************************************
C
C        ELLIPTICAL CONSTELLATION PROPAGATION ALGORITHM
C
C*************************************************************************
```

```
      DO 300 L = 1, NVS
C*********************************************************************
C
C        INCREMENT ANGLE VARIABLES FOR SATELLITE I
C
C*********************************************************************
        CDPHSE = PRR2(L)
        ISVNO = SAT(L)
        IPGPS = SGPSTOV
        FPGPS = SGPSTOV - FLOAT(IPGPS)
        PRSEC = FPGPS - (CDPHSE/CR)
        IF (PRSEC < 0.0) THEN
           PRSEC = 1.0 + PRSEC

        PR = PRSEC*C
        TC = SGPSTOV - PR/C
        TOE = EPHEM(INVERS(ISVNO),15)
        TK = TC - TOE
        ENDIF

        IF (TK > 302400.0) THEN
           TC = TC - 604800.0
        ENDIF
        IF (TK < 302400.0) THEN
           TC = TC + 604800.0
        ENDIF

C*********************************************************************
C
C          COMPUTE MEAN MOTION
C
C*********************************************************************

        DELN = EPHEM(INVERS(ISVNO),9)
        SAS = EPHEM(INVERS(ISVNO),14)
        AS = SAS**2
        N = DSQRT(MU/AS**3) + DELN

C*********************************************************************
C
C          COMPUTE MEAN ANOMALY
C
C*********************************************************************
        M0 = EPHEM(INVERS(ISVNO),10)
        MK = M0 + N*(TC - TOE)

C*********************************************************************
C
C          SOLVE FOR ECCENTRIC ANOMALY
C
C*********************************************************************
        ES = EPHEM(INVERS(ISVNO),12)
        EKNEW = MK
        DO JX=1,100
           EKOLD = EKNEW
        EKNEW = MK + ES*DSIN(EKOLD)
        END DO
        E = EKNEW

C*********************************************************************
C
C          COMPUTE TIME CORRECTION TERM
C
C*********************************************************************
```

```
      DELTR = F*ES*SAS*DSIN(E)

      AF0 = EPHEM(INVERS(ISVNO),7)
      AF1 = EPHEM(INVERS(ISVNO),6)
      AF2 = EPHEM(INVERS(ISVNO),5)
      TGD = EPHEM(INVERS(ISVNO),3)
      TOC = EPHEM(INVERS(ISVNO),4)
      DELT = AF0 + AF1*(TC - TOC) + AF2*(TC - TOC)**2 + DELTR - TGD

      T = TC - DELT
      R = AS*(1.0 - ES*DSIN(E))

C*********************************************************************
C
C        COMPUTE SATELLITE TRUE ANOMALY, NU
C
C*********************************************************************
      NUM = SQRT((1.0 - ES*ES)*DSIN(E))/(1.0 - ES*DCOS(E))
      DENOM = (DCOS(E)-ES)/(1.0-ES*DCOS(E))
      NU = ATAN2(NUM, DENOM)
      W = EPHEM(INVERS(ISVNO),21)
      PHI = NU + W

C*********************************************************************
C
C        COMPUTE CORRECTION TERMS FOR HARMONICS
C
C*********************************************************************
      CUS = EPHEM(INVERS(ISVNO),13)
      CUC = EPHEM(INVERS(ISVNO),11)
      CRS = EPHEM(INVERS(ISVNO),8)
      OMEGA0 = EPHEM(INVERS(ISVNO),17)
      CIS = EPHEM(INVERS(ISVNO),18)
      CRC = EPHEM(INVERS(ISVNO),20)
      CIC = EPHEM(INVERS(ISVNO),16)
      IOTA = EPHEM(INVERS(ISVNO),19)
      IDOT = EPHEM(INVERS(ISVNO),23)

      C2PHI = DCOS(2*PHI)
      S2PHI = DSIN(2*PHI)

C*********************************************************************
C
C        SECOND HARMONIC PERTURBATIONS
C
C*********************************************************************

      DELPHI = CUS*S2PHI + CUC*C2PHI
      DELR = CRS*S2PHI + CRC*C2PHI
      DELI = CIS*S2PHI + CIC*C2PHI

      PHI = PHI + DELPHI
      R = R + DELR
      IOTA = IOTA + DELI + IDOT*(T-TOE)

      OMEGAD = EPHEM(INVERS(ISVNO),22)
      OER = OMEGA0/RAD + (OMEGAD/PI)*(T-TOE) - OMEGAE*T


      ECEFXS(L) = (R*DCOS(OER)*DCOS(PHI))-(R*DSIN(OER)*DCOS(IOTA)*DSIN(PHI))
      ECEFYS(L) = R*DSIN(OER)*DCOS(PHI)+R*DCOS(OER)*DCOS(IOTA)*DSIN(PHI)
      ECEFZS(L) = R*DSIN(IOTA)*DSIN(PHI)


  300   CONTINUE
```

```
      RETURN
      END


C#########################################################################
C   SUBROUTINE XFORM_NAV_COORD (NAVTOV, LAT, LATR. LON, LONR, ALT, ALTR,
C       LLPX, LLPY, LLPZ)
C
C   PURPOSE:  EXTRAPOLATES BOTH SELF AND MASTER NAVIGATION SOLUTIONS INTO ECEF
C             AND LOCAL LEVEL COORDINATE VALUES.
C
C
C   INPUTS:
C
C   NAVTOV        - Time of Validity of MASTER Navigation solution
C   LAT           - Latitude position [rad]
C   LATR          - Latitude rate [rad/s]
C   LON           - Longitude position [rad]
C   LONR          - Longitude rate [rad/s]
C   ALT           - Altitude [m]
C   ALTR          - Altitude rate [m/s]
C
C
C   OUTPUTS:
C   LLPX          - Local level x-coordinate value
C   LLPY          - Local level y-coordinate value
C   LLPZ          - Local level z-coordinate value
C   TLE(3,3)      - Local level transform matrix
C
C
C   STORED CONSTANTS
C   ar            - 2.09257382 E7        Earth semi-minor radius [ft]
C   e22           - 6.74499984 E-3 Earth eccentricity squared
C
C
C
C   LOCAL VARIABLES
C   TEXT          - EXTRAPOLATED TIME
C   LATE          - EXTRAPOLATED LATITUDE
C   LONE          - EXTRAPOLATED LONGITUDE
C   ALTE          - EXTRAPOLATED ALTITUDE
C
C
C
C
C#########################################################################

      SUBROUTINE XFORM_NAV_COORD (GPSTOV, NAVTOV, LAT, LATR, LON, LONR, ALT, ALTR,
     1   LLPX, LLPY, LLPZ, TLE, MS, TE)

      IMPLICIT NONE

      INTEGER*4 MS, TE

      REAL*8    SL, CL, SP, CP

      REAL*8    GPSTOV, NAVTOV, LAT, LATR, LON, LONR, ALT, ALTR, LLPX, LLPY, LLPZ
      REAL*8    TEXT, LATE, LONE, ALTE, ECEFXP, ECEFYP, ECEFZP
      REAL*8    TLE(3,3), ECEFP(3,1), LLP(3,1), REW, AR, RAD, E22

      DATA   AR/2.09257382E7/
      DATA   E22/6.74499984E-3/
      DATA   RAD/57.2957795131/

C**********************************************************************
```

```
C
C        EXTRAPOLATE NAVIGATION TO TIME OF GPS MEASURE
C*********************************************************************

C   TEXT = GPSTOV - NAVTOV
    TEXT = 0

    LATE = LAT !+ LATR*TEXT
    LONE = LON !+ LONR*TEXT
    ALTE = ALT !+ ALTR*TEXT


C*********************************************************************
C
C        COMPUTE ECEF POSITION
C
C*********************************************************************
C   write(44,*) DSQRT(1.0 - E22*DSIN(LAT)**2)
    REW = AR/DSQRT(1.0 - E22*DSIN(LAT)**2)
    ECEFXP = (REW + ALTE)*DCOS(LATE)*DCOS(LONE)
    ECEFYP = (REW + ALTE)*DCOS(LATE)*DSIN(LONE)
    ECEFZP = ( (1.0 - E22)*REW + ALTE)*DSIN(LATE)

    ECEFP(1,1) = ECEFXP
    ECEFP(2,1) = ECEFYP
    ECEFP(3,1) = ECEFZP


C*********************************************************************
C
C       COMPUTE TRANSFORMATION MATRIX BETWEEN ECF AND LOCAL LEVEL
C         AT REFERENCE PLATFORM (WITH MASTER NAV)
C
C*********************************************************************

    IF (MS.EQ.1) THEN
       IF (TE.EQ.1) THEN
       SL = DSIN(LATE)
       CL = DCOS(LATE)
       SP = DSIN(LONE)
       CP = DCOS(LONE)

       TLE(1,1) = -SL*CP
       TLE(1,2) = -SL*SP
       TLE(1,3) = CL
       TLE(2,1) = SP
       TLE(2,2) = -CP
       TLE(2,3) = 0.0
       TLE(3,1) = CL*CP
       TLE(3,2) = CL*SP
       TLE(3,3) = SL
       ENDIF
    ENDIF

C*********************************************************************
C
C        COMPUTE POSITION VECTOR OF PLATFORM IN LOCAL LEVEL COORDINATES
C
C*********************************************************************

    CALL DTMAML(TLE,ECEFP,LLP,3,3,3,1,0,0)

    LLPX = LLP(1,1)
    LLPY = LLP(2,1)
    LLPZ = LLP(3,1)
```

```
      RETURN
      END


C#########################################################################
C   SUBROUTINE COMPUTE_LAB_SAT_VIS ( SAT(NVS),ECEFXS(K),ECEFYS(K),ECEFZS(K),MLLPX,MLLPY,
C       MLLPZ,SLLPX,SLLPY,SLLPZ,TMLLPX,TMLLPY,TMLLPZ,TSLLPX,TSLLPY,TSLLPZ,TLE(),XA(k),
C       SCOUNT, SATVIS(SCOUNT), EMPR(SCOUNT), ESPR(SCOUNT), TMPR(SCOUNT), TSPR(SCOUNT),
c       CXS(SCOUNT), CYS(SCOUNT), CZS(SCOUNT) )
C
C   PURPOSE:  RESTRICTS DGPS SATELLITE USAGE TO A SET OF SATELLITES AT LEAST
C             5 DEGREES ABOVE THE HORIZON.  ALSO COMPUTES DIRECTION COSINES AND
C             PSEUDORANGE VALUES BETWEEN MASTER AND SELF TO ALL SELECTED SATELLITES
C
C
C   INPUTS:
C
C   NVS        - Number visible satellites - common to MASTER and SELF
C   SAT(NVS)   - Ordered array of common visible satellites
C   ECEFXS(K)  - ECEF x-coordinate for satellite K
C   ECEFYS(K)  - ECEF y-coordinate for satellite K
C   ECEFZS(K)  - ECEF z-coordinate for satellite K
C   MLLPX      - ESTIMATED MASTER local level x-coordinate value
C   MLLPY      - ESTIMATED MASTER local level y-coordinate value
C   MLLPZ      - ESTIMATED MASTER local level z-coordinate value
C   SLLPX      - ESTIMATED SLAVE local level x-coordinate value
C   SLLPY      - ESTIMATED SLAVE local level y-coordinate value
C   SLLPZ      - ESTIMATED SLAVE local level z-coordinate value
C   TMLLPX     - TRUE MASTER local level x-coordinate value
C   TMLLPY     - TRUE MASTER local level y-coordinate value
C   TMLLPZ     - TRUE MASTER local level z-coordinate value
C   TSLLPX     - TRUE SLAVE local level x-coordinate value
C   TSLLPY     - TRUE SLAVE local level y-coordinate value
C   TSLLPZ     - TRUE SLAVE local level z-coordinate value
C   TLE(3,3)   - Local Level transform matrix
C   XA(k)      - Refinement Filter solution vector, k = 1,7
C
C
C   OUTPUTS:
C
C   SCOUNT        - Final number of paired satellites visible above ANGLIM
C   SATVIS(SCOUNT)   - Ordered array of common visible satellites above ANGLIM
C   EMPR(SCOUNT)  - PREDICTED Pseudorange Vector to satellites from MASTER
C   ESPR(SCOUNT)  - PREDICTED Pseudorange Vector to satellites from SELF
C   TMPR(SCOUNT)  - TRUE Pseudorange Vector to satellites from MASTER
C   TSPR(SCOUNT)  - TRUE Pseudorange Vector to satellites from SELF
C   CXS(SCOUNT)      - Direction cosine x-component
C   CYS(SCOUNT)      - Direction cosine y-component
C   CZS(SCOUNT)      - Direction cosine z-component
C
C
C   STORED CONSTANTS
C   ANGLIM        - 5 degrees      Angle of inclination that bounds visibility
C
C
C
C#########################################################################

      SUBROUTINE COMPUTE_LAB_SAT_VIS ( NVS, SAT, ECEFXS, ECEFYS, ECEFZS, MLLPX, MLLPY,
     1   MLLPZ, SLLPX, SLLPY, SLLPZ, TMLLPX, TMLLPY, TMLLPZ, TSLLPX, TSLLPY, TSLLPZ,
     2   TLE, XA, SCOUNT, SATVIS, EMPR, ESPR, TMPR, TSPR, CXS, CYS, CZS)


      IMPLICIT NONE
```

```
      INTEGER*4 K, SCOUNT, NVS, IND, SATVIS(NVS)
      REAL*8    SAT(NVS), ECEFXS(NVS), ECEFYS(NVS), ECEFZS(NVS), MLLPX,MLLPY,MLLPZ
      REAL*8    SLLPX,SLLPY,SLLPZ,TMLLPX,TMLLPY,TMLLPZ,TSLLPX,TSLLPY,TSLLPZ,TLE(3,3),XA(8)
      REAL*8    ECEFS(3,1), LLSAT(3,1), LLSATX(NVS), LLSATY(NVS), LLSATZ(NVS)
      REAL*8    SDIFF(3), MDIFF(3), TSDIFF(3), TMDIFF(3), VIS(NVS), ANGLIM
      REAL*8    EMPR(NVS), ESPR(NVS), TMPR(NVS), TSPR(NVS),CXS(NVS), CYS(NVS), CZS(NVS)
      REAL*8    ELEV, ELEVD, RAD

      DATA   ANGLIM/8.726646158107211E-002/

      SCOUNT = 0

      DO 400 K = 1,NVS

         ECEFS(1,1) = ECEFXS(K)
         ECEFS(2,1) = ECEFYS(K)
         ECEFS(3,1) = ECEFZS(K)

         CALL DTMAML(TLE,ECEFS,LLSAT,3,3,3,1,0,0)

         LLSATX(K) = LLSAT(1,1)
         LLSATY(K) = LLSAT(2,1)
         LLSATZ(K) = LLSAT(3,1)


C*********************************************************************
C
C        COMPUTE DIFFERENCE VECTOR BETWEEN SATELLITE AND PLATFORM
C        TEST FOR POSITIVE Z VALUE
C        IF POSITIVE, THEN TEST FOR ELEVATION ANGLE
C
C*********************************************************************

         SDIFF(1) = LLSATX(K) - SLLPX + XA(1)
         SDIFF(2) = LLSATY(K) - SLLPY + XA(3)
         SDIFF(3) = LLSATZ(K) - SLLPZ + XA(5)

         TMDIFF(1) = LLSATX(K) - TMLLPX
         TMDIFF(2) = LLSATY(K) - TMLLPY
         TMDIFF(3) = LLSATZ(K) - TMLLPZ

         IF(SDIFF(3).LT.0.)THEN
            VIS(K) = 0
            GO TO 450
         ELSE
            ELEV = DATAN2(TMDIFF(3), DSQRT(TMDIFF(1)**2 + TMDIFF(2)**2))
            ELEVD = ELEV*RAD


            IF( ELEV.GE.ANGLIM) THEN
               VIS(K) = 1
               SCOUNT = SCOUNT + 1
c              IND = SCOUNT
               SATVIS(SCOUNT) = K    ! without IND, SCOUNT becomes 29314719208749012749021 78
c              SCOUNT = IND

               TMPR(K) = DSQRT(TMDIFF(1)**2 + TMDIFF(2)**2 + TMDIFF(3)**2)

               ESPR(K) = DSQRT(SDIFF(1)**2 + SDIFF(2)**2 + SDIFF(3)**2)


C*********************************************************************

C        COMPUTE DIRECTION COSINES FROM PLATFORM 1 TO ALL VISIBLE SATELLITES
C
C*********************************************************************
```

```
            CXS(K) = SDIFF(1)/ESPR(K)
            CYS(K) = SDIFF(2)/ESPR(K)
            CZS(K) = SDIFF(3)/ESPR(K)

C*****************************************************************
C
C       COMPUTE TRUE RANGE FROM SATELLITES TO MASTER AND SLAVE
C
C*****************************************************************

            TSDIFF(1) = LLSATX(K) - TSLLPX
            TSDIFF(2) = LLSATY(K) - TSLLPY
            TSDIFF(3) = LLSATZ(K) - TSLLPZ
            TSPR(K) = DSQRT(TSDIFF(1)**2 + TSDIFF(2)**2 + TSDIFF(3)**2)

            MDIFF(1) = LLSATX(K) - MLLPX
            MDIFF(2) = LLSATY(K) - MLLPY
            MDIFF(3) = LLSATZ(K) - MLLPZ
            EMPR(K) = DSQRT(MDIFF(1)**2 + MDIFF(2)**2 + MDIFF(3)**2)


         ELSE
            VIS(K) = 0
         ENDIF

      ENDIF

   IF (SCOUNT.GT.10) THEN
      SCOUNT = 10
   ENDIF


 450  CONTINUE

 400  CONTINUE


   RETURN
   END



C#####################################################################
C  SUBROUTINE PROCESS_PR_DATA ( SCOUNT, SATVIS, NVS, PRR1, PRR2, MPGPSTM
C      SPGPSTM, EMPR, ESPR )
C
C  PURPOSE:  PERFORMS A CONVERSION OF SATELLITE CODEPHASE DATA TO GENERATE AN ARRAY
C           OF PSEUDORANGE DATA BETWEEN THE MASTER OR SELF AND COMMON VISIBLE SATELLITES.
C
C
C  INPUTS:
C
C  SCOUNT     - Final paired satellite count satisfying elevation criteria
C  SATVIS(K),    - K=1,SCOUNT Satellite Visibility vector
C  NVS        - Number of satellites paired, but not yet checked for elevation
C  PRR1(j)       - j=1,NVS MASTER sorted code phase data
C  PRR2(j)       - j=1,NVS SELF sorted code phase data
C  MPGPSTM(j)    - j=1,NVS MASTER sorted predicted GPS time vector
C  SPGPSTM(j)    - j=1,NVS SELF sorted predicted GPS time vector
C
C
C  OUTPUTS:
C
```

```
C   EMPR(K)       - MASTER predicted pseudorange vector [m]
C   ESPR(K)       - SELF predicted pseudorange vector [m]
C
C   STORED CONSTANTS
C   C          - 2.99792498 E8 [m/s]      Speed of light
C   CR         - TBD  Code Phase conversion constant
C
C
C###########################################################################

      SUBROUTINE PROCESS_PR_DATA ( SCOUNT, SATVIS, NVS, PRR1, PRR2, MGPSTOV,
     1      SGPSTOV, EMPR, ESPR )

      IMPLICIT NONE

      INTEGER*4 SCOUNT, NVS, I, K, IPGPS
      REAL*8    PRR1(NVS), PRR2(NVS), MGPSTOV(NVS), SGPSTOV(NVS), EMPR(NVS), ESPR(NVS)
      REAL*8    PGPSTM(NVS), PRR(NVS), PGPST, CDPHSE, PRSEC , SATVIS(SCOUNT)
      REAL*8    C, CR, FPGPS

      DATA   C/2.99792498E8/
      DATA   CR/1/


      DO I = 1,NVS
         PGPSTM(I) = MGPSTOV(I)
         PRR(I) = PRR1(I)
      END DO


      DO 500 K = 1,SCOUNT
         PGPST = PGPSTM(SATVIS(K))
         CDPHSE = PRR(SATVIS(K))
         IPGPS = PGPST
         FPGPS = PGPST - FLOAT(IPGPS)
         PRSEC = FPGPS - (CDPHSE/CR)
         IF (PRSEC.LT.0.0) THEN
            PRSEC = 1.0 + PRSEC
         ENDIF

         EMPR(K) = PRSEC*C

500 CONTINUE



      DO I = 1,NVS
         PGPSTM(I) = SGPSTOV(I)
         PRR(I) = PRR2(I)
      END DO


      DO 550 K = 1,SCOUNT
         PGPST = PGPSTM(SATVIS(K))
         CDPHSE = PRR(SATVIS(K))
         IPGPS = PGPST
         FPGPS = PGPST - FLOAT(IPGPS)
         PRSEC = FPGPS - (CDPHSE/CR)
         IF (PRSEC.LT.0.0) THEN
            PRSEC = 1.0 + PRSEC
         ENDIF

         ESPR(K) = PRSEC*C

550 CONTINUE
```

```
      RETURN
      END



C#####################################################################
C   SUBROUTINE GENERATE_KALMAN_OBS_SET ( SCOUNT, EMPR, ESPR, TMPR, TSPR, SATVIS, MLLPX,
C       MLLPY, MLLPZ, SLLPX, SLLPY, SLLPZ, TMLLPX, TMLLPY, TMLLPZ, TSLLPX, TSLLPY, TSLLPZ,
C       BPR2M, BBC2M, TC, RF_STATE, XA, HA, YA, YAEXP, SATVIS, DXTRUE, DYTRUE, DZTRUE )
C
C   PURPOSE:  PERFORMS A CONVERSION OF SATELLITE CODEPHASE DATA TO GENERATE AN ARRAY
C             OF PSEUDORANGE DATA BETWEEN THE MASTER OR SELF AND COMMON VISIBLE SATELLITES.
C
C
C   INPUTS:
C
C   SCOUNT*            Number of selected observations
C   EMPR(SCOUNT)*EST(predicted) pseudorange array from MASTER
C   ESPR(SCOUNT)*EST(predicted) pseudorange array from SELF
C   TMPR(SCOUNT)*TRUE(measured) pseudorange array from MASTER
C   TSPR(SCOUNT)*TRUE(measured) pseudorange array from SELF
C   SATVIS(K)*         K = 1, NS Satellite Visibility vector
C   MLLPX          ESTIMATED MASTER local level x-coordinate value
C   MLLPY          ESTIMATED MASTER local level y-coordinate value
C   MLLPZ          ESTIMATED MASTER local level z-coordinate value
C   SLLPX          ESTIMATED SLAVE local level x-coordinate value
C   SLLPY          ESTIMATED SLAVE local level y-coordinate value
C   SLLPZ          ESTIMATED SLAVE local level z-coordinate value
C   TMLLPX         TRUE MASTER local level x-coordinate value
C   TMLLPY         TRUE MASTER local level y-coordinate value
C   TMLLPZ         TRUE MASTER local level z-coordinate value
C   TSLLPX         TRUE SLAVE local level x-coordinate value
C   TSLLPY         TRUE SLAVE local level y-coordinate value
C   TSLLPZ         TRUE SLAVE local level z-coordinate value
C
C   BPR2M**            Backup measured range to master
C   BBC2M**            Backup measured clock bias to master
C
C   TC             Observation set time of validity
C   RF_STATE       State of the refinement filter
C   X(A)           A = 1, 8  Current RF state vector
C
C
C   OUTPUTS:
C
C   HA(12,8)       Observation matrix for Kalman Filter
C   YA(12)         Inputs to the Kalman Filter
C   YAEXP(14)      Expected Kalman solutions
C   SCOUNT         Number of selected observations
C   TC             Observation set time of validity
C   SATVIS(K)*         K = 1, NS Satellite Visibility vector
C   DXTRUE         TRUE Error composite x-coordinate value
C   DYTRUE         TRUE Error composite y-coordinate value
C   DZTRUE         TRUE Error composite z-coordinate value
C
C
C
C   STORED CONSTANTS
C   C          - 2.99792498 E8 [m/s]      Speed of light
C   CR         - TBD  Code Phase conversion constant
C
C
```

```
C#########################################################################

      SUBROUTINE GENERATE_KALMAN_OBS_SET ( SCOUNT, EMPR, ESPR, TMPR, TSPR, SATVIS, MLLPX,
     1   MLLPY, MLLPZ, SLLPX, SLLPY, SLLPZ, TMLLPX, TMLLPY, TMLLPZ, TSLLPX, TSLLPY, TSLLPZ,
     2   BPR2M, BBC2M, TC, RF_STATE, XA, HA, YA, YAEXP, DXTRUE, DYTRUE, DZTRUE, BCTRUE, CXS, CYS, CZS)

      IMPLICIT NONE

      INTEGER*4 SCOUNT, RF_STATE, I, J, K, INDEX, IND, SATVIS(SCOUNT), IBIAS, IRANGE, IX
      REAL*8    EMPR(24), ESPR(24), TMPR(24), TSPR(24)
      REAL*8    MLLPX, MLLPY, MLLPZ, SLLPX, SLLPY, SLLPZ, TMLLPX, TMLLPY, TMLLPZ
      REAL*8    TSLLPX, TSLLPY, TSLLPZ, BPR2M, BBC2M, XA(8), CXS(24), CYS(24), CZS(24)
      REAL*8    TC, HA(12,8), YA(12),YAEXP(12), DXTRUE, DYTRUE, DZTRUE, BCTRUE
      REAL*8    DELEX, DELEY, DELEZ, DELX, DELY, DELZ, RTRAB, RESAB, RANGEM

      REAL*8 CM, CR, C, ZEROH, RNG1SG, QUANT, EHRNGE, BIASM, RAND

      DATA   CM/0.983571194156611/
C     DATA   C/2.99792498E8/
      DATA   CR/1/
      DATA   QUANT/12.5/
      DATA   IX/12478/
      DATA   RNG1SG/21.0/
      DATA   ZEROH/0.0D0/


C*********************************************************************
C
C       COMPUTE TRUE RANGE BETWEEN PLATFORMS 1 AND 2
C
C*********************************************************************

      DELX = TSLLPX - TMLLPX
      DELY = TSLLPY - TMLLPY
      DELZ = TSLLPZ - TMLLPZ
      RTRAB = DSQRT(DELX**2 + DELY**2 + DELZ**2)


C*********************************************************************
C
C       COMPUTE ESTIMATED RANGE BETWEEN PLATFORMS 1 AND 2 BASED ON
C       NAVIGATIONAL SOLUTIONS, BUT NOT YET INCLUDING CLOCK BIAS
C
C*********************************************************************

      DELEX = SLLPX - MLLPX
      DELEY = SLLPY - MLLPY
      DELEZ = SLLPZ - MLLPZ
      RESAB = DSQRT(DELEX**2 + DELEY**2 + DELEZ**2)


C*********************************************************************
C
C       NAV ERROR RESULTS
C
C*********************************************************************

      DXTRUE = DELEX - DELX
      DYTRUE = DELEY - DELY
      DZTRUE = DELEZ - DELZ


      WRITE(41,600)TC, DXTRUE,DYTRUE,DZTRUE

  600    FORMAT(1H 4(1X,E14.8))
```

```
C*********************************************************************
C
C       (BEGINNING OF MEASUREMENT PROCESSING INNER LOOP)
C
C     COMPUTE EXPECTED AND OBSERVED MEASUREMENTS FOR AUXILIARY FILTER
C     NOTE THAT RANGE AND TIME MEASUREMENTS ARE QUANTIZED TO 12.5ns FOR LINK 16 DERIVED DATA
C
C*********************************************************************

C       RANGEM = RTRAB + BCTRUE*C + GAUSSIAN NOISE TERM

    RANGEM = RTRAB + BCTRUE*CM
    CALL GAUSKX(IX,RNG1SG,ZEROH,RAND)



    RANGEM = RANGEM + RAND

    IRANGE = RANGEM/QUANT
    YA(1) = FLOAT(IRANGE)*QUANT

    EHRNGE = DSQRT(DELEX**2 + DELEY**2)
    YAEXP(1) = EHRNGE + XA(7)*CM  !YAEXP(1) = RESAB

C       BC MEASURED IS BCTRUE + GAUSSIAN NOISE TERM

    BIASM = BCTRUE
    IBIAS = BIASM/QUANT

    YA(2) = FLOAT(IBIAS)*QUANT
    YAEXP(2) = XA(7)


C*********************************************************************
C
C       COMPUTE DGPS MEASURED (YA) AND PREDICTED (YAEXP) VALUES
C
C*********************************************************************
    DO 610 J = 1,SCOUNT
    INDEX = J + 2

    YA(INDEX) =   TSPR(SATVIS(J)) - TMPR(SATVIS(J)) + BCTRUE * CM

    YAEXP(INDEX)= ESPR(SATVIS(J)) - EMPR(SATVIS(J)) + XA(7) * CM


 610  CONTINUE



    DO 620 I = 1,12
    DO 620 J = 1,8
    HA(I,J) = 0.0
 620     CONTINUE

C*********************************************************************
C
C       RANGE PARTIAL DERIVATIVES
C
C*********************************************************************

    HA(1,1) = DELX/RESAB
    IF(ABS(HA(1,1)).LT.1.3)THEN
```

```
      HA(1,1) = 0.0
    ENDIF
    HA(1,3) = DELY/RESAB
    IF(ABS(HA(1,3)).LT.1.3)THEN
      HA(1,3) = 0.0
    ENDIF
    HA(1,5) = DELZ/RESAB
    IF(ABS(HA(1,5)).LT.1.3)THEN
      HA(1,5) = 0.0
    ENDIF

C*********************************************************************
C
C       CLOCK BIAS PARTIAL DERIVATIVE
C
C*********************************************************************

    HA(2,7) = 0.0


C*********************************************************************
C
C       DELTA PSEUDORANGE PARTIAL DERIVATIVE
C     USE SATELLITES INDEXED BY VECTOR SATVIS(24)
C
C*********************************************************************


    DO 650 J = 1,SCOUNT
    INDEX = 2 + J
C   IND = SATVIS(J)
    HA(INDEX,1) =  CXS(SATVIS(J))
    HA(INDEX,3) =  CYS(SATVIS(J))
    HA(INDEX,5) =  CZS(SATVIS(J))

    IF(ABS(CXS(SATVIS(J))).LT..6)THEN
      HA(INDEX,1) = 0.0
    ENDIF

    IF(ABS(CYS(SATVIS(J))).LT..6)THEN
      HA(INDEX,3) = 0.0
    ENDIF

    IF(ABS(CZS(SATVIS(J))).LT..6)THEN
      HA(INDEX,5) = 0.0
    ENDIF

    HA(INDEX,7) =  CM

650   CONTINUE


      RETURN
      END




C ====================================================================

    SUBROUTINE GAUSKX(IX,S,AM,V)

    INTEGER*4 IX
    REAL*8 S,AM,V,A
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   S = STANDARD DEV DESIRED
C   AM = MEAN DESIRED
C   V = VALUE OF COMPUTED RANDOM VARIABLE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

    M = 2147483647
    A = 0.0
    DO 50 I=1,12
    CALL LEHMER(IX)
    A = A+DFLOAT(IX)/M
 50 CONTINUE

    V = (A-6.0D0)*S + AM
    RETURN
    END


C=====================================================================

    SUBROUTINE LEHMER(SEED)
    INTEGER*4 SEED, A,Z,M,R,Q, HI, LO, TEST

    A =16807
    M = 2147483647
    Q = 127773
    R = 2836
    HI= SEED/Q
    LO= MOD(SEED,Q)
    TEST = A*LO - R*HI

    IF(TEST.GT.0) THEN
       SEED = TEST
    ELSE
       SEED = TEST+M
    ENDIF

    RETURN
    END
```

```
C-----------------------------------------------------------------------
C     VERSION NO. 1.0   DATE: 24 FEBRUARY 2003
C
C     PURPOSE:  PERFORMS THE EXTENDED KALMAN FILTERING PROCESSES NEEDED TO ESTIMATE THE NUMERIC
C           VALUES OF THE EIGHT REFINEMENT FILTER ERROR TERMS CONTAINED IN STATE VECTOR XA.
C
C
C   INPUTS:
C
C   PA(8,8)              - Covariance Matrix
C   PHIA(8,8)         - State transition Matrix
C   RA(12,12)         - Measurement Error
C   QA(8,8)              - Process Noise
C   HA(12,8)          - Observation matrix
C   SCOUNT            - Final number of paired satellites visible above ANGLIM
C   YA(INDEX)            - INDEX = 1, 14         Observation measurement vector
C   YAEXP(INDEX)      - INDEX = 1, 14            Observation prediction vector
C   TC                - Reference time at which measurements are valid since GPS time of week [s]
C   DXTRUE
C   DYTRUE
C   DZTRUE            - True error composites.
C   BCTRUE            - TRUE clock bias
C   FCTRUE            - TRUE frequency drift
```

```
C
C
C   OUTPUTS:
C
C   RF_RESULTS_PKG
C
C
C
C
C   STORED CONSTANTS:
C
C   POSX = (1000.0)**2  Initial Position Covariance Constant [m2]
C   BCX =  (100.0)**2   Initial Clock Bias Covariance Constant [ns2]
C   FCX =   (10.0)**2   Initial Freq Drift Covariance Constant [ns2/sec2]
C   PNX = (1000.0)**2   Process Noise Position Constant [m2]
C   BCPNX = (10.0)**2   Process Noise Clock Bias Constant [ns2]
C   FCPNX =  (1.0)**2   Process Noise Freq Drift Constant [ns2/sec2]
C   RNX =   (21.0)**2   Range Measurement Variance Constant [m2]
C   RNX2 =  (12.5)**2   RTT Clock Bias Measurement Variance Constant [ns2]
C   RNX3 =   (4.0)**2   DGPS Measurement Variance Constant [m2]
C   SSLIM =   TBD    Steady State Indication [m]
C
C
C   CALLED BY:
C      RF_EXEC_CTRL
C
C
C
C   SUBROUTINES:
C
C
C   MODICATION HISTORY:
C      V1.0   ORIGINAL VERSION DATE      RESPONSIBLE
C
C             FOR VERSION 1.0     4/03/03       JYH
C
C
C
C
C-------------------------------------------------------------------------


      SUBROUTINE RF_KALMAN_PROC(PA, PHIA, RA, QA, HA, SCOUNT, YA, YAEXP, TC, TOLD,
     1      DXTRUE, DYTRUE, DZTRUE, BCTRUE, FCTRUE, XA)
C*********************************************************************
C
C       DECLARATION OF VARIABLES
C
C*********************************************************************

C     COMMON SCOUNT, TC

      IMPLICIT NONE

      REAL*8 YA(12),YAEXP(12)
      REAL*8 PHIA(8,8),PA(8,8),QA(8,8),RA(12,12),XA(8)
      REAL*8 RSL(12),DTA(8)
      REAL*8 TC, TOLD, TEXT
      REAL*8 DXTRUE,DYTRUE,DZTRUE,BCTRUE,FCTRUE
      REAL*8 SIGX,SIGY,SIGZ,SIGBC
      REAL*8 ERRX,ERRY,ERRZ,ERRBC,ERRFC

      REAL*8 HA(12,8),DUM3(8,8)
      REAL*8 G(8,12),DUM(8,12),DUM1(12,12),DENOM(12,12),DUM2(8,8)
      INTEGER*4 LW(12),NW(12),SCOUNT,DIMLIM, I,J,K
      REAL*8 DUMX(8,8),M(8,8),D
```

```
C*********************************************************************
C
C       COMPUTE KALMAN GAIN MATRIX G(8,12)
C
C*********************************************************************


   CALL DTMAML(PA,HA,DUM,8,8,12,8,0,1)
   CALL DTMAML(HA,DUM,DENOM,12,8,8,12,0,0)
   DO 200 I = 1,12
   DO 200 J = 1,12
   DUM1(I,J) = DENOM(I,J) + RA(I,J)

 200    CONTINUE

   CALL DMINV1(DUM1,12,D)
   CALL DTMAML(DUM,DUM1,G,8,12,12,12,0,0)

C*********************************************************************
C
C       APPLY KALMAN STATE CORRECTION - COMPUTE RESIDUAL
C
C*********************************************************************

   DIMLIM = 2 + SCOUNT
   DO 202 I = 1,DIMLIM
   RSL(I) = YA(I) - YAEXP(I)

 202    CONTINUE


C*********************************************************************
C
C       APPLY KALMAN STATE CORRECTION - COMPUTE RF STATE INCREMENTS
C
C*********************************************************************

   DO 205 I = 1,8
   DTA(I) = 0.0
   DO 205 J = 1,DIMLIM

   DTA(I) = DTA(I) + G(I,J)*RSL(J)

 205    CONTINUE


C*********************************************************************
C
C       APPLY KALMAN STATE CORRECTION - UPDATE RF STATE VECTOR
C
C*********************************************************************

   DO 2205 I = 1,8
   IF(TC.GT.30..AND.ABS(DTA(I)).LT.1.0)THEN
      DO 2206 J = 1,12
C          G(I,J) = 0.0
C          DTA(I) = 0.0
 2206   CONTINUE
   ENDIF
 2205   CONTINUE

   DO 206 I = 1,8
```

```
      XA(I) = XA(I) + DTA(I)
  206 CONTINUE


C*********************************************************************
C
C       GENERATE ERROR TERMS FOR STATUS REPORT
C
C*********************************************************************

      ERRX = XA(1) - DXTRUE
      ERRY = XA(3) - DYTRUE
      ERRZ = XA(5) - DZTRUE
      ERRBC = XA(7) - BCTRUE
      ERRFC = XA(8) - FCTRUE



C*********************************************************************
C
C       OBSERVATION UPDATE OF COVARIANCE MATRIX PA
C
C*********************************************************************

      CALL DTMAML(G,HA,DUM2,8,12,12,8,0,0)
      DO 220 I = 1,8
      DO 220 J = 1,8
      DUM2(I,J) = -DUM2(I,J)
  220 CONTINUE
      DO 225 I = 1,8
      DUM2(I,I) = 1.0 + DUM2(I,I)
  225 CONTINUE
      CALL DTMAML(DUM2,PA,DUM3,8,8,8,8,0,0)
      DO 230 I = 1,8
      DO 230 J = 1,8
      PA(I,J) = DUM3(I,J)
  230 CONTINUE

C*********************************************************************
C
C       COMPUTE COVARIANCE DIAGONAL SQUARE ROOTS
C
C*********************************************************************

      SIGX = DSQRT(PA(1,1))
      SIGY = DSQRT(PA(3,3))
      SIGZ = DSQRT(PA(5,5))
      SIGBC= DSQRT(PA(7,7))


C*********************************************************************
C
C       TIME EXTRAPOLATE COVARIANCE ONE TIME STEP
C
C*********************************************************************

C     TEXT = TC-TOLD

      CALL DTMAML(PHIA,PA,DUMX,8,8,8,8,0,0)
      CALL DTMAML(DUMX,PHIA,M,8,8,8,8,0,1)

      DO 240 I = 1,8
      DO 240 J = 1,8

      PA(I,J) = M(I,J) + QA(I,J)
```

```
 240     CONTINUE

   if(mod(tc,2000.0).lt.1.0e-5)then
      do 1235 i = 1,8
      do 1235 j = 1,8
      pa(i,j) = 0.0
 1235      continue
      do 1236 i = 1,5,2
      pa(i,i) = (10.0)**2
 1236      continue
      pa(7,7) = (10.0)**2
      pa(8,8) = (2.0)**2
   endif

C*********************************************************************
C
C       TIME EXTRAPOLATE FILTER STATES BY ONE STEP
C
C*********************************************************************

   XA(1) = XA(1) + XA(2)*0.25
   XA(3) = XA(3) + XA(4)*0.25
   XA(5) = XA(5) + XA(6)*0.25
   XA(7) = XA(7) + XA(8)*0.25

C*********************************************************************
C       WRITE OUTPUT DATA TO FILE
C*********************************************************************


   WRITE(61,*) TC,ERRX
C  WRITE(71,*) T,SIGX
   WRITE(63,*) TC,ERRY
C  WRITE(73,*) T,SIGY
   WRITE(65,*) TC,ERRZ
C  WRITE(75,*) T,SIGZ
C  WRITE(67,*) T,ERRBC
C  WRITE(77,*) T,SIGBC


   RETURN
   END
```

```
C-----------------------------------------------------------------------
C  SUBROUTINE OCP_INT_PROC (MS_IND,LF_IND,MNAVTOV,MLAT,MNVEL,MLON,MEVEL,MALT,MALTR,
C     GPS_HEALTH_IND, MGPSTOV, BLCK, UMSAT(BLCK),UMPRR(BLCK),TMNAVTOV,TMLAT,TMNVEL,
C     TMLON,TMEVEL,TMALT,TMALTR, SNAVTOV,SLAT,SNVEL,SLON,SEVEL,SALT,SALTR,
C     TSNAVTOV, TSLAT,TSNVEL,TSLON,TSEVEL,TSALT,TSALTR,BCTRUE,FCTRUE)
C
C
C     VERSION NO. 1.0   DATE: 24 FEBRUARY 2003
C
C     PURPOSE:   TO SERVE AS AN EXECUTIVE CONTROL OF THE REFINEMENT PROCESS.  THIS
C             CSC IS THE ONLY INTERFACE TO THE HOST_INTERFACE_PROC AND OCP_INT_PROC
C
C
C
C
C   INPUTS:
C
C  MS_IND           - Master/Slave Indicator
C  CSCI_VERSION     - Version Indicator
```

```
C
C   MASTER_MSG          - RECEIVED MASTER MESSAGE
C   MMS               - Master Message Signal indicator
C
C   TRUE_MNAV
C      TMNAVTOV       - TRUE Time of Validity of MASTER Navigation solution
C      TMLAT          - TRUE MASTER Latitude position [rad]
C      TMNVEL         - TRUE MASTER North Velocity [ft/s]
C      TMLON          - TRUE MASTER Longitude position [rad]
C      TMEVEL         - TRUE MASTER East Velocity [ft/s]
C      TMALT          - TRUE MASTER Altitude [ft]
C      TMALTR         - TRUE MASTER Altitude rate [ft/s]
C
C   EST_SNAV
C      SNAVTOV          - Time of Validity of SELF Navigation solution
C      SLAT           - ESTIMATED SELF Latitude position [rad]
C      SNVEL          - ESTIMATED SELF North Velocity [ft/s]
C      SLON           - ESTIMATED SELF Longitude position [rad]
C      SEVEL          - ESTIMATED SELF East Velocity [ft/s]
C      SALT           - ESTIMATED SELF Altitude [ft]
C      SALTR          - ESTIMATED SELF Altitude rate [ft/s]
C
C   TRUE_SNAV
C      TSNAVTOV       - TRUE Time of Validity of SELF Navigation solution
C      TSLAT          - TRUE SELF Latitude position [rad]
C      TSNVEL         - TRUE SELF North Velocity [ft/s]
C      TSLON          - TRUE SELF Longitude position [rad]
C      TSEVEL         - TRUE SELF East Velocity [ft/s]
C      TSALT          - TRUE SELF Altitude [ft]
C      TSALTR         - TRUE SELF Altitude rate [ft/s]
C      BCTRUE         - TRUE clock bias
C      FCTRUE         - TRUE frequency drift.
C
C
C   OUTPUTS:
C
C   MNAV_DATA (MASTER NAV DATA PACKAGE)
C      MNAVTOV          - Time of Validity of MASTER Navigation solution
C      MLAT           - ESTIMATED MASTER Latitude position [rad]
C      MLATR          - ESTIMATED MASTER Latitude rate [rad/s]
C      MLON           - ESTIMATED MASTER Longitude position [rad]
C      MLONR          - ESTIMATED MASTER Longitude rate [rad/s]
C      MALT           - ESTIMATED MASTER Altitude [m]
C      MALTR          - ESTIMATED MASTER Altitude rate [m/s]
C      TMLAT          - TRUE MASTER Latitude position [rad]
C      TMLATR         - TRUE MASTER Latitude rate [rad/s]
C      TMLON          - TRUE MASTER Longitude position [rad]
C      TMLONR         - TRUE MASTER Longitude rate [rad/s]
C      TMALT          - TRUE MASTER Altitude [m]
C      TMALTR         - TRUE MASTER Altitude rate [m/s]
C      MNAV_VAL_IND   - MASTER navigation validity indicator
C
C   MGPS_DATA
C      MSAT(12)       - MASTER Ordered Satellite List
C      MPRR(12)       - MASTER Ordered measured code phase array
C      MGPSTOV          - MASTER predicted GPS time of validity
C      MVIS           - Number of satellites visible to MASTER
C      INVERS(30)       - Satellite sorting index
C
C   SNAV_DATA (SELF NAV DATA PACKAGE)
C      SNAVTOV          - Time of Validity of SELF Navigation solution
C      SLAT           - ESTIMATED SELF Latitude position [rad]
C      SLATR          - ESTIMATED SELF Latitude rate [rad/s]
C      SLON           - ESTIMATED SELF Longitude position [rad]
C      SLONR          - ESTIMATED SELF Longitude  rate [rad/s]
C      SALT           - ESTIMATED SELF Altitude [m]
```

```
C      SALTR         - ESTIMATED SELF Altitude rate [m/s]
C      TSLAT         - TRUE SELF Latitude position [rad]
C      TSLATR        - TRUE SELF Latitude rate [rad/s]
C      TSLON         - TRUE SELF Longitude position [rad]
C      TSLONR        - TRUE SELF Longitude  rate [rad/s]
C      TSALT         - TRUE SELF Altitude [m]
C      TSALTR        - TRUE SELF Altitude rate [m/s]
C      SNAV_VAL_IND  - SELF navigation validity indicator
C
C
C   STORED CONSTANTS
C   AR            - 2.09257382 E7        Earth semi-minor radius [ft]
C   E22           - 6.74499984 E-3 Earth eccentricity squared
C
C
C
C
C   SUBROUTINE OCP_INT_PROC (MS_IND, MMS, MASTER_MSG, TMNAVTOV,TMLAT,TMNVEL,
C   1   TMLON,TMEVEL,TMALT,TMALTR, SNAVTOV,SLAT,SNVEL,SLON,SEVEL,SALT,SALTR,
C   2   TSNAVTOV, TSLAT,TSNVEL,TSLON,TSEVEL,TSALT,TSALTR,BCTRUE,FCTRUE)
C
C
C
C
C   VERSION 1 - NO MASTER MESSAGE
C   -------------
C   INPUTS: MS_IND, TMNAVTOV,TMLAT,TMNVEL,TMLON,TMEVEL,
C      TMALT,TMALTR, SNAVTOV,SLAT,SNVEL,SLON,SEVEL,SALT,SALTR,
C      TSNAVTOV, TSLAT,TSNVEL,TSLON,TSEVEL,TSALT,TSALTR,BCTRUE,FCTRUE
C
C
C
C   LOCAL VARIABLES:
C
C
C
C   GLOBAL VARIABLES:
C
C
C
C   CALLED BY:
C   RF_EXEC_CTRL
C
C
C   SUBROUTINES:
C   PROCESS_MASTER_MESSAGE
C   PROCESS_RAW_GPS_DATA
C   CONVERT_NAV_DATA
C
C
C
C   MODICATION HISTORY:
C      V1.0   ORIGINAL VERSION DATE      RESPONSIBLE
C
C             FOR VERSION 1.0     4/03/03       JYH
C
C
C----------------------------------------------------------------------

      SUBROUTINE OCP_INT_PROC (CSCI_VERS_IND, MS_IND, MNAVTOV, MLAT, MNVEL, MLON,
      1   MEVEL, MALT, MALTR, TMNAVTOV, TMLAT, TMNVEL, TMLON, TMEVEL, TMALT, TMALTR,
        2 SNAVTOV, SLAT, SNVEL, SLON, SEVEL, SALT, SALTR, TSNAVTOV, TSLAT, TSNVEL,
      3   TSLON, TSEVEL, TSALT, TSALTR, BCTRUE, FCTRUE)
```

```
C*****************************************************************
C
C       DECLARATION OF VARIABLES
C
C*****************************************************************

      IMPLICIT NONE

      COMMON BLCK

      REAL*8 NC_RATE

      INTEGER*4 CSCI_VERS_IND, BLCK, GPS_HEALTH_IND, MS_IND, LF_IND, MMS
      INTEGER*4 SNAV_VAL_IND, MNAV_VAL_IND, MVIS

      REAL*8     MNAVTOV, MLAT, MNVEL, MLON, MEVEL, MALT, MLATR, MLONR, MALTR, MGPSTOV
      REAL*8     TMNAVTOV,TMLAT, TMNVEL,TMLON,TMEVEL,TMALT,TMLATR, TMLONR,TMALTR
      REAL*8     SNAVTOV, SLAT, SNVEL, SLON, SEVEL, SALT, SLATR, SLONR, SALTR, SGPSTOV
      REAL*8     TSNAVTOV,TSLAT, TSNVEL, TSLON, TSEVEL, TSALT, TSLATR, TSLONR, TSALTR
      REAL*8     UMSAT(BLCK),UMPRR(BLCK),MSAT(BLCK), MPRR(BLCK), INVERS(30)

      REAL*8     BCTRUE, FCTRUE


C-----------------------------------------------------------------------
C
C   A CONSTRUCT NEEDS TO BE CREATED FOR THE MASTER MESSAGE
C
C-----------------------------------------------------------------------
      REAL*8 MASTER_MSG

C*****************************************************************

C       BEGIN
C
C*****************************************************************

      IF (CSCI_VERS_IND.EQ.1) THEN
         GOTO 100

      ELSEIF (MS_IND.EQ.1) THEN
         RETURN
      ELSEIF (MMS.EQ.0) THEN
         RETURN

      ELSE
         CALL PROCESS_MASTER_MSG (MASTER_MSG, MNAVTOV, MLAT, MNVEL, MLON, MEVEL, MALT,MALTR,
     1      GPS_HEALTH_IND,MGPSTOV,BLCK, UMSAT, UMPRR, TMNAVTOV, TMLAT, TMNVEL,TMLON,TMEVEL,TMALT,TMALTR)

         CALL PROCESS_RAW_GPS_DATA (BLCK,UMSAT,UMPRR,MSAT,MPRR,INVERS, MVIS)
      ENDIF

      IF (CSCI_VERS_IND.EQ.2) THEN
         GOTO 100
      ENDIF

      CALL CONVERT_NAV_DATA(MLAT, MNVEL, MEVEL, MALT, MALTR, MLATR, MLONR)

      CALL CONVERT_NAV_DATA(SLAT, SNVEL, SEVEL, SALT, SALTR, SLATR, SLONR)

      RETURN

  100 CONTINUE

      CALL CONVERT_NAV_DATA(MLAT, MNVEL, MEVEL, MALT, MALTR, MLATR, MLONR)
```

```
      CALL CONVERT_NAV_DATA(SLAT, SNVEL, SEVEL, SALT, SALTR, SLATR, SLONR)

      CALL CONVERT_NAV_DATA(TMLAT, TMNVEL, TMEVEL, TMALT, TMALTR, TMLATR, TMLONR)
      CALL CONVERT_NAV_DATA(TSLAT, TSNVEL, TSEVEL, TSALT, TSALTR, TSLATR, TSLONR)


      RETURN
      END


C#########################################################################
C     SUBROUTINE PROCESS_MASTER_MSG(MNAVSOLN, MNAVTOV,MLAT,MNVEL,MLON,MEVEL,MALT,MALTR,
C            GPS_HEALTH_IND, MGPSTOV, BLCK, UMSAT(BLCK),UMPRR(BLCK),TMNAVTOV,TMLAT,
C            TMNVEL,TMLON,TMEVEL,TMALT,TMALTR)
C
C     PURPOSE:  DECOMPOSES THE RECEIVED MASTER MESSAGE INTO MASTER NAV DATA AND GPS DATA.
C
C
C     INPUTS:
C        MASTER_MSG
C
C
C     OUTPUTS:
C     MNAVTOV           - Master Message Time of Validity
C     MLAT          - ESTIMATED MASTER Latitude position [rad]
C     MNVEL         - ESTIMATED MASTER North Velocity [ft/s]
C     MLON          - ESTIMATED MASTER Longitude position [rad]
C     MEVEL         - ESTIMATED MASTER East Velocity [ft/s]
C     MALT          - ESTIMATED MASTER Altitude [ft]
C     MALTR         - ESTIMATED MASTER Altitude rate [ft/s]
C
C     GPS_HEALTH_IND    - Indicates the health status of GPS signal. (Binary)
C     MGPSTOV           - Time of Validity of MASTER GPS solution
C     BLCK          - Number of PR reports, size of array
C     UMSAT(BLCK)       - Unordered array of MASTER visible satellite
C     UMPRR(BLCK)       - Unordered array of MASTER code phase
C
C
C#########################################################################

      SUBROUTINE PROCESS_MASTER_MSG(MASTER_MSG, MNAVTOV,MLAT,MNVEL,MLON,MEVEL,MALT,MALTR,
     1   GPS_HEALTH_IND, MGPSTOV, BLCK, UMSAT,UMPRR,TMNAVTOV,TMLAT,
     2   TMNVEL,TMLON,TMEVEL,TMALT,TMALTR)


      INTEGER*4 BLCK, GPS_HEALTH_IND
      REAL*8    MNAVSOLN, MNAVTOV, MLAT,MNVEL,MLON,MEVEL,MALT,MALTR, MGPSTOV
      REAL*8    UMSAT(12),UMPRR(12),TMNAVTOV,TMLAT, TMNVEL,TMLON,TMEVEL,TMALT,TMALTR

      REAL*8    MASTER_MSG

      RETURN
      END

C#########################################################################
C     SUBROUTINE PROCESS_RAW_GPS_DATA (BLCK, UMSAT(BLCK),UMPRR(BLCK),MSAT(BLCK),MPRR(BLCK),
C        INVERS(30))
C
C     PURPOSE:  PROVIDES ORDERED ARRAYS OF INPUTED SATELLLITE ID'S AND CODE PHASE DATA
C
C     INPUTS:
C     BLCK          - Number of PR reports, size of array
C     UMSAT(BLCK)       - Unordered array of MASTER visible satellite
C     UMPRR(BLCK)       - Unordered array of MASTER code phase
C
C
```

```
C   OUTPUTS:
C   MSAT(BLCK)        - Ordered array of MASTER visible satellite
C   MPRR(BLCK)        - Ordered array of MASTER code phase
C
C
C##############################################################################


    SUBROUTINE PROCESS_RAW_GPS_DATA (BLCK, UMSAT, UMPRR, MSAT, MPRR, INVERS, MVIS)


    INTEGER*4    I,J, ID, BLCK, MVIS
    REAL*8     PRR, UMSAT(BLCK),UMPRR(BLCK),MSAT(BLCK),MPRR(BLCK), INVERS(30)

C*****************************************************************
C
C         SORT SATELLITE ID AND CODE PHASE ARRAYS
C
C*****************************************************************

    DO 210 J = 1, BLCK
        ID = UMSAT(J);
        PRR = UMPRR(J)

        DO 211 I = J-1, 0, -1
            IF (UMSAT(I).GT.ID) THEN
                GOTO 211
            ELSE
                UMSAT(I+1) = UMSAT(I)
                UMPRR(I+1) = UMPRR(I)
            ENDIF
211 CONTINUE

        UMSAT(I+1) = ID
        UMPRR(I+1) = PRR

210 CONTINUE

    MSAT(BLCK) = UMSAT(BLCK)
    MPRR(BLCK) = UMPRR(BLCK)

C*****************************************************************
C
C         GENERATE INVERS ARRAY
C
C*****************************************************************

    DO 220 J = 1,12
        ID = MSAT(J)
        INVERS(ID) = J
220 CONTINUE

    MVIS = BLCK

    RETURN
    END

C##############################################################################
C   SUBROUTINE CONVERT_NAV_DATA(NVEL, EVEL, ALT, ALTR, LATR, LONR)
C
C   PURPOSE:  CONVERTS NAV DATA FROM FEET TO METERS TO BE CONSISTENT WITH THE UNITS OF
C             GPS MEASUREMENTS.  GENERATES LATITUDE AND LONGITUDE RATES.
C
C   INPUTS:
C      NVEL          - North Velocity [ft/s]
C      EVEL          - East Velocity [ft/s]
```

```
C      ALT           - Altitude [ft]
C      ALTR          - Altitude rate [ft/s]
C
C   OUTPUTS:
C      LATR          - Latitude rate [rad/s]
C      LONR          - Longitude rate [rad/s]
C      ALT           - Altitude [m]
C      ALTR          - Altitude rate [m/s]
C
C   STORED CONSTANTS
C   ar            - 2.09257382 E7        Earth semi-minor radius [ft]
C   RAD           -
C
C###########################################################################
      SUBROUTINE CONVERT_NAV_DATA(LAT, NVEL, EVEL, ALT, ALTR, LATR, LONR)

      IMPLICIT NONE

      REAL*8 LAT, NVEL, EVEL, ALT, ALTR, LATR, LONR, AR, RAD, FT_M

      DATA AR/2.09257382E7 /
      DATA RAD/57.2957795131/
      DATA FT_M/3.28083991667/

      LATR = NVEL*RAD / AR
      LONR = (EVEL*RAD)/(AR*COS(LAT))
C     ALT = ALT/FT_M
C     ALTR = ALTR/FT_M

      WRITE(44,*)LATR,LONR,ALT,ALTR
      WRITE(44,*)"------"

      RETURN
      END
```

Appendix C:
Link-16 Navigation Simulator (LNS)

The Test Bed 2 studies performed as part of this project relied upon the LNS to provide a realistic representation of the error dynamics of a mobile Link-16 navigation community. The LNS has been under development at BAE SYSTEMS, in steadily more sophisticated versions for the past thirty years. Since the initiation of the JTIDS Class 2 terminal Full Scale Development in January 1981, and for every Link-16 terminal development thereafter, the LNS has been maintained as the reference for the Link-16 operational navigation and synchronization algorithms. By design and convention, the algorithms mechanized within the LNS exactly replicate the navigation algorithms implemented in each Link-16 terminal. The result is a "gold standard" simulation, test and analysis tool which has proven to be an invaluable asset for algorithm development, testing, and troubleshooting of operational navigation problems experienced after terminal delivery. The accuracy and reliability of this remarkable software package have been proven over literally thousands of flight hours in every airborne platform installation of Link-16. Flight test data has been analyzed for, among others, the F-14, F-15, F/A-18, Aegis cruiser navigation and the UK Tornado ADV. The LNS allows the user to analyze individual platform performance as well as community performance. (Figure B-1). The LNS remains a proprietary asset of BAE SYSTEMS.

The use of the LNS as an integral part of the Test Bed 2 simulation package provides assurance that the navigation errors presented to the DGPS RF algorithm for refinement are truly representative of actual flight conditions. The LNS has the following features and capabilities (Figure B – 2) .

- Variable community size (up to 96 JTIDS Units – Jus) in real time on VAX Station 4000 or PC environments.
- Multiple independent trajectories for each JU
- Multiple independent navigation models for each JU, including strapdown INS, gimbaled INS, GPS/INS, ADC/AHRS, Doppler/AHRS
- Multiple independent synchronization models for each JU
- Communication and transmission of RTTs and PPLIs for each JU
- GPS navigation model for each JU
- Independent Link-16 Navigation Kalman Filters for each JU, exactly as mechanized in operational software.
- Other operational capabilities, including Sensor Registration, communication jamming models, and sensor measurement and alignment models.

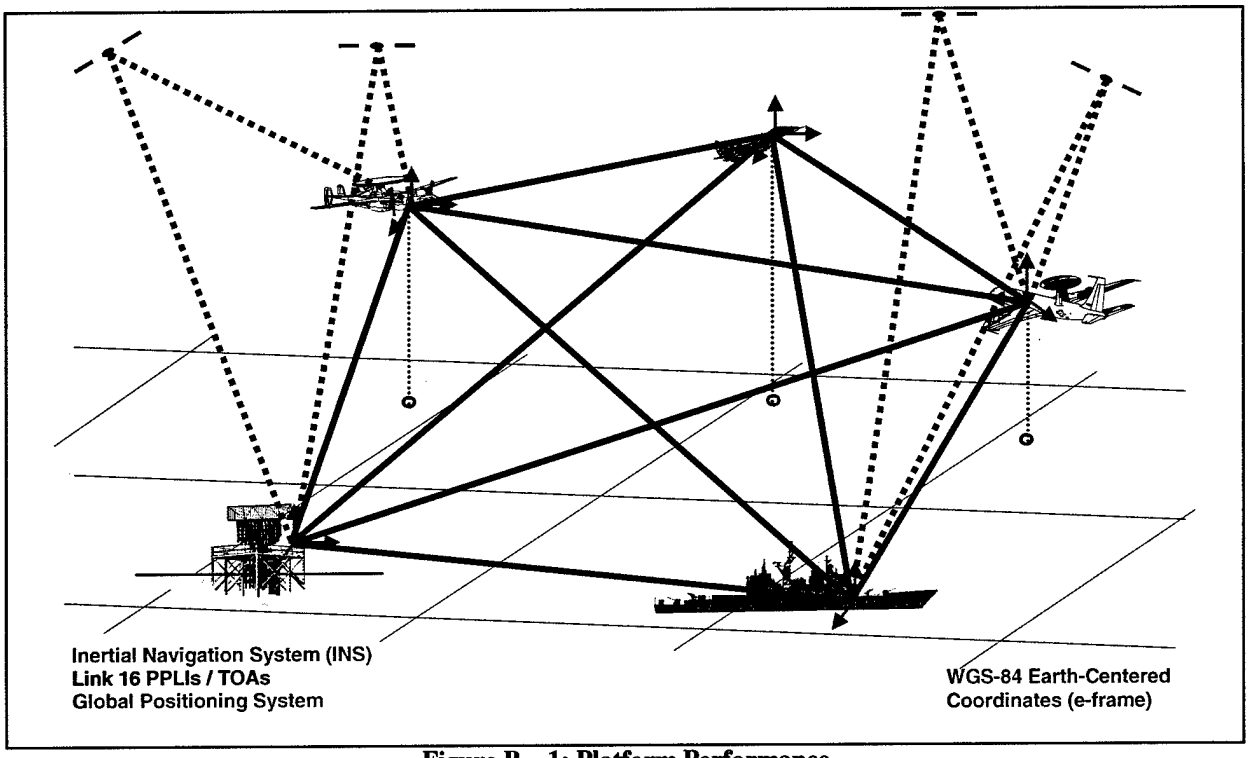
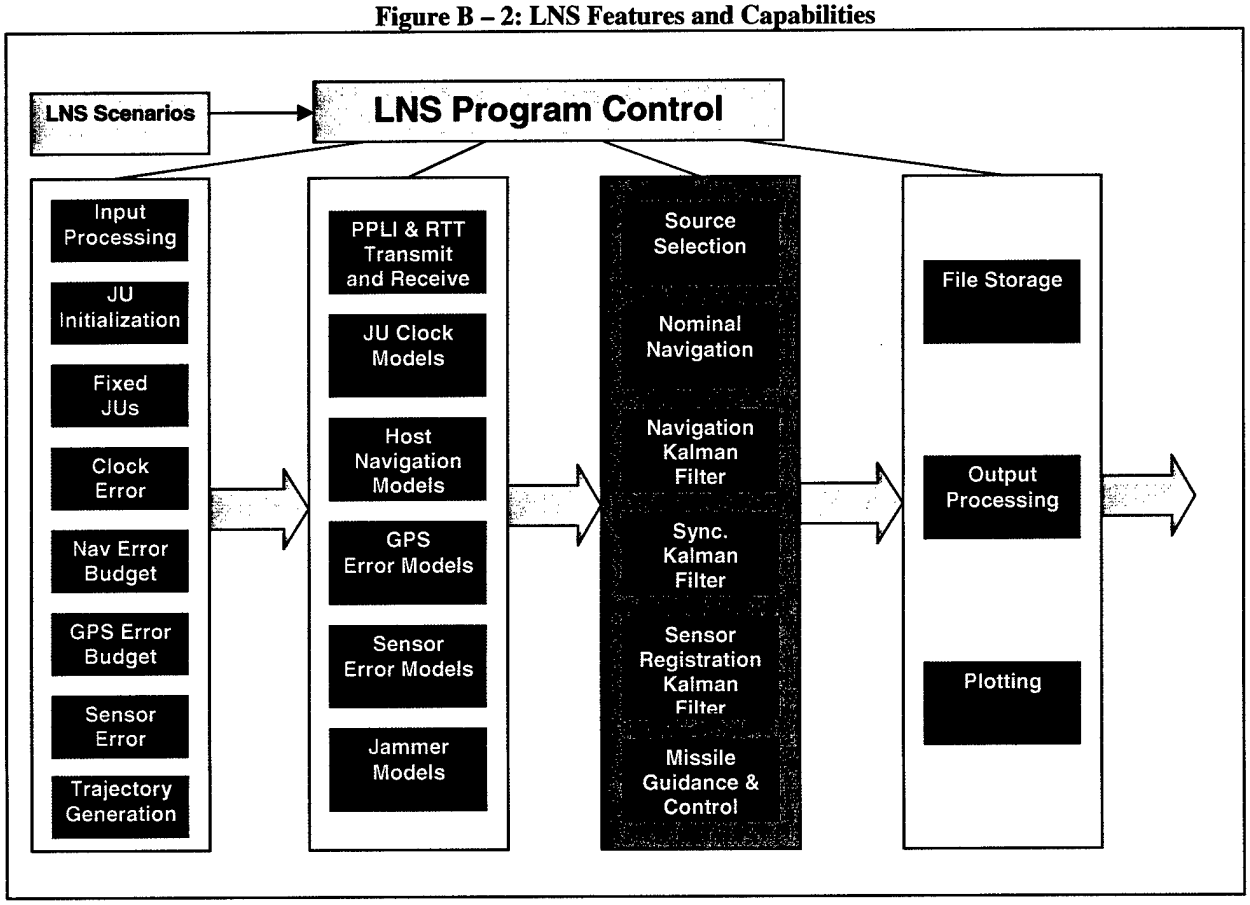


Figure B – 1: Platform Performance

Figure B – 2: LNS Features and Capabilities



As an example of the inherent capabilities of the LNS, we present a complex simulation involving an AEGIS cruiser operating with two airborne members which provide PPLI messages to the ship. The ship is also processing GPS PVT data with an observation update rate of once per 60 seconds. The error budget of the INS on the ship is representative of the WSN-7 INS installed aboard AEGIS-class units. (Figures B-3a through B-3e)

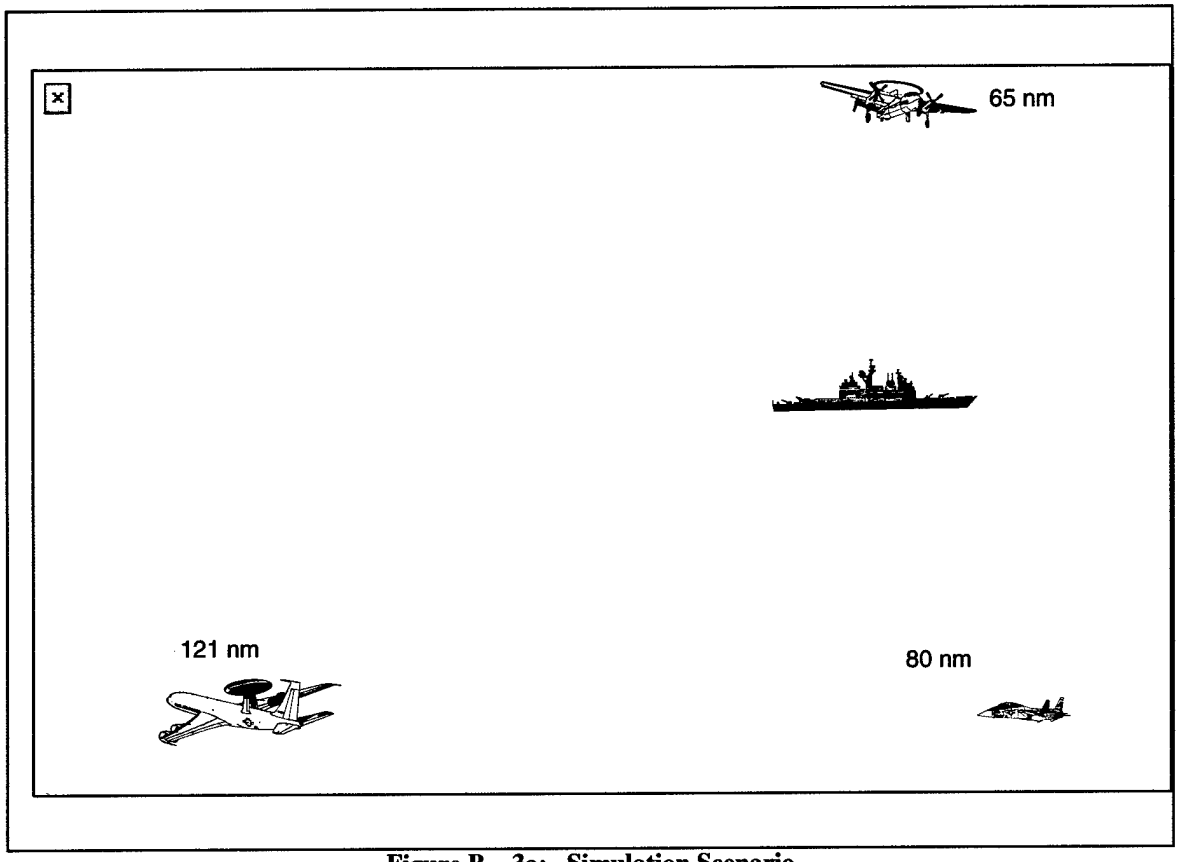


Figure B – 3a: Simulation Scenario

- **INS:**
  - Accelerometer Bias = 25 μg
  - Gyro Drift Bias = .0025°/hr
  - Alignment not modeled
  - Initial Position Error = 0
  - Initial Velocity Error = 1 fps (x and y)
  - Initial Azimuth Error = 0.25°
- **GPS:**
  - Available every 60 sec
  - No error
- **JTIDS:**
  - PPLIs available from each source every 12 seconds
  - No errors in TOA/PPLIs; QP=14
  - Latency modeled as a constant time error ~ 1 second

Figure B-3b: Error Budgets

Case 1A:
Autonomous GPS/INS Navigation Performance
(with accurate GPS)

GPS/INS Kalman Filter:
- GPS Measurement Noise = 0.1 nm (~600 ft)
- GPS available continuously @ 60 sec rate

Figure B-3c

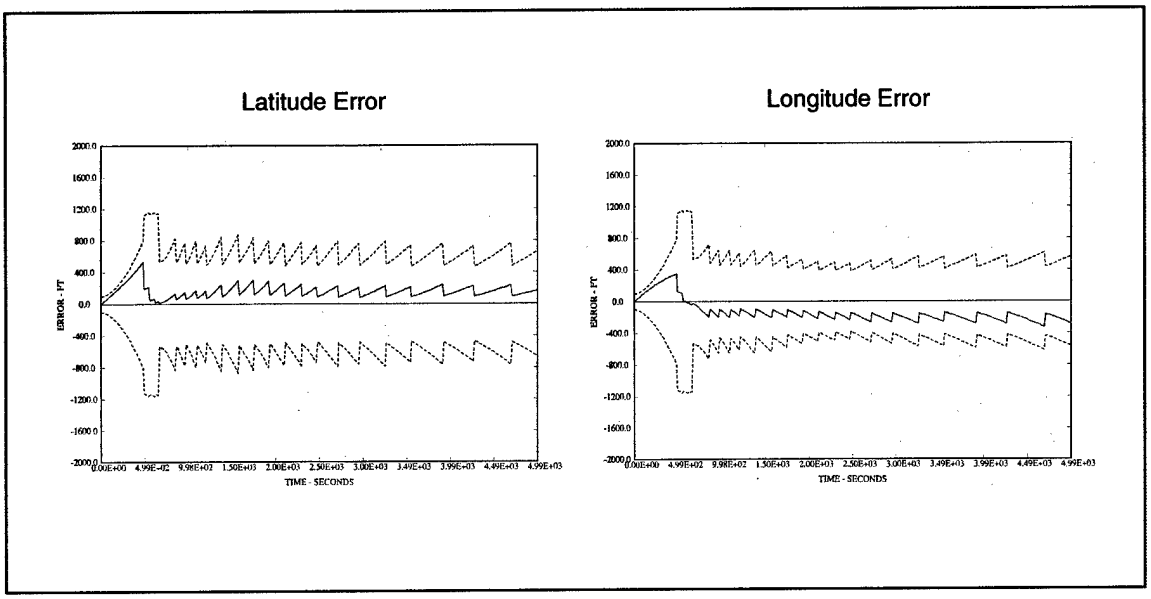


Figure B-3d: Position Errors

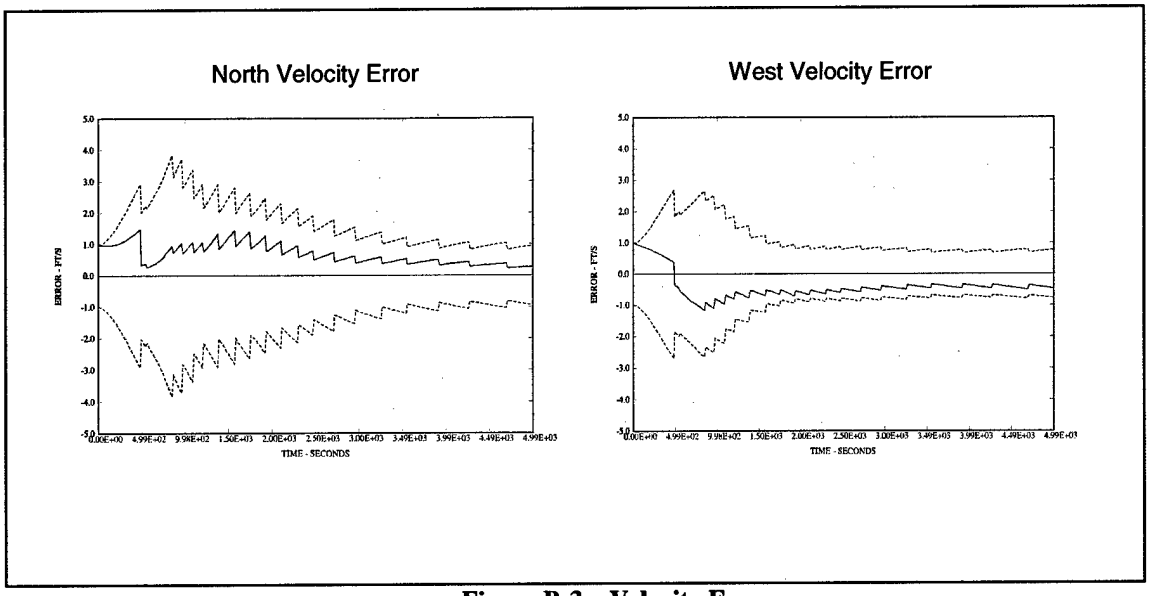


Figure B-3e: Velocity Errors

Appendix D:
Operational Software Flow Diagrams

I. RF_EXEC_CTRL

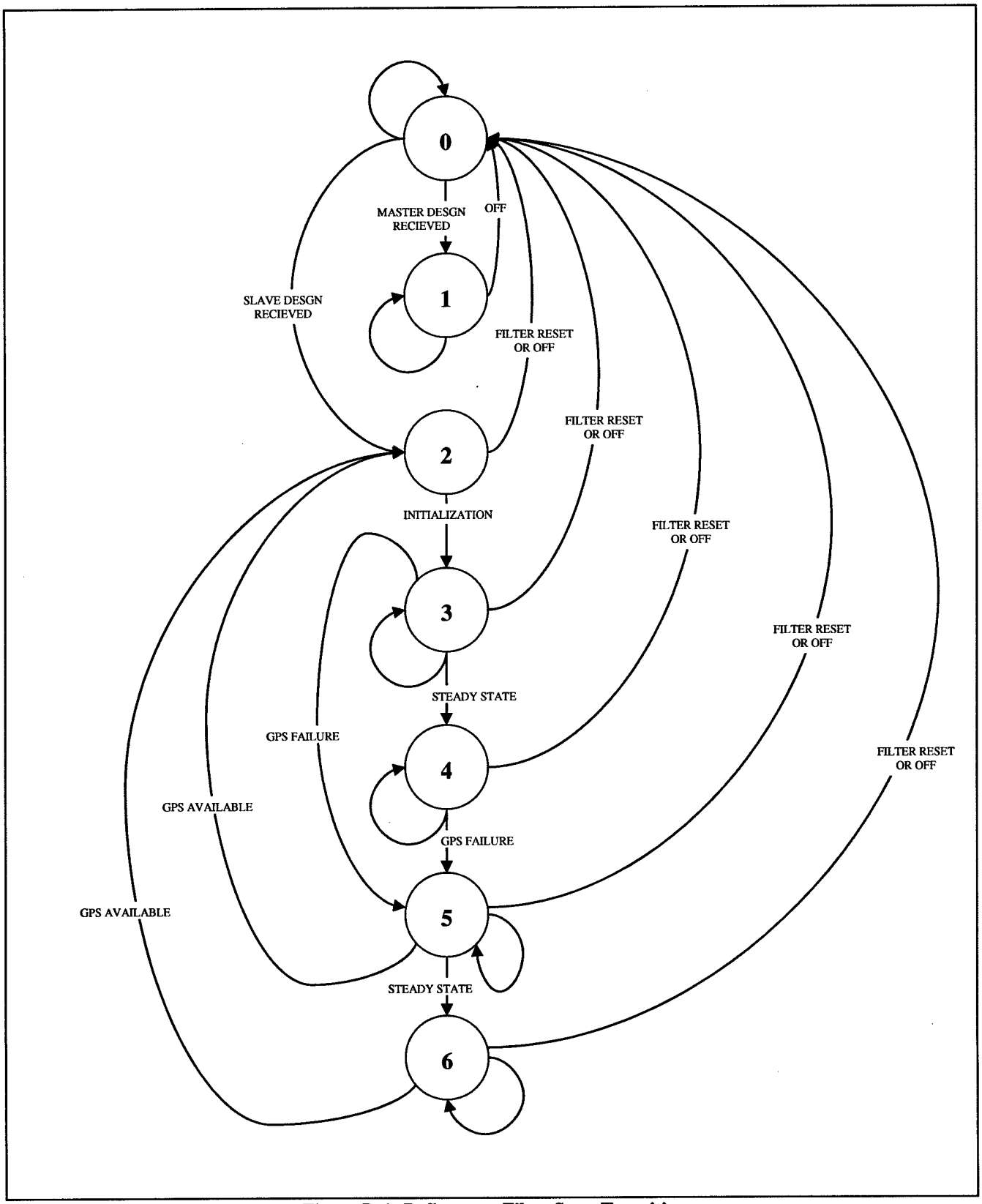


Figure D-1: Refinement Filter State Transitions

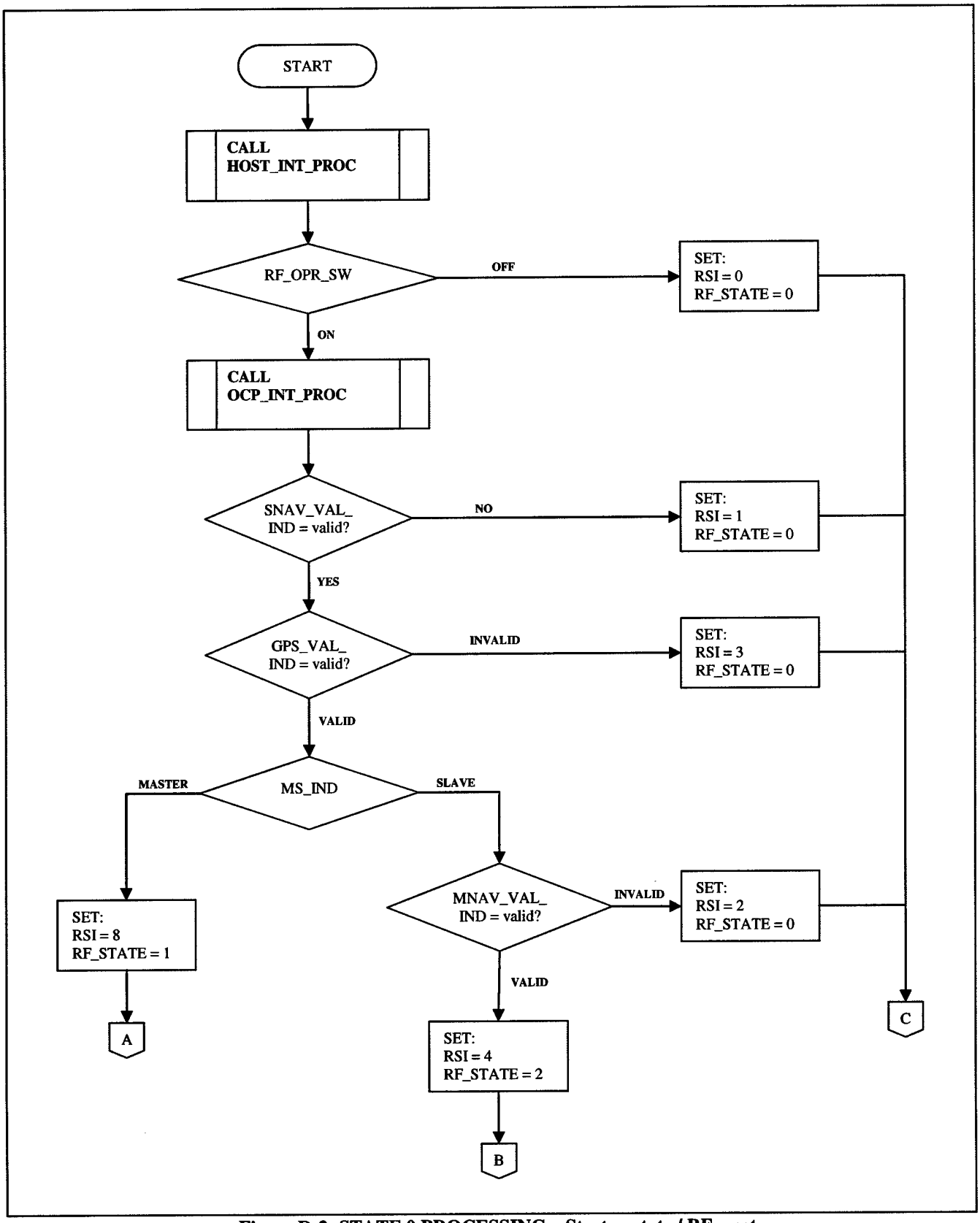


Figure D-2: STATE 0 PROCESSING – Startup state / RF reset

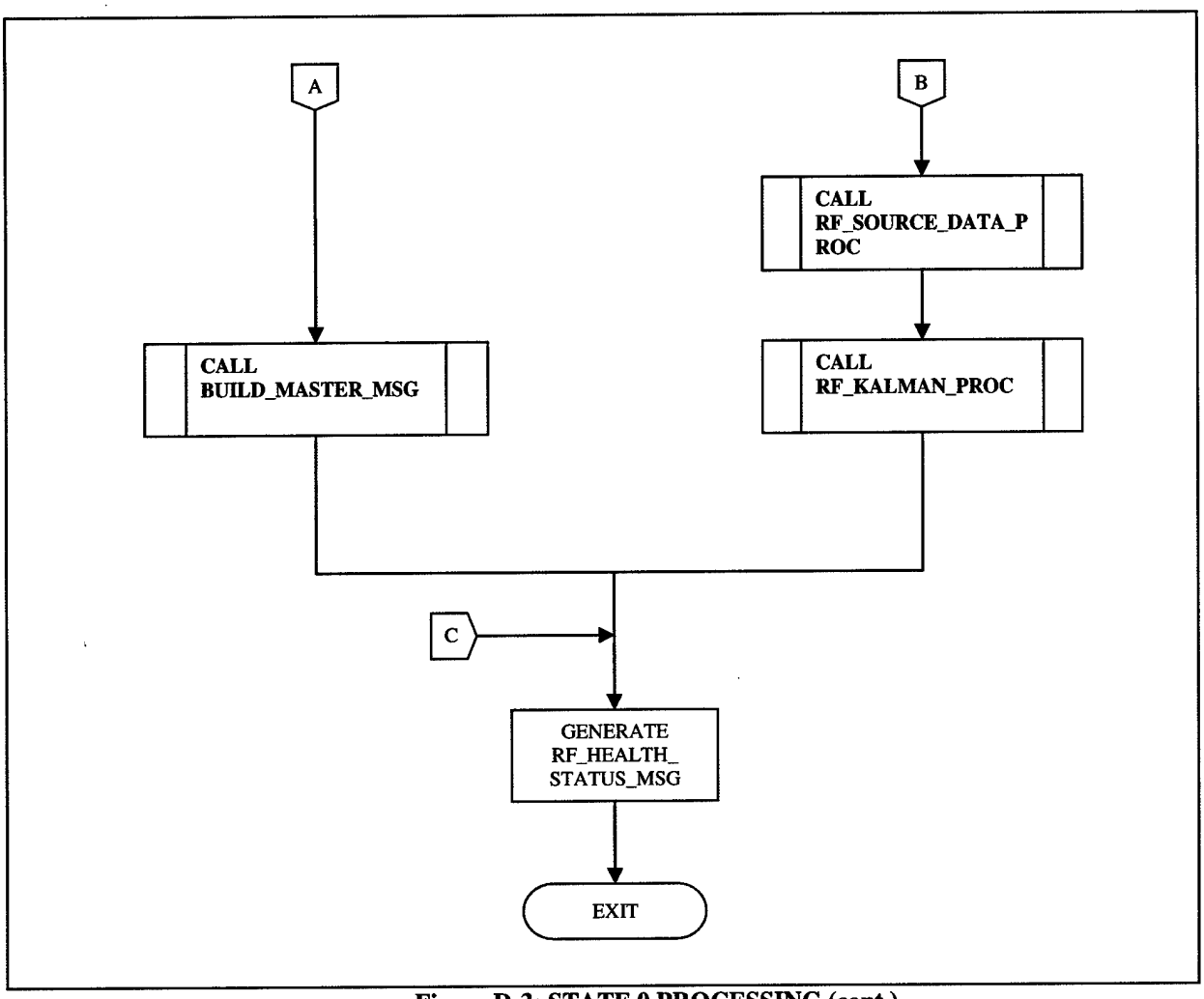


Figure D-3: STATE 0 PROCESSING (cont.)

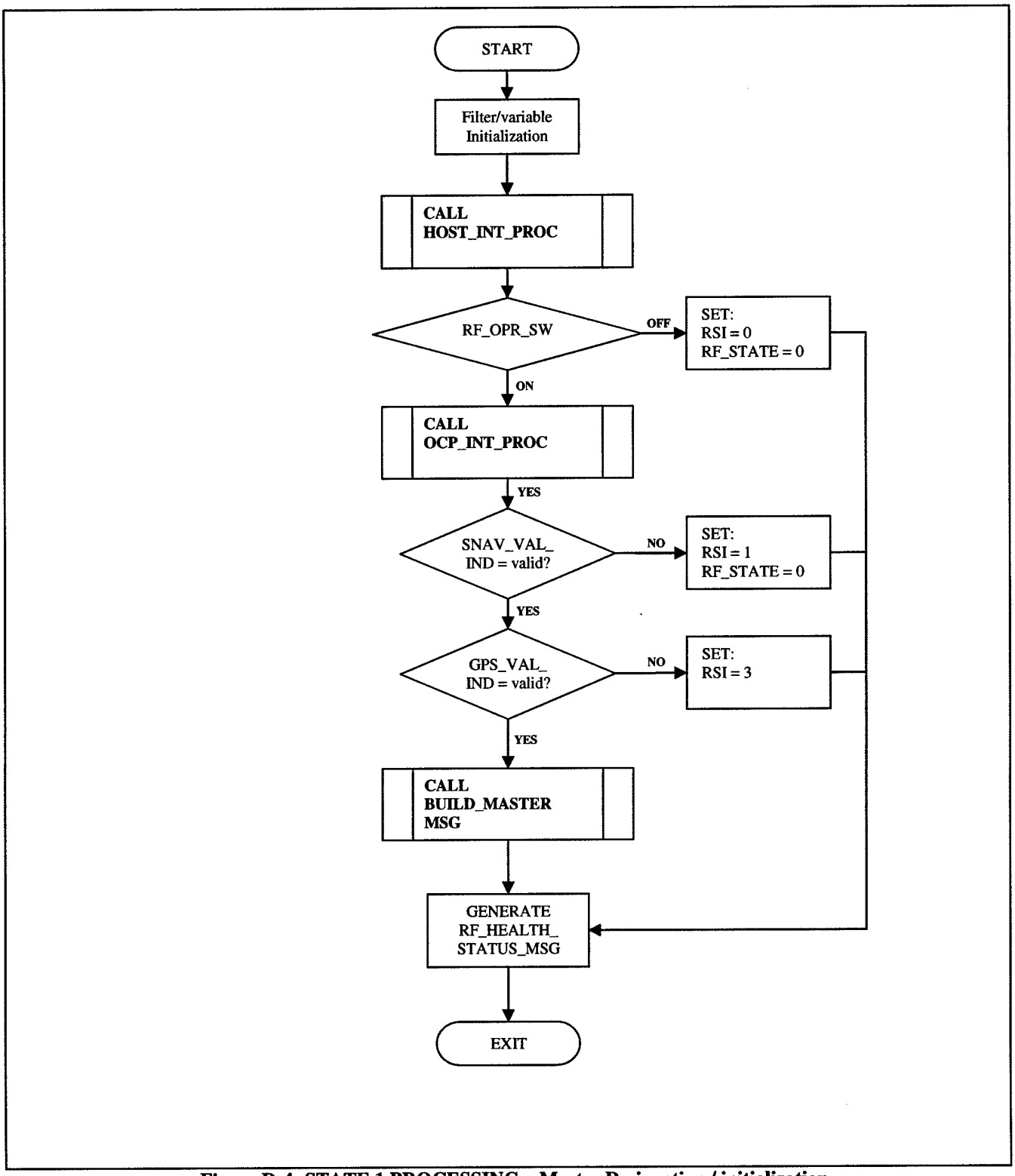


Figure D-4: STATE 1 PROCESSING – Master Designation / initialization

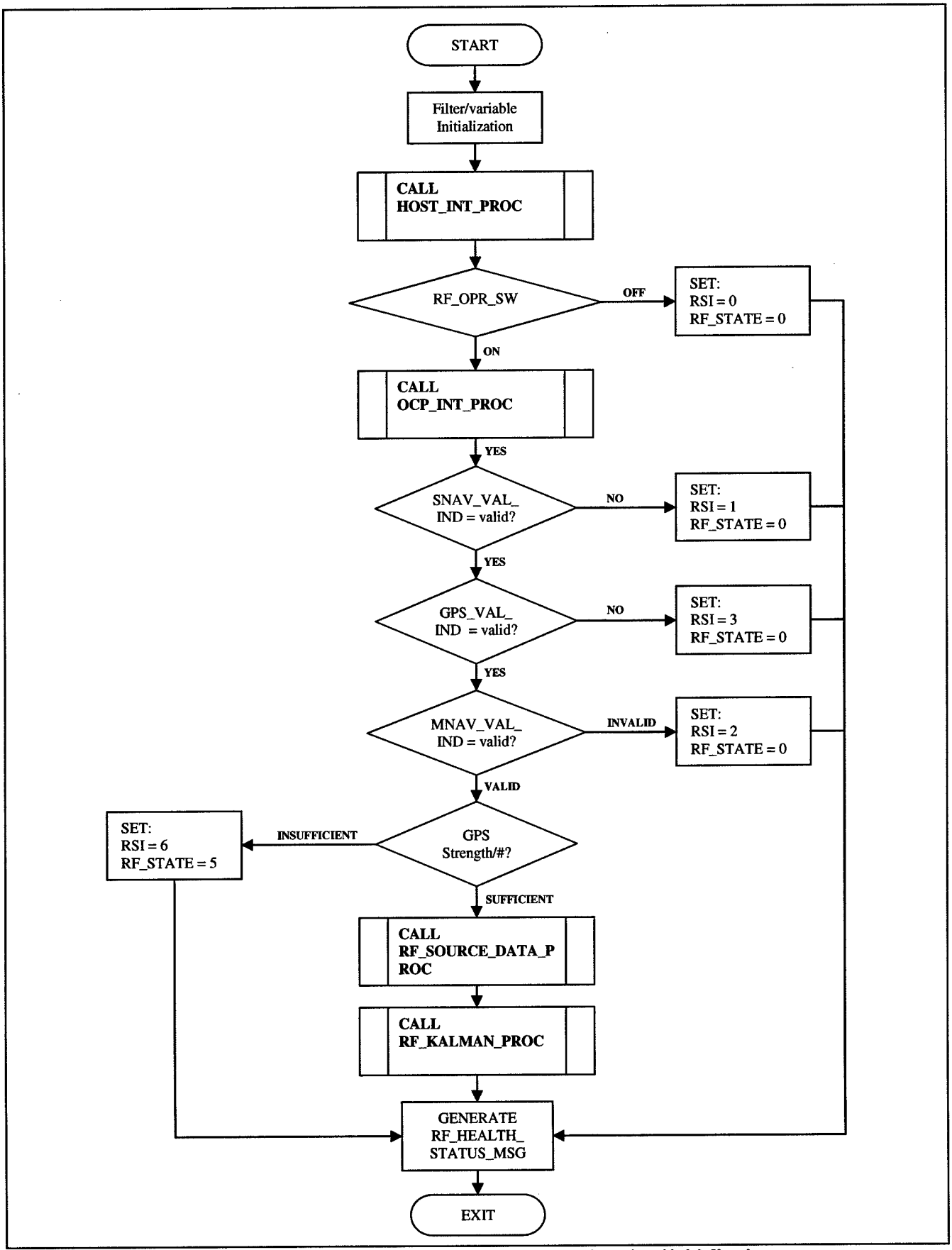


Figure D-5: STATE 2 PROCESSING – Slave Designation / initialization

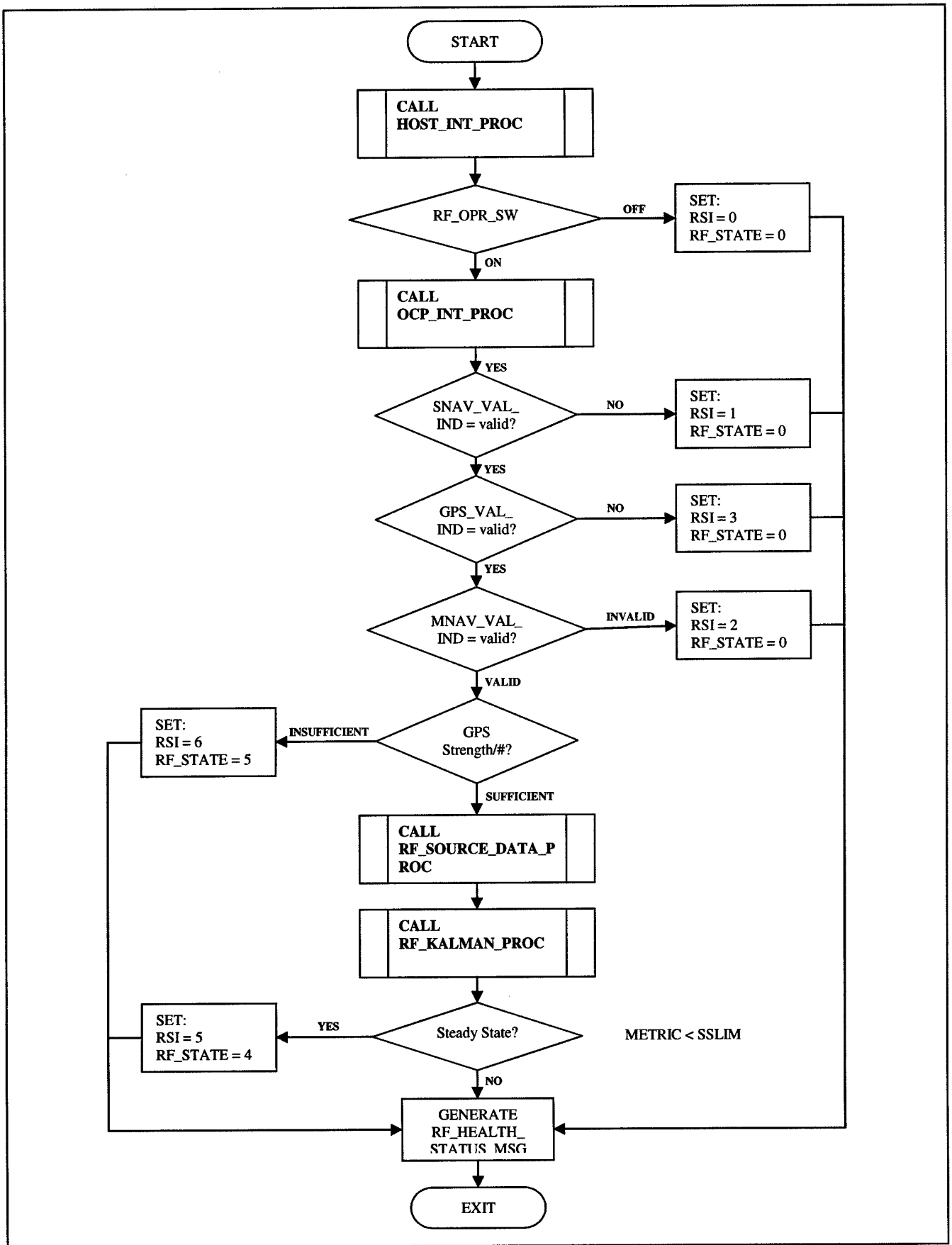


Figure D-6: STATE 3 PROCESSING - Slave Designation / GPS Pseudorange [not steady state]

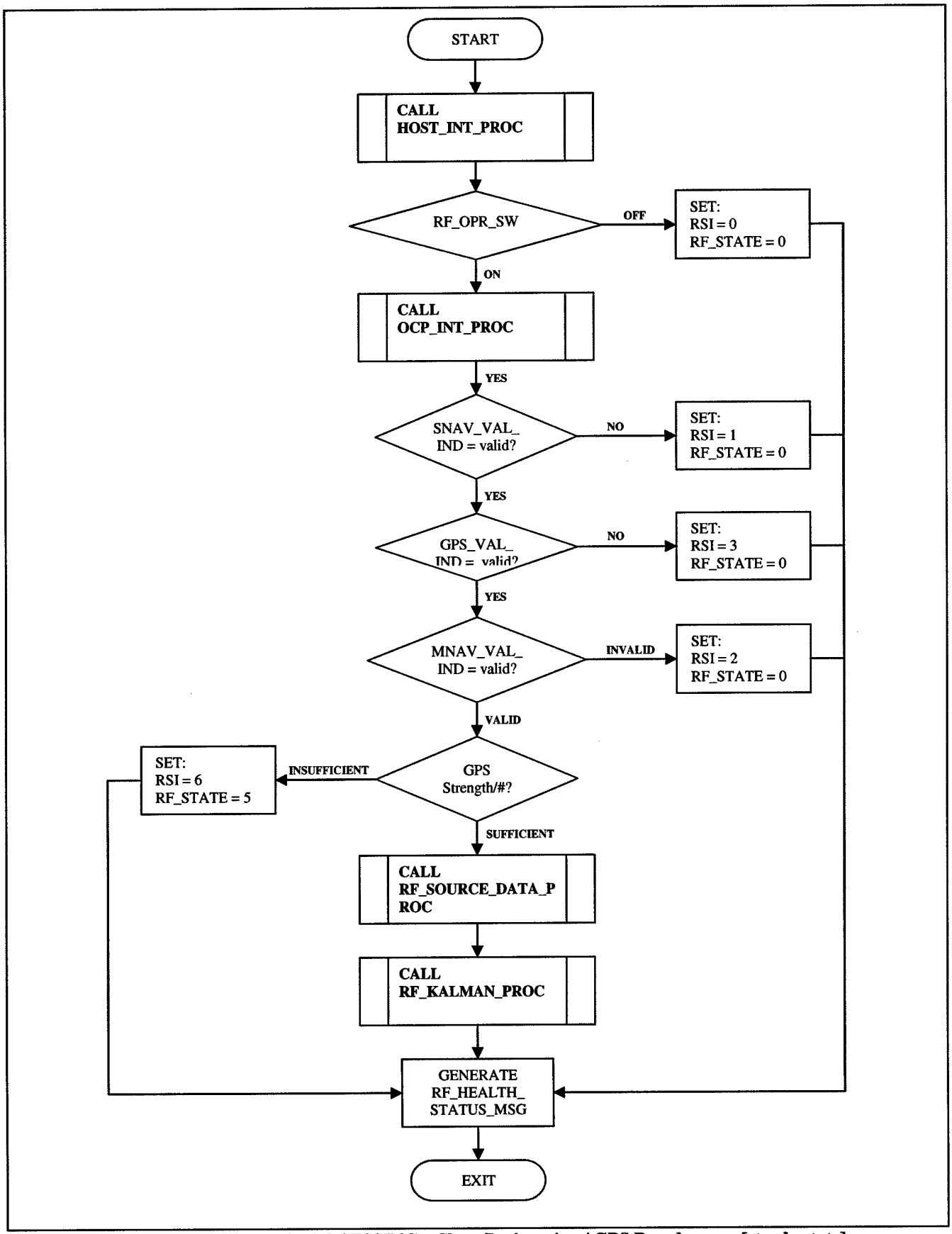


Figure D-7: STATE 4 PROCESSING – Slave Designation / GPS Pseudorange [steady state]

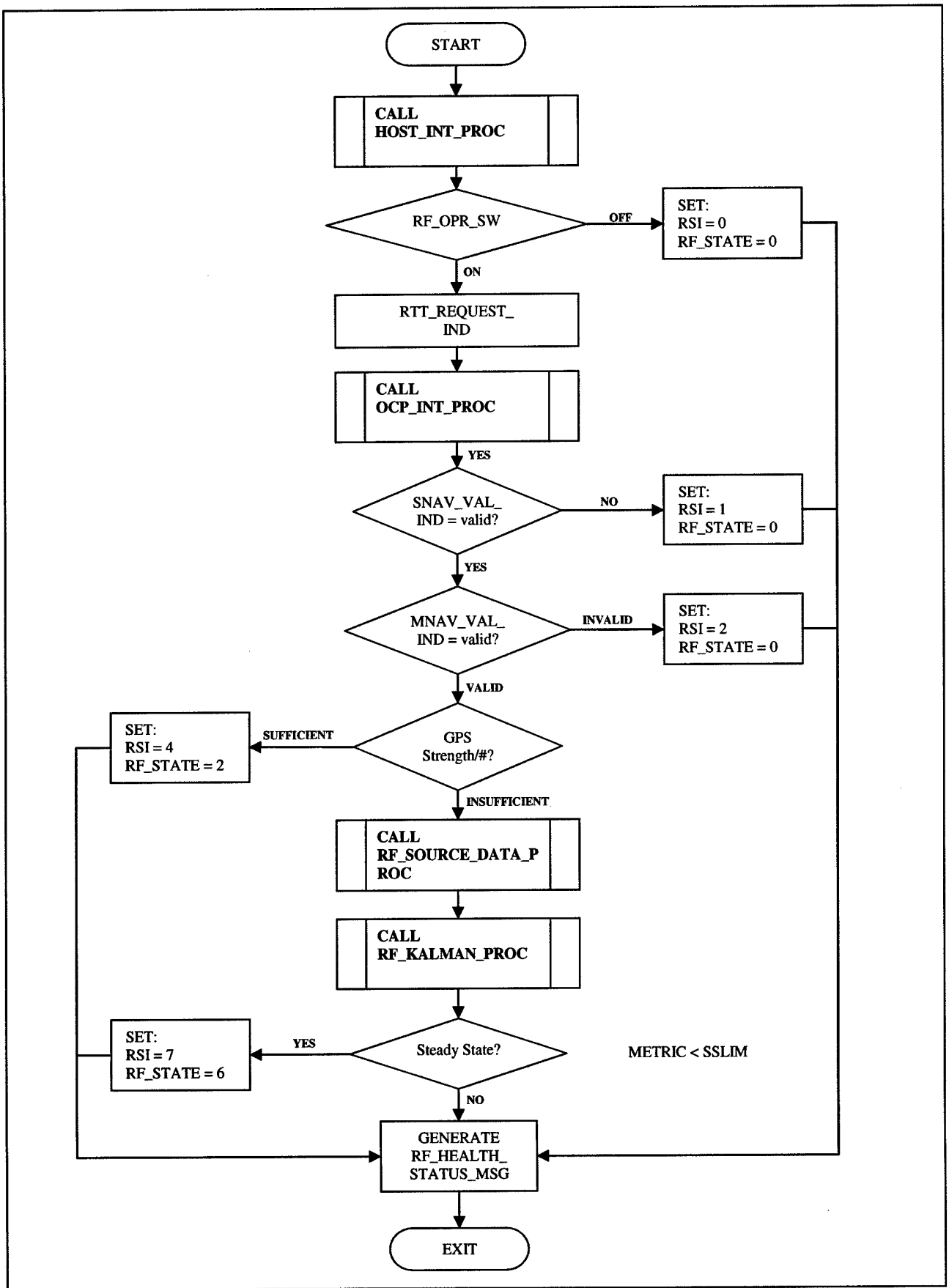


Figure D-8: STATE 5 PROCESSING – Slave Designation / RTT, PPLI [backup mode initialization]

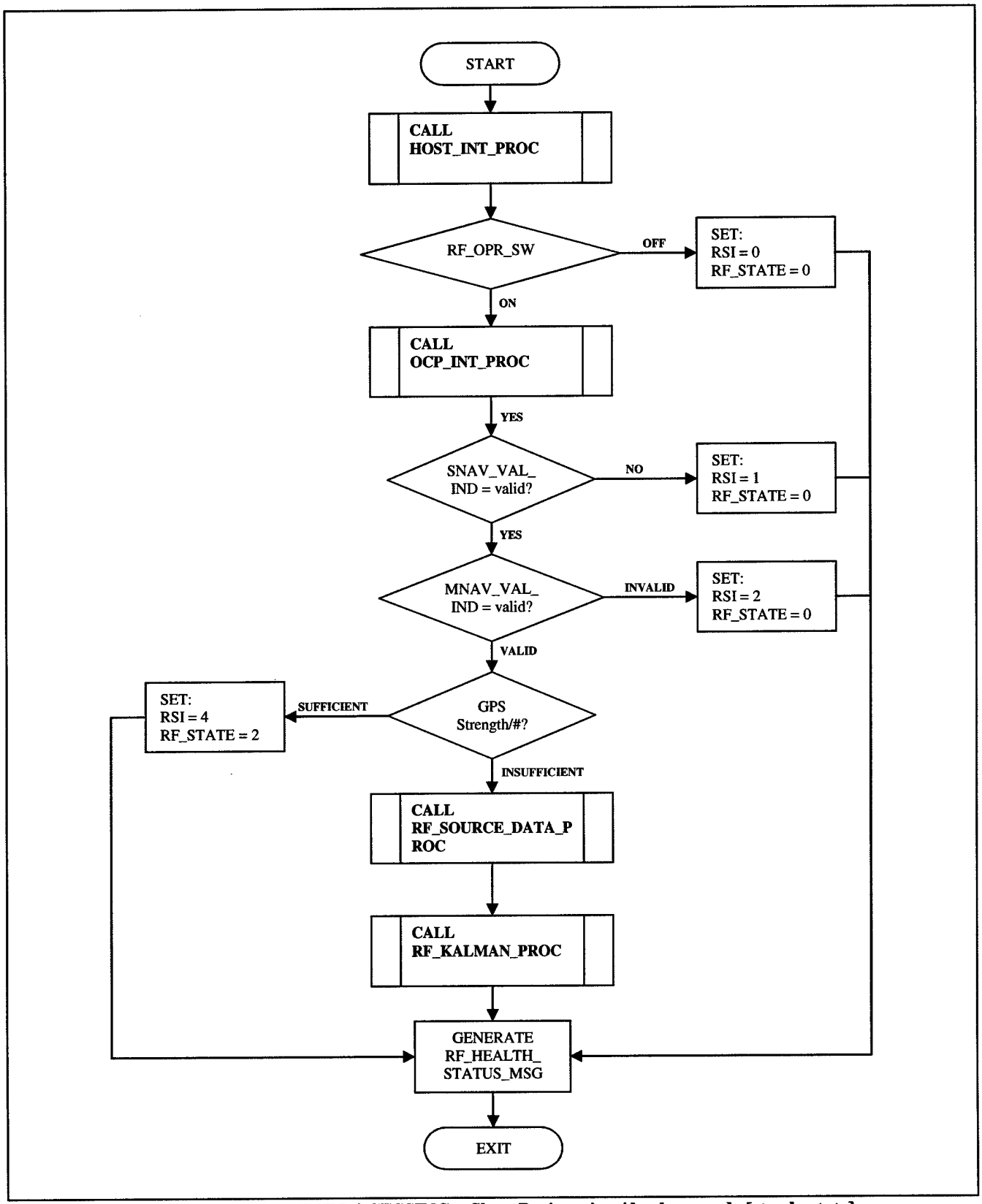


Figure D-9: STATE 6 PROCESSING – Slave Designation / backup mode [steady state]

II. RF_SOURCE_DATA_EXEC

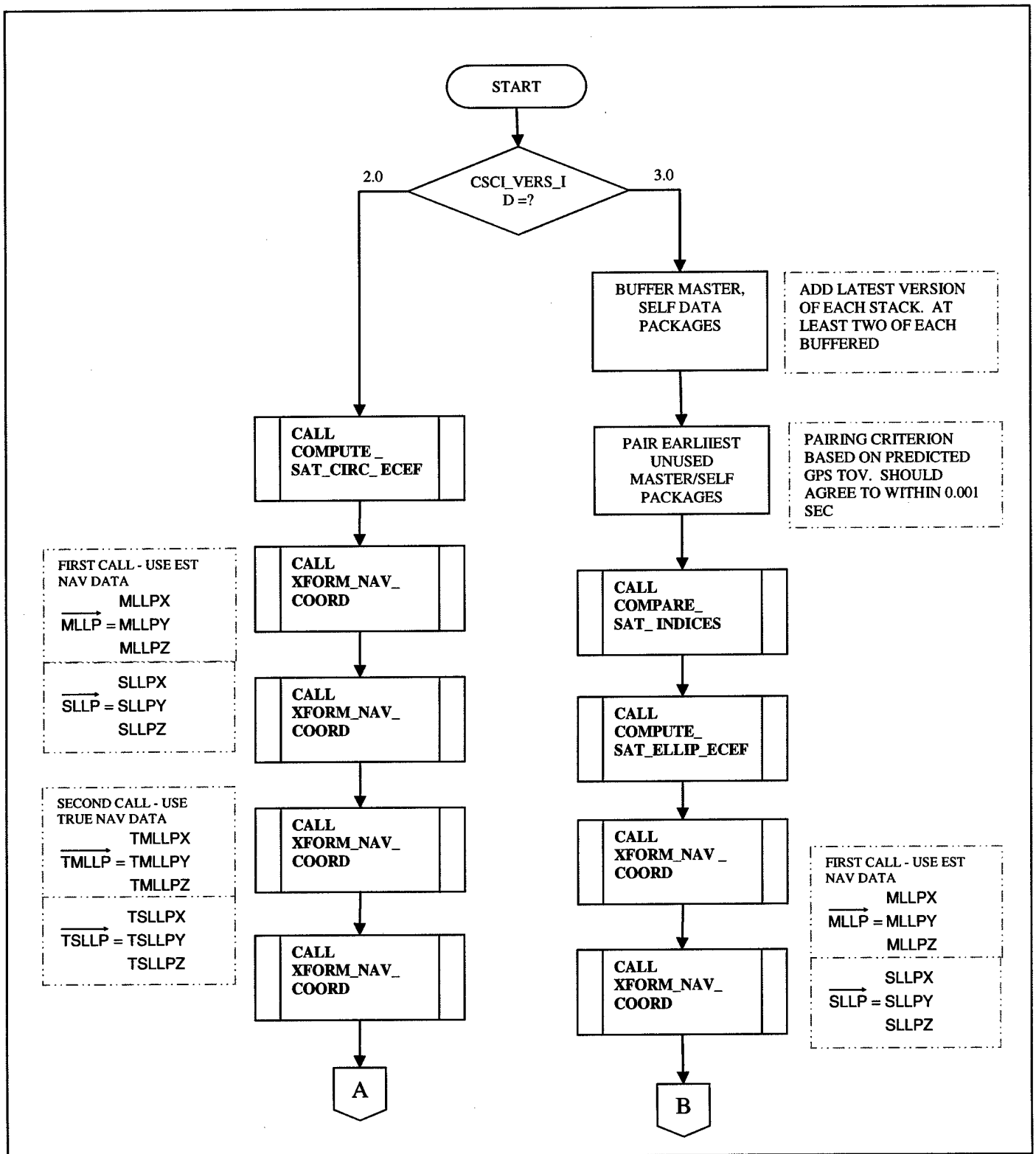


Figure D-10: RF_SOURCE_DATA_EXEC Logical data flow diagram

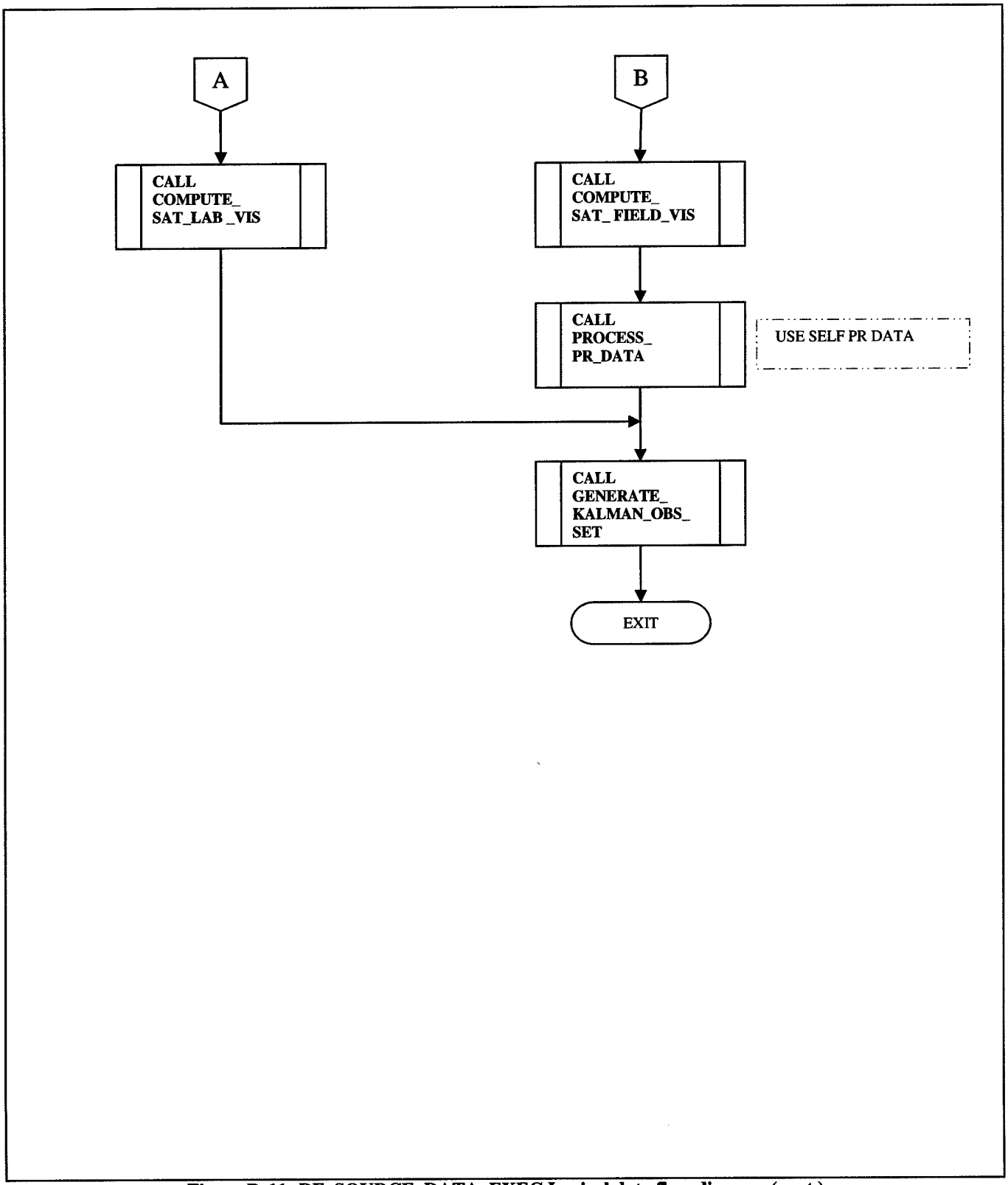


Figure D-11: RF_SOURCE_DATA_EXEC Logical data flow diagram (cont.)

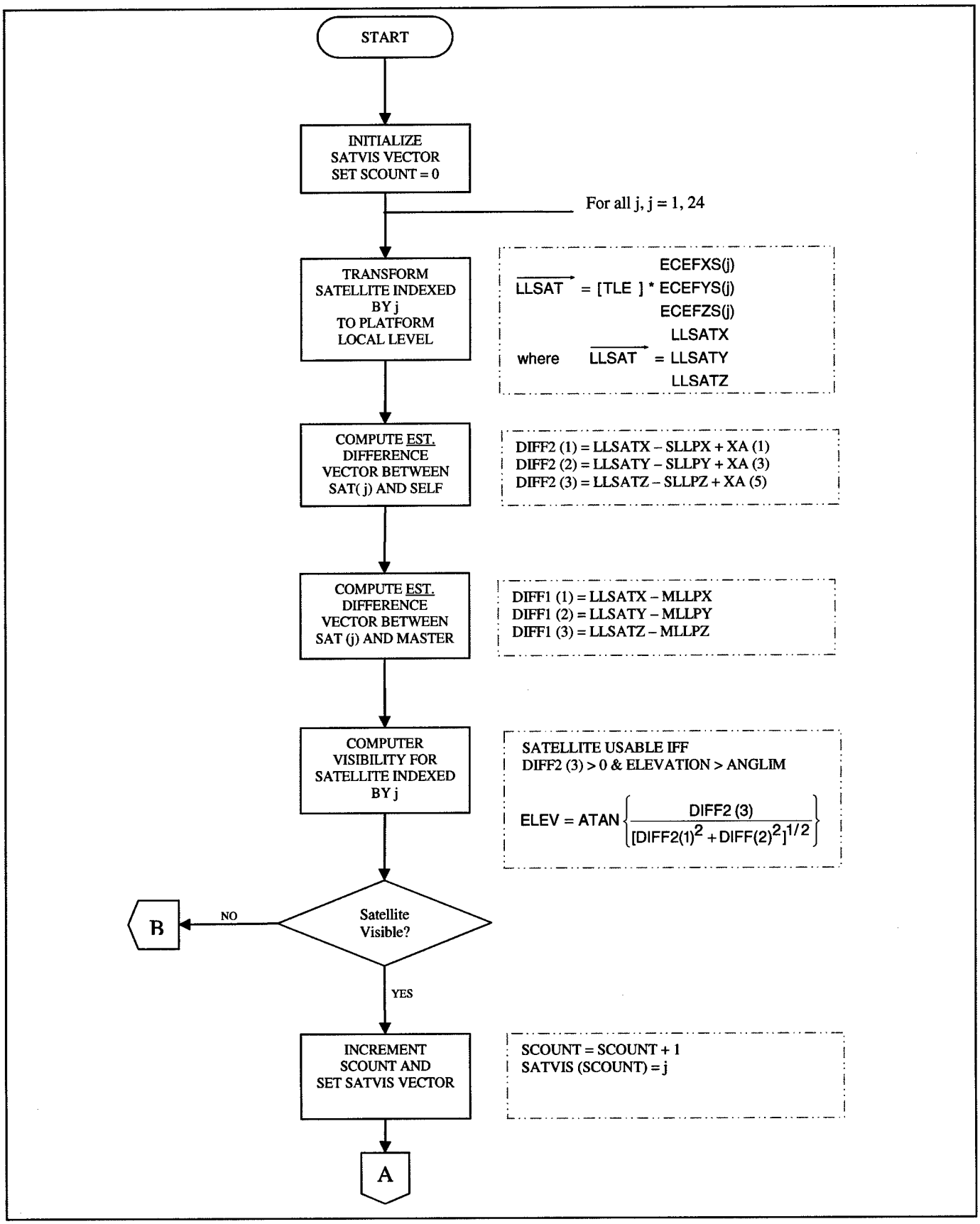
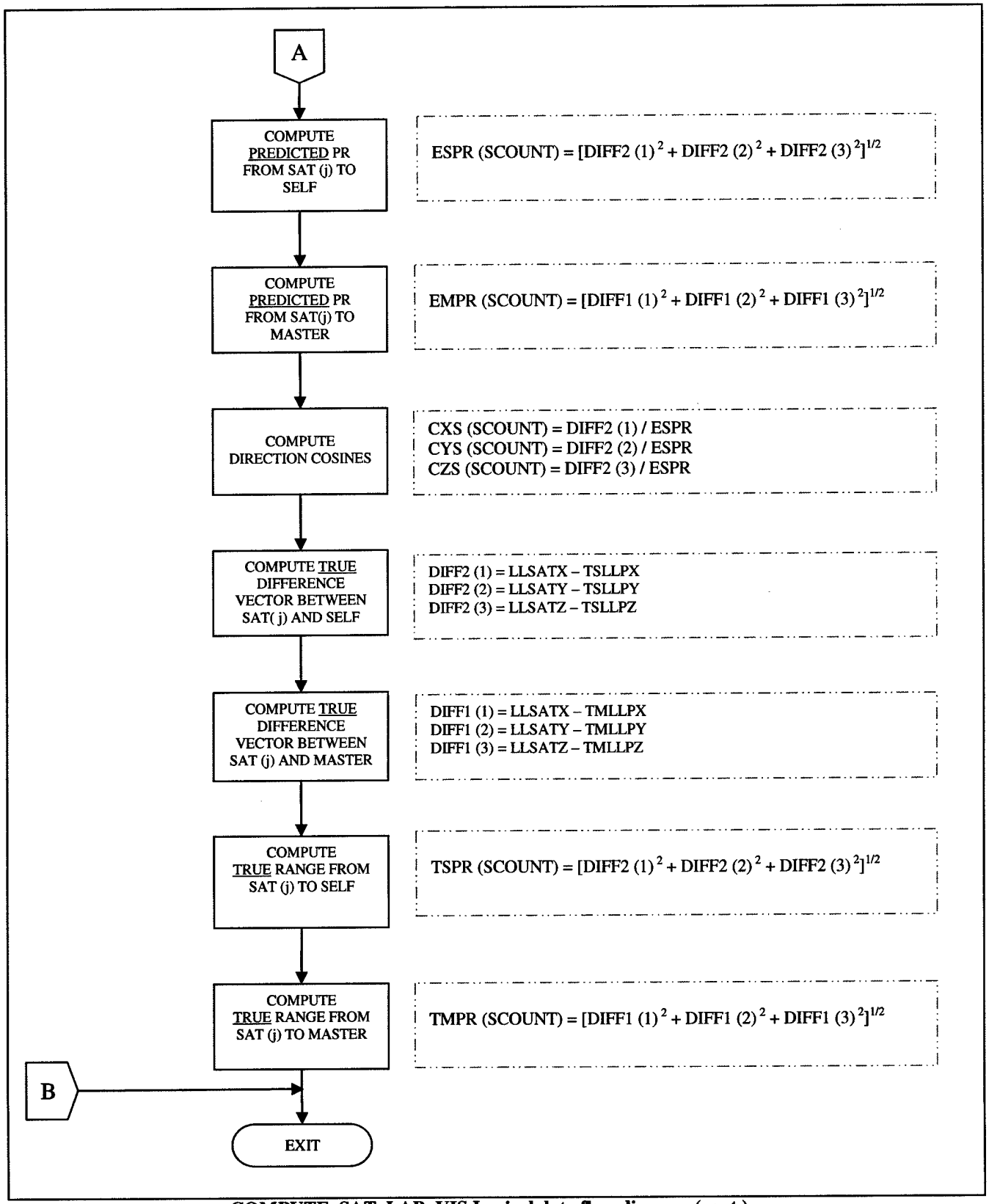


Figure D-12: COMPUTE_SAT_LAB_VIS Logical data flow diagram



COMPUTE_SAT_LAB_VIS Logical data flow diagram (cont.)

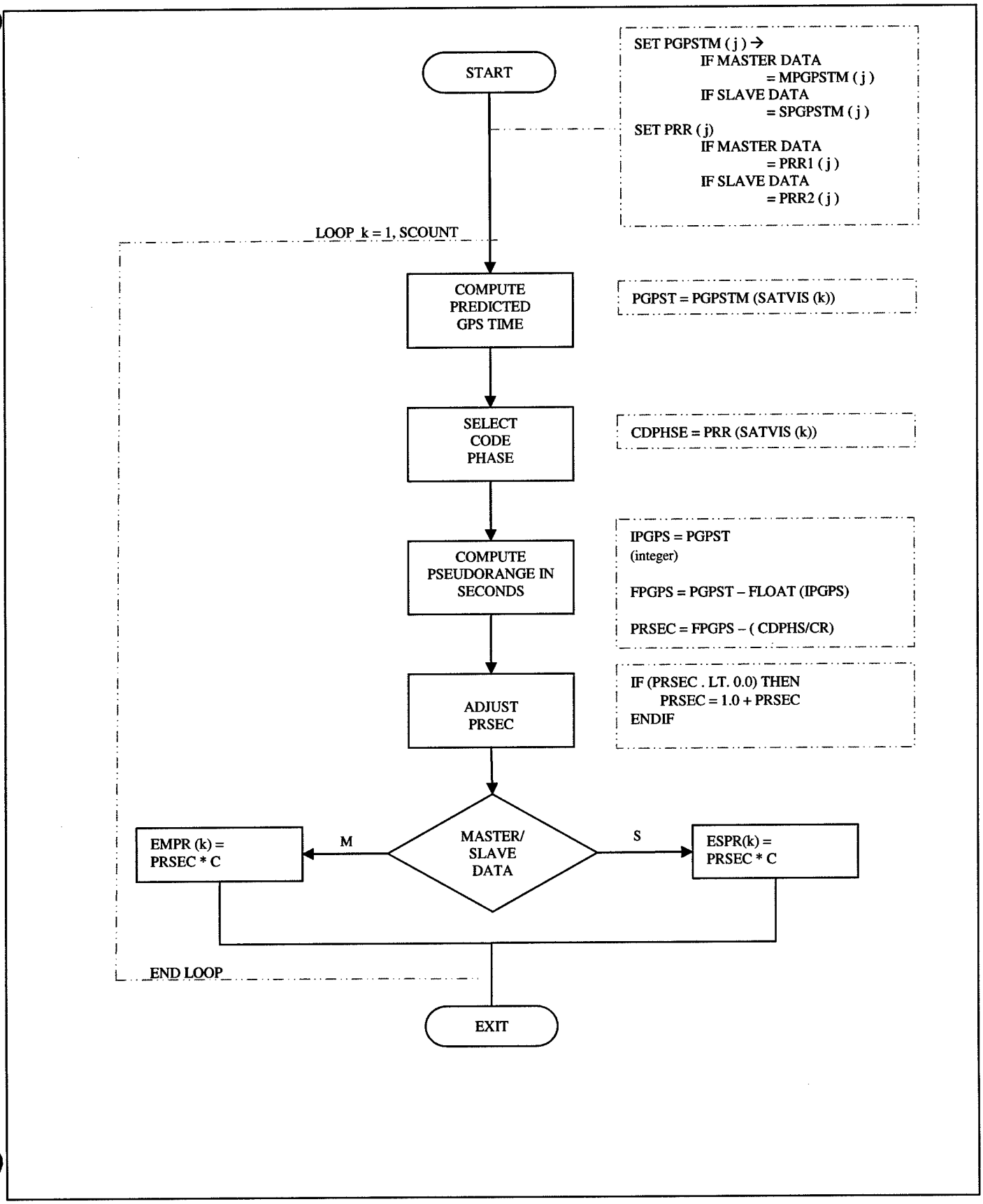


Figure D-13: PROCESS_PR_DATA Logical data flow diagram

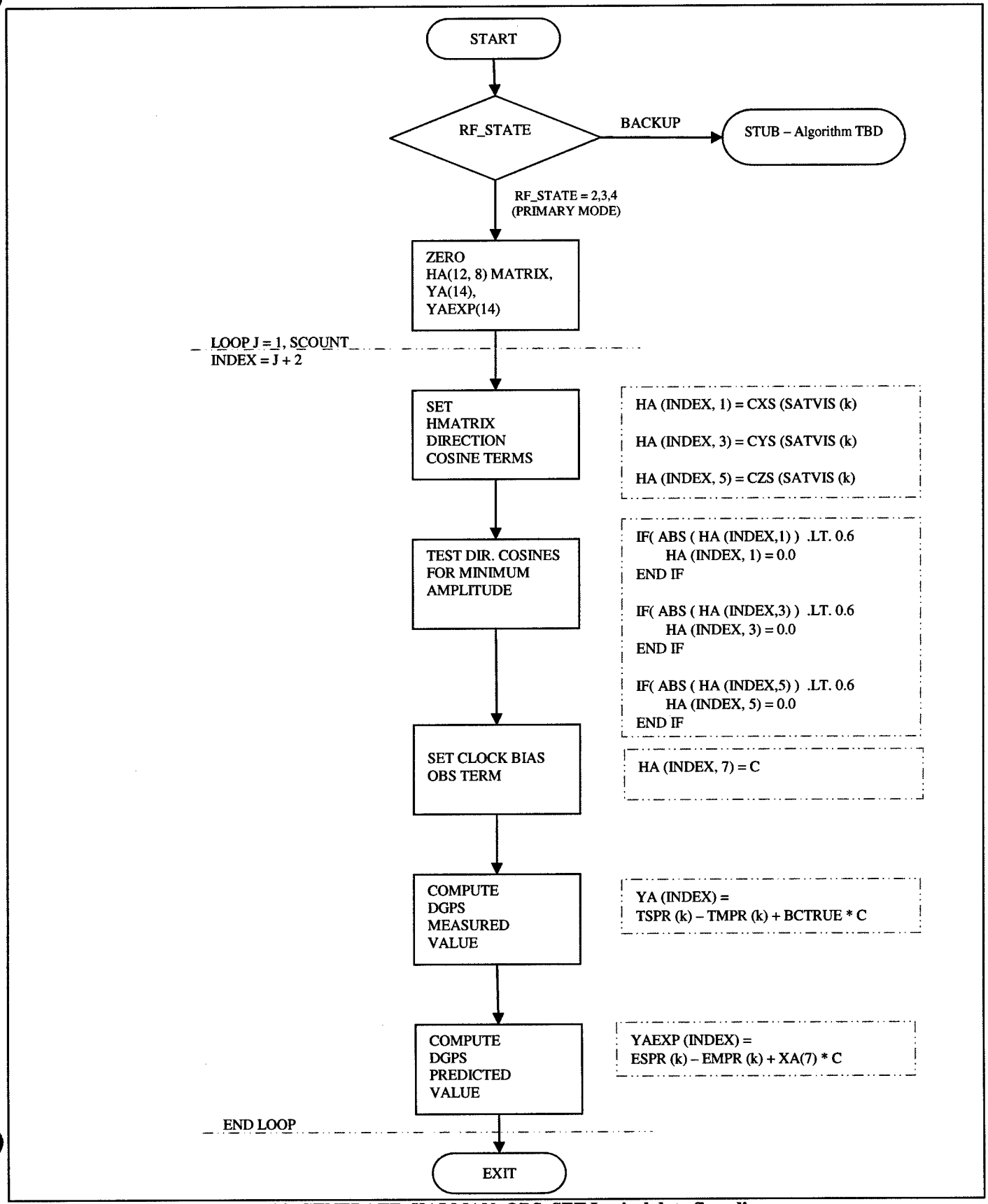
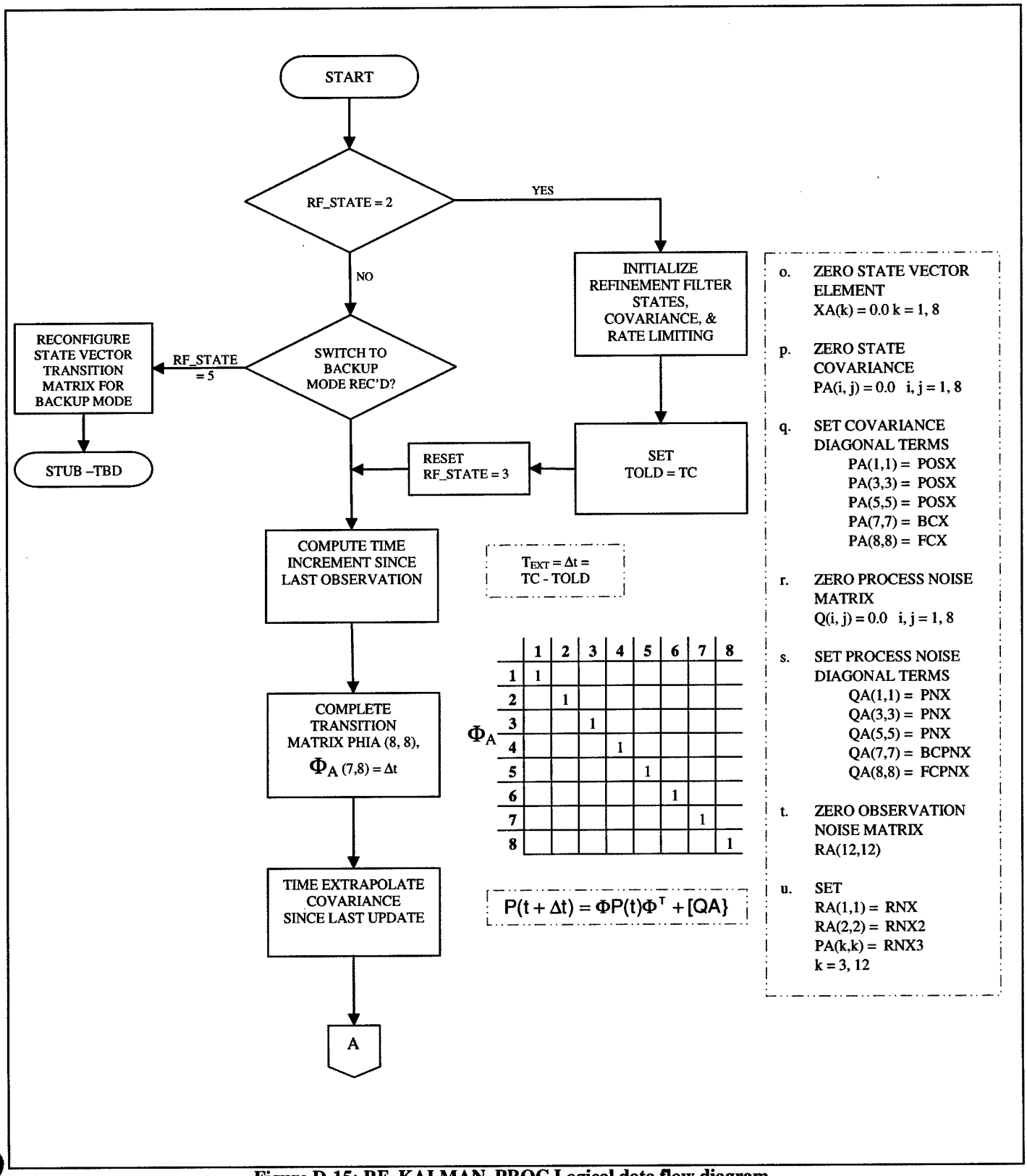


Figure D-14: GENERATE_KALMAN_OBS_SET Logical data flow diagram

III. RF_KALMAN_PROC



	1	2	3	4	5	6	7	8
1	1							
2		1						
3			1					
4				1				
5					1			
6						1		
7							1	
8								1

Figure D-15: RF_KALMAN_PROC Logical data flow diagram

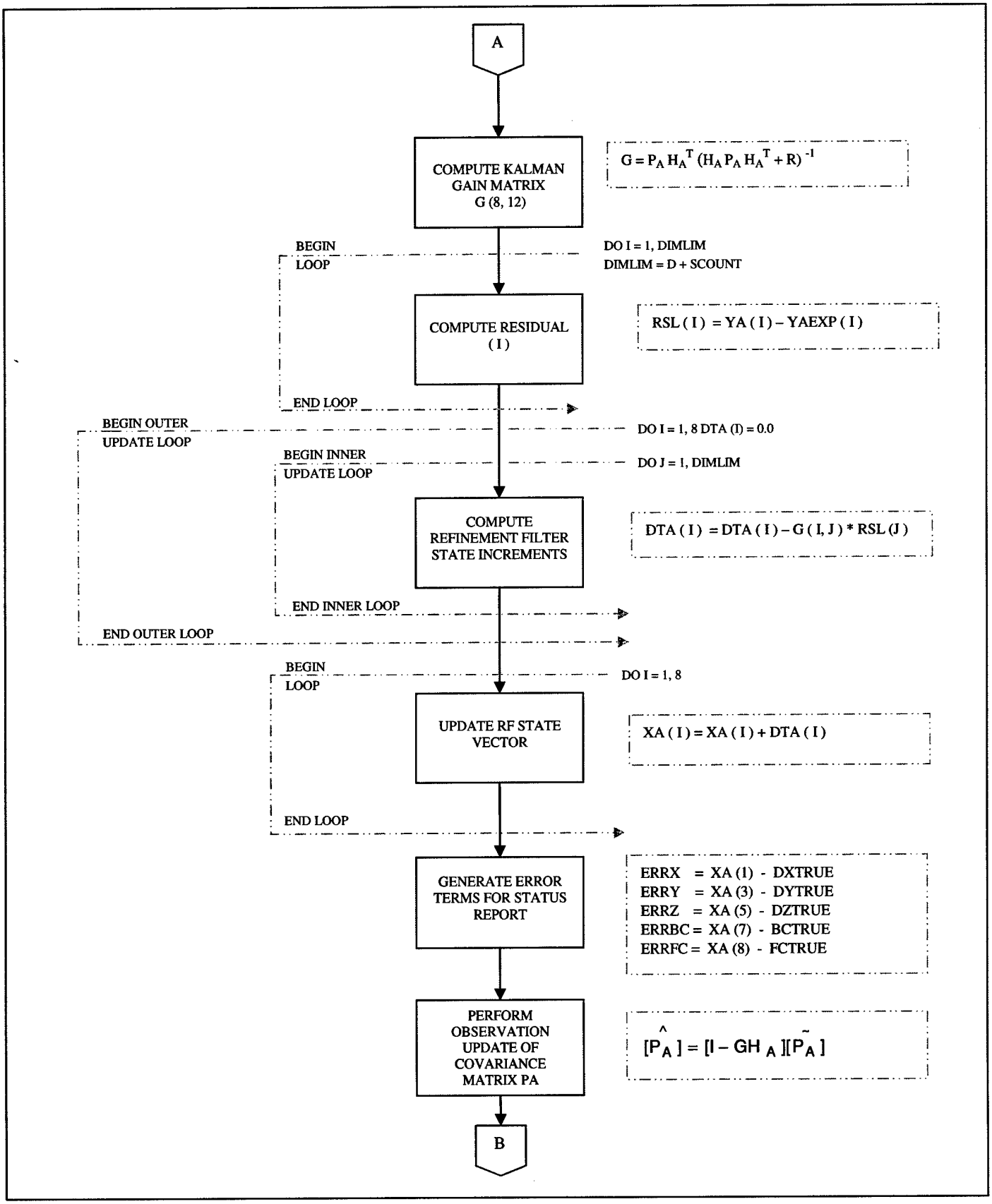


Figure D-16: RF_KALMAN_PROC Logical data flow diagram (cont.)

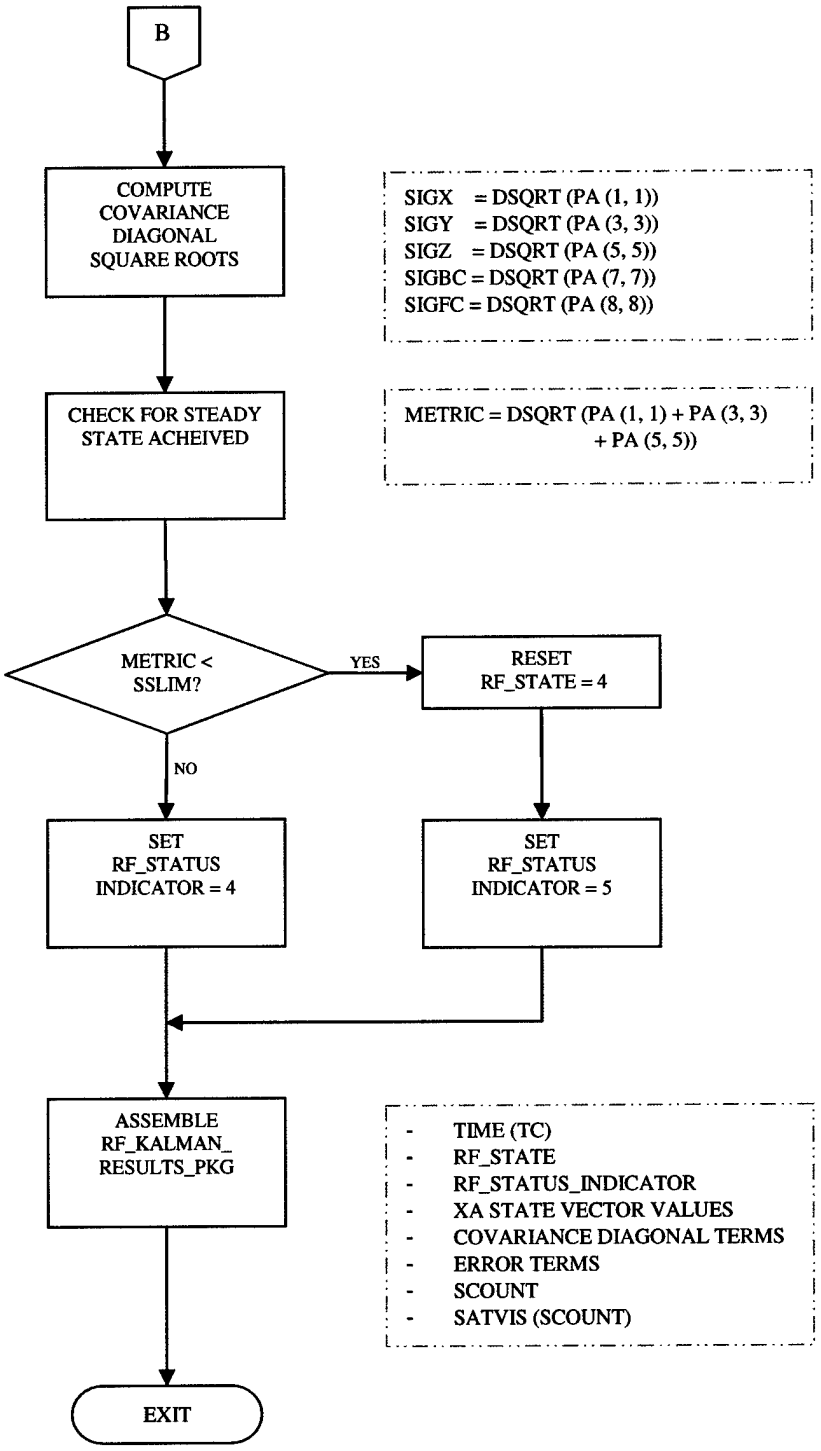


Figure D-17: RF_KALMAN_PROC Logical data flow diagram (cont.)

IV. OCP_INT_PROC

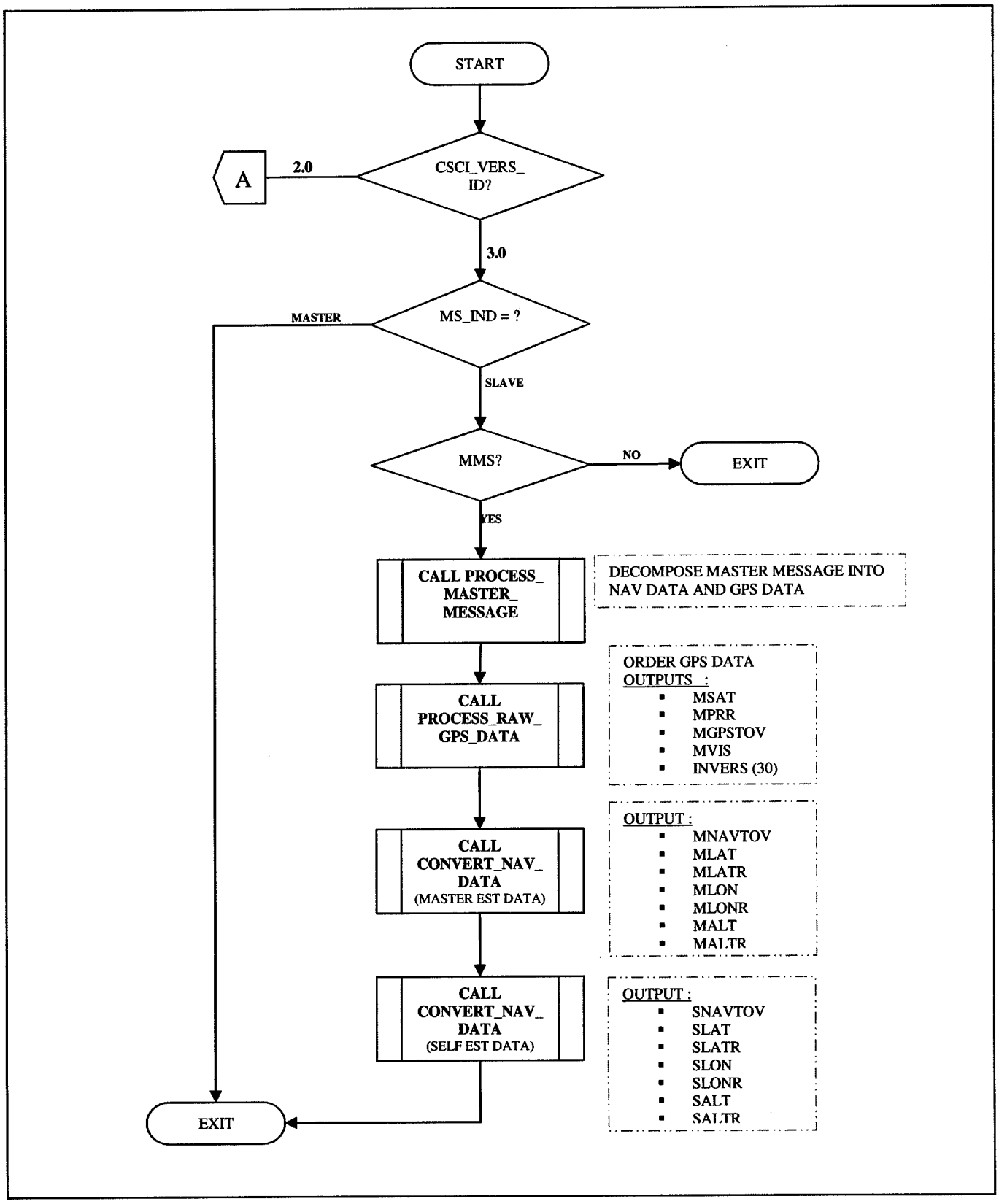


Figure D-18: OCP_INT_PROC Logical data flow diagram

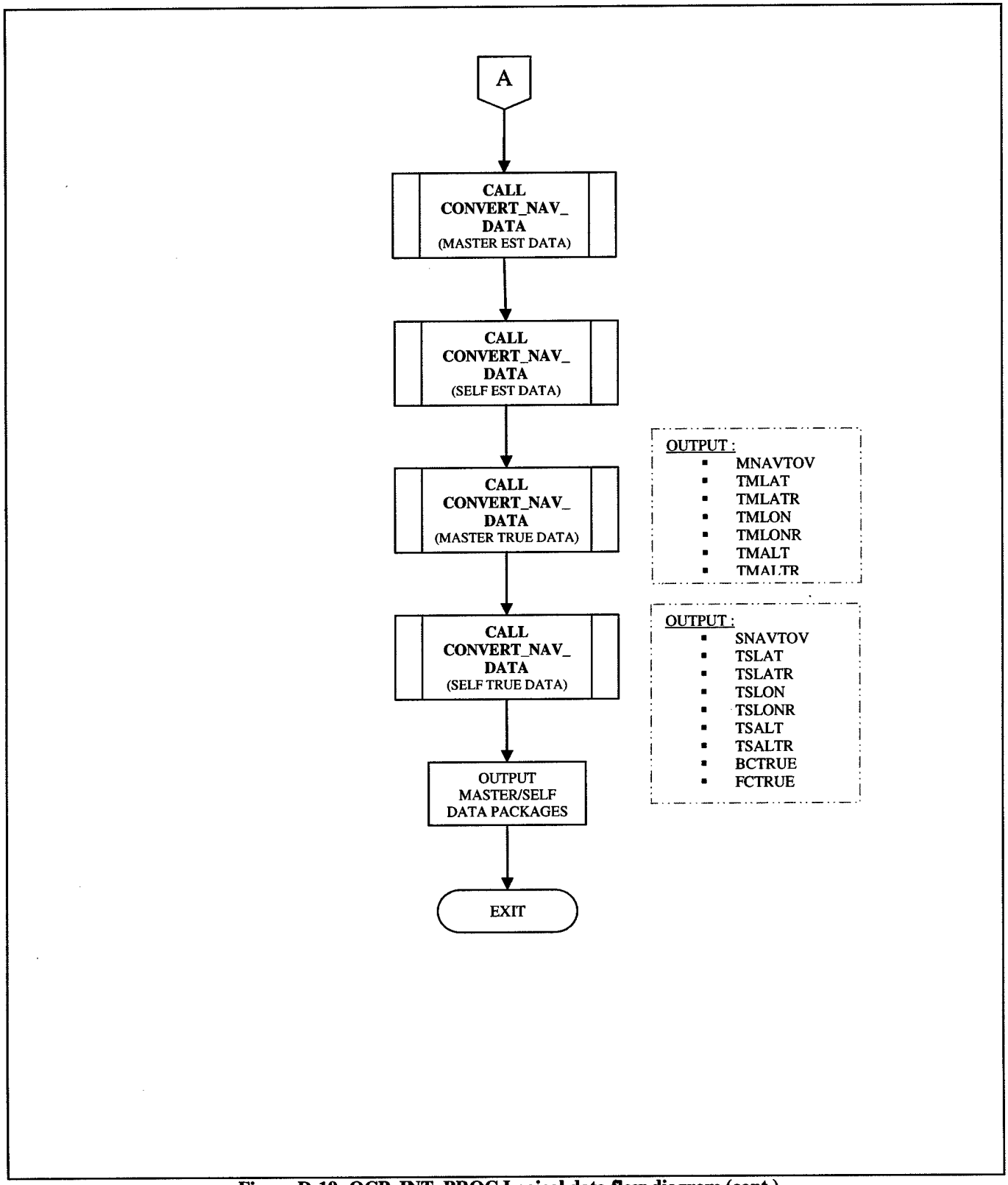


Figure D-19: OCP_INT_PROC Logical data flow diagram (cont.)

[1] Differential Operation of NAVSTAR GPS. R. Kalafus, J.V. Icons, N.Knable. ION Navigation Journal. March 1983. (Reprinted in ION GPS Reprints, Vol. II, June 1984).

[2] A Kalman Filter Approach to Prediction GPS Geodesy. R. Brown, P.Y.C Hwang. ION Navigation Journal, December 1983. (Reprinted in ION GPS Reprints, Vol II, June 1984).

[3] Global Position Systems, Inertial Navigation, and Integration. M.S. Grewal. L. Weill, A. Andrews. Wiley & Sons, 2001

[4] Canadian Marconi Northstar User's Manual. Canadian Marconi Company, Montreal, Canada. April 2000.

[5] Applied Optimal Estimation. Edited by A. Gelb. MIT Press, 1974