ARTSN:
An Automated Real-Time Spacecraft Navigation System

V. M. Pollmeier, P. D. Burkhart, and T. R. Drain

Jet Propulsion Laboratory,
California Institute of Technology
Pasadena, CA 91109

Abstract

ARTSN, the Automated Real-Time Spacecraft Navigation system is a real time system for the automated navigation of interplanetary spacecraft. The system processes the real time DSN radio metric data flow and generates updated spacecraft solutions and the associated covariance. ARTSN is designed modularly with the user interface input and output separated from the data analysis modules. Designed for distributed application, the system provides, for the first time, an automated method of processing radio metric navigation data for interplanetary spacecraft.

The system is divided into three major components which provide the user interface, the data processing, and the display output, respectively. The first version of this system is currently under development and testing, and is scheduled for delivery in September of 1996. This version is intended for use with spacecraft in interplanetary cruise with future versions providing capability of planetary orbiters.

1. Introduction

The navigation of interplanetary spacecraft is a practice which has now entered into its fourth decade. During this period, the accuracy to which the flight path of a spacecraft can be determined and the speed with which a solution can be generated has greatly improved. However, the process used today to perform the operational navigation of missions such as Galileo or Mars Global Surveyor is not dramatically different that that used on Mariner-Mars '64.

All interplanetary missions have made use of ground based radio metric data. In radio metric observations the change in the received spacecraft signal due to the Doppler effect is used to infer the velocity of the spacecraft relative to the Earth or, in the case of ranging data, the amount of time that it takes a specified signal to reach a spacecraft and return to the receiving antenna is used to infer the spacecraft geocentric range. Additionally, Voyager and Galileo have made use of, and Cassini will make use of, optical images of target bodies against a known star field, telemetered to the Earth for processing, to provide target relative position information.

With all of these data types the data are received at ground antenna sites (in the form of telemetered data for optical data or a measured signal phase for radio metric data) and electronically transferred to a ground operations facility, where the data are buffered and stored until processing by navigation analysts. The latency between observation time and processing time may be from as little as 10 minutes to as long as 6 months depending on the needs of the mission. Typically, however, the data is delivered to operations analysts daily.

Newly received data is merged with already analyzed data and the entire data set is processed via a batch-sequential least squares estimator. In this process, the identification and deletion of invalid data as well as the operation of the software is performed by an analyst operating at a workstation console. The process of fitting the data requires the use of multiple software links and the manual examination of pre-fit residuals to determine which points should be fit and which points should be deleted from the solution. After generating the best estimate of the spacecraft trajectory based on the input models, the analyst must determine the appropriate set of output coordinate frames and mappings that are desired to view the solution and use the software to generate post-fit residuals. Typically this process requires approximately one hour of additional processing time after the data is received by the operations analyst. When it is necessary to evaluate multiple models, as is the normal procedure, multiple analysts must work in parallel or additional processing time is required. For these reasons, navigation team staffing in the past has typically been on the order of one to two dozen analysts.

Recent missions have begun to institute greater automation of portions of the process, allowing for a lessened need for staffing and greater performance. The TOPEX/Poseidon operational navigation team has had success with automating some functions. This has allowed the operations support team to dramatically decrease the staffing required for processing the large amount of data received by TOPEX. Additionally, the Mars Pathfinder navigation team is planning to make much greater use of automation in ground operations than has been previously used. However, in both cases the nature of the automation focuses on the use of scripts and automated routines which use the underlying software than on the development of a robust system intended for automated use.

The development of a conceptual prototype for a system designed for automation began in mid 1994. Although such automated systems have been developed for Earth orbiting missions, they had not previously existed for interplanetary missions. This prototype, known as RTAF (Real-Time Automated Filter) [Ref. 1] was intended to be developed quickly using already existing software modules and pieces of code where possible. The purpose of the RTAF development was two-fold: 1) verify that the concept of using an Extended Kalman Filter (EKF) and that the automated processing of interplanetary navigation observables was feasible and 2) identify potential stumbling blocks in the development of a fully operational system. The RTAF development resulted in a number of key lessons learned:

- Modularize the architecture of the system whenever possible.
- The most reusable portion of the existing code was the algorithms.
- Rewriting the code to be more compatible with the automated structure was time well spent.
- Display tools and control user interfaces should be separated from the primary analysis system and generated with commercial software when possible.
- The addition of new force or observable models should be as streamlined as possible.
- The basic concept of the nearly totally automated navigation operations system for interplanetary navigation was feasible and achievable.

The motivation for an operational implementation of a system similar to that prototyped with RTAF was simple economics and efficiency. Changing the operational paradigm from one of constant oversight to one of exception monitoring provides the opportunity to greatly decrease the staffing needed to support such missions. Further, the ability of the automated system to provide faster orbit solution generation than manual systems allows for a greater ability to support missions which have a short turn around between the occurrence of a critical event and the generation of a required response to that event. An example of such a situation is the passage of a planetary orbiter through the atmosphere to circularize its orbit. If the succeeding orbit is too low, then the spacecraft engineers need to respond quickly to this. A fully automated system provides this capability at a potentially much lower cost than a fully developed on-board autonomous system.

For these reasons, the development of a system based on the RTAF prototype was begun in the summer of 1995. Version 1.0 of this system, the Automated Real-Time Spacecraft Navigation (ARTSN, pronounced artisan) is currently scheduled for completion in September 1996. The Mars Pathfinder, Mars Global Surveyor, and Near Earth Asteroid Rendezvous missions have all expressed interest in using the system in operations. The remainder of this paper address the system design and implementation strategy of the software system.

2. System Design

2.1 System Architecture

In keeping with the experience learned from the development of RTAF, a conscious decision has been made to separate the major functions of ARTSN into separate modules. The basic structure of the software system is shown in Figure 1. There are three major components to the system: ARTSN Control, ARTSN Engine, and ARTSN Monitor. The three components serve the basic functions of user input/control, numeric data processing, and output display, with inter-module communication based on TCP/IP.

It is important to note, that by segregating the functions, it is possible to allow for greater flexibility in the appearance and operation of the system. In normal use, a user would start the ARTSN Control module to configure the operation of and specify the model inputs for the filter. Typical control inputs would include data types to be fit, nominal data weight to use, parameters to estimate, etc. ARTSN Control would then pass this information to ARTSN Engine which would process the data stream, generating solutions as data are received. The filter solutions, the associated covariance, and other solution information would be output from ARTSN Engine to the ARTSN Monitor module which would display the resultant information for the user. Since these three basic functions are divided at clean interfaces, the user interface development is made easier, both for the input control and for the output display. Further, as new technologies and capabilities emerge, it is possible to completely change the user interface points without changing the underlying filter. For experienced analysts, the full complexity and options which the system inherently has can be made available via a version of ARTSN Control which allows complete freedom to change all parameters. For more simple operation, an interface which allows only the modification of basic options and uses standard values for all other parameters would require only the modification of the control software, not a change to the underlying filtering system. Similarly, multiple options for ARTSN Monitor exist, depending on the needs of the end users. As the three components are designed to communicate via TCP/IP, it is possible to distribute the components on different systems and, in fact, the user configuring and operating the system may not be the user monitoring the output. This capability has great potential for situations where monitoring of the performance of a spacecraft is done remotely from the navigation team (as is the case with NEAR) or where the real-time monitoring team does not have a navigation background.

To further improve the portability of the entire system across various computing platforms, the entire system is written in ANSI C, except for those portions of the system which use commercial products. This is a major departure from previous JPL navigation software which has been written exclusively in FORTRAN. It is felt that the use of ANSI C, will ease the porting of the system across a number of platforms and allow for easier implementation of a number of the features desired in the system. The decision to use C, however, necessitated the rewriting of all code from previously existing software.

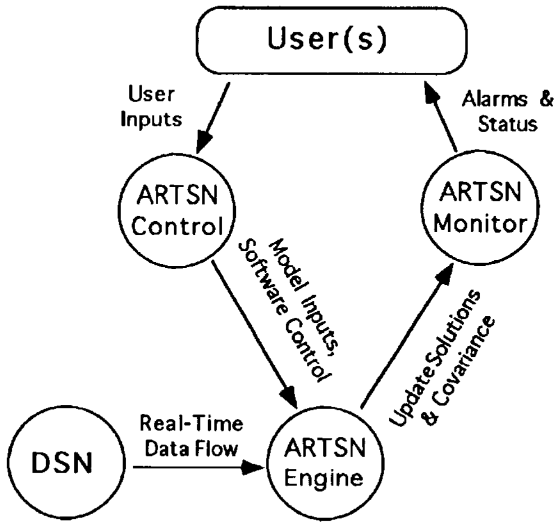

Figure 1: ARTSN system schematic

2.2 Network Data Flow

As indicated in Figure 1, data flow from the DSN to the ARTSN Engine. The real-time radio metric data arrive at the navigation computers in formats governed by DSN interfaces TRK2-15a and TRK2-30 [Ref. 2]. This interface specifies the data as a series of formatted binary data blocks of fixed size. Each word within a block of data has a specified meaning based on a look up table. While the ARTSN Engine will read this format directly, plans are being considered to augment this data flow with a new format which is more extensible and which will eventually supplant TRK2-15a. If this new format is implemented, ARTSN will replace the TRK2-15a readers with those capable of reading the new format. The ARTSN Engine also has the capability to read from a standard binary navigation tracking data file. This capability allows for the processing of previously received data as well as for the testing of the system in comparison with extant navigation operations software.

An additional complication involved in the processing of interplanetary radio metric data, not ordinarily found with Earth orbiter data, is the necessity of maintaining transmitter frequency tables. Since the round trip light times can be very large for interplanetary spacecraft (exceeding 16 hours for Pioneers 10 & 11), it is a non-trivial matter to determine the uplink frequency to a spacecraft that resulted in the received frequency observed by the DSN. Since the uplink frequency reported in real time by the DSN is the current uplink frequency, it is necessary for the system to either wait one round-trip-light-time after initializing to begin processing data or to read an already compiled table of these values from a server. The ARTSN Engine sub-systems responsible for reading the real-time data flow will choose either of these options based on user input and/or availability of the uplink tables. Current operational plans call for the development of a separate server to monitor the DSN data flow and maintain a recent history of the uplink frequency tables for various spacecraft which can be accessed to initialize the filter. Once initialized, the ARTSN Engine will maintain these tables internally and they will be invisible to the operator.

2.3 Components

2.3.1 ARTSN Control

Of the three major components of the ARTSN system, the ARTSN Control interface will be the last implemented. Primarily, this is do to its dependence on the implementation of the ARTSN Engine module. In testing and development the ARTSN Control interface is implemented as a simple namelist input. This namelist is read and input into the database system that underlies the ARTSN Engine and is schematically illustrated in Figure 2. Current plans for Version 1.0 do not involve any upgrade to this basic namelist input. Without any interface usability testing, development of a more complex user interface was deemed premature. The basic design of the ARTSN system is intended to allow for the upgrade of such interfaces as a normal part of the evolution and aging of the system.

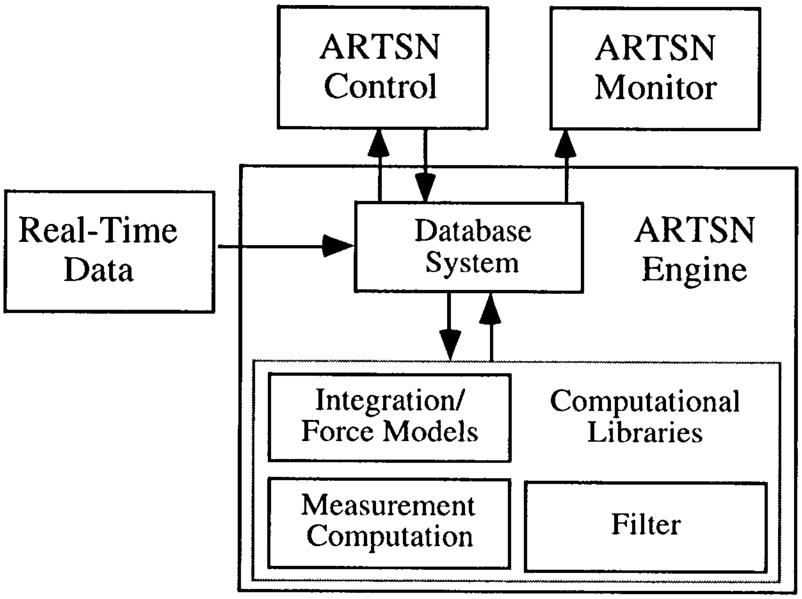

Figure 2: ARTSN Engine Schematic

2.3.2 ARTSN Engine

The ARTSN Engine is by far the most complicated of the three primary ARTSN modules and provides the main processing capability of the system. Included are force modeling, measurement computation, data validation, and the Kalman filter relations. The entire system is built on an underlying layer of computational libraries as indicated in Figure 2. The interface between various sub-systems as well as communication external to the module is via a database system which is discussed in more detail later in this paper. This structure allows for a modular implementation of capabilities and subsystems within the module.

2.3.2.1 Software Implementation Structure

One of the main goals of the ARTSN software implementation is to provide greater supportability and expandability than previous software sets have delivered. With this idea in mind, the concept of the ARTSN database has evolved into it's current structure. The database provides a means of storing and retrieving an unlimited amount of different types of data. Through the use of dynamic memory allocation, there are no fixed array sizes assumed for any of the data types. This allows the program to expand up to the maximum memory available in order to handle larger user inputs. The database provides the functionality needed to implement a modular, expandable, data driven system.

Entries in the database can be stored on disk or in memory with the interface remaining transparent to the calling module. This ability to read and write areas of the database to disk allows the ARTSN computational libraries to be completely insulated from the format of the file system, and to some extent, the files themselves. Binary, ASCII, and other portable file formats can all be supported without changes to the code outside of the database routines.

The database offers an efficient means to expand or change modules in future versions. During initialization, modules notify the database of their existence. ARTSN Control then queries the database and retrieves the list of available modules for the user to select from. In this way, new modules can be added simply by notifying the database of their availability.

The main approach in the coding of ARTSN computational elements is one of object construction and manipulation. Packets of related data are grouped together into an object and tagged with an identifying number or character label. The computational routines were then written to process the data contained in these objects. Since each object's identifier is unique, expanding the system is simply a matter of adding new objects. As an example, future versions of ARTSN may include the ability to process spacecraft to spacecraft data types, GPS data for instance. Since ARTSN groups all of the data associated with a spacecraft into a "spacecraft object", the problem of integrating and tracking the information relating to the GPS constellation becomes very simple. A "spacecraft object" for each GPS satellite and the spacecraft can be created and their positions at a given epoch requested by an identification number. In an object oriented approach, processing multiple objects is no more complicated than processing a single object.

2.3.2.2 Integration Force Modeling

The ARTSN system will initially be built with support for five force models: gravity (including oblate bodies, relativistic effects, and Newtonian point mass), solar radiation pressure, atmospheric drag, finite and impulsive motor burns, and unmodeled accelerations. Additional models will be written as needed and will include a planetary radiation pressure model and a planetary tide model. The implementation of the integration module is designed so that the addition of new force models only requires that the database be notified of their existence. This allows new forces to be added with little or no changes made to the existing software. It should be noted that the mathematical formulation of the ARTSN force models is identical to that used by JPL's DPODP software [Ref. 3]

The ARTSN force models allow inputs in several coordinate systems. The user may specify any number of coordinate systems with axes dependent on planet and satellite body positions, directions in inertial space, or the spacecraft velocity vector. All user coordinate systems are tagged with a label which allows that system to be used by any other ARTSN module. The integration module uses these tags to allow spacecraft geometrical components, finite motor burns, and impulsive motor burns to be input in any of the defined coordinate systems.

The atmospheric drag and solar radiation force models are dependent on the geometrical properties of the spacecraft. The ARTSN system allows multiple geometries to be input for a single spacecraft, each one referenced through a different character label. This allows each force model to use a different geometrical representation of the spacecraft if the user desires. The total spacecraft geometry is built from any number of geometrical components. ARTSN supports one and two sided flat plates, cylinders, and spheres. Each component can be oriented along a direction in any of the user defined coordinate systems. ARTSN also supports an abstract "bus" model. The "bus" model consists of mathematical abstraction of three reference areas along the axes of a user defined coordinate system.

The ARTSN gravitational model includes the affects of Newtonian point-masses, oblate bodies, and relativity perturbations. Each of these models can be turned on or off for any celestial body. The oblate bodies model simulates the perturbations due to a nonspherical gravitational force and is capable of handling any order and degree of gravity field. General and special relativity perturbations are also modeled. Tidal perturbations to the gravitational acceleration will be included in future versions of the software.

The solar pressure model computes the acceleration due to solar radiation pressure on the various components of the spacecraft. The spacecraft geometry is constructed and each component assigned values for the specular and diffuse reflectivity and degradation coefficients. One and two sided plates, cylinders, spheres, and a "bus" are all available as component shapes. Future versions may include a parabolic antenna model. A conical shadow model with the sun as a point light source or a complete umbra/penumbra shadow model is available. The shading of one component by another component is not modeled.

The atmospheric drag model for the initial ARTSN version is a function of a single drag coefficient and the effective area of each component in the direction of the spacecraft velocity vector. Future versions will include a model based on the lift and drag coefficients of each geometrical component. Various atmospheric models can be selected by the user. A scale height exponential model and the DTM [Ref. 4] model are include in the initial release.

The ARTSN system accepts two types of engine burns: finite and impulsive. Each burn vector can be input in inertial space or any of the user defined coordinate systems. The impulsive burn model allows for an impulsive change in the spacecraft mass, position, and velocity at a given epoch. The finite burn models accepts a and mass flow rate and thrust polynomial in each of the three axis directions of the burn coordinate system. The burn can be stopped after a fixed length of time, or after a total delta-V has been imparted to the spacecraft.

The unmodeled accelerations module is used to model perturbations due to unknown forces. Empirical acceleration polynomials may be entered for a given time in order to account for unknown forces, small thruster events (such as momentum dumping), or errors in the other models.

2.3.2.3 Observable Modeling

The primary data type used for interplanetary tracking is Doppler data. This is essentially a measure of the spacecraft velocity along the tracking station-spacecraft line of sight. There are three main types of Doppler data: one-way, two-way and three-way. One-way Doppler is a signal that originates at the spacecraft and is received by a ground tracking station. Two-way Doppler originates at a ground tracking station, is received and rebroadcast by the spacecraft, and received at the transmitting ground station. Two-way Doppler requires a transponder on the spacecraft. Three-way Doppler is a special case of two-way Doppler, where the transmitting and receiving ground stations are not the same. The data type is formulated in the software as the difference of two values of the phase of the received signal divided by the time between the data points. This differenced phase essentially describes the spacecraft line of sight velocity at the midpoint of the observation interval.

The second most common data type used is range data, collected using the DSN Sequential Ranging Assembly (SRA) [Ref 5]. The range observable is a measure of the round trip distance along the line of sight from the tracking station to the spacecraft and back to the station. The data type is formed by measuring the round-trip signal time by correlating a received signal over some period with the transmitted signal to determine the offset between transmit and reception times.

These radiometric data types are primarily broadcast at S, X, or Ka band. However, the formulation of the observables in the ARTSN Engine is such that as long as the uplink frequency and the spacecraft turn-around ratios are known, the system can process the data. The ARTSN Engine will process ramped and unramped Doppler data. Ramped data are those in which the uplink frequency to the spacecraft was not held constant during the Doppler count interval. Unramped data are treated in ARTSN as a trivial case of the more general ramped data.

In addition to the dynamic errors on the spacecraft state discussed earlier, there are also several radiometric error sources that affect the radio signal itself. These include troposphere and ionosphere effects, station location errors, Earth platform errors (pole motion, UT1, precession and nutation errors), and solar plasma errors. The radio signal must travel through the troposphere and ionosphere when it is received at the ground station and (when applicable) transmitted from the ground-station. The effect of the troposphere and ionosphere on the signal is that the path is bent, causing the signal to travel a greater distance than a straight line. The station locations are known to the centimeter level, but due to the extreme sensitivity of the observable to errors in station location, uncertainties due to antenna offsets and various Earth platform uncertainties are included. The Earth platform errors include uncertainties in the orientation of the Earth rotation axis, the rotation rate, and continental plate motion. Additionally, for Earth orbiters, the Earth platform errors will affect the gravity field and thus the dynamics of the spacecraft. Solar plasma will affect the radio signal in the same fashion as the ionosphere. The charged particles will cause the ray to bend, changing the distance traveled. These radio metric error sources are included in the computed measurement as variations in the observed value.

2.3.2.4 Data Validation

Data validation is essentially a simple pre-processor for the data prior to the filter. The extended Kalman filter, discussed later, is sensitive to erroneous data. Since processing a data point that is in error can cause the filter to diverge, a scheme is implemented to ensure that the data processed fall within some regime. The function used to determine the quality of the data is a chi-square density function test. Simply put, if the chi-square density for a specific data point falls within a specified range, then the data point is processed unchanged. If not, then one of two things may be done. In the simplest (and harshest) case, the data outside of the specified range are not processed. In the more complicated case, the data are weighted by some function of the density and then processed. The chi-square density is determined based upon the current best estimate of the spacecraft trajectory and the associated covariance. Consequently, if the position and velocity of the spacecraft are not known well known, the filter will accept and process most data. However, when the position and velocity become better known, the filter will reject or de-weight data which has large pre-fit residuals. This is due to the greater probability of the data being erroneous.

2.3.2.5 Filter

The filter implemented in the system is an extended Kalman filter (EKF) [Refs. 6,7]. The EKF has been used for many years in a wide variety of real-time applications. The primary difference between the EKF and the linearized Kalman filter is that the nonlinear state and measurement functions linearized about the current estimate, not an *a priori* reference trajectory. The advantage is that the linearization reference will be closer to the estimate, reducing linearization error at the expense of computing all system matrices at run time [Ref. 8]. The filtering problem for orbit determination involves modeling the trajectory of a spacecraft including random dynamics and enhancing the model with the inclusion of data collected from the actual spacecraft, which includes effects not modeled at all or modeled as stochastic functions. The EKF is a method for incorporating information from the measurements of the actual system into the solution from the mathematical model to generate the solution that minimizes the mean-square error in the state.

The actual process used in an EKF can be divided into two parts. One is the propagation of the state and its uncertainty from one measurement time to the next. The other is incorporating measurement information into the state and the error covariance at the measurement time.

The EKF operates as follows. At some known initial time, a state value and a state error covariance (the accuracy of the state) are specified. A measurement of the spacecraft position or velocity (as described above) is received with an associated time index. Using the simulated dynamics, the spacecraft state is computed at the measurement time tag, and the simulated state is used to compute a value for the measured quantity. The state error covariance is propagated using the state transition matrix to this measurement time. The difference between the computed and measured values, called the measurement residual, is used to update the value of the state for use in the simulated dynamics. In addition, the state error covariance is updated using the measurement residual. This process is repeated for subsequent measurements until all the measurements have been processed.

There are two main implementations of the extended Kalman filter used in ARTSN. The first is a standard implementation of the Kalman filter equations as derived. This approach, while correct in theory, has some well-documented numerical instability, especially in the computation of the error covariance. The other implementation utilizes factorized equations. These equations are equivalent to the theoretical relations but are more numerically stable, at the expense of added computational cost. The main factorized EKF relations are referred to as the square root EKF relations and the U-D, or square-root free, relations. ARTSN utilizes the U-D form of the factorized KF relations.

2.3.3 ARTSN Monitor

The ARTSN Monitor module provides the primary output display for the ARTSN system. The monitor receives a stream of solution updates, solution covariance information, and other information regarding the solution. This data stream is transferred from the ARTSN Engine to the ARTSN Monitor via TCP/IP, thereby allowing the two processes to reside on different platforms. In fact, it is possible for multiple copies of ARTSN Monitor to operate on data streams from a single ARTSN Engine. This then, allows a navigation analyst as well as others, such as spacecraft engineers or project science personnel to monitor the current spacecraft state. Since it is unlikely that each of the prospective users would desire that the data be presented in the same manner, it is assumed that various version of ARTSN Monitor will be developed for various customer needs and that the display would undergo frequent update based on user and mission requirements. Additionally, it is desirable for the display to be available on various computer platforms including Sun and HP workstations, PC's running Windows, and Macintoshes. For these reasons, it was decided to use a commercial software tool to generate the ARTSN Monitor rather than developing it from scratch.

The environment selected for the development of the baseline ARTSN Monitor is LabView, a product of Analytical Graphics, Inc. LabView is available on each of the desired platforms and displays developed on one platform are readily portable to others. Figure 3 shows the appearance of a general purpose ARTSN Monitor display intended for use by real-time operations personnel. In addition to information concerning the most recently received data, basic statistical analysis of the data is provided as well as comparison of the solution to a reference trajectory. The monitor tool also performs basic checks to verify that the solution is not diverging from the reference trajectory and that an excessive amount of data has not been deleted. Each of these errors generates an alarm message on the Monitor display as well as optionally notifying the cognizant navigation analyst via an alpha-numeric pager. In this manner, a single cognizant navigator can support a large number of missions and the navigation system can be monitored by analysts who are not necessarily skilled in navigation.

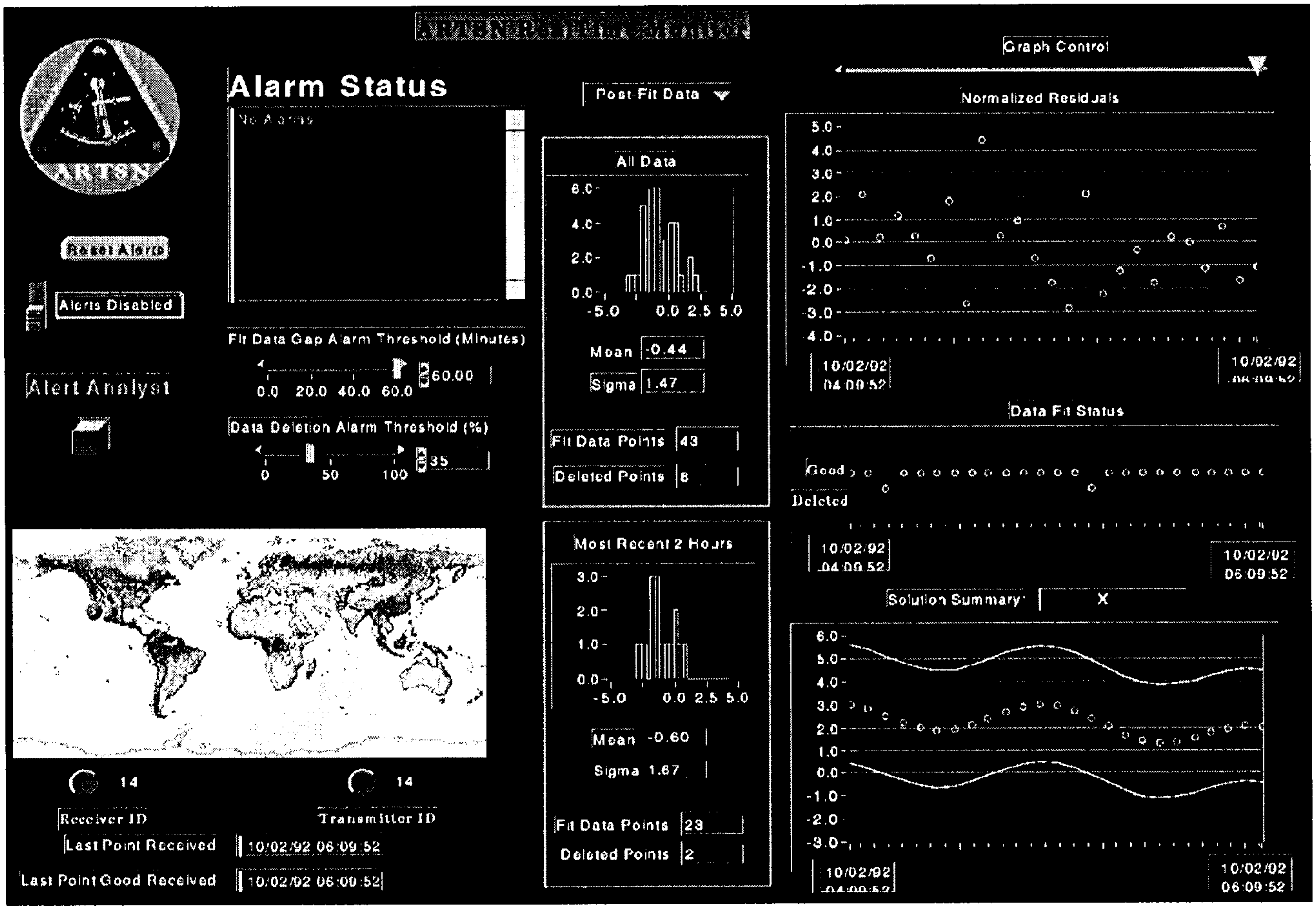

Figure 3: ARTSN Real-Time Monitor

3. Status and Future Plans

Version 1.0 of ARTSN is currently on schedule for delivery in September of this year (1996). The force modeling has been verified and found to match that from the JPL operational orbit determination program. Observation modeling is currently being implemented and tested for Doppler and ranging. Processing with operational NEAR data to validate the system will begin during the summer. Validation tests by the Mars Pathfinder project and Mars Global Surveyor are anticipated in early calendar 1997.

This first version of the ARTSN system is intended primarily for the support of spacecraft in interplanetary cruise. However, model additions to support the precision navigation of planetary orbiters, such as those currently planned for the Mars Exploration program, are identified as high priority items for the second release. Other items identified for implementation in future versions include the processing of radio antenna angle data, optical navigation data, and GPS data types.

The ARTSN system provides a building block on which a number of new capabilities can be developed. Among these are the ability to provide self contained software to support spacecraft navigation using small remote terminals and the ability to generate closed loop metric predicts for controlling DSN antennas. Additionally, ARTSN will reside as the core of an effort to develop adaptive filtering strategies which can further automate the navigation of interplanetary spacecraft [Ref. 9].

4. Acknowledgments

The authors wish to thank and acknowledge the assistance of W. C. Masters, T. P. McElrath, and L. A. Cangahuala in the development of various components of the ARTSN system.

Reference herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise does not constitute or imply an endorsement by the United States government or the Jet Propulsion Laboratory, California Institute of Technology.

The work described herein was performed at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration (NASA).

5. References

1) Masters, W. C. and Pollmeier, V. M., "Development of a Prototype Real-Time Automated Filter for Operational Deep Space Navigation," Proceedings of the Third Int'l Symposium on Space Mission Operations and Ground Data Systems, Greenbelt, Maryland, 1994.

2) *Deep Space Network Requirements/Detailed Interface Design*, JPL Document 820-13, Rev. A, Release 227 (internal document), Module TRK 2-15a, Jet Propulsion Laboratory, Pasadena, California, 15 December 1995.

3) Moyer, T. D., Mathematical Formulation of the Double Precision Orbit Determination Program, JPL Technical Report 32-1527 (internal document), Jet Propulsion Laboratory, Pasadena, California, 15 May 1971.

4) Barlier, F., et al, "A Thermospheric Model Based on Satellite Drag Data," Ann. Geophysics, T. 34 , FASC. 1, 1978.

5) *Deep Space Network/Flight Project Interface Design Handbook Vol. I*, JPL Document 810-5, Rev. D. (internal document), Module TRK-30, Rev. C, Jet Propulsion Laboratory, Pasadena, California, November 1994.

6) Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," Transactions of the ASME, Ser. D, Journal of Basic Engineering, March 1960.

7) Brown, R. G. and Hwang, P.T.C, Introduction to Random Signals and Applied Kalman Filtering, 2nd Edition, John Wiley & Sons, New York, New York, 1992.

8) Gelb, A., Applied Optimal Estimation, MIT Press, Cambridge, Massachusetts, 1974.

9) Burkhart, P. D. and Bishop, R. H., "Adaptive Orbit Determination for Interplanetary Spacecraft," AIAA Journal of Guidance, Control, and Dynamics, Reston, Virginia, sched. for publication May-June 1996.