"SHOE-BOX" ORBIT DETERMINATION SYSTEM FOR SMM PRELIMINARY RESULTS

K. K. Tasaki—NASA Goddard Space Flight Center
and
C. Goorevich—Computer Sciences Corporation

ABSTRACT

Currently, most operational orbit determination functions are performed on large main-frame computers. It has already been demonstrated,that, at least for non-real-time orbit determination, a minicomputer can adequately provide the necessary computing power. The question at hand is: What about the use of microprocessors for operational orbit determination support?

During the past year, a study has been conducted to answer this question. The study involves the implementation of both sequential and batch methods of estimation on National Semiconductor IMP-16 microprocessors. The study used simulated data from a Tracking and Data Relay Satellite (TDRS) whose target satellite was the first Multimission Modular Spacecraft (MMS), the Solar Maximum Mission (SMM). An interesting feature of the hardware was the use of two interconnected ("Shoe-Box") IMP-16's. Some preliminary results from the study, as well as the difficulties and advantages in the use of microprocessors, are presented here.

Introduction

The existing operational orbit determination systems, such as the Goddard Trajectory Determination System (GTDS) and Goddard Real-Time System (GRTS), normally run on a large computer system, (i.e., IBM 360/95 or 75). Because these software systems are rather generalized, multi-purpose systems, and are capable of solving a great variety of orbit-related problems, they require a considerable amount of hardware resources. However, for periodic orbit updates of a given class of satellites (i.e., a given set of orbit characteristics), only a small portion of the software system is utilized. For example, the force models for the orbit propagator do not change from one update to another; and the observation types are limited to half-dozen or less. Thus, the parts of the system which are exercised do not change from run to run for a given type of orbit. With this in mind, the primary goal of the current project was set to develop a microprocessor-based orbit determination system for a particular satellite, namely the Solar Maximum Mission (SMM). Limiting the scope of the software system to always solve for a fixed set of parameters using only two types of observations, the complexity of the software, as well as the amount of code generated, could be minimized. If such a specialized system could be successfully built and tested, then the applicability of microprocessors for orbit determination will have been demonstrated.

In October 1977, the project was initiated. During the first 3 months, two pieces of software were designed: the orbit determination

system (ODS) and the data management system (DMS). ODS was designed based on the requirement of maintaining a 300-meter position accuracy using a sequential estimator. In a 574 km circular orbit inclined 33^{o} to the equator with a period of 96 minutes, the satellite was assumed to be tracked 20 minutes per orbit for 14 revolutions per day with the Tracking and Data Relay Satellite (TDRS). DMS, on the other hand, was designed primarily to manage observation data passed to ODS and updated positions and velocities received from ODS. In addition, DMS had to handle user input from the keyboard of a terminal, and had to provide output to the user terminal. Soon, it became apparent that ODS and DMS functions were separate, and should be targeted to two different processors.

At about the same time, three pieces of hardware were purchased, one printer/keyboard terminal (Silent 700 Model 765) from Texas Instruments and two IMP-16 microprocessors from National Semiconductor. The two IMP-16's were then turned over to Goddard's Microprocessor Development Facility for assembly and testing.

By the beginning of 1978, the software design was completed, and the implementation was initiated. During recent weeks, the software to handle the processor-to-processor and processor-to-terminal communications has been tested. Some of the well-tested code is now being moved to the target processors. The major effort here is to store the programs onto programmable read-only memory chips (PROM chips), and physically install them in the target processors for further testing. It is expected that most of the software will be on PROM's by the very early spring of 1979 for full-scale system testing.

Models and Accuracies

Choice of the models was determined by the original problem statement and the numerical accuracies of the IMP software floating point package. Design was carried out for a select class of orbit, namely the SMM orbit.

A software floating point package was used to do the decimal arithmetic. The package uses three 16-bit words to represent a real number. Two words represent a signed mantissa while one complete word is used for the exponent. This results in approximately 10 significant digits for the IMP real numbers as compared to 7 digits for IBM S/360 single precession and 17 digits for IBM S/360 double precession. Use of a software floating point package as compared to performing scaled arithmetic speeds up the development time of the software while significantly slowing down the actual run time. In the problem being solved, our only criterion was to keep up with real-time (process one observation every 10 seconds). Running the orbit propagator over one period (96 minutes) with a step size small enough to maintain a 300 meter rss accuracy takes 10 minutes, indicating that use of the software package can be justified in the lifetime cost of the software. If more significant digits are required this may not be true, and solutions such as performing scaled arithmetic may be required in selected portions of the code.

To determine the most appropriate models and algorithms for the orbit determination problem at hand, studies were carried out prior to and during the first stages of system design. Different force

models, integrations, and step sizes were tested using IBM S/360-95 FORTRAN simulations of the possible algorithms. The algorithms were evaluated as to complexity, size, speed, and accuracy. Most calculations were done in single-precision floating point, with those calculations needing double-precision accuracy being identified. The FORTRAN simulation programs developed in this pre-design phase were not only important tools in evaluation of the algorithms, but proved to be an aid in the coding and implementation phase. Studies concerning the system software and hardware capabilities of the IMP-16 microprocessor were also carried out in the pre-design phase.

The models used in the "shoe-box" orbit determination system are: orbit propagation model and state transition matrix.

The orbit propagation model will be used to propagate the target satellite (SMM) orbit between observation updates. This period is expected to last no longer than 96 minutes. The components which make up the propagator are as follows:

- Fourth-order Runge-Kutta integrator, with modified Fehlberg coefficients
- Up to 6 x 6 Earth Geopotential model including central earth
- Modified Harris-Priester Drag mode (assuming spherical space-craft)

Integration is performed using the Cowell technique of integrating the cartesian coordinates of acceleration and velocity.

The state transition matrix (required to propagate the covariance matrix of the extended Kalman filter) is a truncated Tayler series and includes the following terms:

- Acceleration partials with respect to position central Earth and J_2
- Acceleration partials with respect to velocity assumed to be very small
- Approximated to terms of order Δt^2

Observation Model

The observation model takes at a given observation time, the current "best" estimate of the target satellite's state (SMM), the known relay satellite's state (ATS-6, TDRS), and the station transmitter position and velocity to calculate the expected observations of satellite-to-satellite range sum (km) and the range sum rate (cycles of Doppler frequency counted). ATS-6 satellite-to-satellite tracking format was used instead of TDRS (as planned) because of the availability of simulated and real observations. The model assumes all signal paths (between station, relay satellite, and target satellite) are straight lines traversed at a constant speed of light. The calculations are performed by backwards tracing of the light path. The light-time corrections require both iteration and propagation of satellite states for times less than one second. A Newton-Raphson method is used as the iteration technique, while a second-order Euler propagator, modeling only the central Earth, is used for the small orbit corrections necessary during the iteration. The model accounts for a single relay satellite and models the nondestruct mode of frequency counting. Along with the model observations, partial derivatives of range and range-rate observations with respect to the target satellite's state are calculated.

State Estimation Algorithm

The state estimation algorithm updates the target satellite's position and velocity (satellite's state) based on errors between the observed observation and the calculation observation. Two approaches have been taken, one is a batch least squares estimator and the other is the extended Kalman filter. The batch least squares estimator is currently being implemented in its standard form. A small modification to the standard Kalman filter implementation was made for computational savings. The modification allows processing of a range and range-rate observation pair before making a state update. The matrix operations were performed in as straight-forward a manner as possible while eliminating as many trivial operations (such as multiplying and addition of zero's). The microprocessor code and the pre-design FORTRAN code were compared to results from the Goddard Trajectory Determination System (GTDS). The GTDS runs contained higher order terms and were assumed to represent the real world. The worst case results showed that the lack of significant digits did not greatly affect the orbit propagator over one period but did have a larger than expected affect on the observation model. The difference has yet to be resolved. The following table presents the errors due to having less significant digits in the IMP.

	Microprocessor	Pre-Design FORTRAN
Propagator Position Error Meters rss	116	32
Observation range sum error km	.9	.24

	Microprocessor	Pre-Design FORTRAN*
Observation range sum rate error cycles	6	.4

*Single precision (7 significant bits) used whenever possible.

Software/Hardware

The Shoe-Box system is composed of two IMP-16 microprocessors. The IMP is a 16-bit microprocessor with extended arithmetic to handle two 16-bit words. Typical memory fetches are 5.5 times slower than the IBM S/360-95. An add of a 32-bit integer takes 19 microseconds.

Processor One is called the data base IMP and contains 5K words of random access memory (RAM) and 28K words of erasable programmable read-only memory (EPROM). The data and calculated values are stored in the RAM while non-changing program code is stored in the EPROM. Processor Two, called the computational IMP, contains 9K words of RAM and 56K words of EPROM. A TI Silent 700 Model 765 is used as the system input/output device. If the system was modified to accept raw tracking data, the terminal would be replaced by an input cpu connected to the data lines.

Keeping in mind that the system uses satellite-to-satellite tracking data, the data flow for the extended Kalman filter is as follows:

1. Observation and TDRS ephemeris data are fetched from the bubble memory of the terminal to the data base IMP.

2. Upon request from the computational IMP, a data point composed of observation time, range sum observation, range sum rate observation,

and tracking satellite position and velocity is moved to the computational IMP.

3. The computational IMP propagates the target satellite's orbit to the observation time and performs the orbit update process. The corrected target orbit is then sent through the data base IMP to the terminal as an output report.

The software was developed using a higher level language SM/PL. The language is a structured procedure oriented language. Part of the development effort was to investigate the use of a higher level language on a microprocessor. We found that the code generated was optimized in the sense of organizing the code to perform operations in registers thus reducing the number of memory fetches. Errors in the language processor documentation and the actual code generated did slow up the development process more than was originally anticipated. At times stepping through the code, an instruction at a time, was required to determine exactly what was happening. One major item lacking in the SM/PL language that caused many problems was the inability to declare separately compiled modules as external. Due to this limitation of the system software, external locations had to be resolved by hand, after various subprograms were compiled separately. This process, although appearing only to be a minor inconvenience, caused a great deal of difficulty and unexpected side-effects. Although unexpected time loss was spent in understanding the SM/PL language in what it can or cannot do, the overall time of system development was not longer than it would have been if assembler language would have been used. Future programs coded in

SM/PL should proceed faster.

The data base IMP was divided into the following five modules which perform the following functions:

1.) Initialization

2.) Messages to Computational IMP

3.) Decoding of teletype input

4.) Messages to teletype

5.) Respond to computation IMP request for next observation

The computational IMP was divided into the following five modules which perform the following functions:

1.) Initialization of data, GHA, and station model

2.) Observation retrieved

3.) Orbit propagation

4.) Observation model and partials

5.) Orbit estimation processes

Advantages and Disadvantages

From our experience of developing the microprocessor-based orbit determination, a number of advantages and disadvantages to using microprocessors for mathmatical applications can be pointed out. The greatest disadvantage, especially from the software developer's point of view, has been the lack of adequate system support software, such as a good compiler with floating point arithmetic which produces truly relocatable object code, and a reliable linker or loader which can correctly link-edit the

object code. Due to this lack of adequate support software, programmers must constantly worry about absolute addresses for referencing and debugging. The hardware could also be a potential problem. In most cases, one must assemble, test, and maintain various hardware components, because, unlike the minicomputer and main-frame world, there are usually no installation and/or maintenance services provided by the manufacturers or other electronic firms. The lack of relative speed and accuracy is still a problem in using microprocessors for mathematical applications. This is especially true in performing floating point calculations which are normally accomplished by software.

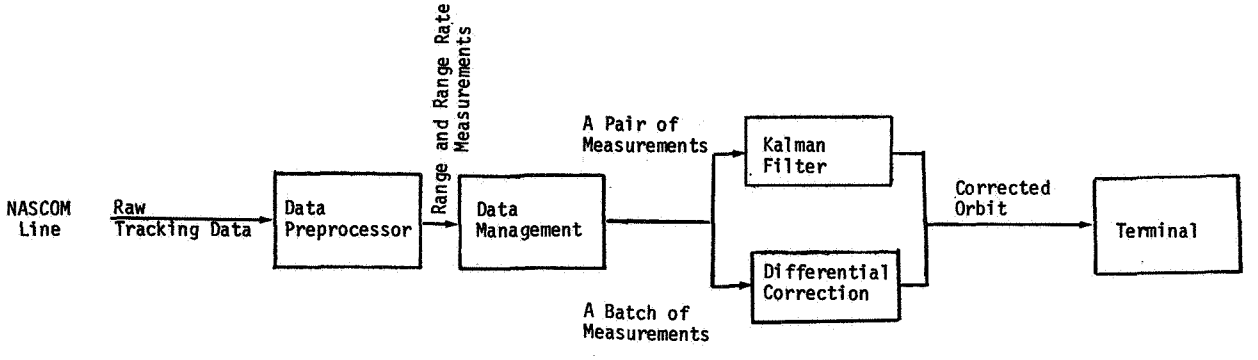
Of course, there are a number of advantages to using microprocessors. First of all, the hardware is very inexpensive compared to the cost of larger processors. The cost of the Shoe-Box system with two IMP-16's and enough memory to accommodate the data base and orbit determination software was about $3K. The second notable advantage is that the hardware/software system can be totally portable. For example, the shoe-box system can be taken to any site where SMM tracking data is available, and begin processing the data (provided that there exists an appropriate set of initial conditions). Another important advantage is modularity. A microprocessor-based system can be quickly upgraded to meet new project requirements which may be placed after the completion of the initial implementation. Due to modularity, this type of system upgrade can be accomplished either by adding more memory or adding more processors.

Future Considerations

A number of future considerations are now discussed in order of probable implementation. The first enhancement to the Shoe-Box system is to develop a data preprocessor and a data postprocessor. The objective of the preprocessor is to provide the capability to accept raw tracking data from the NASCOM line and to perform the necessary preprocessing functions before the data is passed to the data base IMP. This program is scheduled to be completed in March 1979. The data postprocessor, on the other hand, is envisioned as a set of interface routines which will make the computed orbit available to the user computer. Because these interface routines will have to be user processer-dependent, only a general design of the routines will be completed.

Another area of enhancement is reconfiguration of the Shoe-Box system so that a user will have the choice of executing either the Kalman filter program of the differential correction program, depending on the initial input. Each of these programs will reside on a separate IMP-16, and will be parallel to each other, as shown on the following diagram.

PLANNED SHOE-BOX HARDWARE CONFIGURATION



Another possible improvement to the Shoe-Box system may be to solve for both the target and the tracking satellites' position and velocity vectors simultaneously. It is currently believed that this will increase the accuracy of the target satellite's orbit.

Acknowledgements

Messrs. Charles Shenitz and Carl Rabbins of Computer Sciences Corporation have provided much of the software design and programming support for this project, whereas Messrs. Bill Holmes and Ed Zenker of Goddard's Microprocessor Development Facility have contributed a considerable amount of their time in the hardware area. Without these key persons, this project could not have been a success.

OBJECTIVES

1. DEMONSTRATE THE APPLICABILITY OF MICROPROCESSORS FOR ORBIT DETERMINATION
2. INVESTIGATE THE OPERATIONAL ASPECTS OF MICROPROCESSOR-BASED ORBIT DETERMINATION SYSTEM
3. DETERMINE SYSTEM DEVELOPMENT AND MAINTENANCE CHARACTERISTICS

Figure 1

SMM MICROPROCESSOR-BASED ORBIT DETERMINATION

PROBLEM:

MAINTAIN A 300 METER (POSITION-COMPONENT) ACCURACY AT ALL TIMES

GIVEN:

- A 547 KM CIRCULAR ORBIT
- 33° INCLINATION
- 96 MINUTE PERIOD
- 20 MINUTES OF SATELLITE-TO-SATELLITE TRACKING DATA PER ORBIT

Figure 2

DEVELOPMENT OF MICROPROCESSOR ORBIT DETERMINATION SYSTEM

- PURCHASE IMP SOFTWARE DEVELOPMENT SYSTEM
- SELECTION OF MATHEMATICAL MODELS AND ALGORITHMS
- DECISION ON HARDWARE CONFIGURATION
- SOFTWARE DESIGN
- PURCHASE AND ASSEMBLE HARDWARE ("SHOE-BOX")
- IMPLEMENT SOFTWARE ON DEVELOPMENT IMP
- TEST HARDWARE/SOFTWARE COMMUNICATION BETWEEN CPU's
- MOVE SOFTWARE ONTO TARGET SYSTEM
- SYSTEM TESTING

Figure 3

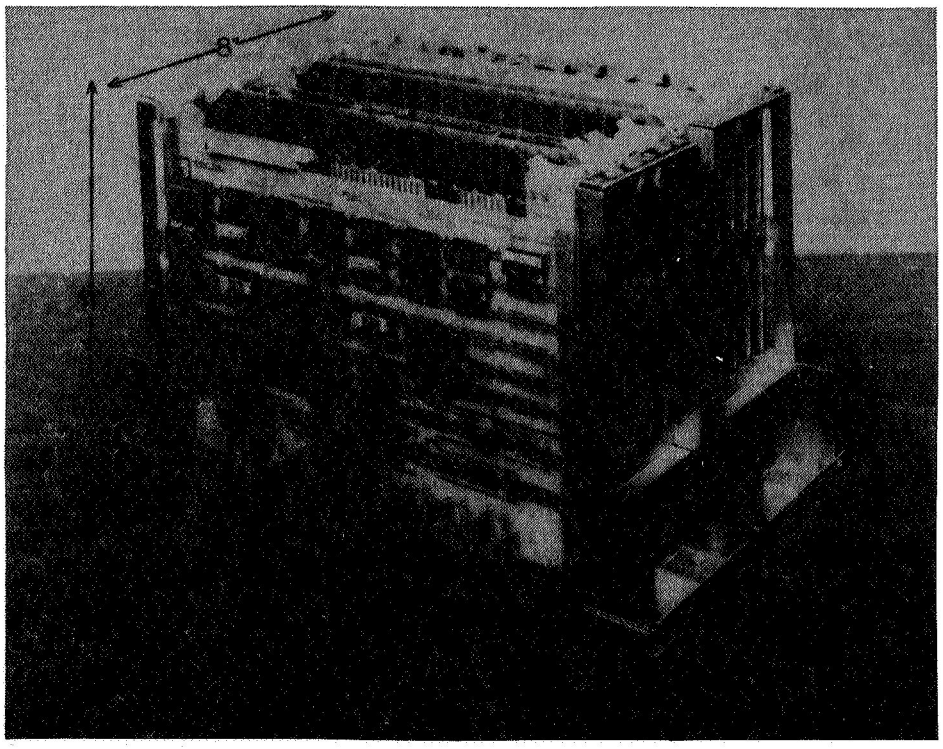


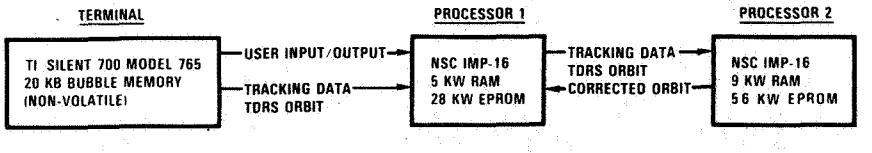
Figure 4

MICROPROCESSOR-BASED ORBIT DETERMINATION SYSTEM

DISADVANTAGES	ADVANTAGES
● LACK OF ADEQUATE SYSTEM SOFTWARE	● CHEAP HARDWARE
● MACHINE-LEVEL WORRIES	● SYSTEM PORTABILITY
● HARDWARE HEADACHES (IN GENERAL, BUT NOT IN OUR CASE)	● FLEXIBILITY FOR HARDWARE ADDITIONS
● LACK OF RELATIVE SPEED AND ACCURACY	● CHIP AND/OR BOARD LEVEL MAINTENANCE AND MODIFICATION

Figure 5

HARDWARE CONFIGURATION AND DATA FLOW



- STORAGE FOR TDRS POSITIONS AND VELOCITIES
- STORAGE FOR TRACKING DATA
- USER INTERFACE TO THE PROCESSORS

- TERMINAL COMMUNICATIONS
- DATA HANDLING

- ORBIT PROPAGATION
- SEQUENTIAL BATCH ESTIMATION

Figure 6

MATHEMATICAL MODELS

ORBIT PROPAGATOR

INTEGRATOR:

FEHLBERG RUNGE-KUTTA (4)

FORCE FIELD:

CENTRAL FORCE FIELD (EARTH)
UP TO 6×6 EARTH FIELD
HARRIS-PRIESTER ATMOSPHERIC DRAG

ORBIT ESTIMATOR

SOLVE FOR PARAMETERS:

TARGET SATELLITE POSITION AND VELOCITY

METHODS:

EXTENDED KALMAN FILTER
AND
BATCH LEAST SQUARES

OBSERVATION MODEL

OBSERVATIONS:

RANGE SUM
RANGE SUM RATE

MODEL:

ATS-6 (TDRS) SATELLITE-TO-SATELLITE TRACKING
LIGHT TIME CORRECTIONS TAKEN INTO ACCOUNT
ONLY ONE TRACKING SATELLITE VISIBLE AT ONE TIME

GROUND STATION

MODEL:

GEODETIC STATION MODEL

Figure 7

NUMERICAL ACCURACIES

- **SIGNIFICANT DIGITS**

IMP-16 FLOATING POINT PACKAGE	10 DIGITS
IBM FORTRAN IV	7 DIGITS FOR SINGLE PRECISION 17 DIGITS FOR DOUBLE PRECISION

- **GTDS COMPARISONS**

ORBIT PROPAGATOR

	IMP MICROPROCESSOR	IBM 360*
POSITION ERROR (METERS RMS)	91-116	32

OBSERVATION MODEL

	IMP MICROPROCESSOR	IBM 360*
RANGE SUM (KM)	.9	.24
RANGE SUM RATE CYCLES	6	.4

*SINGLE PRECISION USED WHENEVER POSSIBLE

GOO-10-78

Figure 8

BASELINE DIAGRAMS OF SOFTWARE

- SOFTWARE DEVELOPED USING HIGHER LEVEL LANGUAGE (SM/PL)
- FLOATING POINT PACKAGE USED FOR NON-INTEGER ARITHMETIC

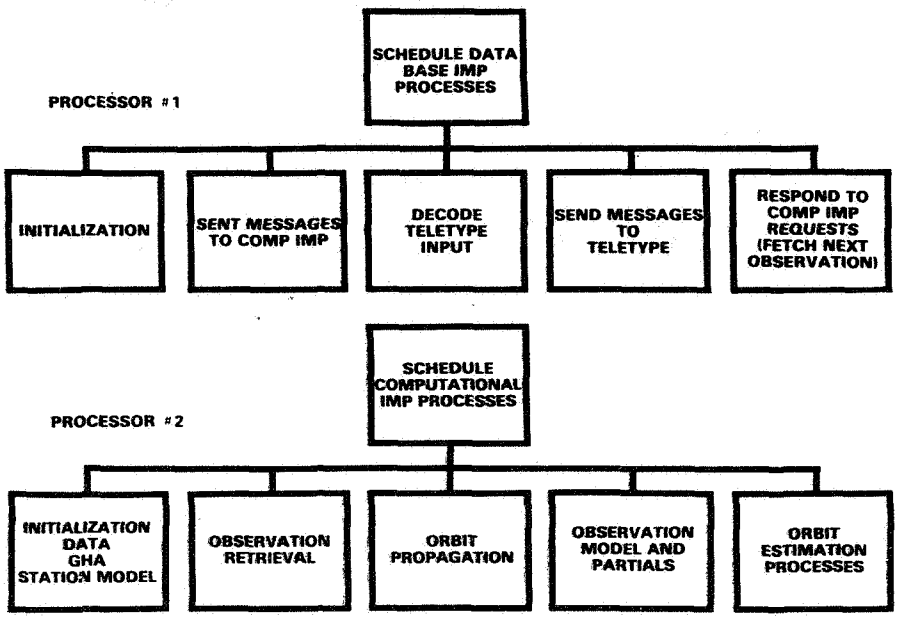


Figure 9

RESULTS OF MICROPROCESSOR SYSTEM DEVELOPMENT

- DEMONSTRATED USING SOPHISTICATED COMPUTATIONAL MODELS ON MICROPROCESSORS
- DEMONSTRATED USE OF HIGHER LEVEL LANGUAGE, BUT FURTHER WORK ON SYSTEM SOFTWARE IS NEEDED
- DEMONSTRATED PRACTICALITY OF USING SEPARATE MICROPROCESSORS (CPU's) TO PERFORM INDIVIDUAL FUNCTIONS
- SPEED AND MEMORY REQUIREMENTS FOR ORBIT DETERMINATION PROBLEM APPEARS NOT TO BE A PROBLEM; ACCURACY IS THE REAL PROBLEM

Figure 10

FUTURE CONSIDERATIONS

- **DATA PREPROCESSOR AND POSTPROCESSOR**

  ACCEPT RAW TRACKING DATA, DETERMINE ORBIT AND STORE THE RESULTS ON A USER PERIPHERAL DEVICE

- **HARDWARE RECONFIGURATION**

  USE BATCH AND SEQUENTIAL FILTER IN PARALLEL

- **GRARR TRACKING DATA PROCESSING**

  ACCEPT RANGE, RANGE RATE, AND ANGLE MEASUREMENTS FROM CONVENTIONAL TRACKING

- **SOLVE FOR BOTH TARGET AND TDRS ORBITS**

Figure 11