Draft

IN-43
524756

AAS 99-064

AMERICAN ASTRONAUTICAL SOCIETY

LANDSAT-7 Simulation and Testing Environments

E. Holmes, K. Ha, K. Hawkins, J. Lombardo, R. Phillips, M. Ram, P. Sabelhaus S. Scott

NASA Goddard Space Flight Center

21st ANNUAL AAS GUIDANCE AND CONTROL CONFERENCE

February 4-8, 1998
Breckenridge, Colorado

Sponsored by
Rocky Mountain Section

AAS Publications Office, P.O. Box 28130 - San Diego, California 92198

AAS 99-064

LANDSAT-7 Simulation and Testing Environments

E. Holmes, K. Ha, K. Hawkins, J. Lombardo,R. Phillips, M. Ram, P. Sabelhaus S. Scott

NASA Goddard Space Flight Center

21st ANNUAL AAS GUIDANCE AND CONTROL CONFERENCE

February 4-8, 1998
Breckenridge, Colorado

Sponsored by
Rocky Mountain Section

AAS Publications Office, P.O. Box 28130 - San Diego, California 92198

Landsat 7 Simulation and Testing Environments

E. Holmes*, K. Ha†, K. Hawkins‡, J. Lombardo§, R. Phillips**, M. Ram††, P. Sabelhaus‡‡, S. Scott§§

ABSTRACT

A spacecraft Attitude Control and Determination Subsystem (ACDS) is heavily dependent upon simulation throughout its entire development, implementation and ground test cycle. Engineering simulation tools are typically developed to design and analyze control systems to validate the design and software simulation tools are required to qualify the flight software. However, the need for simulation does not end here. Operating the ACDS of a spacecraft on the ground requires the simulation of spacecraft dynamics, disturbance modeling and celestial body motion. Sensor data must also be simulated and substituted for actual sensor data on the ground so that the spacecraft will respond by sending commands to the actuators as they will on orbit. And finally, the simulators is the primary training tool and test-bed for the Flight Operations Team.

In this paper various ACDS simulation, developed for or used by the Landsat 7 project will be described. The paper will include a description of each tool, its unique attributes, and its role in the overall development and testing of the ACDS. Finally, a section is included which discusses how the coordinated use of these simulation tools can maximize the probability of uncovering software, hardware and operations errors during the ground test process.

INTRODUCTION

Landsat 7 is part of NASA's Earth Science Enterprise (ESE). The ESE is committed to developing an understanding of the total Earth system, the effects of natural and human-induced

* Senior Control Engineer, Lead Landsat 7 Attitude Control Subsystem Project Engineer for NASA/Goddard Space Flight Center. Mr. Holmes is affiliated with Swales Aerospace, 5050 Powder Mill Road, Beltsville, Maryland, 20705

† Senior GN&C Engineer. ACSIML7 Developer and Landsat 7 Attitude Control Subsystem Support Engineering Staff to NASA/Goddard Space Flight Center, Mr. Ha is affiliated with Jackson & Tull, .Maryland,

‡ Senior Software Engineer, Landsat 7 Flight Software Support Staff to NASA/Goddard Space Flight Center, Ms. Hawkins is affiliated with Computer Sciences Corporation. , Maryland

§ Senior Control Engineer. Lead Landsat 7 Attitude Control Subsystem Engineer for Lockheed Martin Missiles and Space, P.O. Box 8555, Philadelphia, Pennsylvania 19101

** Senior Software Engineer. Lead LSIM Software Developer, Mr. Phillips is affiliated with ****, , Maryland

†† Senior GN&C Engineer. ACSIML7 Developer and Landsat 7 Attitude Control Subsystem Support Engineering Staff to NASA Goddard Space Flight Center, Mr. Ram is affiliated with Jackson & Tull, ,Maryland,

‡‡ NASA Landsat Project Manager, Mr. Sabelhaus is affiliated with NASA/Goddard Space Flight Center, Landsat Project Office. Code 430. Greenbelt, Maryland 20771

§§ NASA Landsat Spacecraft Systems Manager & Software Manager, Mr. Scott is affiliated with NASA/Goddard Space Flight Center. Landsat Project Office, Code 430, Greenbelt, Maryland 20771

changes on the global environment, and how natural processes affect humans and how humans affect them.

The Landsat 7 satellite is comprised of the spacecraft bus which is provided under a NASA contract with Lockheed Martin Missiles and Space in Philadelphia, PA., and the Enhanced Thermatic Mapper Plus (ETM+) instrument, procured under a NASA contract with Santa Barbara Remote Sensing in Santa Barbara, CA.

The Landsat 7 ACDS provides many essential functions for the operation of the spacecraft bus and for ETM+. The ACDS maintains the required attitude and orbit at the degree of accuracy necessary for power generation, command and telemetry, thermal balance, image acquisition, Gimbaled X-Band Antenna (GXA) Pointing and data for image post-processing. The proper operation of the ACDS is required immediately after ascent in order to assure safe spacecraft operation and subsequently to satisfy mission objectives. However, it is highly impractical to fully exercise all of the ACDS functions on the ground. Because of this, Landsat 7 has been largely dependent on the fidelity of simulation tools to verify the operation of the ACDS. Seven independent simulation tools and test environments were used throughout all phases of the program to mitigate ACDS design and implementation risks. Following a brief description of the Landsat 7 mission and ACDS structure and function, each simulation tool will be described in terms of its structure, development, origin, usage and unique contribution to the verification process. Finally, a section is included which demonstrates how the coordinated use of these simulation tools minimizes the risk of the occurrence of an ACDS design, software or implementation error.

LANDSAT HISTORY AND MISSION DESCRIPTION

For over 25 years, the Landsat series of spacecraft have continuously provided calibrated Earth science data to a diverse group of users world wide. Landsat 7 is the seventh in a family of imaging satellites which provide multi-spectral land and coastal images. The first Landsat satellites (1, 2 & 3) were originally called the Earth Resources Technology Satellite or ERTS. They provided images, including the first composite multi-spectral mosaic of the 48 contiguous United States, between July 1972 and March 1978. ERTS 1,2 & 3 were followed by a second generation of imaging satellites which were called Landsat 4 & 5. Landsat 4 was launched in July 1982 and Landsat 5 in March 1984. Landsat 5 continues to receive images to date. Landsat 6 failed to reach orbit. Landsat 7 is scheduled to launch on April, 15, 1999.

ETM+ DESCRIPTION

The ETM+ instrument is an eight-band multi-spectral scanning radiometer capable of providing high-resolution imaging information of the Earth's surface. It detects spectrally-filtered radiation in the visible, near infrared, short-wave infrared, and thermal frequency bands from the Sun-lit Earth. The spatial resolution is 60 meters in the thermal infrared band, 30 meters in the visible, near infrared, and short-wave infrared bands, and 15 meters in the panchromatic band. The ETM+ images the Earth in 185 kilometer swaths when orbiting at the mission altitude of 705 kilometers.

LANDSAT 7 POINTING PERFORMANCE REQUIREMENTS

The overall pointing requirements for the primary (imaging) mode of Landsat 7 are as follows:(entries are in arcseconds and arcseconds per second, 3 sigma, with respect to the navigation frame of reference)

	Attitude		
	Roll	**Pitch**	**Yaw**
Primary Mode Attitude Knowledge	135.0	135.0	135.0
Primary Mode Attitude Control	120.0	120.0	120.0
Primary Mode Attitude Rate Control	15.0	15.0	15.0

LANDSAT 7 ACDS DESCRIPTION

The ACDS controls spacecraft attitude and orbit, following separation from the launch vehicle and controls propulsive maneuvers for orbit adjustment and maintenance. It also maintains a stable pointing platform to support payload science. In addition, the ACDS controls the motion of the solar array, and three GXA's relative to the spacecraft.

The ACDS is comprised of sensors, actuators, software and support hardware. Included in the sensor package is a Honeywell Inertial Measurement Unit (IMU) and Celestial Sensor Assembly (CSA), nine Adcole Coarse Sun Sensors (CSS), two NASA triaxial magnetometers (TAM), and a Barnes Solid State Earth Sensor Assembly (ESA). The actuator package includes four Honeywell Reaction Wheel Assemblies (RWA), two Ithaco Magnetic Torquer Rods (MTR), and 12 Olin Rocket Engine Assemblies. There are nominally two independent software packages available on the spacecraft. These are the Safehold Package (SHP) which resides on an Electrical Erasable Programmable Read Only Memory chip (EE PROM) and the Flight Load Package (FLP) which is loaded into one or both Spacecraft Controls Processor (SCP). The SHP is used during initial spacecraft activation and as a backup to the FLP. The FLP is the nominal mission software package, which can be modified by way of uploads from the ground. Support hardware required for operation of the ACDS includes the SCPs and the Controls Interface Unit (CIU) which are the heart of the Command and Data Handling Subsystem (C&DH).

The ACDS has two nominal modes of operation (primary and backup) and a number of submodes. Landsat 7 also has a Sun Pointing Safehold mode. The submodes of the backup mode include Rate Nulling, Local Vertical Acquisition (LVA), Yaw Gyrocompassing (YGC), and Earth Search.

Rate Nulling is only performed during the end of the ascent profile, immediately after launch vehicle separation. In this submode rates about all three axes are nulled using only the IMU to sense body rates with respect to inertial space. The initial separation attitude is used as the attitude reference and the RWAs are the primary actuators. The REAs are also enabled and are ready for use if higher than nominal tip off rates are encountered. The rate null submode resides in the SHP only. After successful rate nulling and solar array deployment, the mission transitions from the ascent mode to the orbit mode. The flight software makes the transition from the SHP to the FLP at this time and the ACDS enters the LVA submode.

In LVA, the ACDS acquires the Earth by using ESA data to calculate roll and pitch attitude errors and IMU data for rate damping about all three axes. In LVA the RWAs are the only actuators used for attitude control and magnetic momentum unloading is activated. In the case of an anomalous separation orientation in which none of the ESA quadrants have acquired the Earth, the flight operations team (FOT) may elect to command the ACDS into the Earth Search submode which is essentially LVA with a rate bias added to the roll and pitch axes. The spacecraft remains in this submode until the ESA has acquired the Earth in at least one of its quadrants. At that time the submode autonomously re-enters LVA. Once the pitch and roll attitude errors and the roll, pitch, and yaw rates are below the prescribed thresholds, the submode autonomously transitions to YGC. YGC can be thought of as LVA with the addition of attitude control in the yaw axis. Yaw attitude error is derived using a gyrocompassing technique. Once the spacecraft has successfully achieved this submode, it is power and thermal safe indefinitely assuming that the array is articulating properly. Once the attitude errors and rates have settled below the prescribed thresholds, YGC is considered complete and the Precision Attitude Determination System (PRADS) may be initialized at the FOT's discretion. PRADS is a Kalman Filter which computes attitude errors using star transit data from the CSA. The IMU provides rate data. Once PRADS has converged, the Primary mode may be entered which places the spacecraft in the Precision submode. The Precision submode will maintain local geocentric pointing of the spacecraft to a very high degree of accuracy. In the Precision submode, the RWAs are used to control spacecraft attitude and angular rate, and the MTRs are used as the primary actuators for RWA momentum unloading. The ESA no longer provides attitude error data used by the controller. It is used in an error detection role.

While in the Precision submode, other submodes of the Primary mode may be selected. The Slew submode may be commanded in preparation for an orbit inclination maneuver. In slew submode, a desired rotation about any navigation axis may be commanded. The Maneuver submode may be commanded from either the Precision or Slew submodes depending on the nature of the desired maneuver. Once the maneuver is complete, the ACDS submode autonomously returns to Precision.

LISTING OF ACDS SIMULATION TOOLS AND TEST ENVIRONMENTS

The following is a brief list of Simulation Tools/Test Environments to be described in this paper:

Landsat 7 Simulation
PLATSIM
Software Development Facility
1750 SPECIAL TEST EQUIPMENT CSCI
Landsat Simulator
ACSIMLS7
Self-Test Software

Each of these simulation tools or test environment serves a unique role in the overall verification of the spacecraft performance.

LANDSAT 7 SIMULATION

The Landsat 7 Simulation tool (LS7SIM) was developed at Lockheed Martin Missiles & Space (LMMS) for the purpose of designing and assessing the performance of Landsat 7 attitude control / attitude determination subsystem control algorithms. LS7SIM is written in FORTRAN and runs on VAX VMS Workstations. It models the controllers for each ACDS mode and submode, spacecraft equations of motion, flex-body dynamics, sensors, actuators, and environmental torques. It does not emulate flight software structure or hardware / software interfaces, nor does it model timing.

LS7SIM is an adaptation of a FORTRAN simulation tool that was originally developed for the Solar Max Mission which launched in 1980. The core of the simulator (equations of motion, environment models and flex body dynamics) was also used to develop the Landsat 4 and Landsat 5 spacecraft. The simulation tool was then generalized for easier adaptation to other spacecraft. The first project to use this updated version was the Upper Atmosphere Research Satellite (UARS). In addition to using LS7SIM for design and analysis, the UARS mission successfully correlated LS7SIM data with flight data. Most recently, LS7SIM has been adapted for use on the Landsat 7 project. Modeling of the Landsat 7 ACDS requires the addition of CSA and ESA models. The templates for these sensor models have been taken from the Defense Meteorological Satellite Program (DMSP). DMSP's TIROS spacecraft also use these sensors.

The core components and Landsat 7 specific components of LS7SIM have all been verified on successful flight programs and therefore serves as a very solid engineering tool for the basic Landsat 7 ACDS design and performance. It also serves well as a performance benchmark for some of the other independently developed simulation tools which were used on the Landsat 7 project.

PLATSIM

In addition to meeting the pointing accuracy requirements relating to controller performance and rigid body spacecraft response, the spacecraft must also meet pointing error specifications resulting from jitter caused by internal disturbance sources. The jitter amplitude requirements are as follows: (amplitudes are measured peak to peak, 3 sigma, measured at the spacecraft/instrument interface)

Low Frequency Range (0.01 to 2.0 Hz)	30 arcseconds
High Frequency Range (2.0 to 125 Hz)	24 arcseconds

The simulation environment used to demonstrate acceptable jitter levels is called PLATSIM. This software simulation tool was developed by NASA Langley for verification of EOS-AM jitter study results and uses a "sparse matrix" formulation to allow detailed analysis of very large dynamic systems. PLATSIM was verified and validated at NASA Goddard against EOSSIM (a tool which preceded PLATSIM). PLATSIM runs the Sun Workstation. This is the only simulation environment on the Landsat 7 that can perform highly detailed jitter analysis. In fact most of the other simulation environments (particularly those that run in real time) do not model any flexible body dynamics or internal jitter producing disturbances. These unique capabilities make this simulation environment an important one for ensuring a stable platform for image collection.

PLATSIM is built around a NASTRAN model of the deployed spacecraft. This model contains 326 vibration modes (up to 200 Hz) and contains 42 physical spacecraft locations (6 degree-of-freedom nodal points). An attitude control subsystem is modeled with an approximate 0.01 Hz bandwidth which is incorporated to control the spacecraft rigid body rotations. However, PLATSIM is really a collection of interconnected MATLAB M-file subroutines which model disturbances that excite the NASTRAN model. These M-files are generic in nature and can be readily adapted to simulate most large-scale linear dynamic system.

The disturbance models are developed using MATLAB/PLATSIM language. These models are used as inputs to the NASTRAN/Controller linear dynamics model. The output responses are generated and mathematically combined to the rigid body response to obtain jitter performance data. Disturbances modeled for Landsat 7 include the ETM+, Magnetic Unloading, GXA movement, Solar Array Normal Operations, Solar Array Speed Changes, RWAs, RWA zero crossings and Solar Array snap.

The ETM+ disturbance model simulates the 7 Hz scan mirror oscillation which is characterized by a 249.0 in-lb amplitude torque transient to the spacecraft structure as the mirror impacts a hard rubber stop at each extreme of the mirror oscillation.

Magnetic unloading is provided by the use of the MTRs controlled with a bang-bang controller. This type of controller imparts significant torque impulses to the spacecraft. Therefore, during MTR unloading, a feedforward torque command is sent to the RWAs in order to counter the anticipated impulse associated with MTR turn-on. This torque can only be estimated based on TAM readings and assumed MTR dipole. For the purposes of analysis, an 80% effective feedforward procedure is assumed. Therefore, the disturbance associated with magnetic unloading is defined by the difference between a the actual impulse torque applied to the spacecraft and the feedforward torque commanded.

The GXA disturbance model is defined as a 10.7 lb mass with a center of mass offset of 7 in with respect to the inboard gimbal (cross track). Maximum slew rate (1.0 deg/sec) and a maximum acceleration of 0.16665 deg/sec^2. The individual stepper motor pulses are not modeled.

Solar array normal operations is defined as the solar array rotating at orbit rate. The solar array inertia is modeled as well as the center of mass migration due to the 20 degree cant angle of the array (solar array center of mass is not along axis of rotation). The stepper motor drive with a nominal pulse rate of 3.64 pulses per second is modeled as is the 108:1 gear drive and friction. Additionally, solar array speed change is also modeled. This is the transient caused by a 1.7% speed change in the nominal solar array rate. The change in rate requires 0.5 seconds. Solar array thermal snap torque is modeled in this simulation environment. The models are developed using a technique which was successfully used to predict UARS thermal snap disturbances from Landsat-4 flight data. However, thermal snap is only modeled for completeness, since this phenomenon occurs at times when images are not being collected.

RWA jitter is also modeled as a disturbance. The two largest disturbances associated with RWAs are typically static and dynamic imbalance. However, in this modeling effort, only static imbalances have been modeled. This is because of the relatively small impact that the dynamic imbalance has on pointing error. Both static and dynamic imbalance disturbances are a function of wheel speed. Therefore, a wheel speed of 2550 RPM was chosen to excite a large structural mode that exists at 42.5 Hz. RWA zero crossing is modeled even though this phenomenon will not be allowed to occur during the normal mission unless one of the four RWAs fails.

The data obtained from this simulation tool indicates that only RWA zero speed crossing and solar array snap exceed the jitter budget. Fortunately, under four RWA control (roll, pitch, yaw and skew) the wheel speeds can be biased such that zero crossings will not occur. As mention earlier, solar array snap occurs when the solar array heats up as it enters sunlight and as it cools during the eclipsed portion of the orbit. These disturbances settle out before any image collection occurs.

SOFTWARE DEVELOPMENT FACILITY

The Software Development Facility (SDF) is the simulation tool used to verify software requirements which require a high degree of fidelity, such as the ACDS requirements and their related algorithms. This verification process is known as the Formal Qualification Test (FQT).

The SDF consists of hardware, such as a DEC VAX Server which hosts the SDF Computer Software Configuration Item (CSCI) and all of the software tool sets except the Requirement/Verification Tractability Tool (RVTM) , one or more DEC VAX Workstations, Sun Workstation which hosts the RVTM and related network hardware. The SDF also consists of a number of software items which were employed in the software test process. These include the 1750A Integrated Tool Set, SDF CSCI, the Flight Software (FSW) CSCI and various other support tools.

The 1750A Integrated Tool Set consists of the Jovial/J73 Compiler and 1750A Assembler/Linker software development tools. The Integrated Tool Set is used for compiling and linking the FSW into an executable image. It may also be used to support debugging, recompiling and re-lining during Unit Level Test, CSC Integration Test and CSCI Test.

The SDF CSCI provides the tools necessary for the development and testing of the FSW, and runs on a DEC VAX workstation. It performs high fidelity, though not real time, closed loop simulations of the satellite subsystems. The SDF CSCI consists of the Package/Module Development Software (P/MDS) and the P/MDS Support Program (PSP).

The SDF satellite subsystem simulations include simulations of actuators, sensors, commanding, telemetry processing, ranging, the 1750A processors (one or two processors can be simulated), and environment/orbit/dynamic behavior. The SDF is also capable of collecting timing measurements which are used to ensure timing requirements are met.

The other support tools that support the FSW test activities include the Change Control/Build Tool (DEC CMS), Software Problem Reporting Tool (DEC CMS), RVTM, and Documentation Generation Tool (Interleaf).

1750 SPECIAL TEST EQUIPMENT CSCI

The 1750 Special Test Equipment (STE) (sometimes referred to as the FSW STE) is the other test environment which is used to demonstrate that the software requirements specifications have been met. It's purpose is to provide the capability to test and validate FSW in a real-time environment with flight like hardware prior to spacecraft integration and test. The STE is the first place that the FSW is tested with flight like hardware operating with the timing and I/O characteristics of the actual flight components.

The STE runs on a host Hewlett Packard 1000 A990 host computer under control of a real-time operating system (RTE-A). This software provides input or stimulus data to FSW as would be provided by actual spacecraft devices, such as actuators and sensors, along with simulations of the spacecraft orbital behavior and dynamics to provide closed-loop stimulus data. The STE includes a simulation of the FSW command stream, and collects and records telemetry and hardware commands output by the FSW. The STE environment also includes other pieces of hardware such as the 1750A SCP, CIU (including the Telemetry Data Formatter (TDF)), a HP tape drive and a HP line printer.

In addition to being a test environment used to perform FQT. the STE also serves as a test-bed on which to test new versions of FSW and FSW patches prior to loading and running on the spacecraft. This will be its primary task after the FQT process is completed, and throughout the Landsat 7 mission.

LANDSAT SIMULATOR

Landsat Simulator (LSIM) is a simulation tool / test environment developed for use by the Landsat 7 FOT and Flight Support Team (FST). LSIM is primarily used as a training tool. It provides the FOT and FST with a virtual spacecraft which responds to commands and returns telemetry as the actual spacecraft would on orbit. It runs in real-time and simulate/emulates all spacecraft functions and some instrument functions. LSIM is used in all of the FOT and FST supported operational rehearsals and ground station rehearsals prior to launch. It is also used in the development of operational procedures (both nominal and contingency). Post-launch, it will be used to diagnose spacecraft problems, develop and rehearse new procedures, and train new FOT personnel.

LSIM is written in C++ on a SUN UltraSparc workstation. It uses two subroutine libraries, RougeWave tools and Kinesix SAMMI. The former provides basic classes for C++ (lists, collections, etc.) the latter provides the GUI. There are about 200,000 lines of C++ code in the system. The Rational Rose design tool was used to develop and maintain the design documents for the program. Source control is handled by the Rational APEX system. The program has been under development for approximately three years. and has generally had a staff of 4 to 8 programmers assigned to it at any given time.

The LSIM working environment is a Sun UltraSparc 2 workstation with a pair of 300 MHz processors. The code is designed to use multiple threads of execution and to provide real-time telemetry for the Mission Operations Center (MOC). The system provides independent (truth) models for attitude and orbital mechanics (AOM), attitude control systems, reaction control systems, sensors, the solid state recorder, the telemetry system, the RF systems (X-band antennas and power supplies), and rudimentary control information related to the ETM. Orbital calculations can be performed with a force model propagator (which allows prediction of orbital motion during thruster firings) and a Precision mode which uses tables of ephemera calculated using the Satellite Tool Kit (STK) program. To assist in testing and in development of FSW, the system has an emulator of the two 1750A processors on the spacecraft. LSIM executes the same compiled and linked code as the spacecraft. New versions of FSW can be uploaded to LSIM in the same manner as they would be uploaded to the spacecraft. Operation of the spacecraft with different versions of FSW in the two processors is fully supported.

ACSIMLS7

ACSIMLS7, developed by Jackson & Tull, is a high fidelity spacecraft dynamics and control simulation software program derived from a generic simulation tool designed to simulate a wide range of spacecraft configurations, sensors, and actuators. ACSIMLS7 is a Win32 application software program, running under the Windows operating system, and is used as part of an independent verification and validation effort of the Landsat 7 ACS design and implementation.

ACSIMLS7 simulates all Landsat 7 ACS sensors and actuators, solar array motion and deployment scenarios, as well as spacecraft orbit configuration. As a tool to verify and validate the ACS design, the models of sensors, actuators and dynamics implemented in ACSIMLS7 were derived independently from any Lockheed-Martin simulation effort, and based only on system specifications, requirements, and test results. The simulation of the Landsat 7 ACS algorithms is developed closely in parallel to the development of the Landsat 7 flight software in order to maintain ACSIMLS7 with the latest updates in the ACS design and flight software implementation.

The simulation software is structured into three distinct components: (a) the model component which contains the spacecraft dynamics, disturbance, sensor, and actuator models, (b) the ACS component which simulates the ACS flight software code, and (c) the Graphical User Interface (GUI) component which provides the interactive control of the simulation. The interfaces between these components are designed to be generic and follow a general format and convention that is independent of a particular simulated spacecraft configuration. This design feature allows for independent and parallel development of these components. Below are descriptions of some of the highlights in each of the three components of the simulation.

ACSIMLS7 is an interactive software program controlled and monitored by the user through an extensive set of tools in the GUI component. These tools basically allow the user to change simulation parameters in real-time, to see what happens immediately, and to permit user to access to all variables in the simulation to look at current values either numerically or graphically.

The GUI component essentially is the main driver of the simulation in that it controls the flow of the simulation, and is responsible for creating, synchronizing, and handling the two main processes in the simulation. One process is for executing the GUI component and the other is for executing the model and ACS components.

As part of the initialization of the simulation, the GUI is responsible for setting up the simulation initial conditions at the start of a simulation run. It reads from a series of ASCII files the data defined in various modules in the three simulation components, and populates them in the global memory in the module's own data structure formats. These input data contain initialized values of variables defining the simulation initial conditions. These ASCII files are created manually by the user and can not be modified by the simulation. However, during the course of a simulation run assigned values in memory may be changed either as results of computations or by the user through the GUI variable-editing feature.

A graphical tree structure created by the GUI provides pointers to all data variables using the same memory addresses that defined by the component modules. This tree structure is available to the user in all subsequent data processing interfaces to select variables for plotting, displaying, and updating.

During a simulation run, not all variables defined and contained in the tree structure are allowed by the GUI to be modified by the user. Those that derived and manipulated by the dynamics and models can only be changed by the equations that define them. Otherwise, physically non-realizable scenarios can occur leading to inconsistent and invalid simulation results. Other variables such as ACS variables and simulation control parameters can be changed by the user during the simulation to affect scenario and the simulation setup.

The user can start and stop the run at any time in a simulation run to examine or change values of variables, or to start/stop output data stream, or to perform a host of other data processing tasks provided by the GUI tools. The simulation can run in either continuous mode or single step mode useful for checking system response during transient period. The simulation can also be started and stopped by timer defined by the user. This feature enables scheduling of the simulation to enter various operational modes at precisely specific times for test comparison purposes.

The GUI component also has all the standard features developed in most application GUI software such as to create, save, and restore configurations of display pages and plots. Capabilities such as plot magnification and multiple plotting of arrays and automatic scaling of axes are also available.

The dynamics and model component in ACSIMLS7 contains high fidelity models of the spacecraft dynamics, its environment, and sensors and actuators on-board Landsat 7 spacecraft. These models are based on the established and accepted models developed at NASA, for various space missions including the recent XTE and TRMM missions. These models are developed to cover both the linear and non-linear operating ranges and have all the significant noise sources. Also, with the simulation software structure designed to accommodate generic ACS spacecraft system, different models for spacecraft dynamics, external and internal disturbances, and sensors and actuators can easily be added or replace those models already in the simulation.

The model component is built around a Fifth-order Cash-Karp Runge-Kutta integration scheme. This integration technique utilizes an adaptive step-size approach, which monitors local truncation error to determine, in an iterative process, the appropriate step-size for achieving the required accuracy. During initialization, a number of user-defined parameters for the integration such as the maximum and minimum step sizes, the relative and absolute error tolerance levels are read in. All these values can be adjusted during the simulation for efficiently optimizing the simulation time as well as maintaining the result accuracy.

Any module from the three software components that requires the integration will report with number of states, the address of callback derivative functions, and the initial conditions. The integration software then organizes and allocates memory for all the states and their corresponding derivative functions. This integration scheme allows the handling of variable number of states, which can be dynamically modified during run-time so that modules or models in the simulation can easily be added or replaced.

In addition to the integration scheme described above, this truth model component also handles the creation and maintenance of a circular linked-list structure containing all the time events occurring during a control cycle. This linked-list is organized during the initialization by gathering from all the model modules and the ACS components the time events and corresponding callback routines. This feature not only eases the adding and removal of simulation processes associated with different sampling time, also makes the system more general in accommodating changes in spacecraft configuration and ACS design.

In general, while the simulation is running the truth model component interfaces with the GUI essentially through common memory and to others such as the ACS component through function calls to addresses that are already set during the simulation initialization and recorded in the linked-list structure.

For Landsat 7, the spacecraft dynamics, in terms of attitude, is implemented with equations of motion for a system of two constrained rigid bodies, representing the spacecraft main body and the solar array, with one rotational degree-of-freedom joint between them. The solar array is treated in the formulation of the equation of motion as a rigid body with varying inertia and dimensions. This approach allows for the simulation of the solar array from stowed, through deployment, and to fully deployed configurations. For the rotational motion of the solar array during normal operation, a simple solar array torque control loop is assumed to provide a reasonably smooth transition between various allowable rates when commanded, and to account for the internal disturbance to the spacecraft motion.

For orbital dynamics, displacement due to external forces, such as thruster and aerodynamic force, is taken into account by the equations of motion of the systems center of mass. Perturbation effects due to non-spherical earth including terms up to the J4 in the harmonic expansion are also implemented.

Detailed models of coarse sun sensors mounted on the rotating solar array, earth sensors with 4 detectors in each of the four quadrants, reaction wheels with ripple, friction, and bearing noise terms, and thrusters operating with varying system center of mass are also included. A gyro model that contains initial drift rate, white, and random walk noise terms, as well as quantization noise and the gyro dynamics bandwidth are provided. The high fidelity of this model is important to support an in-depth evaluation of both the short and long-term performance of precision attitude determination normal mode. Also, a detailed model of the CSA is provided for the analysis and evaluation of false star identification issue which is an inherent drawback in the use of the CSA in place of standard star trackers. The CSA model contains accurate slit geometry and timing clock with respect to the gyro sampling rate. It also keeps track of stars with visual magnitude down to 4.9 that cross each of the 6 CSA slits. It determines detected transits based on the probability of detection curve and star magnitude.

The ACS component in ACSIMLS7 contains code, in C, corresponding to the Landsat 7 attitude control flight software written in Jovial. This component is developed very closely in structure to the Landsat 7 ACS flight software. It, like the flight software, is divided into two packages: a Safehold Package and a Flight Load Package. It has all the ACS-related interrupts at 2, 8, and 10 Hz. In addition to maintaining the same software structure, it has the same naming convention, and separate data sets for the packages. The ability to switch from one package to the other is also maintained within the simulation software. In short, all routines and components of the flight software deemed to be directly related to the operation of the Attitude Control Subsystem and have impact on the control design performance have been coded in the simulation.

SELF-TEST SOFTWARE

Self-Test Software (SELTS) was developed by LMMS and serves in the performance verification process. SELTS resides in the SCPs of the Landsat 7 spacecraft and the software is used to facilitate on the ground closed-loop testing of the FSW CSCI running on the SCP during the Integration and Test (I&T) phase of the program.

To the extent possible, the verification process requires satellite testing that emulates on-orbit satellite operations in the presence of a simulated on-orbit environment. The goal is to create a test environment that simulates the actual flight conditions and flight configuration of the satellite as possible.

ACDS testing uses SELTS quite heavily during the I&T process because of the desire to operate the subsystem in a close-loop fashion. At the satellite level, no special test equipment is available to close the loop on the FSW CSCI to emulate flight-like conditions. Instead, SELTS, running on the SCPs interacts with the FSW CSCI to close the loop. This is SELTS' primary role. SELTS will simulate of all sensor and actuator data on the satellite except for the CSA. This allows for the use of pure SELTS data, combined hardware and SELTS data or pure hardware data in any combination. The use of SELTS data is transparent to the FSW. SELTS is not meant to validate ACS sensor function or performance. Many of its models have been simplified to allow for real-time execution on the flight SCPs on top of the burden of running FSW. However, in conjunction with the other simulation environments in this paper, SELTS can be a powerful tool in verifying spacecraft performance at the spacecraft level.

SUMMARY

Validation of ACDS design and implementation is heavily dependent upon simulation because of the subsystem's need to be in a dynamic orbital environment in order to perform its function. Subsystem performance and stability are completely dependent upon simulation. Proper phasing of all of the subsystem's actuators and sensors are also largely dependent upon simulation, though some open-loop tests can measure certain aspects of phasing. Therefore, the accuracy of simulation models and the validity of simulation results is crucial. This concern is even greater when it is not possible to independently verify and validate all of the simulation tools, as is the case on the Landsat 7 project. However, coordinated properly, the use of the various Landsat 7 simulation tools and test environments can minimize the probability of design and implementation errors in the ACDS after launch.

All of the simulation tools described in this paper have a common thread. They all functionally simulate the Landsat 7 ACDS. However, each simulation tool models each aspect of the subsystem to a lesser or greater degree of fidelity. Each tool has a unique capability, so every component of the subsystem and every subsystem function is modeled to a high degree of fidelity. Every flight component of the subsystem is also exercised in closed loop simulation.

Risk of a carrying a design problem into orbit is mitigated through the use of thoroughly tested design tools. Though not specifically verified and validated for Landsat 7 application, variants of LS7SIM have been successfully used as an ACDS design tool on other flight projects. This is also true of the individual sensor and actuator models used in the simulation tool. Likewise, PLATSIM has been successfully used to predict jitter on other flight projects. Its components, NASTRAN and MATLAB are each industry standards in their respective disciplines. Therefore, there is overall confidence in the controller design because of the track record of the design tools.

The proper implementation of the ACDS design in flight software and flight hardware is handled by a different set of tools. The SDF is both a software development tool and a software test environment. It runs the actual FSW in an environment of simulated C&DH hardware, simulated sensors and actuators, and simulated orbital dynamics. Flight-like commanding and telemetry processing are also simulated on the SDF. All FSW functionality can be tested in this environment. The FSW STE also uses the actual FSW, but it runs in real-time on an actual flight-like processor hardware and flight-like CIU. Therefore commanding and telemetry processing take place as they would on the actual spacecraft. The STE is the closest software processing environment to the actual spacecraft. It's sensor, actuator, and dynamics models (like the SDF) were developed independently from the LS7SIM models and therefore are in a sense validated when STE and SDF simulation results match LS7SIM results.

The command and telemetry portion of the ACDS design are more thoroughly exercised using LSIM. The LSIM simulation tool can be used to validate the ground system as well as to train the FOT. The LSIM emulates the spacecraft/ground station command interface and spacecraft response. Its dynamic, sensor and actuator models were also independently developed.

The ACSIMLS7 tool was developed as a completely independent, high fidelity simulation environment to assess the performance of the ACDS design and FSW implementation. The ACSIMLS7 has the highest fidelity models of the sensors and actuators of any of the Landsat 7 simulation tools. It also is unique in its ability to simulate the varying inertia properties of the spacecraft as the solar array deploys. ACSIMLS7 uses the actual FSW structure, but translated into C++. ACSIMLS7 does not attempt to simulate the C&DH hardware or hardware/software timing.

Finally, the SELTS/Spacecraft test environment ties all of the previous testing and simulation together. The hardware is the actual flight hardware and FSW is the flight code. SELTS allows the FSW and hardware to be operated in a closed-loop fashion on the ground. This test environment is flexible enough to perform open-loop testing of the spacecraft without SELTS active, closed-loop with SELTS generated dynamic response and sensor/actuator data, or any combination or SELTS sensors/actuators and flight sensors/actuators. It is in the spacecraft test environment (with and without SELTS activated) that all end to end phasing tests and comprehensive performance tests are performed in which every data path and hardware/software interface are tested.

CONCLUSION

The coordinated use of specialized simulation tools and test environments has made it possible to design and thoroughly test the Landsat 7 ACDS. Consistent simulation and test results across independently developed tools and test environments is a convincing indicator that the design and implementation of the Landsat 7 ACDS has been performed successfully. Mission success will be the ultimate validation.