AUTOMATION OF ORBIT DETERMINATION FUNCTIONS FOR NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA)-SUPPORTED SATELLITE MISSIONS*

H. Mardirossian, K. Heuerman, A. Beri, and M. Samii
Computer Sciences Corporation (CSC)

C. E. Doll
Goddard Space Flight Center (GSFC)

ABSTRACT

The Flight Dynamics Facility (FDF) at Goddard Space Flight Center (GSFC) provides spacecraft trajectory determination for a wide variety of National Aeronautics and Space Administration (NASA)-supported satellite missions, using the Tracking Data Relay Satellite System (TDRSS) and Ground Spaceflight and Tracking Data Network (GSTDN). To take advantage of computerized decisionmaking processes that can be used in spacecraft navigation, the Orbit Determination Automation System (ODAS) was designed, developed, and implemented as a prototype system to automate orbit determination (OD) and orbit quality assurance (QA) functions performed by orbit operations. Based on a machine-resident generic schedule and predetermined mission-dependent QA criteria, ODAS autonomously activates an interface with the existing trajectory determination system using a batch least-squares differential correction algorithm to perform the basic OD functions. The computational parameters determined during the OD are processed to make computerized decisions regarding QA, and a controlled recovery process is activated when the criteria are not satisfied. The complete cycle is autonomous and continuous.

ODAS has been extensively tested for performance under conditions resembling actual operational conditions and found to be effective and reliable for extended autonomous OD. Details of the system structure and function are discussed, and test results are presented.

*This work was supported by the National Aeronautics and Space Administration (NASA)/Goddard Space Flight Center (GSFC), Greenbelt, Maryland, under Contract NAS 5-31500.

1. INTRODUCTION

Operational orbit support for many current National Aeronautics and Space Administration (NASA) missions involves a well-defined sequence of activities leading up to the generation and transmission of estimated dynamic states and definitive and predictive ephemerides for supported spacecraft. These activities can be separated into three stages: tracking data preprocessing (TDP), orbit determination (OD), and orbit product generation and transmission (OPGT). Figure 1 provides an overview of this type of orbit support at the Goddard Space Flight Center (GSFC). Only the OD stage is described in detail. The TDP and OPGT stages are included to show their relationships with the OD stage. This paper presents the Orbit Determination Automation System (ODAS), which is designed to automate activities involved in the OD stage (References 1 and 2). Automation is achieved in ODAS through replacement of functions that are normally performed by an analyst. A brief description of the functions involved in OD is useful in understanding the nature of the automation processes in ODAS and is provided below.

Current trajectory determination systems process tracking measurements and use them in conjunction with parameterized dynamic models to update the estimate of the dynamic states of supported spacecraft (References 3 and 4). As a specific example, the Goddard Trajectory Determination System (GTDS), employed regularly at GSFC, employs a differential correction (DC) algorithm to fit the tracking measurements to the models and estimate a solution state for the spacecraft orbit. The estimated solution state is used to generate trajectories and other orbit-related products. In Figure 1, the orbit maintenance schedule serves to provide information about when spacecraft are due for OD; the tracking data base represents the collection of tracking measurements, a subset of which is used for OD. The operational OD at GSFC involves the following steps, some of which require intervention by the analyst:

Step I. The analyst scans the orbit maintenance schedule at regular intervals to determine if an orbit update is scheduled for a spacecraft at a time close to the time of the scan.

Step II. The analyst appraises the tracking measurements for the particular spacecraft in the tracking data base to establish sufficiency in quantity and distribution.

Step III. The analyst determines initial parameters to be supplied to the trajectory determination system as control and data information and incorporates the values into OD control/input data sets (CIDS), which the analyst retrieves from the control and input parameters data base.

Step IV. The resulting set of OD processing commands sets up the trajectory determination system in a specific processing mode. The analyst initiates orbit estimation in this processing mode.

Steps V and VI. If the estimation process converges, a solution state for the spacecraft is generated by the computational system. The analyst examines the computed results, including state vectors, other estimated parameters, and ephemerides.

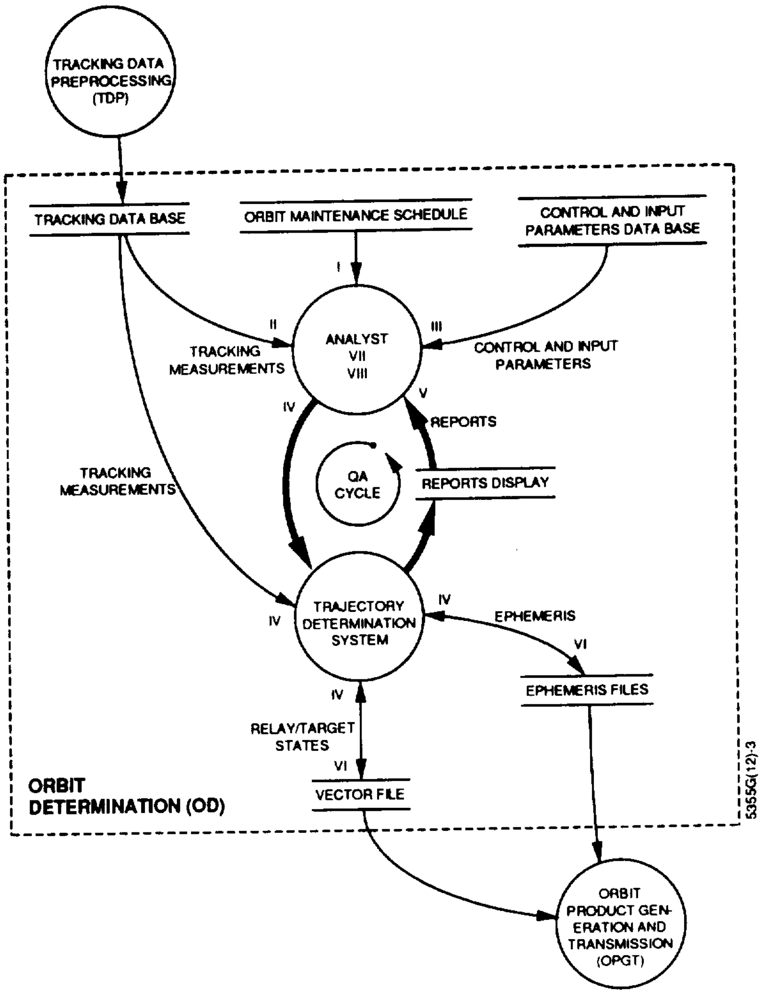

Figure 1. Orbit Determination Activities

Step VII. The analyst determines the pass/fail condition of the estimation result on the basis of a comparison of specific quality assurance (QA) parameters available from reports generated by the trajectory determination system with mission-dependent QA criteria. This is the QA failure detection process.

Step VIII. In case of QA failure, the analyst determines specific changes in processing modes that might lead to improved orbit estimation. This is the QA failure recovery initiation process.

The analyst modifies the processing modes in which the trajectory determination system is set up and repeats the OD and QA operations described in steps II through VIII as often as necessary to generate a satisfactory solution. This represents OD QA through a cyclical recovery process represented by the bold circle in Figure 1.

Since automation of the OD process potentially provides benefits—such as reduced analyst intervention, reduced demand on system resources, improved operational flexibility, improved reliability, automatic accumulation of historical experience, and a ready source of operational and analytical training—ODAS was developed to create an autonomous analog of the processing environment depicted in Figure 1. Some of the OD steps have been previously automated by systems such as the Orbit Production Automation System (OPAS), which incorporates aspects of steps I, III, and IV (Reference 5), and the Automatic Orbit Determination IV (AOD-IV) system (Reference 6), which specializes in applications of step III. However, these systems require analyst intervention for all other phases of the OD process. The ability of ODAS to sustain autonomous operation of the entire OD process indefinitely without any analyst intervention is the system's primary distinguishing feature.

The remainder of this paper concerns the functional and structural aspects of ODAS. The specific prototype described in this paper was developed within the GTDS environment. In Section 2, primary functions of ODAS that provide autonomous analogs of the steps in Figure 1 are discussed. In Section 3, the structural configuration and coordination of the primary ODAS functions in performing the overall OD process is described. In Section 4, selected system tests are described and the test results presented to illustrate some of the operational aspects of the system. In Section 5, the significant conclusions resulting from this prototyping study are summarized, and directions for future enhancements are discussed.

2. ODAS FUNCTIONS

The functional objective of ODAS is to provide an autonomous analog of the overall OD process represented by the boxed area in Figure 1. The eight-step OD process has been described in Section 1. The functional design of ODAS consists of logical functions that accomplish tasks corresponding to each of the eight steps in the proper sequence without any analyst intervention. Additional logical functions in ODAS provide the capability to perform this autonomous OD continuously for an indefinite period. These functions are discussed in the remainder of this section.

Table 1 lists all the primary ODAS functions and establishes a mapping between each ODAS function and an operational step. Each ODAS function is briefly described.

OD Update Scheduling. This function schedules spacecraft for OD. ODAS makes periodic queries of a generic scheduling data (GSD) file for information related to the update frequency and processing parameters. OD updates are then performed according to these specifications.

Tracking Data Sufficiency Checking. The DC process of GTDS operates on tracking measurements from a chosen period denoted as the "data arc" (step IV of Figure 1). Typically in step II, the analyst considers the number of distinct trackers, the number of tracking data batches, and the presence and disposition of large periods containing no measurements (gaps) in qualifying the measurement set as sufficient or insufficient for achieving a reliable OD solution. In case of insufficiency, the analyst can extend the data arc further back in time (arc retrocession) to access more data or "better" data. In ODAS, the data arc is specified generically in the GSD file and is converted into a specific data arc.

Automated data tracking sufficiency checking and arc retrocession capabilities analogous to step II are present in ODAS.

Table 1. ODAS Functions and Corresponding Routine Operational Orbit Determination Steps

ODAS FUNCTION	ROUTINE OD OPERATION*
OD UPDATE SCHEDULING	I
TRACKING DATA SUFFICIENCY CHECKING	II
CIDS CREATION	III
CIDS SUBMISSION	IV
DC/STATISTICAL OUTPUT REPORT (SOR) EXTRACTION	V
DC FAILURE DETECTION	VII
DC FAILURE RECOVERY	VIII, III, IV, V
EPHEMERIS QA	VI
TIME CONTROL	NR
PROCESSING SUSPENSION/RESUMPTION	NR
SYSTEM STATUS REPORTING	NR

*NR: NOT REPRESENTED IN FIGURE 1.

CIDS Creation. In close analogy to step III, the CIDS creation function of ODAS retrieves a skeleton CIDS, which represents a specific GTDS processing mode from the CIDS file and incorporates processing information from the GSD file into the data set.

CIDS Submission. This function submits the CIDS to the processing queue of a host computer system to initiate the corresponding DC process. The CIDS submission step is the automated version of step IV.

DC/SOR Extraction. This function extracts certain parameters, which includes the parameters specified in Table 2, from the DC/SOR output reports for analysis. For several of the GSFC-supported spacecraft, acceptable limits are specified for these parameters (Reference 7). In ODAS, several additional quantities are included in the DC/SOR subset because of their potential values in DC recovery in case of DC failure. The DC/SOR extraction is analogous to step V.

DC Failure Detection. This function determines whether DC was successful or failed established QA criteria. The QA parameters are retrieved from the DC/SOR subset and are compared with predetermined limits/tolerances from a user-defined QA criteria file. The current design of ODAS recognizes a fixed set of seven DC failures listed in Table 2. This function of ODAS corresponds to step VII in Figure 1.

DC Failure Recovery. The last step in the overall OD process is step VIII, for which the analyst decides whether to repeat the estimation under different processing conditions if a DC failure is detected. The analyst may implement one or more recovery procedures,

Table 2. ODAS DC Failures

ODAS NAME	DC FAILURE TYPE
F1	NONCONVERGENT DC
F2	FINAL WEIGHTED ROOT MEAN SQUARE (WRMS) OF OBSERVATION RESIDUALS EXCEEDS CRITERION
F3	ESTIMATED ATMOSPHERIC DRAG SCALING PARAMETER (ϱ_1) OUTSIDE NOMINAL RANGE
F4	ESTIMATED SOLAR RADIATION PRESSURE SCALING PARAMETER (C_R) OUTSIDE NOMINAL RANGE
F5	STANDARD DEVIATION OF RANGE OBSERVATION RESIDUALS (σ_R) EXCEEDS CRITERION
F6	STANDARD DEVIATION OF RANGE-RATE/DOPPLER OBSERVATION RESIDUALS ($\sigma_{\dot{R}/D}$) EXCEEDS CRITERION
F7	ESTIMATED ABSOLUTE POSITION ERROR IN A PRIORI STATE (ΔR) EXCEEDS CRITERION

involving repeating the estimation under different processing conditions. Typical examples of recovery procedures are using a different selection of batches of tracking data, a different range of values for the atmospheric density, or a different convergence criterion. The choice is dictated by the type of DC failure detected. The overall failure recovery process may involve more than one recovery procedure. This process is automated in ODAS by the DC Failure Recovery function. Currently, ODAS provides five distinct recovery procedures, which are listed in Table 3. In general, each recovery procedure can generate a different set of failed criteria and different magnitudes of departures from the criteria. ODAS computes a weighted sum of the magnitudes to use as an average indicator of the overall degree of failure and implements recommended recovery procedures in an attempt to reduce this indicator to zero. In addition, the overall recovery process is controlled through limits on the maximum number of recovery attempts and the minimum relative improvement in the indicator.

Table 3. Procedures Employed in ODAS to Attempt Recovery From DC QA Failure

ODAS NAME	RECOVERY PROCEDURE
P1	CHANGE HARRIS-PRIESTER DENSITY TABLE
P2	EXTEND DATA ARC BACKWARD
P3	ELIMINATE BIASED OR NOISY BATCHES
P4	INCREASE INITIAL WRMS
P5	USE FINAL ELEMENTS AS INPUT

Ephemeris QA. Operationally, OD consistency is measured through a point-by-point comparison of adjacent overlapping ephemerides. The magnitude of the maximum difference, $|\Delta R_{ephem}|$, relative to a tolerance specified in Reference 7 provides the required measure of OD consistency. This function is included in step V of Figure 1 and is performed in ODAS using an algorithm that involves extraction of the ephemeris comparison results from the GTDS reports and the tolerances from the criteria file.

Time Control. The time control function of ODAS is a device for biasing and scaling the time variable, with respect to actual clock time, to allow the autonomous operations of ODAS to be performed for arbitrary times (past and present) and to proceed at accelerated schedules. This function is not a replication of any single step in Figure 1 because routine operational OD is performed in real time.

Processing Suspension/Resumption. To provide continuous operation for indefinite periods of time, portions of ODAS must be active continuously. The current operational OD support, on the other hand, involves periods (for tracking measurement accumulation, the TDP stage of Figure 1) during which OD is not actively performed. A capability is devised to detect the onset and duration of such a processing mode, suspend activities for the required period, release resources, and resume processing automatically at the end of the suspension period. A short suspension mode is also provided to handle lull periods during a series of clustered OD updates.

System Status Reporting. ODAS generates status reports both for transmitting results between ODAS components and for informing the analyst who monitors the automated OD operation. The chronological progress reports data and archival capabilities can be utilized to support operational analysis.

The functions described above represent the primary building blocks of an OD automation system. The ODAS prototype discussed in this paper is constructed with the specific requirements of the GSFC FDF environment in mind, such as compatibility with the global trajectory computation and orbital products support system. The construction is embodied in a particular configuration of the functions as components of larger units, namely subsystems of ODAS, and of the sequential and hierarchical relationships among the subsystems. This specific configuration of ODAS is the subject of Section 3.

3. ODAS CONFIGURATION

The primary ODAS functions described in Section 2 can be designed in several ways, depending on the host computational system and operational/development requirements. At GSFC, operational orbit support is based on background batch processing with programs that are available in GTDS. "Batch" here is the computational term describing a noninteractive processing of complete, predefined jobs and must be distinguished from the batch estimation technique in OD, referred to in Sections 1 and 2. The requirement for prototype development that greatly influenced the design of the primary ODAS functions in the current version of ODAS was the use of GTDS components as black boxes: no modifications were made to the GTDS programs. The resulting configuration, shown in Figure 2, incorporates the DC, Ephemeris Generation (EPHEM), and Ephemeris

Comparison (COMPARE) programs of GTDS without modifications. One of the prominent features of ODAS, namely the presence of numerous interface input/output (I/O) files, is a direct consequence of this requirement.

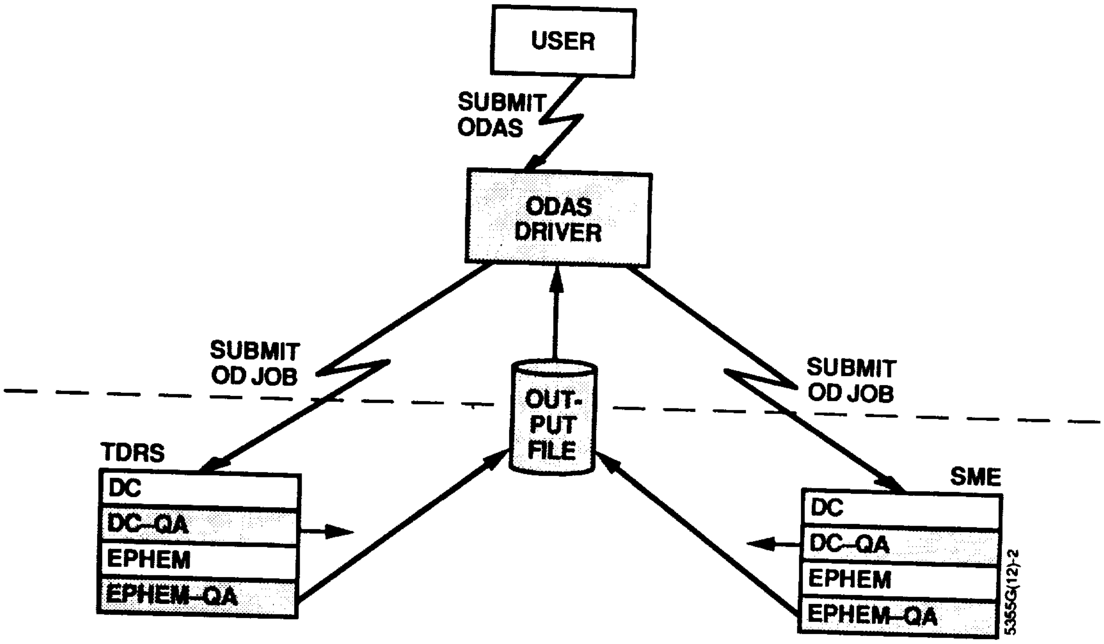

Figure 2. ODAS Configuration

The design of the ODAS prototype consists of the following five coordinated subsystems and the interfaces between the subsystems as defined in Figure 2:

- The ODAS Driver subsystem
- The DC subsystem
- The DC QA subsystem
- The EPHEM subsystem
- The EPHEM QA subsystem

The user initiates ODAS by submitting the ODAS Driver subsystem, which executes indefinitely until terminated by the user. The ODAS Driver periodically submits a series of OD jobs, each consisting of the other four subsystems, for all spacecraft scheduled for OD update. Figure 2 represents a typical situation in which the ODAS Driver has submitted two OD jobs, one for the Tracking and Data Relay Satellite (TDRS) and another for the Solar Mesopheric Explorer (SME). The DC QA subsystem analyzes the DC results and, in the case of OD failures, communicates the result to the interface output file and terminates execution. In the case of successful DC, the EPHEM and EPHEM QA subsystems are executed, and the result is communicated to the interface output file. The ODAS

Driver monitors the output file to determine the status of the OD process and make decisions affecting subsequent processing. The remainder of this section describes the logical functions being performed within each subsystem.

ODAS Driver Subsystem. The ODAS Driver is responsible for the overall initiation and control of the computational functions within ODAS. It resides permanently on the host system and is designed to be in perpetual execution, monitoring all automation functions, and effectively synchronizing tasks. The ODAS Driver is functionally separated into five major components:

- The scheduler component
- The tracking data sufficiency checking component
- The job submission component
- The timer component
- The suspension/resumption component

After scheduling spacecraft for OD for the day, the Driver suspends its activities, resuming at scheduled OD times for each spacecraft. It then checks the tracking data for sufficiency; if the data do not meet the user-defined criteria for sufficiency, the Driver extends the data arc backward in time to obtain additional data. If arc extension still does not meet the criteria, the Driver stops processing that spacecraft for that day. If the sufficiency checking passes, the Driver then prepares CIDS for that spacecraft and submits jobs involving the four ODAS subsystems for OD and QA processing. After submission of OD jobs, the Driver periodically checks the output file for the results of OD. If the OD results indicate a DC failure and a directive to reexecute the DC to recover from the failure, the Driver prepares new CIDS and resubmits the four ODAS subsystems for another round of OD processing.

DC QA Subsystem. The DC QA subsystem performs quality assurance of DC results and consists of four components. The DC/SOR subset extraction component extracts subsets of parameters from the DC reports and the SOR analysis. The failure detection component diagnoses specific DC failures, based on computational parameters generated by the DC processing in the SOR. It determines whether a DC solution has met the spacecraft acceptance criteria determined by the user. In the absence of DC failure, the DC QA subsystem terminates and initiates processing of the EPHEM subsystem. In the presence of DC failure, the DC QA subsystem activates the recovery component. For a specific DC failure, the recovery component invokes a corresponding recovery procedure, which translates into system control modifications that have been prescribed by expert analysts. The results transmission component transmits the decisionmaking information from the DC QA subsystem to the ODAS Driver subsystem.

EPHEM QA Subsystem. The EPHEM QA subsystem performs QA on the results from the EPHEM subsystem by comparing the maximum difference between the previous definitive ephemeris and the currently computed definitive ephemeris. The EPHEM QA subsystem consists of three components. The compare extraction component takes the computational parameters for a spectrum of different solutions between two sequential ephemerides and provides the data to the EPHEM QA subsystem for further analysis. The failure detection component determines if ephemeris comparison results meet specific mission-dependent requirements. The results transmission component transmits the decisionmaking information from the EPHEM QA subsystem to the ODAS Driver subsystem.

Certain characteristics of the configuration are crucial to reliable, continual operation of ODAS. The logical separation of the individual OD jobs from the Driver, for example, ensures that problems arising during the DC and post-DC processing will not affect any ODAS Driver functions, thereby enabling processing of the remainder of the OD jobs to proceed normally. The maintenance of a single interface (see Output File in Figure 2) between the ODAS Driver and all OD jobs allows efficient monitoring, coordinating, and scheduling by the ODAS Driver. Of great significance is the generic table-driven nature of the DC failure detection and recovery components, which allows convenient modification of the actual choices of recovery algorithms to be associated with particular DC failures. In this area, ODAS requires continuous evolutionary enhancements, as indeed does any mode of operational processing requiring complex decisionmaking regarding options for improving OD.

4. TEST RESULTS AND DISCUSSION

The presence of scheduling, time scaling, and suspension/resumption functions in ODAS allows extensive system and performance testing in a relatively short time (Reference 8). All tests involve the automated analog of the OD process of Figure 1. Many tests address additional ODAS-unique functions, such as extraction of a preselected set of parameters from the DC/SOR characterizing the measurements and the measurement residuals. The tests were performed using an ODAS prototype implemented in VS FORTRAN on an IBM-compatible host computer.

Test results presented in this section are grouped by individual ODAS functions. Only selected tests that typify test categories are presented in this section. Eight spacecraft were used in testing ODAS:

- TDRS-E
- SME
- Solar Maximum Mission (SMM)
- Landsat-4
- Landsat-5
- Meteorological Observation Satellite (NIMBUS-7)
- Earth Radiation Budget Satellite (ERBS)
- Dynamics Explorer (DE)-A

One or more tests were performed in the course of OD for each of the spacecraft, as shown in test case matrix entries for each function. Test result summaries are presented in the remainder of this section.

Initiation. The objective of testing the initiation function was to check the validity of ODAS response with respect to different system start parameters (see Table 4). The tests consisted of initiating ODAS as a cold start (first initiation of ODAS), terminating its processing, and reinitiating it as a warm start (subsequent initiations of ODAS within the same day) and varying speed ratio as compared to clock time.

Table 4. Test Matrix for the Initiation Function of ODAS

INITIATION SUBFUNCTIONS	TDRS	SME	SMM	Landsat-4	Landsat-5	NIMBUS-7	ERBS	DE-A
COLD START ODAS	●	●	●	●	●	●	●	●
WARM START ODAS	●	●	●	●	●	●	●	●
SPEED RATIO	●	●	●	●	●	●	●	●

ODAS was initiated as a cold start with all eight spacecraft scheduled for OD. After performing OD for TDRS, execution was halted by the user for a short period and then reinitiated as a warm start. ODAS resumed processing activities that were continuations of those interrupted at the time its operation was halted. ODAS was also initiated using a speed ratio of six (six times faster than normal clock time), where a 24-hour cycle was compressed into 4 hours of real clock time. All tests were successful.

Scheduler. The goal of testing of the scheduler function was to confirm the ODAS scheduling ability under various conditions (see Table 5). This included transforming the generic schedule of spacecraft provided by the user through the GSD to a specific schedule for a given day of ODAS operation.

Using a generic schedule for all spacecraft, ODAS created a specific schedule for the current test day (September 11, 1987) and the next day for all spacecraft. Additionally, OD for DE-A was rescheduled for September 11, 1987, at 02 hours according to the GSD entry. ODAS successfully scheduled DE-A for 2 a.m. on the current test day. The timer component accurately kept track of the ODAS time, and the suspension/resumption component suspended ODAS activities and resumed at scheduled spacecraft OD times. All tests were successfully performed.

Tracking Data Sufficiency Checking. The goal of testing the tracking data sufficiency checking function of ODAS was to verify that ODAS would only perform OD when sufficient tracking data were in the 60-byte metric tracking data base for a given data arc. As shown

Table 5. Test Matrix for the Scheduler Function of ODAS

SCHEDULER SUBFUNCTIONS	TDRS	SME	SMM	Landsat-4	Landsat-5	NIMBUS-7	ERBS	DE-A
SCHEDULE SPACE-CRAFT	●	●	●	●	●	●	●	●
RESCHEDULE SPACE-CRAFT								●
TIMER COMPONENT	●	●	●	●	●	●	●	●
SUSPEND/RESUME ODAS	●	●	●	●	●	●	●	●

in Table 6, seven subfunctions were tested, based on the specific parameters defining sufficiency. The parameters are

- Number of distinct trackers
- Total number of tracking data batches
- The largest gap in data
- The total number of observations

Table 6. Test Matrix for the Tracking Data Sufficiency Checking Function of ODAS

TRACKING DATA SUFFICIENCY CHECKING SUBFUNCTIONS	TDRS	SME	SMM	Landsat-4	Landsat-5	NIMBUS-7	ERBS	DE-A
SUFFICIENCY CHECK PASSED	●	●	●	●	●	●	●	●
INSUFFICIENT TRACKERS					●			
INSUFFICIENT BATCHES				●				
DATA GAP TOO LARGE	●			●				
INSUFFICIENT OBSERVATIONS					●			
ARC RETROCESSION FAILURE			●					
NO TRACKING DATA							●	●

In addition, the case of an unsuccessful attempt at tracking data improvement using arc retrocession was also tested.

To test the case of insufficient trackers, the criterion for minimum number of distinct trackers for Landsat-5 was set to four. The tracking data contained only three distinct trackers, which did not satisfy the criterion. ODAS extended the data arc back to obtain an extra distinct tracker to meet the criterion, then continued with the processing of OD for Landsat-5. To test the case of insufficient batches, the criterion for the minimum acceptable number of batches was set to 14 for Landsat-4. The data contained only 13 batches for the given arc. ODAS extended the data arc back to obtain an extra batch to meet the criterion. To test the case of no tracking data, a 60-byte metric tracking data base that contained no tracking data for ERBS and DE-A spacecraft was chosen. The sufficiency tracking function detected this, generated warning messages, and aborted OD processing for DE-A and ERBS.

Job Submission. The goal of testing the job submission function was to verify that ODAS did possess the ability to set up and submit GTDS DC, EPHEM, and their associated QA subsystems. Testing the job submission function of ODAS involved checking the accuracy of job control language (JCL) and input for all the jobs submitted (see Table 7).

Table 7. Test Matrix for the Job Submission Function of ODAS

JOB SUBMISSION SUBFUNCTIONS	TDRS	SME	SMM	Landsat-4	Landsat-5	NIMBUS-7	ERBS	DE-A
CIDS MODIFICATION	●	●	●	●	●	●	●	●
JOB SUBMISSION	●	●	●	●	●	●	●	●

Using the skeleton CIDS files set up by the user for all spacecraft, ODAS created updated CIDS files for the specific test day and submitted them for all spacecraft. All tests were successful.

DC Failure Detection. The goal of testing the DC failure detection function was to verify the ODAS capability to detect and respond to DC failures (see Table 8). The tests were performed by setting the failure criteria to unreasonable numbers to guarantee the failures of certain desired parameters. In Table 5, the subfunctions listed represent all of the single-failure cases.

ODAS was executed using sufficient data for all spacecraft to test the case of DC convergence. The case of nonconvergent DC was generated by using a small data arc and an extremely stringent convergence criterion. Divergence of the DC was detected, and a recovery attempt was initiated. The remaining subfunctions in the table refer to individual DC failures, all of which were successfully detected.

DC Failure Recovery. The goal of these tests are twofold. The tests described here are designed to verify whether the recommended recovery procedure would be initiated when a particular DC failure was detected. However, other tests within this category have a performance aspect to them for which the ultimate effectiveness of the specific recovery procedure in resolving that specific failure is to be verified. This aspect of testing relies

Table 8. Test Matrix for the DC Failure Detection Function of ODAS

DC FAILURE DETECTION SUBFUNCTIONS	TDRS	SME	SMM	Landsat-4	Landsat-5	NIMBUS-7	ERBS	DE-A
CONVERGED DC	●	●	●	●	●	●	●	●
NONCONVERGENCE		●						
FINAL WRMS TOO LARGE	●	●		●	●			
ϱ_1 OUT OF RANGE		●	●					
C_R OUT OF RANGE	●							
σ_R TOO LARGE			●	●	●			
$\sigma_{R/D}$ TOO LARGE				●		●		
ΔR TOO LARGE						●		

most heavily on complete decisionmaking processes reliant on qualitative human experience, requirements which are difficult to emulate in the ODAS-type development environment. This area properly belongs in the realm of basic research and is not addressed here. Testing the DC failure recovery component of the DC QA subsystem involved a complicated set of tests to check the performance of the five ODAS recovery procedures. It involved creating DC failures and verifying attempts at using the proper recovery procedure(s) to recover from the failure(s) (see Table 9).

Table 9. Test Matrix for the DC Failure Recovery Function of ODAS

DC FAILURE RECOVERY SUBFUNCTIONS	TDRS	SME	SMM	Landsat-4	Landsat-5	NIMBUS-7	ERBS	DE-A
MODIFY H-P DENSITY TABLE NUMBER		●						●
EXTEND DATA ARC BACKWARD		●						
DELETE BIASED/NOISY BATCHES	●		●	●				
INCREASE INITIAL WRMS VALUE					●			
USE FINAL ELEMENTS AS INPUT						●		
UNRECOVERABLE		●						

Testing the recovery procedure to modify the H-P density table number involved the case of ϱ_1 failures. ODAS successfully computed a new density table number to meet the QA criterion and recover from the failures. For example, performing DC for SME spacecraft produced a solution with $\varrho_1 = -0.724$, using H-P density table number 8. This failed the QA criterion set at an acceptable range of -0.7 to 0.7. ODAS computed a new H-P density table number 7, reexecuted DC, and obtained a solution with $\varrho_1 = -0.645$, which passed the QA criterion.

Testing the recovery procedure to extend the data arc backward involved using TDRS with a data arc of 34 hours. ODAS successfully extended the data arc backward by half an arc length (17 hours).

Testing the recovery procedure to eliminate biased or noisy batches involved using TDRS, SMM, and Landsat-4. For example, a WRMS value of 2.47 was obtained for TDRS, where the QA criterion was set at 1.45. ODAS used the recovery procedure to eliminate biased or noisy batches and identified a set of batches that need to be deleted. This successfully brought the WRMS value to 1.43, which passed the QA test.

Ephemeris QA. The goal of testing the Ephemeris QA function of ODAS was to verify the successful comparison of ΔR_{ephem} with the criterion for maximum EPHEM overlap difference (see Table 10).

Table 10. Test Matrix for the Ephemeris QA Function of ODAS

EPHEMERIS QA SUBFUNCTIONS	TDRS	SME	SMM	Landsat-4	Landsat-5	NIMBUS-7	ERBS	DE-A
PASS COMPARE CRITERIA	●							
FAIL COMPARE CRITERIA	●	●			●		●	

For the case of TDRS spacecraft, the ΔR_{ephem} value met the QA criterion. The criterion was changed, and ODAS reported an EPHEM QA failure. For the initial day of ODAS execution, all EPHEM QA failed since no previous overlap data arcs were available for comparison.

ODAS Reporting. The goal of testing the reporting function of ODAS was to verify that proper information was sent to the designated files, and the ODAS activities could be monitored (see Table 11). This involved executing ODAS and monitoring the output files.

ODAS successfully processed its output files. The log files contained different levels of detailed reporting on ODAS activities. The DC/SOR output files contained summaries of DC results for few executions for each spacecraft.

Miscellaneous Functions. The goal of testing the miscellaneous functions of ODAS was to validate other embedded functions. These are defined in Table 12.

Table 11. Test Matrix for the Reporting Function of ODAS

REPORTING SUBFUNCTIONS	TDRS	SME	SMM	Landsat-4	Landsat-5	NIMBUS-7	ERBS	DE-A
ODAS LOG FILES	●	●	●	●	●	●	●	●
DC/SOR SUBSET FILES	●	●	●	●	●	●	●	●
OTHER OUTPUT FILES	●	●	●	●	●	●	●	●

Table 12. Test Matrix for Miscellaneous Functions of ODAS

MISCELLANEOUS SUBFUNCTIONS	TDRS	SME	SMM	Landsat-4	Landsat-5	NIMBUS-7	ERBS	DE-A
DC/SOR OUTPUT EXTRACTION	●	●	●	●	●	●	●	●
DIFFERENT DATA TYPES	●					●		
CONTINUOUS EXECUTION	●	●	●	●	●	●	●	●

The DC/SOR output extraction function successfully extracted all required output parameters from the DC/SOR output file for all spacecraft. This involved searching through the output files, locating the required parameters, and extracting them. ODAS successfully processed different data types, e.g., TDRS System (TDRSS) data with TDRS spacecraft, and only Spaceflight and Tracking Data Network (STDN) Ranging Equipment (SRE) data for NIMBUS-7 spacecraft. ODAS was also executed on a continuous basis for 3 days without any problems to check the durability of the system for long periods on uninterrupted execution. All tests were successful.

The test results summarized above demonstrate the viability of autonomous routine OD operation for extended periods of time without analyst intervention. Several types of situations, e.g., host system failure and unacceptable DC solution (DC failure unrecoverable by the ODAS DC QA subsystem), will require analyst intervention. It is possible to enhance ODAS to extend the range of situations that may be handled autonomously. Feasibility studies of several enhancements are in progress.

5. SUMMARY

The development and testing of a working prototype ODAS has established the feasibility of reliable continuous autonomous routine operational OD, especially for situations where successful DC solutions are obtained in the first attempt, representing the major fraction of operational situations. In addition, the inclusion of a generic subsystem capable of accepting direct instructions on specific recovery procedures from an analyst allows ODAS to stay abreast of current levels of expertise, while providing an archival function for past expertise on operational OD techniques. As described in Section 4, preliminary tests of the performance of particular recovery options applied to certain types of DC failure have already been successfully demonstrated in an ODAS testbed. Continued refinement in this area is in progress and represents a definite future direction for ODAS. Associated with this concept is the application of artificial intelligence methodologies to the quality assurance component to exploit the efficient learning algorithms of the latter (Reference 9). Future concepts also include applications to onboard navigation, particularly of refined recovery procedures to provide improved solution reliability for onboard processor-based OD.

APPENDIX

This appendix contains brief descriptions of the five ODAS recovery procedures. Detailed descriptions are available in Reference 2.

Recovery Procedure P1 (Change H-P Density Table). Recovery procedure P1 provides a new modified H-P atmospheric density table corresponding to a different $F_{10.7}$ solar flux. The H-P atmospheric density model currently used in GTDS consists of 10 numerical tables specifying minimum and maximum densities as a function of spacecraft height. Each table corresponds to a specific 10.7-centimeter (cm) solar flux. The recovery procedure uses the estimated value of ϱ_1, an atmospheric density scaling parameter used during DC to compute a more appropriate H-P atmospheric density table. The QA criterion for ϱ_1 is a range, i.e.,

$$\varrho_{1,\min} < \varrho_1 < \varrho_{1,\max}$$

The P1 algorithm employs an analytical representation of the tabulations to determine a higher flux table if ϱ_1 is larger than $\varrho_{1,\max}$ and a lower flux table if ϱ_1 is smaller than $\varrho_{1,\min}$.

Recovery Procedure P2 (Extend Data Arc Backward). Recovery procedure P2 extends the data arc backward in time by one half the current arc length to obtain additional tracking data.

Recovery Procedure P3 (Eliminate Biased or Noisy Batches). Recovery procedure P3 detects biased or noisy batches in the tracking data and creates directives to delete these batches when performing OD. P3 is used to modify the set of observations that is accepted for input to GTDS DC and uses statistics for the observation residuals from the SOR. Since the SOR editor has editing criteria that are different than those used in the DC process, the SOR data are supplemented by additional statistical data. The procedure is based on a modeled range of acceptable means and standard deviations for individual batches of tracking measurements and eliminating any that fall outside this range in a subsequent DC.

Recovery Procedure P4 (Increase Initial WRMS). P4 modifies the initial WRMS specification to accept additional observations during DC. During the DC process, a number of iterations are performed that correct a least-squares fit of the observation residuals. Elimination of observations that correspond to a residual in the first DC iteration that lies outside a user-specified range (proportional to a user-specified initial WRMS) is a mechanism used to eliminate certain measurements for DC. P4 operates on the premise that the user-specified WRMS may have been inappropriately small and computes an increment based on the fraction of accepted measurements and a simple model for the initial statistical distribution of the observation residuals.

Recovery Procedure P5 (Use Final Elements as Input). Recovery procedure P5 is a means for performing further DC iterations in an attempt to achieve better convergence characteristics. The procedure detects the behavior of the WRMS of residuals as a function of DC iterations to determine if the direction is toward convergence or divergence. If the DC process is convergent, this procedure recommends additional iterations, starting with the elements from the last iteration of the earlier DC.

REFERENCES

1. Computer Sciences Corporation, CSC/SD-86/6714, *System Definition and Requirements Analysis of the Orbit Determination Automation System*, June 1986

2. --, CSC/SD-89/6013, *System Description for the Orbit Determination Automation System (ODAS), Revision 1*, H. Mardirossian, K. Heuerman, and A. Anderson, March 1989

3. Goddard Space Flight Center, X-582-76-77, *Mathematical Theory of the Goddard Trajectory Determination System*, J. O. Cappellari, Jr., C. E. Velez, and A. J. Fuchs, April 1976

4. Computer Sciences Corporation, CSC/SD-85/6738, *Goddard Trajectòry Determination System (GTDS) User's Guide, Revision 2*, D. Squier and K. Byers, December 1987

5. --, CSC/TM-87/6706, *Orbit Operations Section CLIST System Description and User's Guide*, A. Drew and J. P. Amenabar, November 1987

6. --, CSC/SD-84/6854, *Automated Orbit Determination-IV (AOD-IV) System Description and User's Guide*, W. L. Steger, III and M. V. Samii, July 1984

7. --, CSC/SD-86/6001, *Operational Orbit Determination Procedures and Quality Assurance Manual*, November 1986

8. --, SEAS (550)/IM-88/-26/(58-472), *Orbit Determination Automation System (ODAS) System Test Results Report*, H. Mardirossian and F. Anzola, May 1988

9. --, CSC/TM-88/6096, *Orbit Determination Automation System (ODAS) Future Development Concepts*, H. Mardirossian, October 1988