JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 32, No. 6, November–December 1995

Spacecraft Autonomous Operations Experiment Performed During the Clementine Lunar Mission

Trevor C. Sorensen*
Allied-Signal Technical Services Corporation, Alexandria, Virginia 22314
Dean C. Oswald†
Software Technology, Inc., Alexandria, Virginia 22303
Rodney M. Shook‡
Space Applications Corporation, El Segundo, California 90245
and
James Van Gaasbeck†
Interface Control Systems, Inc., West Melbourne, Florida 32904

The Clementine mission was the first space-flight demonstration of autonomous operations scheduling using a rule-based script written in Spacecraft Command Language. This capability was demonstrated as an experiment during the lunar postmapping phase. On orbit 303 around the moon, the spacecraft computer propagated a ground-supplied state vector to determine the time of orbital events upon which the mapping command sequences are based. These event times were used to trigger the rules in the script, which in turn issued all the commands required to successfully complete the mapping orbit. The experiment conducted on orbit 303 was completely successful in meeting the mission objectives for the demonstration of autonomous operations. Several other functions of the Clementine spacecraft were routinely performed autonomously, including attitude determination and most of the guidance, navigation, and control pointing modes.

Introduction

THE Deep Space Program Science Experiment (DSPSE) Clementine spacecraft was designed, built, and operated by the Naval Research Laboratory (NRL) for the Ballistic Missile Defense Organization (BMDO). It was launched on a Titan IIG from Vandenberg Air Force Base on Jan. 25, 1994, and entered lunar orbit on Feb. 19. During the 73 days spent in lunar orbit, Clementine returned nearly 1.8 million images of the moon, providing the first global multispectral lunar map suitable for mineral identification.

The primary purpose of the DSPSE Clementine mission was to test new technologies for the BMDO.[1] In addition to the hardware technologies tested, there were two new software technology tests planned—the autonomous position estimation (APE) experiment and the autonomous operations scheduling (AOS) experiment. This paper presents the results of the latter experiment.

The mission objective of the AOS experiment was to demonstrate functioning of algorithms for autonomous spacecraft operation to determine the extent of successful autonomous operation based on position, orbit, and attitude determinations by processing data from onboard sensors. The operational objective was to have the spacecraft autonomously determine the timing of commands and the required spacecraft pointing to accomplish various subtasks of lunar operations, and eventually to autonomously control the operations for an entire lunar mapping orbit. The successful execution of autonomous control for an entire lunar orbit was to be considered the ultimate demonstration of successfully accomplishing this BMDO objective.[1]

Although previous spacecraft have exercised some degree of autonomy (e.g., an autosafing mode to recover from serious spacecraft subsystem anomalies, and a lifeboat mode to regain communication with Earth after attitude knowledge has been lost), the current generation of spacecraft use uploaded sequences of commands for mission operations.[2,3] The command sequences are written at the ground station and tell the spacecraft at what time it should execute each of the desired commands. These times of predicted orbital or mission events are calculated using a ground-based orbit propagator that extrapolates the spacecraft trajectory from the last known ephemeris or state vector. In the case of deep-space missions, the command sequence for several days is often uploaded to the spacecraft days or weeks in advance of execution. This usual method of spacecraft mission operations control has some drawbacks:

1) It requires an extensive ground support team for mission planning and sequencing.

2) It is inflexible and nonadaptive to unforeseen changes in the spacecraft state or mission, which is a disadvantage for deep-space missions with their inherent long communications delays.

3) It requires uploading new command sequences on a regular basis.

This pro tem method was used for standard lunar mapping operations on the Clementine mission. However, one of the design requirements for spaceborne missile interceptors, such as envisioned for the Brilliant Pebbles constellation, is to have autonomous spacecraft operations, including position and attitude determination, without dependence on ground control. The Clementine mission was the first space-flight test conducted by the U.S. to determine the feasibility and methodology of accomplishing autonomous operations. The Air Force's Phillips Laboratory built another satellite, called Technology for Autonomous Operational Survivability (TAOS), which was launched on March 13, 1994. Its objectives also included a test of autonomous operations. Autonomous operations are also desirable for other applications, especially NASA's deep-space probes, such as the upcoming Pluto flybys.[4]

Autonomous Operations on Clementine

The full end-to-end autonomous spacecraft operations can be divided into distinct, but connected, segments (see Fig. 1). The feasibility demonstration on Clementine was divided into separate distinct experiments without any provision for a full end-to-end test combining the segments. In Fig. 1, the level of autonomy used in the Clementine mission is shown by the shading of the boxes and the notation above each box. The specific subject of this paper is

Received Oct. 25, 1994; revision received May 16, 1995; accepted for publication June 19, 1995.

*Clementine Lunar Mission Manager. Senior Member AIAA.

†Software Engineer, Clementine Flight Software Group.

‡Software Engineer, Clementine Guidance, Navigation, and Control Group.

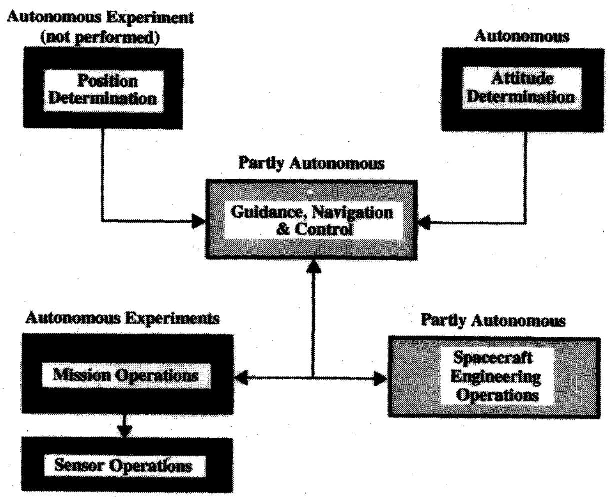

Fig. 1 Segments of spacecraft autonomous operations.

the mission operations segment, but it is appropriate to first briefly discuss the implementation and results of the other segments.

Position Determination

Autonomous position determination was to be tested during the Clementine mission by the APE experiment. It was planned to run this experiment after the spacecraft left lunar orbit, and in cislunar space prior to the encounter with the asteroid. Unfortunately, the spinup failure of the spacecraft occurred after lunar departure and before the APE software could be tested. However, ground tests of the software and algorithms using actual Clementine images have been successfully conducted.

Attitude Determination

Autonomous attitude determination was routinely used during the Clementine mission. The two star trackers built by Lawrence Livermore National Laboratory were actually stellar compasses that provided an autonomous inertial reference for the spacecraft. They were able to take an image of a star field without any a priori knowledge of attitude and provide an attitude quaternion to the guidance, navigation, and control (GNC) software, which fed in into a Kalman filter to update the current calculated attitude sate. To achieve the desired pointing accuracy for the lunar mapping mission, star-tracker updates were provided about every 10 s. This function proved to be highly successful during the mission.

GNC

The GNC function had several autonomous modes that were used extensively during the Clementine mission. These modes were activated by commands issued by either the ground station or the onboard Spacecraft Command Language (SCL) scripts.

The Earth-pointing mode was used during the postmapping phase of each orbit. The Earth pointing mode slewed the spacecraft, generally using reaction wheels although attitude control jet usage was an option, and pointed the high-gain antenna (HGA) at a selected ground station on the Earth while maintaining proper solar-panel orientation and avoiding passage of the mission sensors across the sun. This mode was used whenever the solid-state data recorder (SSDR) was to be dumped to the ground. The Earth-pointing mode maintained an Earth-pointing attitude until the mode was changed.

The star-tracking mode first used the star-pointing command to point the ultraviolet–visible camera at a particular star. This was achieved by looking up the position of the star in an onboard table and slewing the spacecraft to the correct attitude. Once the star was in the field of view (FOV) of the camera, a command was issued to change to the star-tracking mode. This mode used a centroiding algorithm in the image-processing system to track the position of the brightest object in the camera FOV. Corrections were automatically made by the onboard GNC function until the centroid was centered in the FOV of the camera. The star-tracking mode then tracked the star as long as centroids were sent to the GNC function. The star-tracking mode was also used to track the solid rocket motor after separation.

The lunar mapping mode kept the sensors pointed to the center of the moon. This mode permitted an offset angle to be specified, which would point the sensors at the moon's surface, keeping a fixed offset angle between the center of the moon and the orbital plane in a cross-track direction.

Spacecraft Engineering Operations

Autonomous spacecraft engineering operations were routinely used during the Clementine mission. SCL rules were used to monitor the current battery pressure and execute the appropriate safing commands upon detecting that the battery pressure had dropped below an operationally defined threshold. Additional rules were defined to detect the actuation of the paraffin actuator on the primary mission sensor door and to turn off the paraffin-heater timer when actuation occurred. These rules were used throughout the mission.

Sensor Operations

The main mission sensors had two methods of autonomous exposure control available. The first used an onboard analysis of the most recently taken image to adjust the exposure settings (gain and integration time) for the next image from that sensor. This method was expected to be used during the asteroid encounter, but not during lunar mapping. The second method was called the lunar autoexposure control (LAC). It used algorithms that factored in the current spacecraft position, altitude, and solar incidence angle to automatically adjust the camera settings. The LAC was tested on several orbits during the first month of orbit around the moon, and although it was finally adjusted so that it produced good images at the redundant mapping latitudes, it was never used as the primary exposure control. The standard method used on Clementine was to load in exposure setting tables every 10 deg of latitude, and since this method was working so well, it was decided not to risk the prime mapping mission with the experimental LAC.

Architecture of Autonomous-Operation-Scheduling Flight Software

The Clementine flight software provided the capability to perform an experiment on autonomous spacecraft operations for an entire lunar mapping orbit. The autonomous operations integrated the spacecraft command, telemetry, attitude determination and control, orbit determination, and sensor control and processing systems in order to complete an entire lunar mapping orbit.

The autonomous-operation-scheduling flight software was segmented into the following Computer Software Units (CSUs):

1) SCL real-time engine (RTE) task.
2) GNC autonomous scheduler (GNC/AS).
3) Sensor image processor (SIP) interface task.
4) Command and telemetry (C&T) database.
5) Scripts and rules.
6) Autonomous orbit propagator.

The SCL RTE provided the spacecraft stored command capability through the execution of scripts and rules. Throughout the Clementine mission the primary means of stored commanding was the execution of scripts. An additional SCL capability used extensively for the AOS experiment was the execution of forward-chaining rules based on events occurring onboard the spacecraft. Rules allowed the SCL RTE to monitor onboard telemetry values and issue spacecraft commands to control sensor configurations and attitude pointing based on spacecraft events. The SCL RTE executed onboard the housekeeping processor (HKP), which utilized a radiation-hardened Honeywell 1750A processor.

The GNC/AS provided the capability to propagate a state vector for a lunar orbit and record in a table the Universal Time Constant (UTC) time at which an uploadable list of events would occur. For the AOS experiment those key events were based primarily on specific latitude crossings at which different sensor sequence tables and exposure parameters were required. The GNC/AS executed on the SIP, which utilized a commercial IDT R3081 32-bit RISC processor.

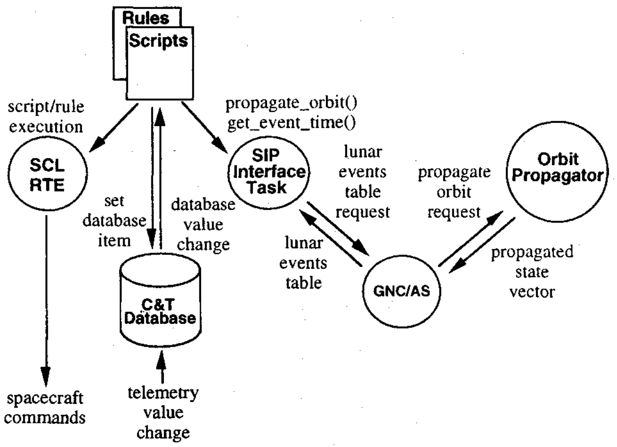

Fig. 2 Software data flow for autonomous flight scheduling.

The SIP interface task also executed on the HKP and provided the procedural interface between the SCL scripts and the GNC/AS. This task provided functions invoked by an SCL script to trigger the propagation of the next lunar orbit by the GNC/AS and to retrieve the UTC time for key lunar events during that orbit. The task received the lunar mapping events table from the SIP via a standard asynchronous message (SAM) and stored it into local memory on the HKP.

The C&T database was used onboard the spacecraft to store the UTC time of the lunar mapping events generated by the GNC/AS. This was necessary so that the UTC times could be referenced by the scripts and rules. Each database record held a 48-bit UTC time.

SCL scripts and rules were developed to support autonomous scheduling of a complete lunar mapping orbit. The scripts and rules were developed in such a manner that the scripts could execute repetitively for each lunar orbit without requiring modification or ground intervention. SCL scripts were used to initiate the generation of the lunar mapping event table by the GNC/AS and store the event times into the C&T database. SCL rules were used to autonomously trigger spacecraft commands on the occurrence of the lunar mapping events. Figure 2 illustrates the autonomous scheduling data flow.

Implementation of Autonomous Operation Scheduling

AOS Scripts

The primary script used to implement the autonomous scheduler was the BuildEventTable script. This script initiated the propagation of the next lunar orbit and waited for the event table to be transferred from the SIP to the HKP. The script then called the SetDBTimes script to retrieve the lunar mapping event times and set the appropriate database values for the events specified.

The BuildEventTable script invoked the propagate_orbit C function, which triggered a SAM to be sent to the GNC/AS for the next lunar orbit. The script would then wait until the orbit propagation was complete and the table was successfully transferred to the SIP interface task on the HKP. After the event table had been successfully built and transferred to the HKP, the SetDBTimes script repetitively invoked the get_event_time C function to retrieve the time for each lunar mapping event and store the value into the appropriate database record. After the database values were set, the SetDBTimes script scheduled the activation of all the rules for the lunar orbit.

Autonomous Scheduling Rules

The SCL rules that implemented the autonomous scheduling for a lunar orbit were simple, single-condition rules, which were evaluated according to changes in the telemetry item utc. This telemetry item provided the current UTC time to the SCL RTE every second. Rules were then evaluated according to the value of this item and the database values set by the SetDBTimes for the key lunar mapping events. This approach was chosen over a more dynamic alternative because events needed to be scheduled some time before and after key events (e.g., South Pole crossing, North Pole crossing), in addition to a particular event occurring. Each rule was defined as a standalone atomic unit, which was activated by the ActivateLunarRules script and deactivated within the rule itself after it executed. This would allow the rule to be executed once and only once for each lunar orbit. Within the rules, attitude pointing commands were sent, spacecraft hardware commands issued, sensor sequence tables selected, and sensor exposure tables loaded. To support an entire lunar orbit, 38 rules were defined. Table 1 shows a sample of the SCL rules used for a lunar mapping orbit.

Table 1 Example of an SCL script

```
—wrule RULE_StartImg—Go to Lunar Mapping mode and start imaging
rule     RULE_StartImg
subsystem    SCL
category    Sched_b
priority    4
activation    no
continuous    yes

  if
    utc >= DCMEXE31
  then
    msg "R:EVT_SPOLE."
    msg "Initiate GNC Lunar Mapping Mode···"
    cexl GNC14POSRW    —Issue the GNC command to go to lunar
                              mapping mode
    cexl DHUSEL21    —start imaging-sequence table
    deactivate RULE_StartImg    —Deactivate rule
  end if

end RULE_StartImg
```

GNC Autonomous Scheduler

The GNC/AS was designed to provide information to the onboard scripts without requiring commands from the ground. The primary function (as used by the Clementine mission) was to provide the times at which specific latitude crossings would occur. Although the specific events were predetermined, the values of these events could be set by loading new values from the ground via a memory load. The GNC/AS also provided data on specific current values for the spacecraft (e.g., current altitude and current longitude).

The GNC/AS required access to the spacecraft position as determined by he onboard navigation function and the onboard orbit propagator. The navigation function and orbit propagator were available only on the SIP, although both were heavily used by the GNC flight software. The orbit propagator was written to be re-entrant, allowing access by both the GNC/AS and the onboard attitude determination software for on-demand state-vector propagation. However, SCL could only communicate to the HKP and had no access to the SIP. Subsequently, the onboard SCL scripts talked to the SIP interface task (part of the GNC/AS) on the HKP, which in turn called a GNC/AS task on the SIP to calculate the appropriate data. The resulting event table was sent back to the HKP via message. SCL access to the data in the event table was via subsequent functions calls to retrieve data from the event table. Communication between the two GNC/AS tasks was done via asynchronous message.

Event Table

The main purpose of the GNC/AS was to fill the event table. The event table consisted of a time of occurrence for each event and also a Boolean flag indicating whether or not the event was found during the calculated orbit. To fill the event table, the GNC/AS was sent a message containing the start time of the orbit desired, the number of hours to propagate the orbit, and a flag determining whether or not the resulting event table should be telemetered to the ground. Once the message had been received, the GNC/AS called the navigation function on the SIP to determine the current spacecraft position. This was the same software module used by the guidance function for attitude control. The spacecraft position was then fed to the orbit propagator to propagate the orbit in 30-s intervals. For each 30-s propagation, the GNC/AS checked to see if any events had been triggered. This cycle would continue until either all the events were found or the propagation time sent up via message had expired.

Once the event table had been filled, it was sent to the SIP interface task via message, where it was stored in local memory until used. Any user of the event table had to make individual C function calls to retrieve a UTC time for a particular event.

The GNC/AS calculated times for the significant events of a particular orbit, such as entering and exiting the solar and rf shadows, crossing the terminator, and reaching a particular altitude or latitude. The event criteria were defined in an internal table and could be changed from the ground via memory upload. Once the events had been checked for the spacecraft position, the GNC/AS checked for the latitude crossing events. The user could specify up to 20 increasing and 10 decreasing latitude events to be predicted. Any time prior to completing the event table, the ground could upload new latitude event positions for all 30 latitudes not yet calculated.

For all times in the event table, a simple linear interpolation was used for calculating event times that fell between the 30-s boundaries of the propagated orbit.

Orbit Propagation

An onboard orbit propagator was used to determine spacecraft position during future orbits. The primary function of the orbit propagator was to provide target pointing information for other modules. Every 30 s, the module executed and generated a message that contained the position and velocity vectors for the Earth, sun, and moon relative to the spacecraft. The vectors were given for two times that were 30 s apart and close to the current time. Other modules that received the message could compute position and velocity at the current time by using an interpolation function. The vectors were defined in an inertial J2000 reference frame.

The pointing vectors were generated by propagating the spacecraft orbit from an initial spacecraft state vector that was uploaded from the ground. The state vector consisted of time, position, and velocity in one of three reference frames: Earth-centered, moon-centered, or sun-centered. All were parallel to the J2000 reference frame. Orbit propagation was done in the reference fame of the uploaded state vector. The propagated state vector was combined with estimates of the sun and moon positions relative to the Earth as computed by separate functions.

Orbit propagation was done with a classic fourth-order Runge–Kutta algorithm. Three forces were included:

1) The gravitational field of the central body (point source for sun-centered).

2) Third-body gravitational force due to the two noncentral bodies among the Earth, sun, and moon.

3) Solar radiation pressure.

Current-Value Function

The current-value function was designed to return specific current values of the spacecraft GNC state. The user of the current-value function could poll a particular event and trigger action when a specific value had been reached. This function returned the current value for such spacecraft parameters as spacecraft position, angular distance to the terminator, solar and rf shadow state, and the angle between the omni antenna boresights and the Earth center.

To access the current-value function, which ran on the SIP, the user had to make a function call to the SIP interface task code indicating which current value was wanted and an address for depositing the value. The SIP interface task then sent a message to the GNC/AS residing on the SIP and waited for a return value. To ensure that the user module waiting for the return of the function call did not hang indefinitely, a timer was built into the software to expire at a preset value if the current value of the parameter was not received from the SIP. The timer value was arbitrarily set at 1 s. Once the message was received by the SIP interface task, the value was deposited into the provided address, the timer was canceled, and control was returned to the calling module.

Experiment Test Plan

The AOS experiment was conducted on orbit 303, which occurred after the completion of systematic mapping. Figure 3 shows the mission and orbital events planned for that orbit, which were typical for a second-month mapping orbit. The time in hours and minutes from aposelene, which was the start of the orbit, is shown in brackets next to each event. At aposelene the spacecraft has the HGA pointed to Earth while dumping the images from the previous orbit. During this dump period the infrared cameras and their cryocoolers are turned on. Shortly before reaching the South Pole, the spacecraft stops the data dump, switches to omni antennas, and slews its cameras to the constellation Crux to perform dark-field calibration imaging. It then turns on the other cameras and slews to nadir to start mapping from pole to pole on the sunlit side of the moon. Laser ranging is performed while the spacecraft altitude is less than 640 km. At the North Pole the imaging is stopped and the spacecraft slews to point the sensors to the star Vega for calibration imaging. Once this is completed, the cameras are turned off and the HGA is slewed to the Earth to start the data dump. The primary difference between the activities of this orbit and standard mapping orbits was the addition of a period between 10°S and 30°N where the spacecraft used an offset to the east from its normal nadir mapping mode to fill a gap in the images that was left from orbit 36.

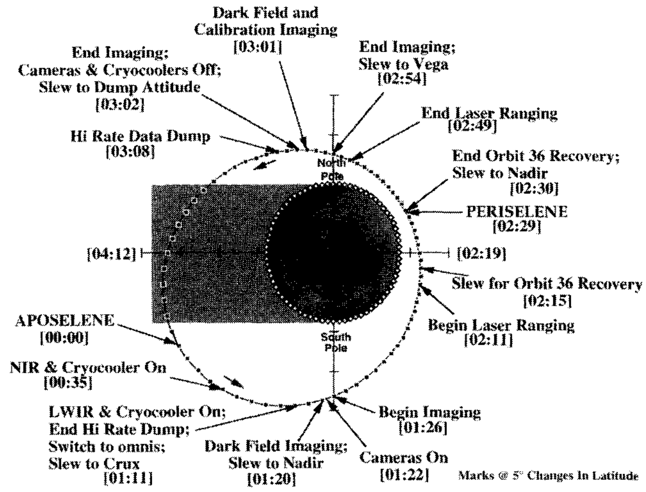

Fig. 3 Major mission and orbital events for orbit 303.

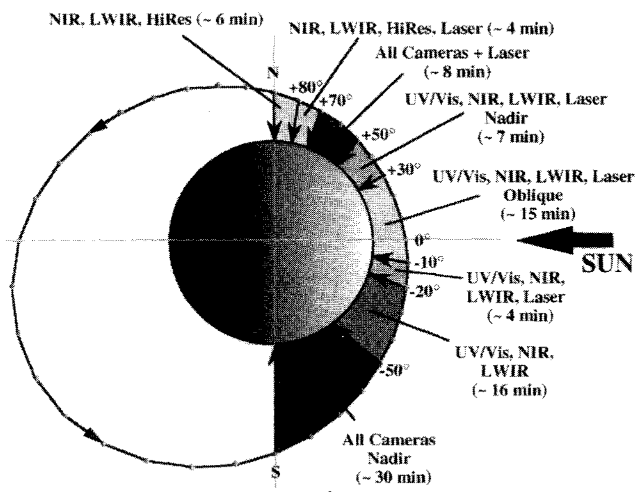

Fig. 4 Mapping activities for orbit 303.

The major imaging activities for mapping this orbit, which were controlled by the AOS scripts, are shown in Fig. 4. The sensors used for lunar mapping were the ultraviolet–visible (UV/Vis) camera, high-resolution (HiRes) camera, near-infrared (NIR) camera, long-wavelength infrared (LWIR) camera, and the LIDAR laser altimeter. A series of tables in the data-handling unit (DHU) computer was used to command the cameras in the correct sequence to obtain the required mapping coverage for the given latitude and altitude band of the spacecraft. The duration of the different types of camera sequencing is shown in parentheses. The HiRes cameras did not take images between 50°S and 50°N, to reduce the amount of data to be downlinked.

The SCL script was written, compiled, and tested in the DSPSE Operational Test Bed (DOTB) by the Science Mission Operations & Planning (SMOP) and Flight Software team, and uploaded to the spacecraft for scheduled execution during the orbit 302 dump period prior to the activation of the NIR camera and cryocooler. The GNC/AS would propagate the orbit and fill the event table. This table would then be dumped to the DSPSE Mission Operations Center (DMOC), where the values would be compared with the planning values. If there were problems with the GNC/AS or there was a large discrepancy between the spacecraft and ground solutions, then the experiment could be aborted. The AOS script would then control all the spacecraft activities from the turning on of the NIR camera, through mapping, and until the spacecraft had slewed HGA to Earth and switched from the omni antennas to the HGA, thus being ready for the ground-initiated playback of the SSDR. This latter function was normally controlled from the ground because of the preference of the spacecraft controllers, but had successfully been done autonomously on previous orbits.

Flight-Test Results

On April 24, 1994, the tested AOS scripts were uploaded to the Clementine spacecraft. The BuildEventTable script was started during the orbit 302 dump period. This called the GNC/AS, which propagated the orbit and filled the event table. This table was downloaded to the DMOC, where the calculated times were compared with the ground-computed times. All the times matched within a couple of seconds. The slight differences were expected, since the state vector used by the spacecraft was more recent than the one used by the mission planners in developing the timeline used for comparison. Since the times were in sufficient agreement, the AOS experiment was allowed to proceed.

During the entire mapping Orbit 303, the execution of spacecraft commands and pointing were controlled by the AOS rules and closely monitored in the DMOC. Every command occurred on time and as expected, and the spacecraft pointing was nominal. The data recovery procedure using offset pointing that had been added to the plan almost literally at the last minute was also successful. Postflight analysis showed that the orbit 36 gap had been filled.

At 0521 UTC, the spacecraft switched from the omni antennas to the HGA after having slewed the HGA to Earth. At this time the AOS script ended, having successfully controlled by spacecraft for the mapping phase of an entire orbit. The DMOC then downloaded the data stored on the SSDR. The quality of the images received was nominal.

Conclusion

The AOS experiment conducted during orbit 303 of the Clementine lunar mapping mission successfully achieved the basic mission objective that had been set by the BMDO. The SCL scripts and rules to support autonomous spacecraft mission operations performed flawlessly, as did the GNC AutoScheduler module that propagated the orbit and filled the event table used by the SCL scripts. However, due to concerns about CPU performance on the HKP (1750A) caused by the need for SCL to poll for the current value, which was only known on the SIP, the current-value function of the AOS was not able to be used during this mission. Operationally this had little effect—all events were triggered by precalculated times instead of currently calculated orbital event values, such as latitude. Since the capability of using current values will be important in future space missions, this feature will have to be tested on future spacecraft. It was the conclusion of the Clementine operations and planning team that if the AOS capability had been used for standard mapping orbits during systematic mapping, the workload on the team would have been significantly reduced, allowing more time and attention to nonstandard observations and orbits, as well as greatly reducing the amount of overtime worked by the team.

Acknowledgments

The authors would like to acknowledge the contributions of R. Prescott of Protasis, Inc., C. Wilderman of NRL, E. Andrews, P. Cappalaere, and Leon Shi of Interface Control Systems, Inc., M. Adams of Software Technology Inc., R. Stapleford and R. Wagner of Space Applications Corp., and M. O'Brian of Mnemonics, Inc., to the success of the Clementine AOS experiment.

References

[1]Horan, D. M., "DSPSE Program Mission Requirements Document," Naval Research Center, SSD-D-DS001 Rev 2.0, Washington, DC, March 1993.

[2]Erickson, K. D., and McKinley, E. L., "Mars Observer Mission Operations," *Journal of Spacecraft and Rockets*, Vol. 28, No. 5, 1991, pp. 542–551.

[3]Young, Carolynn (ed.), *The Magellan Venus Explorer's Guide*, JPL Publication 90-24, NASA, Pasadena, CA, Aug. 1990, pp. 75, 131.

[4]Carraway, J., Henry, P., Herman, M., Kissel, G., Price, H., Staehle, R., and Underwood, M., "Lessons Learned from Clementine on the Way to Pluto," *Proceedings of the AIAA/USU Conference on Small Satellites*, AIAA, Washington, DC, 1994.

A. L. Vampola
Associate Editor