DSENDS - A High-Fidelity Dynamics and Spacecraft Simulator for Entry, Descent and Surface Landing[1][2]

J. Balaram, R. Austin, P. Banerjee, T. Bentley, D. Henriquez, B. Martin, E. McMahon, G. Sohl
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, MS 198-219
Pasadena, CA 91109
+1 (818) 354-6770
balaram@jpl.nasa.gov

Abstract—The Jet Propulsion Laboratory (JPL) is developing a high-fidelity spacecraft simulator for Entry, Descent and Landing (EDL) on planetary and small-bodies. This simulator, DSENDS (Dynamics Simulator for Entry, Descent and Surface landing), is an EDL-specific extension of a JPL multi-mission simulation toolkit (Darts/Dshell) which is capable of modeling spacecraft dynamics, devices, and subsystems, and is in use by interplanetary and science-craft missions such as Cassini, Galileo, SIM, and Starlight. DSENDS is currently in use by the JPL Smart Lander project [1] to provide a high-fidelity testbed for the test of precision landing and hazard avoidance functions for future Mars missions.

We first briefly describe the core tool capabilities in dynamics, instrument/actuator device models, and real-time simulation engineering. These allow the modeling of the flight-train elements during various phases of EDL and all of the spacecraft responses. We then address the various EDL-specific aspects of the simulator. These include the high-fidelity entry-body and parachute aerodynamic models and supporting atmospheric models. In addition the simulator provides real-time terrain and instrument simulations for terrain imaging lidars and radars. The simulator hosts guidance/navigation simulation modules for hypersonic steering and powered descent. Models for landing kinematics and dynamics are being incorporated to determine contact and impact forces. We also discuss the automated state-machine driven model switching used to handle spacecraft separations and re-configurations, as well as specific visualization tools developed to support EDL execution interpretation.

The DSENDS system is capable of supporting the entire mission lifecycle use of simulators. This includes the facility to embed the simulator into a Matlab/Simulink environment where control analysts can use the same high-fidelity simulation used in real-time operations within a familiar analysis environment. DSENDS also provides stand-alone simulations to provide mission visualization and support for Monte-Carlo analysis. Using the real-time features of the underlying Darts/Dshell toolkit, DSENDS can be made to operate in a VxWorks/VxSim real-time testbed.

A key aspect of building such a simulator is the verification of its capabilities. We discuss our approach to verification which includes comparisons with off-line simulators (e.g. POST), mission data (e.g. Mars Pathfinder, NEAR asteroid mission), as well as data from terrestrial test programs.

TABLE OF CONTENTS

1. INTRODUCTION

The National Aeronautics and Space Administration (NASA) plans to conduct a series of challenging in-situ missions at a number of planetary and small-bodies [2]. These include a variety of Mars rover and sample return missions, a Venus sample return mission, a Europa lander, a Titan organics explorer, and a comet nucleus sample return mission. These new missions are being conceived of as being more sophisticated in their capabilities, especially with regard to their precision in landing and their ability to handle a variety of landing hazards.

For planetary bodies, the EDL portion of these missions are vastly more sophisticated than earlier and current missions which utilize relatively straightforward ballistic entry methods with only mechanical hazard accommodation capabilities. Precision landing is to be achieved using lifting-body aero-maneuvering methods during the hypersonic atmospheric entry phase to reduce the large landing dispersions associated with ballistic atmospheric entry.

Active landing and beacon sensors are being contemplated for precision approach and hazard avoidance, with scanning lidar and radar sensors actively used in closed-loop fashion to control powered descent during the terminal phases of descent and landing.

All of these closed-loop control actions, together with the machine intelligence algorithms used to select a safe landing site based upon sensor data, are embedded into the on-board flight software and require thorough verification at all stages of development. A high-fidelity simulation, with the capability to realistically capture the relevant physics and device interactions, can provide a level of verification and validation of the algorithms, flight software, and embedded system real-time performance.

For small-body missions, the EDL portion contemplates active hovering and precision landing onto specific targets determined from earlier mapping orbits. These capabilities depend on precision navigation in a microgravity environment, active sensing to select viable landing sites, together with control actions to overcome external disturbances in the small-body proximity environment. Unlike a large planetary body, atmospheric effects are fundamentally different, as comet out-gassing is transitory rather than continuous. Descent and landing necessarily includes orbital periods during which the gravity field of the body is characterized, and a descent that is significantly slower than for planetary bodies. Due to this, sequence timing is not as critical, however an offsetting factor is the increase in uncertainty of gravitational effects, and the increased requirements for autonomy due to the typically large distances from Earth. From a simulation perspective, high-fidelity nonlinear gravity models, and the ability to model a variety of spacecraft environment interactions are key to verifying EDL performance.

An activity at JPL was initiated in the summer of 2000 to provide a high-fidelity simulation toolkit for EDL applications. The DSENDS system is our attempt to develop a tool to address these needs for a wide variety of planetary and small-body missions across the entire mission life-cycle. The foundation for this effort is the Darts/Dshell spacecraft simulator tool used in a variety of JPL planetary and science-craft missions. Darts (Dynamics Algorithms for Real-Time Simulation) is a JPL computational engine that provides high-fidelity, physics-based simulation of multi-body and flex-body dynamics [3]. Dshell (Darts shell) provides the ability to wrap spacecraft devices and environments around the dynamics engine to allow the simulation of spacecraft systems interacting with their environment [4][5][6]. DSENDS is thus a Darts/Dshell adaptation that has EDL-specific models and simulation facilities.

2. CORE TOOL CAPABILITIES

Darts/Dshell Tool Capabilities

Darts is the dynamics engine with the capability to model the dynamics of complex multi-body systems with flexible modes. An extensive Dshell library of models is available to the user to allow the instantiation of various spacecraft bodies, spacecraft devices such as actuators (e.g. thrusters) and sensors (e.g. Inertial Measurement Units), and environments (e.g. gravity, ephemeris). Dshell also provides a modeling and scripting language based on Tcl [7]. Extensive capabilities for debugging, check-pointing, 3D visualization, real-time graphing, and dynamic peek/poke of variables, are directly built into the core Darts/Dshell toolkit. Dshell also provides the capability to interface to external models and compiled model libraries to allow the use of legacy models as well as external simulations that do not need to be tightly coupled (i.e. at the differential equation solver level) to the multi-body dynamics (e.g. external sensor models).
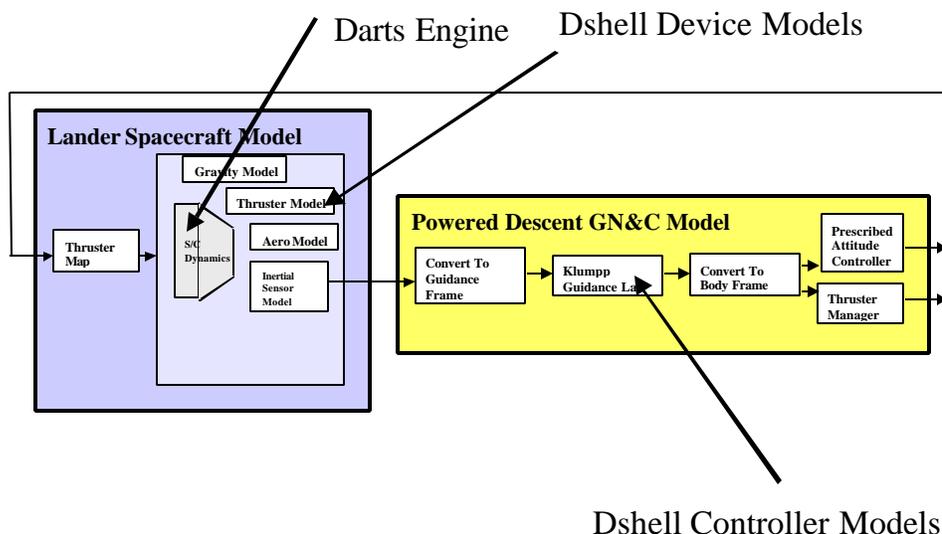

Figure 1 – Simplified block diagram of Darts/Dshell models for powered descent portion of EDL

Lifecycle Engineering Capabilities

Darts/Dshell can be used for standalone mission simulation on an engineering workstation. An example of this is shown in Figure 1. It can also be automatically embedded into the Matlab/Simulink simulation toolkit to support system design and analysis as shown in Figure 2. Since Darts/Dshell supports the modeling and simulation of all the spacecraft devices, it can serve as a virtual spacecraft that is exercised by flight software during test. An example of this is illustrated in Figure 3. A key feature here of Darts/Dshell is that the underlying libraries fully port to real-time operating systems such as VxSim/VxWorks, thereby allowing the simulation exercised in a non real-time standalone computer or in Matlab/Simulink, to be executed in a real-time testbed. This is especially useful for hardware-in-the-loop testing during spacecraft Assembly Test Launch Operations (ATLO). Finally, the tool may also be used in mission operations to support the design and verification of spacecraft command sequences.

3. EDL EXTENSIONS

DSENDS implements a number of EDL-specific extensions that go beyond the basic Darts/Dshell capability. The relationship between the core Darts/Dshell models and those specific to EDL is shown in Figure 4. We now discuss some of these extensions in more detail.

Aerodynamic Models

The Darts/Dshell simulation core has an object-oriented design with a well-defined class structure for its models. Aerodynamics, at any fidelity, fits within the DSENDS *actuator class* definition because it is a forcing function with dependence on the current state of the EDL system's trajectory. Thus, given the state of an EDL mechanical system model and the environment, the aerodynamics model should output the state-dependent aerodynamic forces and moments experienced by the EDL system. Ambient atmospheric properties along the EDL system's trajectory are assumed to come from an encapsulated atmospheric model object which has both spatial and temporally dependent data.

This broadly defined aerodynamics actuator model can serve as a base class to encapsulate high-fidelity aerodynamics libraries that determine the detailed aero-coefficients across various spacecraft attitude configurations and fluid flow
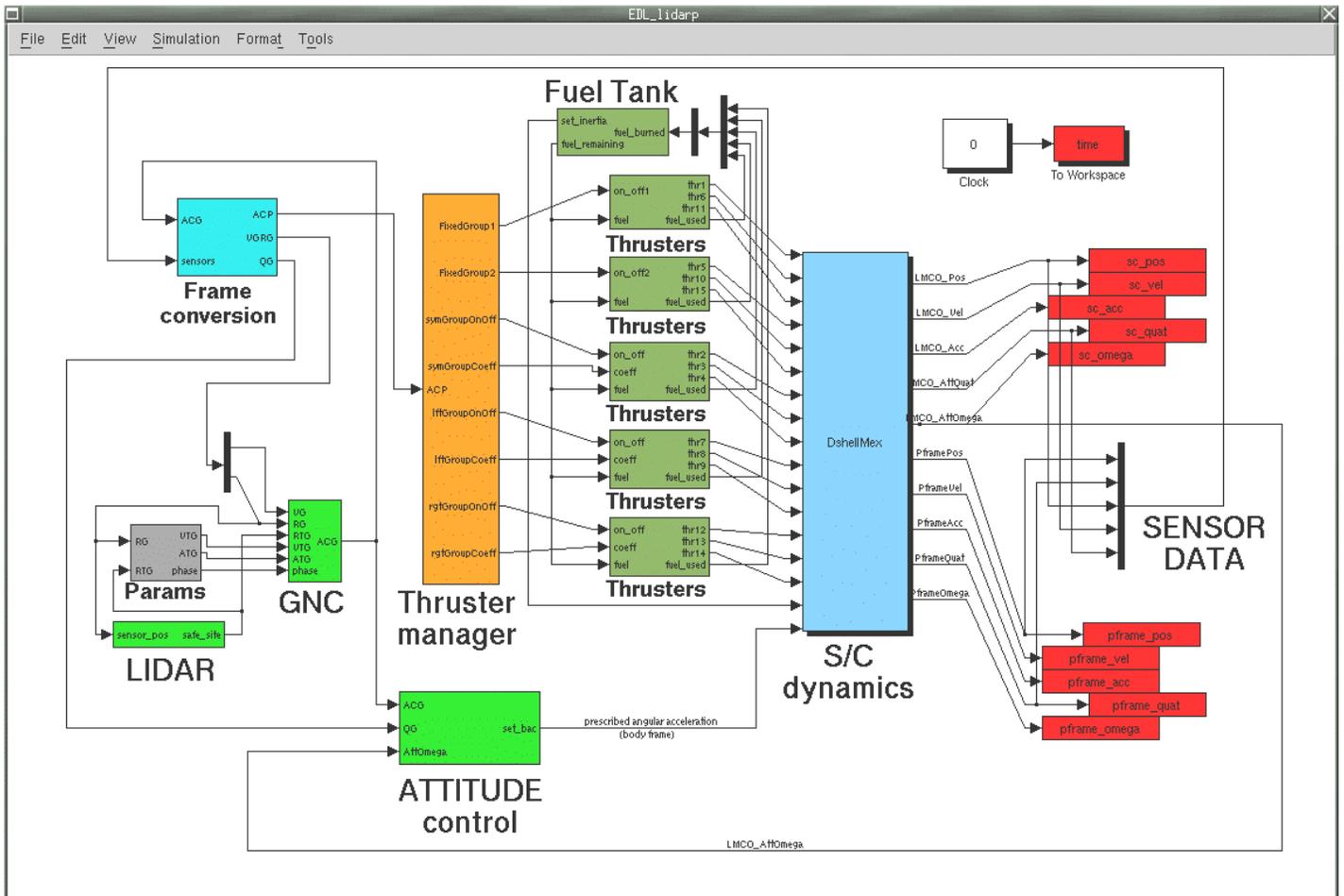

Figure 2 –Matlab/Simulink embedding example of Darts/Dshell dynamics engine and model

Darts/Dshell Simulator

Darts Dynamics Engine

Controller Models or Flight Software/Systems

Bus Or IPC

Entry/Descent/Landing Software

Sensor Processing

G N & C

Environment/Flight Models
- Aerodynamic Forces
- Orbital/Gravity Dynamics
- Digital Terrain Map
  - Wind-Fields
- Lander/Terrain Dynamics

Vehicle & Device Models
- Multibody Solid Models
- Actuators
- Sensors
- LIDAR

Scene Generator
Terrain Visualisation

StethoScope
Real-Time Data Monitoring

Dview
Spacecraft Visualisation

Console | Fault Injection | GUIs
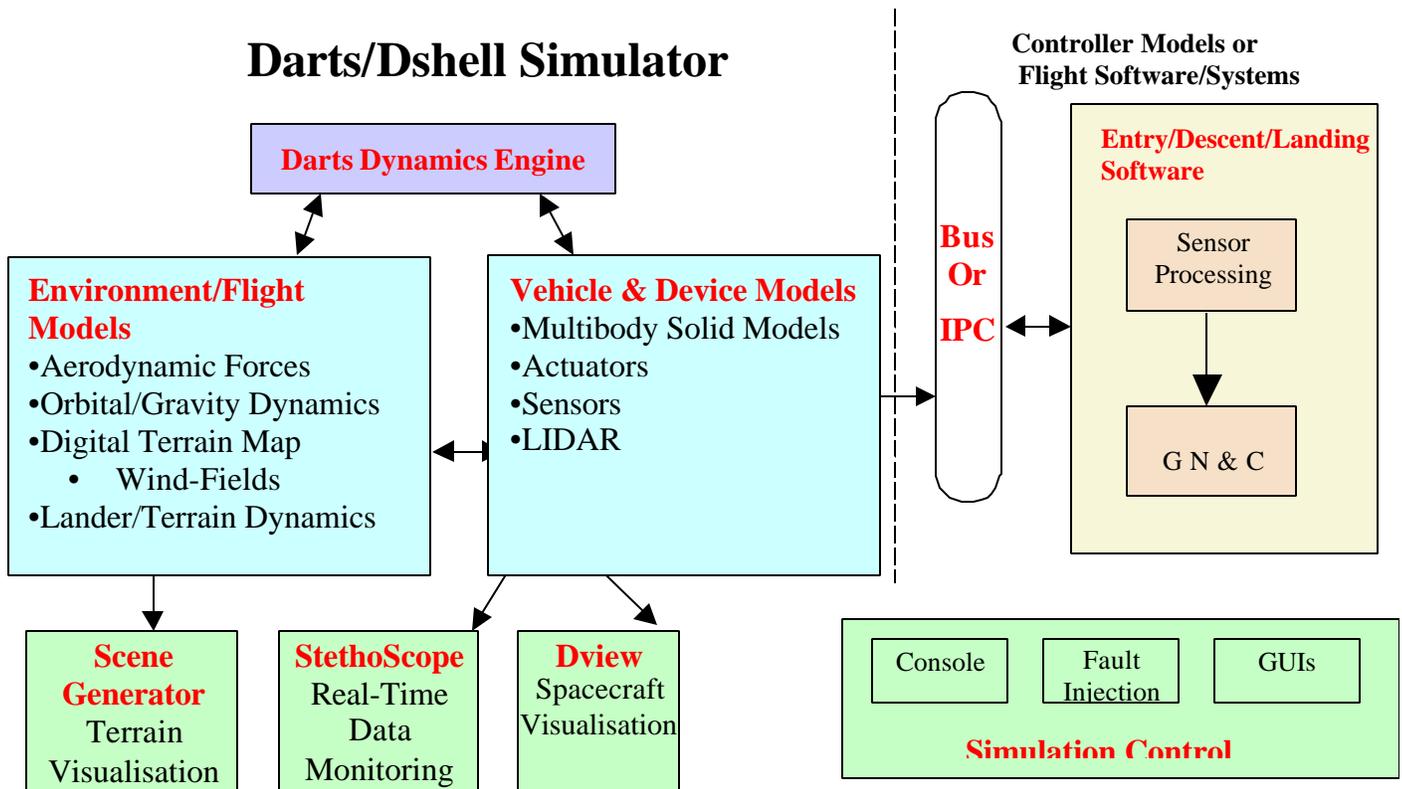
Simulation Control

Figure 3 – Darts/Dshell architecture for flight software testing

regimes. They may also be used as the starting point for developing medium-fidelity aerodynamics models, which are approximations to the high-fidelity models and suffice for early engineering simulation use. DSENDS currently provides a medium-fidelity generic axi-symmetric aero-coefficient model utilizing interpolation routines on a table of aerodynamics coefficients. This capability can be used to provide medium-fidelity entry-body and simple parachute models.

Entry-Body Aerodynamic Models

High-fidelity aerodynamics models better characterize the flight dynamics of an EDL system during its atmospheric traversal and thus tests the adequacy of the GN&C (Guidance, Navigation and Control) algorithms for EDL (e.g. the stability margins for bank-angle control of the vehicle lift vector). For example, post-flight analysis of the Mars Pathfinder (MPF) mission showed that high-fidelity aerodynamics modeling was successful in predicting the MPF 6-DOF (degrees-of-freedom) entry trajectory static instabilities prior to parachute deployment [8][9]. As a consequence, incorporating high-fidelity aerodynamics models into DSENDS is a necessary extension of its core capability for robustly testing simulated EDL systems and their GN&C strategies.

DSENDS currently has a medium-fidelity, axi-symmetric aerodynamics models that applies interpolation routines on a table of aerodynamic coefficients to determine the appropriate coefficient for the given angle-of-attack. The

model then makes a hydrostatic atmosphere assumption to utilize the coefficient in computing forces and moments. DSENDS also has a high-fidelity aerodynamics model developed at the NASA Langley Research Center (LaRC) for the Mars Smart Lander mission that uses Mars-GRAM 3 [10] as its atmospheric model. This same model is also used in LaRC's high fidelity mission analysis tool called POST [11]. This common model library usage between POST and DSENDS ensures the reproduction of the model behavior across both mission analysis (POST) and real-time test (DSENDS) environments.

Atmospheric models

Given an aerodynamic coefficients data object, an EDL system can be flown for different planetary bodies simply by varying the atmospheric model objects (assuming that the flight regimes do not have significant ambient gas dissociation). The selection of the atmospheric model also defines the complexity of the atmospheric structure traversed by the EDL system and thereby bounds the kinds of aerodynamic forces and moments encountered by the EDL GN&C algorithms.

In order to limit the model complexity solved at each simulation time step, the atmospheric models used by DSENDS do not solve meteorological equations of state or motion. The atmospheric models are instead data objects that parametrically model atmospheric conditions, or interpolate on a table of atmospheric properties. Hence, the
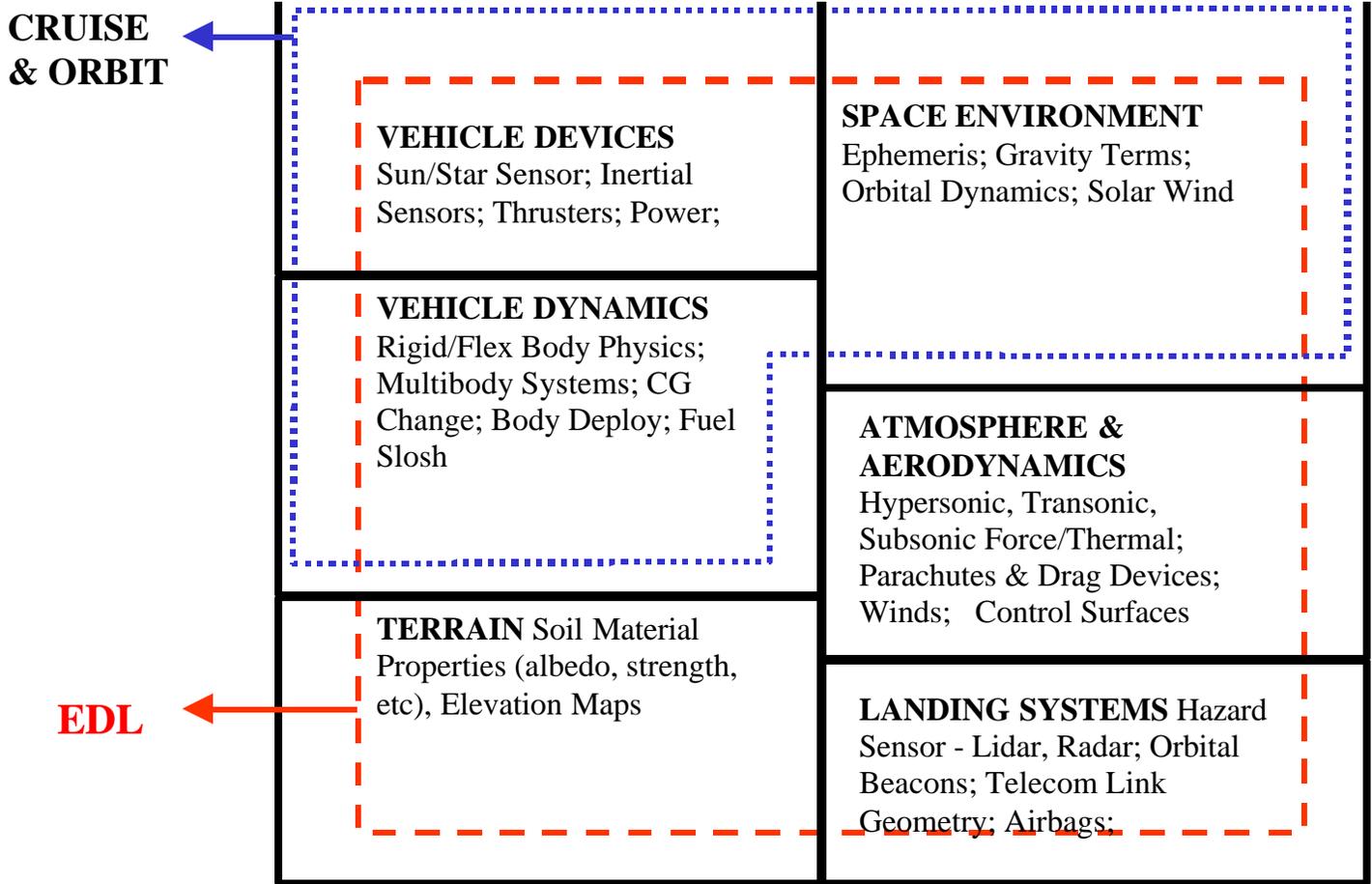
CRUISE & ORBIT

VEHICLE DEVICES
Sun/Star Sensor; Inertial Sensors; Thrusters; Power;

SPACE ENVIRONMENT
Ephemeris; Gravity Terms; Orbital Dynamics; Solar Wind

VEHICLE DYNAMICS
Rigid/Flex Body Physics; Multibody Systems; CG Change; Body Deploy; Fuel Slosh

ATMOSPHERE & AERODYNAMICS
Hypersonic, Transonic, Subsonic Force/Thermal; Parachutes & Drag Devices; Winds;   Control Surfaces

TERRAIN Soil Material Properties (albedo, strength, etc), Elevation Maps

EDL

LANDING SYSTEMS Hazard Sensor - Lidar, Radar; Orbital Beacons; Telecom Link Geometry; Airbags;

Figure 4 – Darts/Dshell model libraries extended to handle EDL simulation.

DSENDS aerodynamic actuator model object can input its position and the atmospheric model will output local density, temperature, global circulation, and local wind gusts. The DSENDS atmospheric model may, in turn, derive some of its properties from other atmospheric models (e.g. Mars-GRAM) encapsulated within it.

Parachute models

The simplest DSENDS parachute model applies a constant drag, with no lift or dependence on angle of attack. To more accurately describe the descent of the entry body, a higher-fidelity parachute model is being developed. Design elements that will be captured in this model come from a variety of sources in the literature [12][13][14]. These include many phenomena that would be desired in a model such as forebody wake effects and aero-elasticity. The aerodynamic properties (tangential, normal and moment coefficients) of a parachute would be obtained from wind-tunnel tests and mechanized in DSENDS using the medium-fidelity actuator model described earlier, and would capture effects of angle-of-attack. Additional information to account for the mass of the air trapped inside the canopy, as well as the air being accelerated along with the parachute (known as included and apparent mass effects, respectively) could be captured by considering the canopy as behaving as a planetary ellipsoid immersed in the flow. More complicated dynamics

such as inflation dynamics, canopy ripple effects due to wind gusts, parachute line twisting, and viscoelastic behavior of the canopy lines requires further parachute model development.

Terrain Models

DSENDS can incorporate terrain models that are represented as Digital Elevation Maps (DEMs). These models can be either a synthetic fractal terrain map [15] or DEMs derived from actual mission data, or combinations of the two.

A key architectural element in DSENDS is a design to allow the rapid switching and fetching of terrain maps as the simulation evolves. To this end, an Instrument Terrain Server will interface the DSENDS simulation model elements (e.g. lidar, radar, altimeter) to a high-speed Terrain Server. The Terrain Server is capable of generating and uploading terrain upon request. Generation of terrain is a relatively slow process and must be requested sufficiently in advance of its use. Uploading of terrain to the requestor at a specified resolution and extent is a relatively fast process. In DSENDS the Instrument Terrain Server constantly monitors the current usage of the DEM by the instrument modules and pre-fetches, based upon a prediction model, the next piece of the DEM that is expected to be used. These pre-fetched DEM segments are located in shared memory and available

for simultaneous use by the various instrument models implemented in the DSENDS system. As the instrument models make requests at the edge of the currently loaded DEM, a fast switch of DEM memory buffers allows the instrument models to transition to the next DEM segment in real-time. If the DEM requests are predicted to be outside of the existing generated set, then a new generation requested is queued to the server.

Lidar/Radar Instrument Simulations

The lidar simulation used by DSENDS is provided as a library of routines that determines the instrument's interaction with a DEM. DSENDS uses a socket interface to a lidar simulation process incorporating this library. The lidar model and associated hazard avoidance algorithms are described in considerable detail elsewhere [16].

Landing Kinematics and Dynamics

DSENDS does not yet implement landing kinematics and dynamics. However, existing and ongoing Darts/Dshell developments in Rover simulation [17] is expected to form the basis of a new landing kinematics and dynamics capability in DSENDS in the near future. This includes the notion of Configuration Kinematics to predict the geometric accommodation and interactions between the lander and the ground, traction and stability measures to assess landed system state and the ability to withstand external forces, and computational geometry algorithms to determine the penetration and separation distance geometry [18]. Empirical models utilizing a generalized spring/damper Dshell actuator will be used to model the reaction forces during impact.

GN&C Modules for Hypersonic and Powered Descent

In stand-alone simulations, it is useful to exercise the simulation by means of simulated GN&C elements. These are implemented as Dshell models and encapsulate the GN&C algorithms in software. They can be easily replaced with either an alternative GN&C model, or with an interface to actual flight algorithms or software.

For the powered descent portion of the simulation, DSENDS incorporates a GN&C guidance/control algorithm based on the Apollo lunar landing algorithm [19]. The implementation provides a simple form of re-targeting based on hazard detection using lidar-derived data. A similar implementation of a hypersonic entry guidance algorithm and bank-angle control is planned for the near future [20].

Automated state-machine model switching

One complicating factor in synthesizing an EDL simulation is keeping track of the potentially large number of configuration-related state changes in the simulation. From transitioning dynamic models, to maintaining the visualization, to performing automated tasks depending on the current state, these changes are often conceptually simple, yet the implementation can be complex. An automated method of maintaining parallel, potentially interrelated states and their required tasks was developed to solve these problems.

The state machine developed for use in DSENDS was based on recreating a subset of the capabilities of existing commercial state-machine products. This facilitates training users in developing the state machine within the simulation and understanding of the operation. The state machine is currently implemented in the Tcl language and is activated at the end of each simulation time-step to manage the required transitions. The state-machine definition is in the form of a text table defining the state names followed by definitions of the associated state transitions and actions. As the Tcl scripting language is used to specify the state transition actions, a powerful and flexible mechanism is provided for the simulation developer to express the desired simulation system behavior. Since the typical state-transitions actions are not computationally intensive, we have found that the interpreted language implementation of the state-machine does not result in an excessive penalty in computing resources.

Visualization

Visualization of the simulated dynamics and environment/spacecraft interactions is an essential part of EDL system design, verification and validation. More so than spacecraft simulations during cruise/flyby mission phases, the number and complexity of EDL spacecraft separation and reconfiguration events (e.g. parachute release, bridle lowering), the complexity inherent in active terrain scanning by onboard instruments (e.g. lidar scans), and spacecraft interactions with a dynamic environment (such as the flight-train buffeted by the ambient wind-fields), requires a high-fidelity, high-performance visualization tool. The visualization environment used by DSENDS is called Dspace, and is based on the open source scenegraph API (Application Programming Interface) called Open Inventor, and takes full advantage of hardware accelerated OpenGL under Solaris, IRIX, and Linux on Intel compatible architectures. Open Inventor was chosen because of its efficient representation and rendering of 3D scene-graphs. Besides, the Open Inventor file format is widely accepted by the computer graphics industry and there are many free and commercial modeling applications that support this format. Availability of full source code was important to us because of the obvious advantage of having the ability to customize the API according to our requirements and Open Inventor was a natural fit under this constraint. The custom-built visualization environment used by DSENDS was designed following a client-server paradigm. This allows physical separation of the visualization system from the dynamics simulator, which is useful for implementing a remote display

over a computer network. The server is written in C++ and has a Tcl wrapper. It utilizes a custom- built communication library with TCP and named pipe support.

4. LIFECYCLE SUPPORT

Standalone simulations are provided on standard engineering workstations running the Solaris, Irix and Linux operating systems. The DSENDS implementation for these simulations requires models for both the spacecraft

Matlab is a multi-platform commercial tool with capabilities including visualization/graphing, advanced integration methods, optimization tools, control system design tools, etc. Simulink provides an interactive, drag-and-drop interface for rapid prototyping of simulations utilizing the so-called s-function blocks that incorporate either Matlab code or compiled external C/C++ code libraries to define each simulation element. Together these constitute a very popular standard analysis tool for system control design,
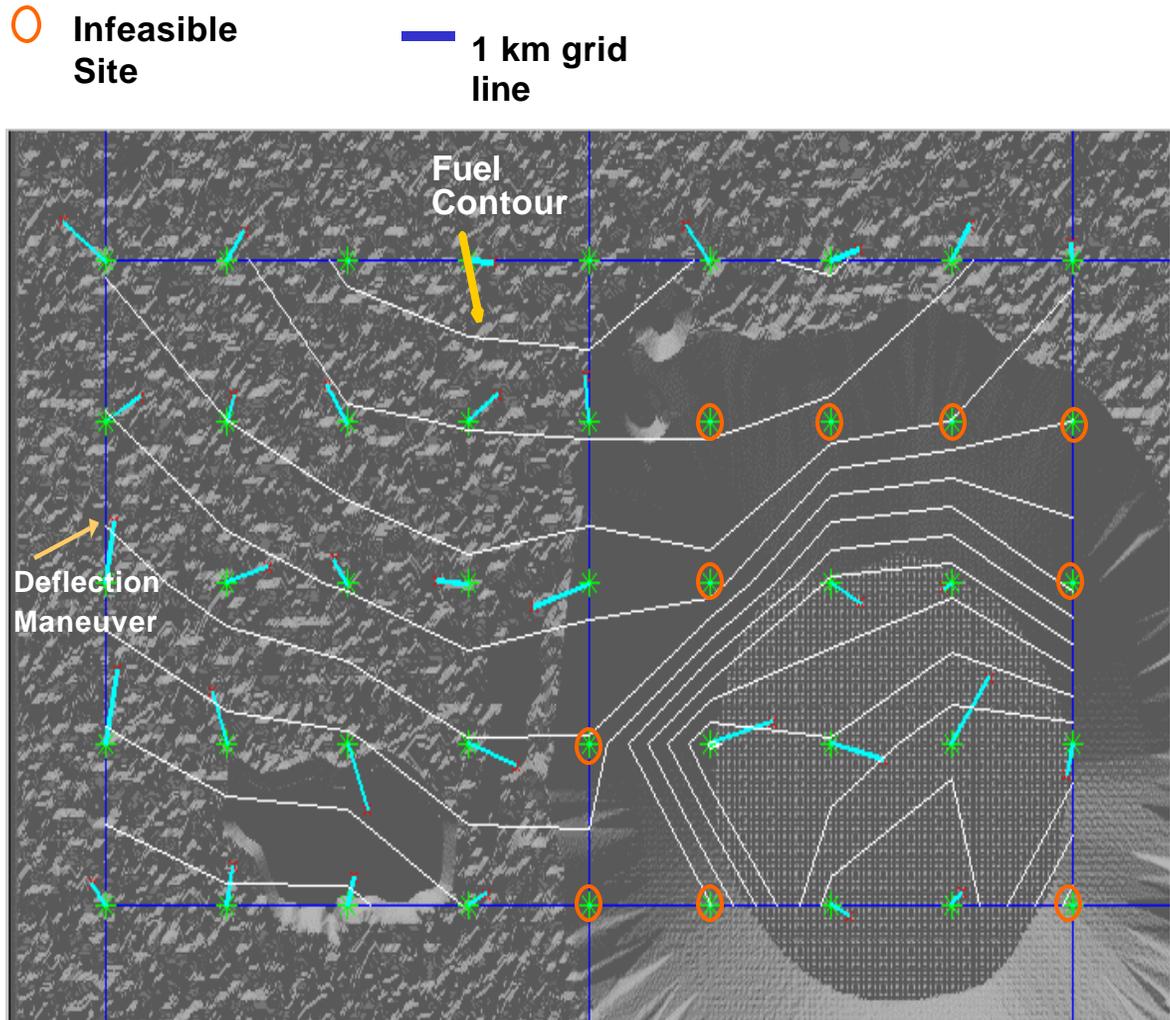

Figure 5 – DSENDS Simulation of Hazard Avoidance Maneuvers and Fuel Consumption.

dynamics and device elements as well as the GN&C components required to operate the spacecraft. DSENDS can also be made to execute in an embedded emulated environment such as VxSim or on actual real-time hardware using VxWorks. In addition to these modes of operation DSENDS supports other modes which have application in other parts of the simulation lifecycle.

Matlab/Simulink Embedding

development, simulation and algorithm testing. By interfacing to Matlab/Simulink, the same Darts/Dshell models can be used throughout the lifecycle of the simulation (from analysis to real-time, hardware in the loop) - simplifying the validation of testbeds by allowing a single set of models to migrate between testbeds. In order to accomplish this, two Darts/Dshell tools are used to embed DSENDS models in Simulink as s-function blocks.

DshellMex is a tool that is used to generate an s-function block representation of Darts/Dshell dynamics and device models from a definition file. This file describes the spacecraft dynamics along with attached instruments and actuators. The inputs and outputs of the s-function block are chosen from a list of available Dshell signals. Internal signals (which are not selected as input/outputs) can be peeked at from the Matlab command line.

ModelMex is used to create an s-function block representation of an individual instrument/actuator model, thereby not requiring that the Darts dynamics engine itself be embedded into Simulink. Inputs and outputs of this block are defined in the device description file.

These s-function blocks, whether they are obtained from ModelMex or DshellMex, are then connected together inside Simulink with other Simulink block elements. An example of these embeddings was shown in Figure 2.

Batch/Monte-Carlo Support

Darts/Dshell supports a batch processing capability where parameters controlling the simulation are systematically varied over a desired range, and state and performance metric variables from the simulation are captured for post-processing analysis. The facility allows for concurrent processing of the simulation on all available processors identified on a master processor list. By choosing the appropriate batch specification, a Monte-Carlo simulation run can be supported.

An example of the results of such a batch run is shown in Figure 5. Here the DSENDS simulation has been repeatedly exercised to obtain information about hazard avoidance maneuvers and fuel consumption for landing operations near a crater. Such data is useful during mission analysis to understand the performance envelope of the system. A grid of prescribed landing sites is given to the DSENDS batch processor and statistics are obtained on the size of a one-shot sense/deflect operation and the fuel required to reach each landing site. The deflections prescribed by the onboard GN&C algorithms are indicated in blue and the white lines represent constant fuel consumption contours. Note that a one-shot sense/deflect maneuver is inadequate for avoiding infeasible sites along the crater wall because of limited lidar field-of-view size.

5. TOOL VERIFICATION

POST

DSENDS is a new tool that will prove valuable for real-time and hardware-in-the-loop simulations of EDL systems. It extends the simulation tools available by incorporating high-fidelity multi-body models, detailed engineering devices, and the ability to support real-time operations. However, until the DSENDS aerodynamics output is verified against a known standard, confidence in the results will be limited. It is for this reason that DSENDS plans to verify its output with POST. POST is a high-fidelity, primarily 6-DOF trajectory simulation tool that has been used since 1970 by LaRC. POST has been used extensively in mission analysis for a variety of missions and was used in the MPF reentry studies [8]. Since DSENDS has been designed to be able to use the same aerodynamics models used by POST, comparative runs against POST will verify the DSENDS output and utility.

Mission Data

DSENDS will also be verified by comparing simulation data outputs with various mission data sets.

In the next year, we will verify the planetary mission EDL simulation capability by modeling the Mars Pathfinder mission and system elements in DSENDS. The simulation will be exercised and the resulting data will be compared against Pathfinder return telemetry and trajectory reconstruction. Since the aerodynamics capability in DSENDS will be independently verified as described earlier, the emphasis here is on verifying the multi-body dynamics and device models. To this end, certain standalone multi-body simulations performed in support of Pathfinder and the Mars 03 mission will be compared to the DSENDS outputs.

We will validate our small-body gravity model and NEAR spacecraft model by simulating orbital and descent paths for the Eros asteroid, and comparing against NEAR mission data archives. The orbital simulations will be used to validate the aggregate properties of the gravity model and the mass properties of the spacecraft. During descent, pre-programmed delta-V maneuvers were used to slow the descent. We will add these as timed open-loop commands to the simulation. The descent simulation will therefore validate both the thruster and mass properties of the spacecraft, as well as the close-in behavior of the gravity model.

Experimental data from Rocket Sled tests

JPL is currently performing ground testing of the lidar instrument and hazard detection algorithms in order to demonstrate that the system will function successfully when traveling at typical Mars descent speeds. Successful operation includes the generation of a terrain map, use of that map for hazard detection, and demonstration that the terminal guidance software works with the sensor data that is generated [21].

To achieve these goals a sled-test program was developed with flight-like and ground-truth sensor mounted atop a vibration isolation bench on a rocket sled. As the sled

travels down the track the lidar images a three dimensional "Mars-like" terrain, and the onboard algorithm determines a safe landing site.

Due to the high cost of sled-test runs, a simulation of the sled-test was developed in Darts/Dshell to supplement the program. Moreover, the sled-test simulation incorporates a number of elements common to the flight system simulation in DSENDS. Cross validation of the sled-test simulation with sled-test data serves to increase the confidence in the models and capabilities of the DSENDS system.

The simulation consists of a dynamic model of the vibration isolation system and socket connection to an external lidar instrument simulation and hazard avoidance software. The simulation is driven by a prescribed motion that was characterized by data from actual sled tests. A Digital Elevation Map of the target is used by the lidar simulation to provide information to the hazard detection algorithm to determine the safe landing site.

Darts/Dshell was used to simulate the dynamics of the two-stage passive vibration isolation system. The first stage consist of a 6-DOF table mounted to the sled base by 8 simulated springs. Four springs are vertically mounted at the corners of the table, one set of opposing springs is horizontally mounted parallel to the rocket sled thrust axis, and another set of opposing springs is horizontally mounted perpendicular to the thrust axis. The spring constants were selected to tune the natural frequencies of the rigid body modes to be around 5 Hz. The actual vibration isolation system has these frequencies ranging from 3 to 8 Hz.

The second stage of the simulated isolation system consists of a bench, which is mounted to the first isolation system via 5 springs. Four are vertically positioned at the corners of the bench and a fifth is a "virtual" zero-length spring located at the center of mass of the bench. The purpose of this spring is to resist translation of the bench, whereas the other four resist rotation and, coupled with the fifth spring, also assist in resisting vertical translation. The springs constants were selected such that the natural frequencies of the second stage were about 25 Hz. For reference, the second stage of the actual system has natural frequencies that range from 20 to 30 Hz.

Simulated accelerometers were placed at spring locations to allow for monitoring of dynamics. These locations are similar to that of the four ground -truth sensors used in the actual test runs. The simulation is driven by an acceleration profile that was based on sled-test data. To provide this stimulation, the Darts/Dshell simulation was embedded within Matlab/Simulink via a custom s-function block and the Dmex facilities described earlier. Simulink was then used to generate the profile and execute the simulation. The acceleration profile includes a 4-second boost phase analogous to the firing of the solid rocket, which accelerates

the sled to a peak of 1.3g. Following that is an exponential roll-off to a sustaining phase where the sled continues to accelerate, but at about a level of 0.3g, for 25 seconds. Finally, the coast phase is entered where friction dominates and brings the sled to rest after 58 seconds.

Output from the simulated accelerometers shows that the dynamics of the vibration isolation system are similar to that of the actual system. Mass property and geometry updates will allow the simulated and actual results to converge better. Once the dynamics are tuned with real data comparing results from actual sled tests including time-domain plots and power spectral densities will validate the simulation.

The lidar simulation communicates with the simulated sled dynamics via the UNIX socket connection. At this time the lidar simulation only samples the dynamics at 1 Hz and uses linear interpolates to model the base trajectory over a 1-second interval. This is inadequate to capture the higher frequency dynamics of the vibration isolation system, but is an interim step before its replacement with a higher fidelity simulation.

6. TOOL USAGE EXAMPLES

Mars Pathfinder & Smart Lander Simulation

DSENDS is being used by the Mars Smart Lander project (currently slated for the 2007+ timeframe) to provide a real-time simulation capability to allow extensive checkout of the flight software in both a workstation as well as an embedded VxWorks test environment. In order to develop this simulator, an end-to-end Mars entry/descent simulation has been completed to exercise the DSENDS system's capabilities. The scenario consists of a Pathfinder style ballistic entry followed by a powered descent landing using active lidar sensing of the terrain. As the mission definition gets refined, the scenario will be updated with the hypersonic GN&C functions as well as specific details of a Mars Smart Lander system configuration. In addition, a phased-array imaging radar simulation will also be incorporated. In the meantime, the existing simulation suffices to develop the real-time system interfaces and conduct the computational performance studies in preparation for the development of the real-time testbed.
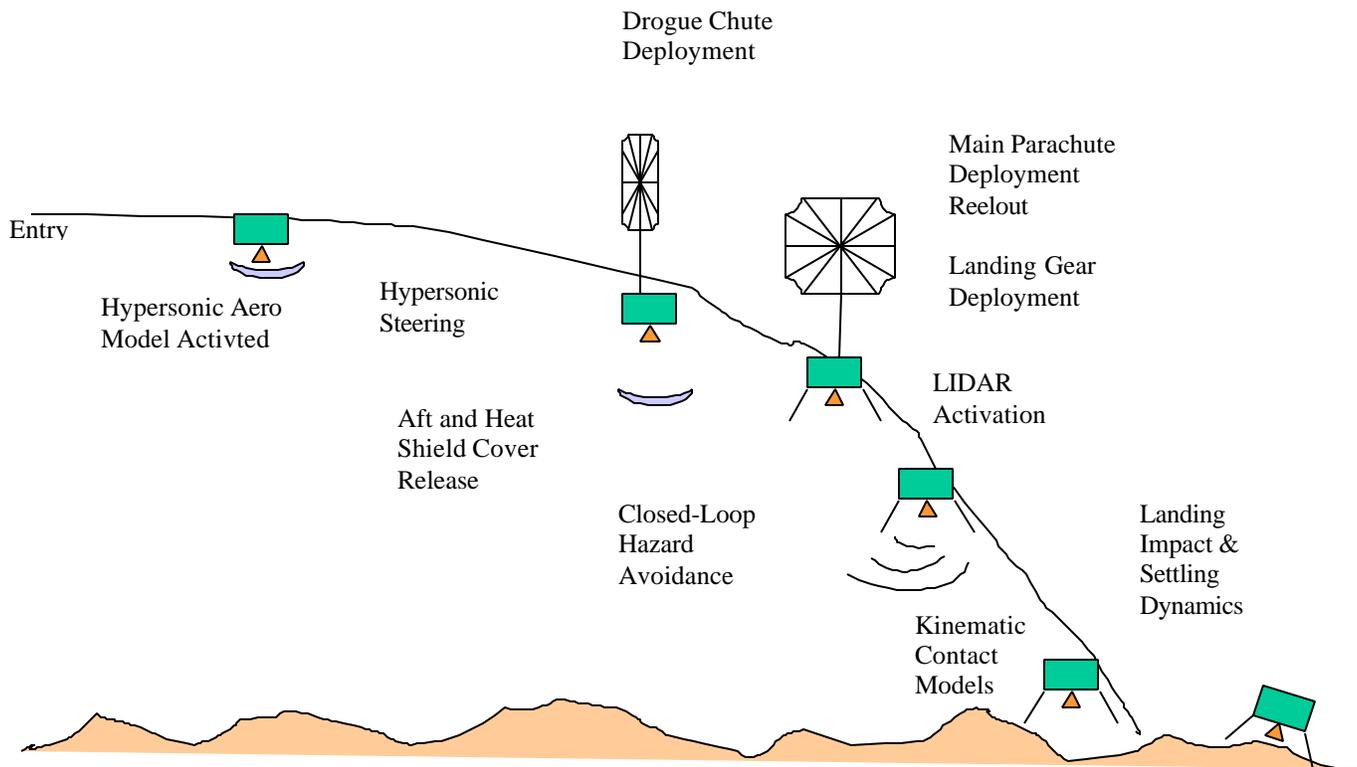
Figure 6 - This represents the sequence of high-level events during an example Mars entry sequence DSENDS simulation

The specific scenario is illustrated in Figure 6 which depicts key events along the descent. The specific sequence has been chosen to exercise the simulator's capabilities in Multibody Dynamics, Pendulum Dynamics, Center-of-Gravity Shifts, Thruster Effects, Shield/Cover/Leg Deploy, Lidar In-the-loop, Terrain Models, Landing Geometry, Fuel Depletion, Trajectory Propagation, Closed-Loop Control, Entry Aerodynamics and Parachute behavior. Landing forces and kinematics have not yet been implemented. The succession of spacecraft models that are active during the simulation are depicted in Figure 7. A cruise model transforms after parachute deployment to include a tether pendulum multi-body system. During the EDL a bridle lowering of the lander is represented by adding additional multi-body elements. The heatshield and the lander under powered descent are also modeled as separate entities. A series of screen shots in Figures 8 – 13 from the simulation illustrate various aspects of the entry and descent.

Titan EDL Mission Technology Infusion Study

DSENDS has been used to study the applicability of advanced, next-generation EDL technologies to a Titan lander mission. The nominal mission scenario calls for a ballute decelerator system used in an aerocapture maneuver, followed by a smart delivery system using a controlled parafoil and lidar/radar guided powered descent. The 400 kg orbiter separates from the ballute/lander combo at 1800 m/sec at an approximate altitude of 640 km. A periapsis raise maneuver completes aero-capture. The inflatable ballute

(see Figure 14) represents the next generation EDL technology, and consists of a 20 meter radius, "donut" shaped inflatable device. The effective area is 400 m^2 and the nominal coefficient of drag is 2. The ballute is deployed at an altitude of 1000 km. Entry flight-path angle is approximately 28 degrees with an entry velocity of 6 km/sec. The ballute is jettisoned from the lander at an altitude of about 400 km. Then a small parachute (4-6 m^2) is deployed for fast descent through the thick Titan atmosphere and is jettisoned around 40 km. A parafoil (approx. 100 m^2) is then deployed and used for correction of entry errors for precision landing. It has a glide ratio of 4:1. A descent imager is activated at parafoil deploy and images are obtained every 3 to 4 seconds. Pattern matching algorithms are used to identify the landing site and provide a vector-to-target input to the onboard GN&C for parafoil control. Powered descent is initiated at the terminal phase with lidar for hazard avoidance.

GN&C Sensor Testbed

This represents a new use of DSENDS in support of the development of a GN&C Sensor testbed. The objective is to provide new capabilities for mission planning and trade studies. The testbed can be used to fully explore the capabilities of a sensor, and also quantify its performance characteristics in a realistic environment and under expected conditions. This exploration may be performed either with pre-determined mission profiles, or using closed-loop control using representative guidance algorithms to validate performance and feedback issues during maneuvers.

Utilizing this capability and applying it to a wide range of descent sensors, the testbed will provide tabulated results comparing various sensors against a selected set of metrics. Test scenarios can either be selected to represent a wide range of possible conditions (and therefore provide good aggregate behavioral comparisons), or to simulate expected nominal and off-nominal profiles for a few mission profiles (and thus provide metrics for a narrow, but focused subset of requirements and disturbances).

Small Body GN&C

This also represents a new use of DSENDS to support the development of a testbed in support of developing small-body mission GN&C algorithms. The testbed will be developed in collaboration with the team developing a hovering control system for small-body exploration.

MER Trajectory Reconstruction

Another proposed use of DSENDS is to support the verification and validation of trajectory reconstruction algorithms and filters that will be deployed for reconstructing the trajectories of the Mars 03 spacecrafts. The Mars 03 mission calls for twin spacecraft arriving at Mars within an interval of a few weeks. Reconstructing the trajectory and events after the first landing allows valuable insight into system performance that could be used to guide the landing of the second lander. DSENDS will provide a variety of simulated engineering data as well ground-truth products in support of the trajectory reconstruction algorithm and software development. The data products also include those resulting from exercising the DSENDS simulation to develop sensitivity matrices that can be used within the reconstruction filters.
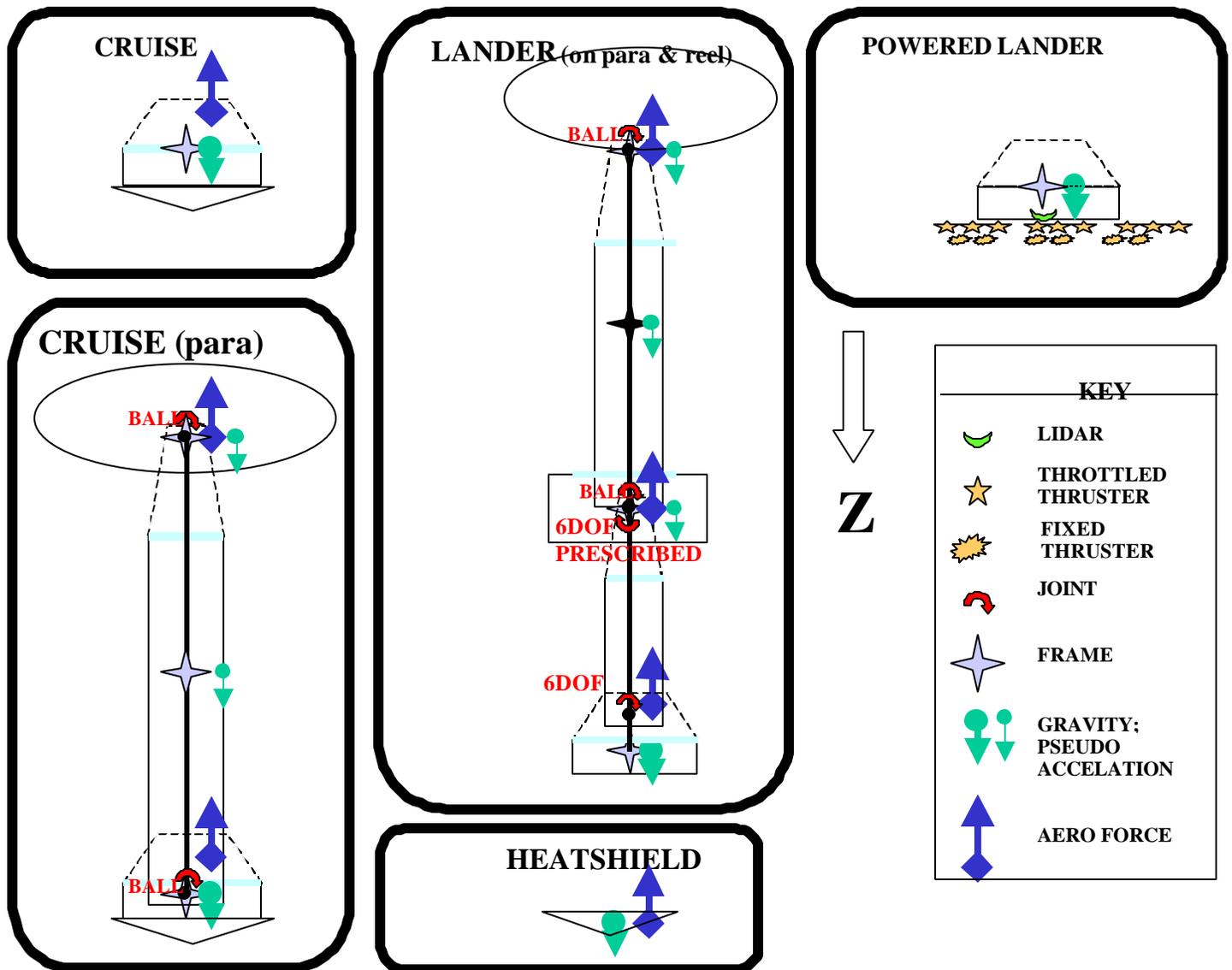

Figure 7 - This schematic representation shows the various spacecraft dynamics model that are active during different phases of EDL. Darts/Dshell manages the correct transition dynamics and system state during all of these model transformations.

7. FUTURE WORK

Future work in DSENDS will address the following areas:
- Tool verification and validation
- Small-body EDL capability
- Model libraries for aerodynamics and aero-thermal effects
- Interfaces to a variety of environmental models including updated versions of MarsGram and small-body environments
- High-fidelity implementations of flight-train dynamics including complex parachute models and viscoelastic line elements
- Preliminary landing kinematics and dynamics models

8. CONCLUDING REMARKS

We have presented a description of a multi-mission, high-fidelity, physics-based simulator that can support spacecraft EDL system and mission development across the entire design lifecycle of a mission. As EDL missions evolve in sophistication to accommodate the next generation of challenging missions, the availability of such simulations allows for the robust testing of flight software and a reduction in mission risk.

ACKNOWLEDGMENTS

REFERENCES

[1]. NASA/JPL Mars Program Website. *Mars: Missions – 2005 and Beyond,* available at the following URL: http://mars.jpl.nasa.gov/missions/future/2005-plus.html

[2]. NASA Office of Space Science. *Space Science Enterprise 2000 Strategic Plan* available at the following URL: http://spacescience.nasa.gov/admin/pubs/strategy/2000/index.html

[3]. A. Jain and G. Rodriguez, "Recursive Flexible Multibody System Dynamics Using Spatial Operators," *Journal of Guidance, Control and Dynamics,* vol. 15, pp. 1453-1466, Nov. 1992.

[4]. J. Biesiadecki, D. Henriquez and A. Jain, "A Reusable, Real-Time Spacecraft Dynamics Simulator," in *6th Digital Avionics Systems Conference*, (Irvine, CA), Oct 1997.

[5]. J. Biesiadecki, A. Jain, and M.L. James, "Advanced Simulation Environment for Autonomous Spacecraft," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'97)*, (Tokyo, Japan), Jul. 1997.

[6]. J. Biesiadecki and A. Jain, "A Reconfigurable Testbed Environment for Spacecraft Autonomy," in *Simulators for European Space Programmes, 4th Workshop*, (Noordwijk, The Netherlands), ESTEC, Oct. 1996.

[7]. J. K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley, 1994.

[8]. Braun, R. D. et al, "Six Degree-of-Freedom Atmospheric Entry Analysis for the Mars Pathfinder Mission," *AIAA* 95-0456, 1995

[9]. Gnoffo, P. A. et al, "Prediction and Validation of Mars Pathfinder Hypersonic Aerodynamic Data Base," *AIAA* 98-2445, 1998

[10]. C.G. Justus et al. "Mars Global Reference Atmospheric Model 2000 Version (Mars-GRAM 2000): Users Guide," *NASA* TM 2000-210279. May 2000.

[11]. Brauer, G. L.., Cornick, D. E. & Stevenson, R., "Capabilities and Applications of the Program to Optimize Simulated Trajectories (POST)," *NASA CR-2770*, 1977.

[12]. Fallon, E.J., "Parachute Dynamics and Stability Analysis of the Queen Match Recovery System," *AIAA 91-0879*, 1991.

[13]. Fallon, E.J., "System Design Overview of the Mars Pathfinder Parachute Decelerator Subsystem," *AIAA-97-1511*, 1997.

[14]. Cockrell, D.J., "The Aerodynamics of Parachutes," *AGARD*, 1987.

[15]. R. Gaskell, "Martian Surface Simulation," *Journal Geophysical Research – Planets*, 98 (E6), pp. 11099-11103, 1993.

[16]. A. Johnson, A. Klumpp, J. Collier and A. Wolf, "Lidar-based Hazard Avoidance for Safe Landing on Mars," *AAS/AIAA Space Flight Mechanics Meeting*, February 2001.

[17]. J. Yen, A. Jain and J. Balaram, "ROAMS : Rover Analysis, Modeling and Simulation Software," *Fifth International Symposium on Artificial Intelligence and Automation in Space,* Noordwijk, The Netherlands, 1-3 June 1999.

[18]. Stephen A. Ehmann and Ming C. Lin, "Accurate and Fast Proximity Queries between Polyhedra Using Surface Decomposition," To appear in *Computer Graphics Forum Proc. of Eurographics*, 2001.

[19]. A.R. Klumpp, "Apollo Lunar-Descent Guidance*," Report R-695, 1971, MIT Charles Stark Draper Laboratory*, Cambridge, MA.

[20]. G. Carman, D. Ives and D. Geller, "Apollo-Derived Mars Precision Lander Guidance," *AIAA Atmospheric Flight Mechanics Conference,* Aug 10-12, 1998, Boston, MA.

[21]. Skulsky, E.D., Johnson, A.E., et al, "Rocket Sled Testing of a Prototype Terrain-relative Navigation System," *AAS 01-026.*

J. (Bob) Balaram received his Ph. D. in Computer and Systems Engineering from Rensselaer Polytechnic Institute. He works at the Jet Propulsion Laboratory in the area of telerobotics and spacecraft simulation technology development in the Avionics Systems Division. He is active in the

technology programs for Mars EDL, Rovers, and multi-mission, high-fidelity Spacecraft & Rover simulators.

Richard Austin received his BS degree in Honors Physics from the University of Massachusetts, Amherst. He is currently employed in the Mission and Systems Architecture Division, at the Jet Propulsion Laboratory, in Pasadena, California. Mr. Austin has worked on a variety of space and radar missions, primarily in the areas of science and mission planning. He was Deputy Team Chief of the Magellan mission operations and science team, and the architect of the mission and data acquisition plan for the joint NASA-Canadian space agency Antarctic Mapping Mission for which he received a NASA Group Achievement Award. He has also worked in the science and mission planning on the Galileo, Mars Pathfinder, and Voyager missions. Prior to coming to JPL, he was a Senior Engineer at the Hughes Aircraft Co. where he did extensive work in the design and development of the Ku-Band Space Shuttle Radar.

Pranab Banerjee received his B.Tech. in Electronics and Electrical Communication Engineering from Indian Institute of Technology, Kharagpur, India and MS in Computational Physics from Kansas State University, USA. He is currently a Member of Information Systems and Computer Science Staff at the Jet Propulsion Laboratory. His research interests include scientific data visualization, distributed computing, interactive visualization of large data sets, parallel rendering algorithms, multiresolution techniques and digital image processing.

Timothy Bentley is a Senior at the University of Southern California majoring in Aerospace Engineering. He is an Academic Part Time employee at the Jet Propulsion Laboratory.. He is the president of the USC AIAA Student Section and a member of the USC aeronautical design team. His interests include aerodynamics and fluid flows.

David Henríquez received B.S. and M.S. degrees in Aerospace Engineering from the University
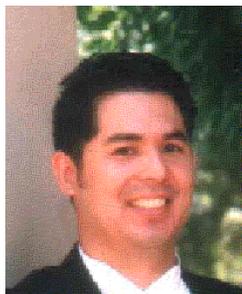
of Southern California in 1994 and 1999. He started his career at The Aerospace Corporation in 1992, where he was a developer on a GPS autonomous navigation simulator. He has been at the Jet Propulsion Laboratory since 1996 as a member of the Software Simulation and Modeling Group in the Autonomy and Control Section. While at the Jet Propulsion Laboratory, he has been involved in development of spacecraft simulations for a variety of flight projects (e.g. Cassini, Europa Orbiter, SIM, Stardust). He is currently collaborating to develop a simulator for a joint JPL and CNES autonomous rendezvous technology demonstration project (CNES '07) and aerodynamics modeling capability for use in Entry, Descent and Landing missions like the Mars Smart Lander flight project.

Bryan Martin is a Senior Engineer at the Jet Propulsion Laboratory. He received his Ph.D. in Mechanical and Aerospace Engineering at the University of California, Irvine in 1996. He currently works in the Simulation and Modeling group, and is active in technology programs for Mars EDL, Mars Rendezvous and Sample Capture, simulation of large formations, and multi-mission, high-fidelity spacecraft simulators. His research interests include: advances in simulation, optimization of dynamic motions, and analytical dynamics.

Eli McMahon received his M.S. in 1989 and B.S. in 1987, both in Mechanical Engineering from the Massachusetts Institute of Technology. He is a member of technical staff at the Jet Propulsion Laboratory in Pasadena, California where he is currently the lead simulation developer for the Europa Orbiter project and supports Mars entry, descent and landing (EDL) simulation tasks. He was the task leader for the Deep Space One simulation effort, and has also served as Group Leader and Analysis Team Leader for the Galileo Attitude, Articulation and Control Subsystem Group.

Garett Sohl received a B.S. in mechanical engineering from Rice University in 1993. He then attended the University of California, Irvine and received a masters (1996) and Ph.D.(2000) in

mechanical engineering. His Ph.D. work employed a Lie algebra based multi-body dynamics formulation to generate optimal motions for underactuated mechanisms. He is currently an employee of the Jet Propulsion

Descending Status:	Time(s)	Velocity(m/s)
	89	−333 −3273 −1123
Parachute Release		
Heat Shield Separation		
Tether Reel−out	−	−
Tether Reel−out Complete	−	−
Heat Shield Impact	−	−
Landing Body Separation	−	−
Powered Descend Phase		
Target Redesignation	−	−
Terminal Landing Phase	−	−

StartWithLidar   StartWithoutLidar   Stop   Init Dvi

"North Knob"   inder
"Twin Peak   nall Crater
                          g Cra

AxialGLoad simtime(0)

Height(0) simtime(0)

Descending Status:	Time(s)	Velocity(m/s)
	144	−105 −319 −108
Parachute Release	137	−149 −545 −185
Heat Shield Separation		
Tether Reel−out	−	−
Tether Reel−out Complete	−	−
Heat Shield Impact	−	−
Landing Body Separation	−	−
Powered Descend Phase		
Target Redesignation	−	−
Terminal Landing Phase	−	−

StartWithLidar   StartWithoutLidar   Stop   Init Dvi

inder
nall Crater"

"North Knob"   inder
"Twin Peak   nall Crate
                          g Cra

AxialGLoad simtime(0)

pendulumOmega(0) simtime(0)   pendulumOmega(1) simtime(0)   pendulumOmega(2) simtime(0)
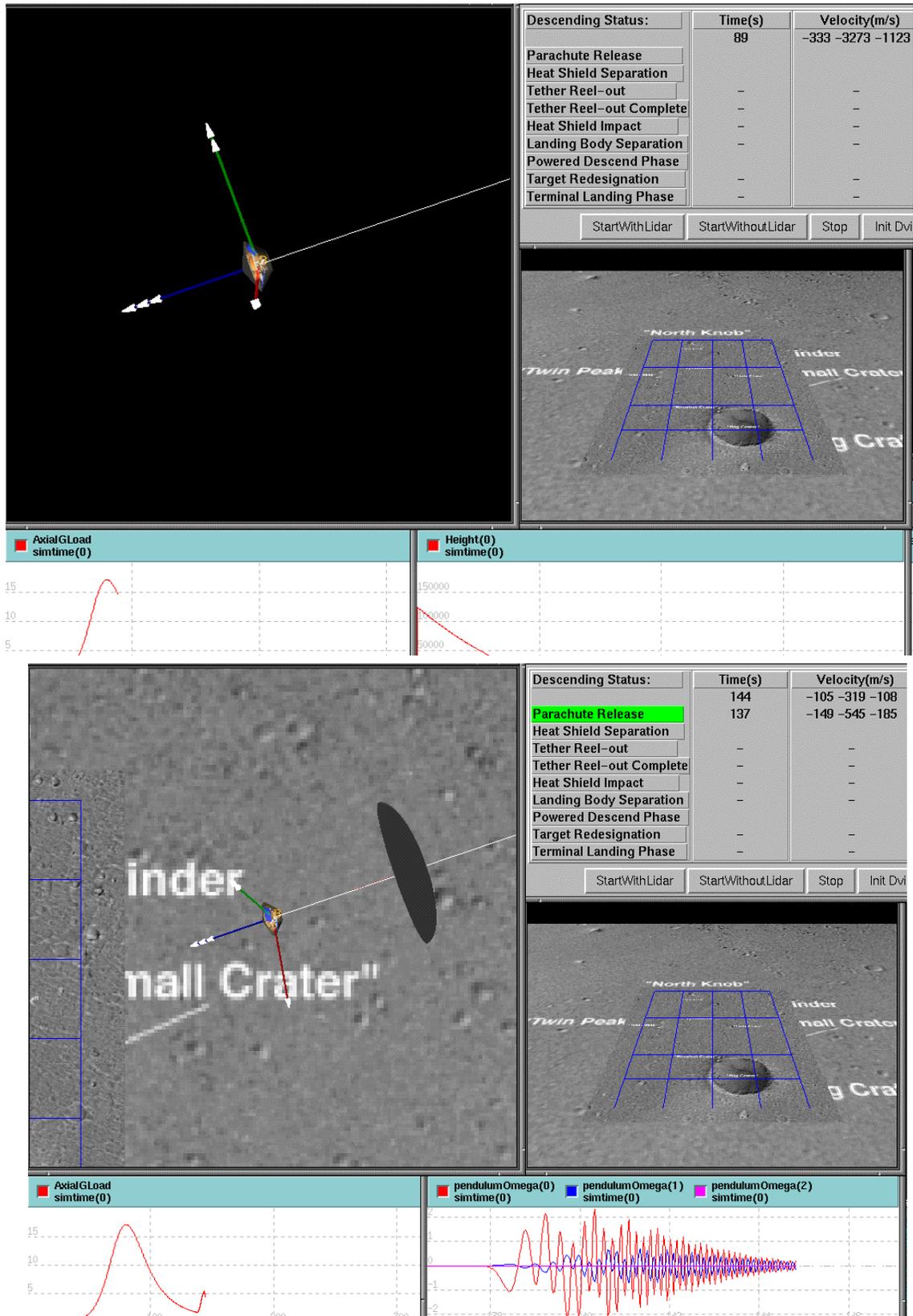
Figure 9 - The chase-camera view shows an iconic representation of the parachute deployment. Notice the the induced g-forces caused by the parachute release. In the lower right window a depiction of some of the pendulum dynamics is shown that result from a simulated off-axis deployment of the parachute.
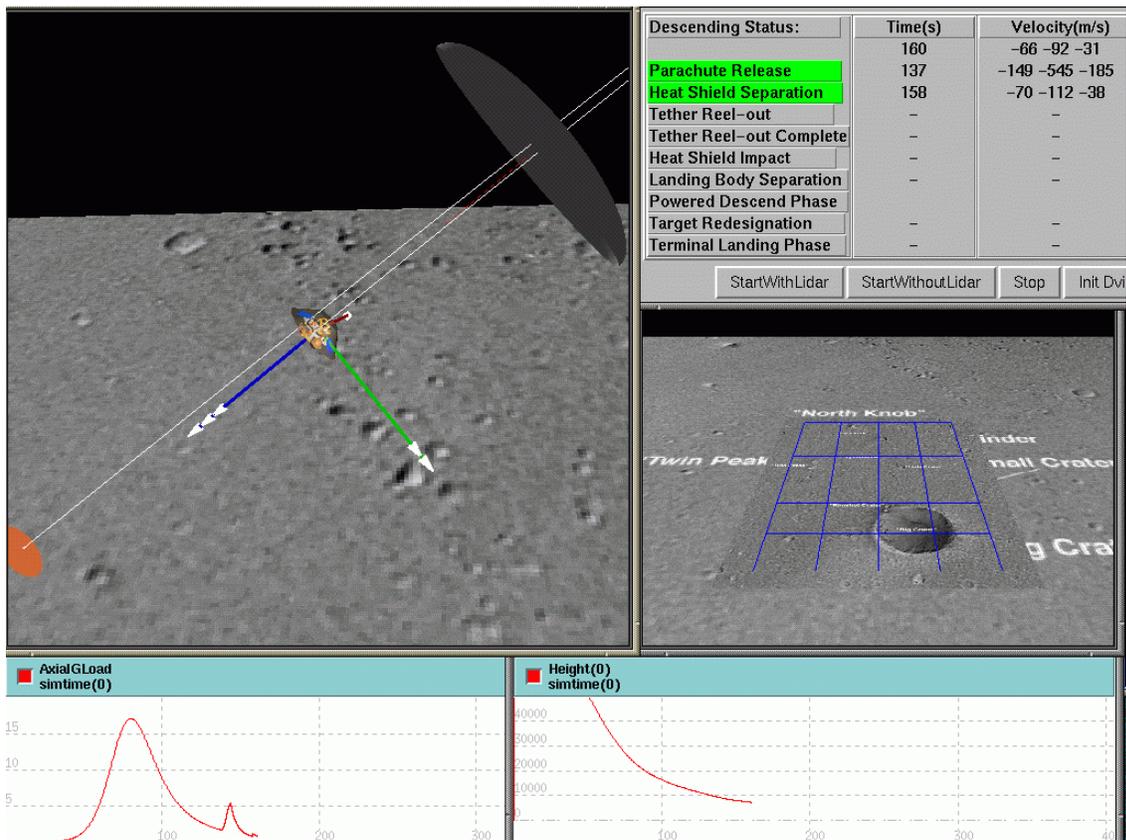
Figure 10 - The heatshield has been jettisoned as indicated by trails in the chase-camera view. The separation dynamics results in a g-force transient that is noticeable in the lower-left plot.
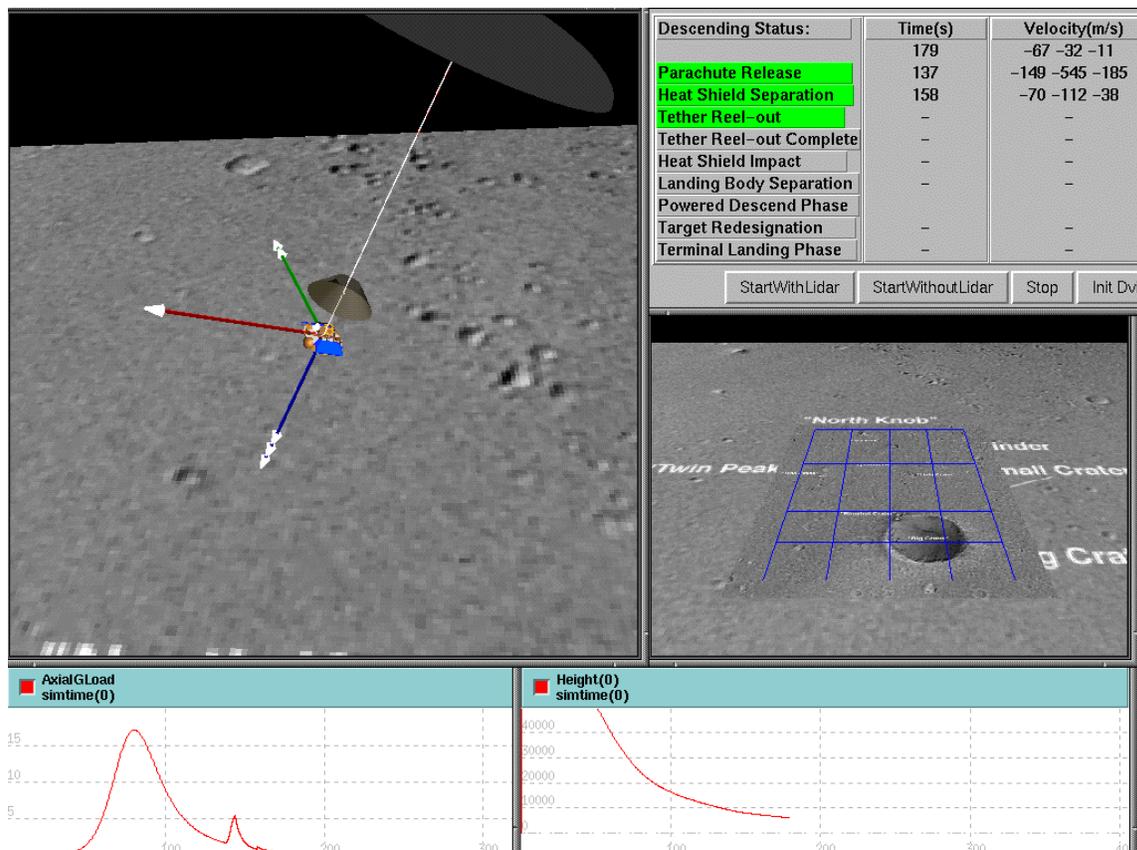

Figure 11 - Here DSENDS is simulating the lowering of the lander from the aft-cover using a bridle.

Descending Status:	Time(s)	Velocity(m/s)
	221	−73 −1 −4
Parachute Release	137	−149 −545 −185
Heat Shield Separation	158	−70 −112 −38
Tether Reel−out	−	−
Tether Reel−out Complete	−	−
Heat Shield Impact	−	−
Landing Body Separation	−	−
Powered Descend Phase	209	−67 −6 −2
Target Redesignation	−	−
Terminal Landing Phase	−	−

StartWithLidar StartWithoutLidar Stop Init Dvi

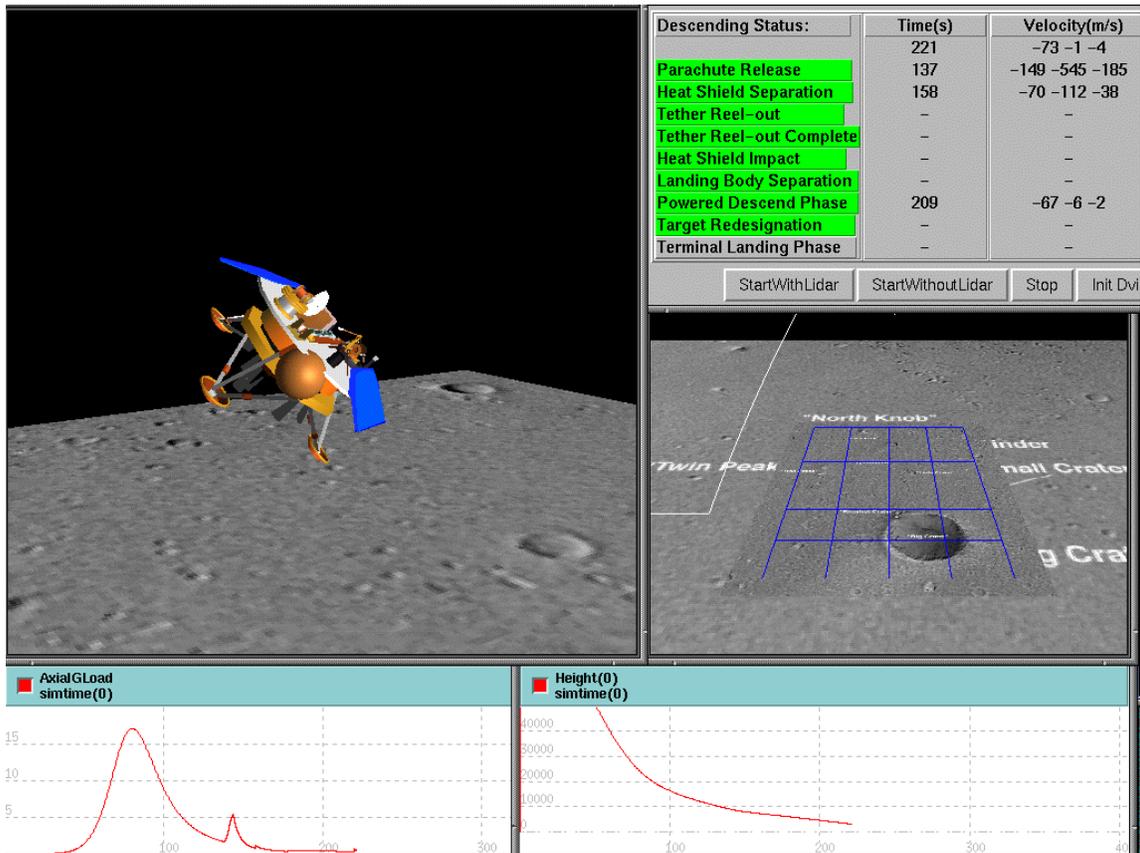AxialGLoad simtime(0)

Height(0) simtime(0)

Figure 12 - After spacecraft separation, a tip-up maneuver is performed to change the thrust vector direction. Note the white trail in the landing-site view that depicts the trajectory of the heatshield that was released earlier.

Descending Status:	Time(s)	Velocity(m/s)
	252	−55 18 −19
Parachute Release	137	−149 −545 −185
Heat Shield Separation	158	−70 −112 −38
Tether Reel−out	−	−
Tether Reel−out Complete	−	−
Heat Shield Impact	−	−
Landing Body Separation	−	−
Powered Descend Phase	209	−67 −6 −2
Target Redesignation	−	−
Terminal Landing Phase	−	−

StartWithLidar StartWithoutLidar Stop Init Dvi

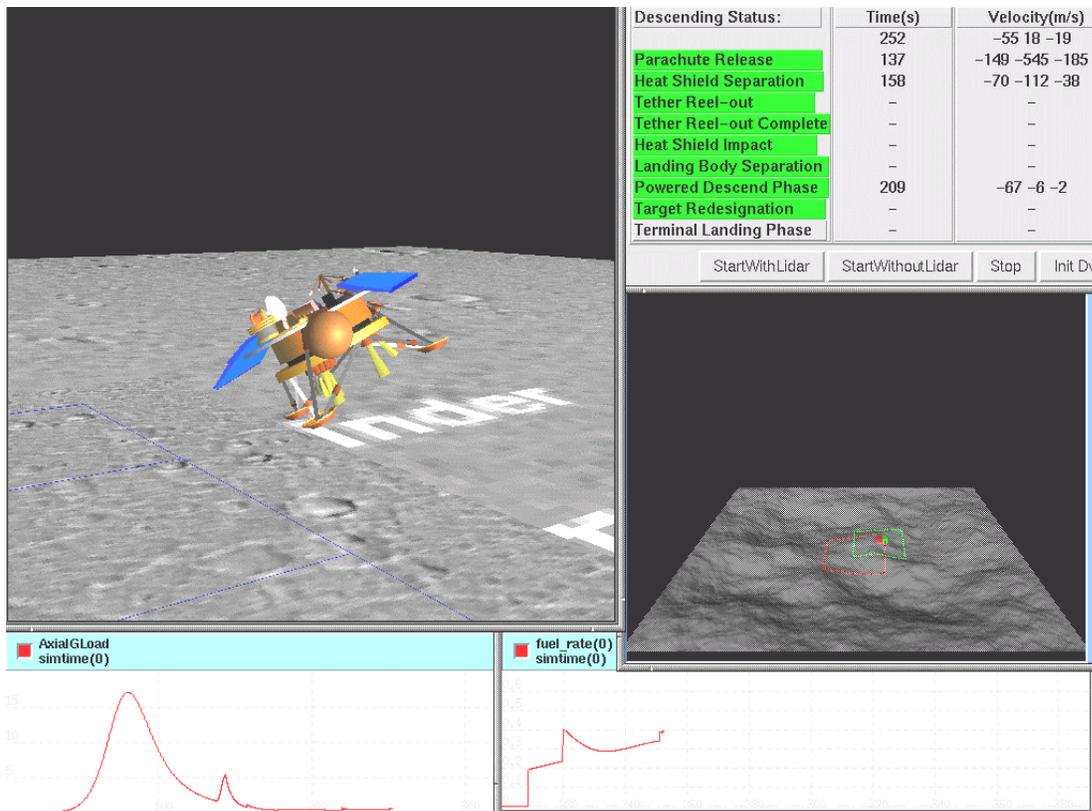AxialGLoad simtime(0)

fuel_rate(0) simtime(0)

Figure 13 - In this figure we provide a close-up view of the lidar scan footprints on the landing site. Note that the initial scan (in red) is displaced from the second scan (in green) indicating the attempts by the on-board GN&C to select a better landing site from the initially targeted point. Note too the fuel-consumption rates that are depicted in the lower-right window as the spacecraft performed various tip-up and throttle maneuvers to achieve the re-designated landing sites.
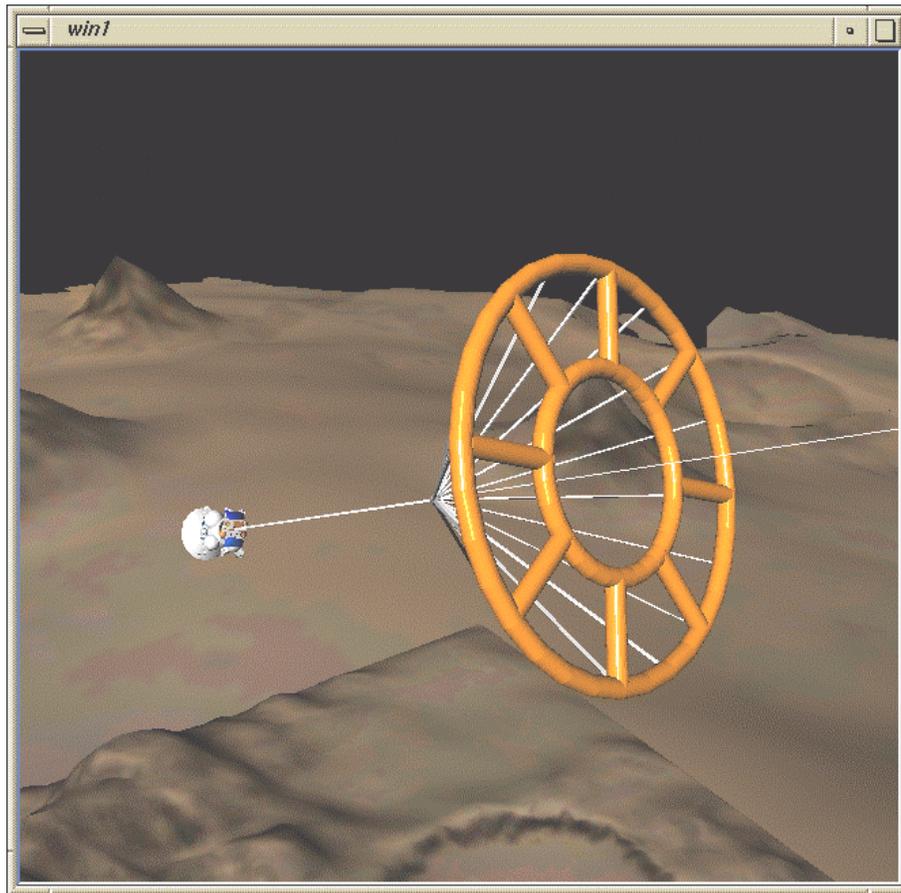
Figure 14 - Titan EDL advanced technology demonstration simulation showing entry using an inflatable ballute aero-decelerator.