

MARS SMART LANDER SIMULATIONS FOR ENTRY, DESCENT, AND LANDING

S. A. Striepe,* D. W. Way,* A. M. Dwyer,* and J. Balaram†

Abstract

Two primary simulations have been developed and are being updated for the Mars Smart Lander Entry, Descent, and Landing (EDL). The high fidelity engineering end-to-end EDL simulation that is based on NASA Langley's Program to Optimize Simulated Trajectories (POST) and the end-to-end real-time, hardware-in-the-loop simulation testbed, which is based on NASA JPL's Dynamics Simulator for Entry, Descent and Surface landing (DSEENDS). This paper presents the status of these Mars Smart Lander EDL end-to-end simulations at this time. Various models, capabilities, as well as validation and verification for these simulations are discussed.

NOMENCLATURE

AGL	Above Ground Level
CFD	Computational Fluid Dynamics
DEM	Digital Elevation Map
DOF	Degree of Freedom
DSEENDS	Dynamics Simulator for Entry, Descent and Surface landing
EDL	Entry, Descent, and Landing
ETPC	Entry Terminal Point Controller
GCM	Global Circulation Model
IMU	Inertial Measurement Unit
JPL	Jet Propulsion Laboratory
Mars-GRAM	Mars Global Reference Atmosphere Model
MOLA	Mars Observer Laser Altimetry
MSL	Mars Smart Lander
POST	Program to Optimize Simulated Trajectories

INTRODUCTION

The Mars Smart Lander (MSL) mission will deliver an advanced rover to a specified Martian site with an accuracy never before achieved by using a guided aeromaneuvering lander. [Ref. 1] Currently slated for a

*Aerospace Engineer, Vehicle Analysis Branch,
NASA Langley Research Center.

†Principal Member, Avionics Systems Division,
NASA Jet Propulsion Laboratory.

Copyright © 2002 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental Purposes. All other rights are reserved by the copyright owner.

2009 launch, this mission uses an onboard guidance during entry and terminal descent phases to direct the spacecraft to the desired landing site, while avoiding any surface hazards detected by the onboard systems. The guidance controls the vehicle through lift vector modulation during the entry phase and thrust vector pointing during the terminal descent. Figure 1 shows an illustration of the Entry, Descent, and Landing (EDL) for the Mars Smart Lander. This mission will demonstrate some of the precursor steps required for a Mars sample return mission.

Current approaches to EDL at Mars have focused on unguided and uncontrolled ballistic entry with no precision landing or hazard avoidance capabilities. The new generation of "Smart" Landers with their lifting body designs, aerodynamic steering, active terrain sensing, and powered-descent/precision-landing capabilities requires a new generation of simulation technology to support their development. The EDL systems are required to successfully deliver the surface systems through the entry, descent, and landing phases, which begin at lander separation from the cruise stage and end at touchdown on the surface. For MSL, these EDL systems will be designed, developed, tested, and evaluated using end-to-end high fidelity computer flight simulations developed by a team of engineers from NASA-JPL, NASA-JSC, NASA-Langley, and the University of Texas at Austin.

Two primary EDL simulations have been developed from existing tools and are being updated specifically for the MSL project. The first of these simulations is the high fidelity engineering end-to-end EDL simulation, which is based on NASA-Langley's Program to Optimize Simulated Trajectories (POST) [Refs. 2 and 3]. The second program is the end-to-end real-time,

hardware-in-the-loop simulation testbed, which is based on NASA-JPL's Dynamics Simulator for Entry, Descent and Surface landing (DSEENDS) [Ref. 4]. Models of various MSL EDL systems are provided by groups responsible for their technology or flight development. Initially, these models are incorporated into the high-fidelity engineering (POST-based) simulation for evaluation and development. Eventually, these models, as well as flight-ready algorithms and hardware, will be incorporated into a real-time simulation capable of supporting detailed dynamics, device models and hardware-in-the-loop simulations (Mars-DSEENDS) for flight operation evaluation, testing, and refinement. During the Mars Smart Lander mission, both simulations will support flight operations and anomaly resolution. This paper describes these two end-to-end simulations as well as the models, algorithms, and hardware they incorporate to test and evaluate the Lander during simulated Mars entries.

BACKGROUND

Traditionally, disparate simulation tools have been exercised in an ad-hoc manner on the separate portions of a typical EDL profile. This approach has resulted in a patchwork of tools, with complex, hand-crafted interfaces requiring manual transfer of data products across simulation systems. Such an approach was difficult even for the relatively simple requirements of the Mars Pathfinder style missions. This approach will be totally inadequate for the Smart Lander missions where extensive, closed-loop actions by the spacecraft requires intimate integration of all the supporting simulation elements.

The objective of the MSL EDL simulations is to support Lander systems design, trade studies, development, testing, and operations by establishing end-to-end simulations that include high fidelity models of the Mars Smart Lander systems and the Mars environment. To support this objective, a capability to provide end-to-end engineering simulation of all phases of EDL flight throughout the MSL project lifecycle is required. This engineering simulation is needed to enable rapid initial screening to define critical mission and lander parameters and sensitivities for conceptual design and system level trades. Not only is this high fidelity simulation necessary for detailed lander design, mission statistics, and operations support, but also to verify integration of EDL subphases performance (such as the control system during entry or the parachute during descent) through evaluation of lander systems and configurations in an end-to-end environment. Also needed

to support the objective above is a real-time, hardware-in-the-loop end-to-end EDL simulation to allow systems level tests of flight hardware and software components, flight software checkout prior to upload, and systems level troubleshooting during operations. Overall Lander system risk is reduced by fully integrating not only detailed engineering models of the subsystems, but the actual hardware for testing in the simulated mission environment from cruise stage separation through lander touchdown. These end-to-end system level simulations also provide inputs as well as independent validation and verification of subphase simulations used for subsystem design, algorithm development, and detailed component level analyses. Additionally, the simulations receive input conditions from interplanetary trajectory and cruise stage system simulations while providing inputs to Rover egress simulations. These simulations enable designers and mission managers to evaluate specific and overall lander systems performance in the expected Mars environment. These objectives and simulation requirements are met for MSL by the high-fidelity engineering (POST-based) and real-time, hardware-in-the-loop testbed (DSEENDS-based) end-to-end EDL simulations.

The high-fidelity engineering (POST-based) end-to-end EDL simulation provides mission level feasibility, design trades, guidance and control algorithm development and analysis capability, as well as success criteria evaluation of flight systems using rapid Monte Carlo analyses. The POST tool developed at NASA Langley has been successfully used for mission design, operations, trajectory determination/ optimization and Monte-Carlo analyses in many recent Mars missions. These missions include the Mars Pathfinder, Polar Lander, Odyssey Orbiter, proposed 2001 Lander, and MER 2003 Lander missions. POST has also been used for a number of non-Mars missions involving atmospheric entry (e.g., Stardust, Genesis, etc.). [Refs. 5-19]

In turn, the real-time, hardware-in-the-loop (DSEENDS-based) end-to-end EDL simulation allows detailed system integration analyses and testing for the Mars environment. The DSEENDS (Dynamics Simulator for Entry Descent and Surface landing) real-time simulation tool is an extension of the Darts/Dshell multi-mission spacecraft simulation tool developed at NASA-JPL. This tool has been used on the Cassini mission, which will study Saturn and send an entry probe into Titan's atmosphere. DSEENDS is capable of providing detailed spacecraft system/device simulation (sensor, actuators, communication devices, flex/rigid multi-body dynamics, aerodynamics, etc). Additionally, flight soft-

ware can be embedded in the simulation creating a system integration testbed to support flight system validation, ATLO, and mission operations.

The current MSL EDL simulation strategy has several strong advantages. The use of POST-based and DSENDS-based end-to-end simulations in conjunction with independent subphase simulations enables each critical segment of EDL to be covered by three simulations for synergistic purposes. The independent development of these simulations provides a strong basis for independent validation and verification of all simulations. End-to-end simulations provide coordination with specialized subphase simulations leading to early identification of interface issues as well as subsystem conflicts from one phase of EDL to the next. Figure 2 illustrates the coverage that the two end-to-end simulations provide and their interaction with more specific subphase simulations. Although these end-to-end simulations are undergoing further improvements, the design engineers and project managers are receiving useful simulation results even as the simulations are being updated and their fidelity is increasing. By providing these results in the early phases, the project schedule is less likely to be impacted by system level EDL design issues late in the design cycle.

The following two sections describe the POST-based and DSENDS-based simulations for Mars Smart Lander Entry, Descent, and Landing in more detail. The third section includes information about the validation and verification approach for these simulations.

PART 1. HIGH FIDELITY ENGINEERING **(POST-BASED) END-TO-END EDL** **SIMULATION**

The Program to Optimize Simulated Trajectories was initially developed in the 1970's to support Space Shuttle development. [Ref. 2] It has been continually upgraded and modified since then to support a large variety of aerospace vehicle development and operations through trajectory simulation, analyses, and system performance assessments. [Ref. 3] POST contains many basic models (such as atmosphere, gravity, and navigation system models) that are used to simulate a wide variety of launch, orbital, and entry missions (see Fig. 3).

However, exploiting the modular nature of the POST program by adding mission specific models in concert with the existing POST architecture allows for the development of higher fidelity, mission specific simulations. These simulations support design, devel-

opment, testing, and operations of vehicles for particular missions. Simulation complexity varies from first-order trades (e.g. parachute size and deployment conditions, terminal descent engine size, etc.) to all-up Monte-Carlo simulations to day-of-entry operations.

The models required for these simulations depend on the desired fidelity of the analysis. In the initial phases of mission definition and vehicle conceptual design, basic models already available in POST are used without modification to provide a tool for top level trades and conceptual level design. By using existing models, rapid turnaround vehicle assessment and design simulations are possible. These engineering models can be rapidly adapted for performance evaluation and top-level trades of new designs. As the mission and systems get better defined and higher fidelity models become available, they are incorporated into the POST simulation to perform more mission specific trades and analyses of the updated systems. Eventually, three and six degree-of-freedom (3- and 6-DOF) simulations which span an entire phase of a mission (such as entry, descent and landing at Mars from the final exoatmospheric trajectory correction maneuver to lander touchdown) using the latest engineering models of onboard systems are available for detailed mission trades, system analyses, system testing, and mission operations. This approach has been and is being applied to the Mars Smart Lander mission for the Entry, Descent, and Landing high fidelity engineering simulation using POST as the main simulation engine.

The POST-based simulation tools have been used to support all elements of the design life cycle for a wide variety of missions. Early conceptual studies have been conducted using models in the basic production version of POST. [Refs. 5-9] Higher fidelity simulations have included many mission specific models and data including aerodynamic parameters from wind tunnel testing and Computational Fluid Dynamics (CFD) runs, vehicle mass properties, parachute, control systems, and onboard propulsion systems as these data and models became available. [Refs. 10-15] POST-based simulations have been exercised for extensive Monte-Carlo runs including those for "stress tests" that determine the limits of system capability. [Refs. 13-18] The technical capabilities of POST have already been validated against other Mars mission data. [Refs. 16-19] Note that these references focus mainly on Mars missions, whereas a much larger set of references exists for other applications in which POST is used, such as Earth launch and entry vehicle development as well as entry systems for other planetary missions.

The Mars Smart Lander mission is illustrated in Fig. 1. The current EDL timeline is shown in Table 1. As noted in the timeline, two configurations are currently included in the POST-based MSL high fidelity end-to-end EDL engineering simulation. The simulation starts at cruise stage separation, whereas the actual entry begins at atmospheric interface when aerodynamic forces (albeit small at altitudes over 60 km) are acting on the vehicle. During atmospheric entry, the flight path is controlled by the entry guidance until it commands supersonic parachute deploy (nominally around 9 km above the ground). Next, the backshell and supersonic parachute separate from the lander and deploy the subsonic parachute around Mach 0.8. Ten seconds later, the heat shield is jettisoned. The radar begins to get usable altitude and velocity data about three seconds later when the heat shield is clear. Upon command of the terminal descent guidance, the main engines are started at 20 percent thrust. After two seconds, the subsonic parachute is released and the preliminary touchdown target is identified. When hazard avoidance is included, the system will use LIDAR to scan the surface to determine if the preliminary target is suitable. If not, a new target will be established and the lander will be diverted to it. In the simulation, a single divert maneuver of 100 m (horizontal distance) at 300 m above ground is used in the Monte Carlo analyses to simulate the divert capability. During terminal descent, guidance is commanding the thrust vector (magnitude and direction) to reach the target point such that a constant velocity, vertical motion only phase is started at 5 m above the ground. The radar stops returning useful data at about 10 m above the surface. All engines are shut off when the lander is one meter above the surface. To assess the current configurations under consideration, an EDL Challenge Site has been identified (which is about 2500 m above the MOLA areoid) as the nearly 20 km wide crater at 41.45° S latitude and 286.5° E longitude. The entry and terminal descent guidances are configured to land inside the Challenge crater.

The MSL high fidelity engineering simulation of these EDL events currently includes various engineering models at varied levels of complexity. Both 3 DOF and 6DOF simulations, which start at lander separation from the cruise stage and finish at touchdown on the surface, are under development, with the 3DOF simulation the most mature at this point. As mentioned above, the latest engineering models are incorporated into these simulations, or existing models in POST are used until the engineering model is developed. The vehicle specific models include terminal descent and entry guidance algorithms, flight navigation filter, sensors,

inertial measurement unit (IMU), vehicle aerodynamics (entry, parachute, and terminal descent configurations), parachute dynamics, as well as various control system and propulsion system models. The environment models in the simulation include high fidelity gravity models, Mars-GRAM atmosphere models, and surface topography based on MOLA data. Table 2 shows the responsible groups for various vehicle specific models. Several of these models are briefly described below, whereas references are provided for details of other models.

The POST-based simulation has been used to provide a variety of products to the Mars Smart Lander project. These products include: entry and terminal descent vehicle designs and trades; entry aeroshell configuration trades; guidance and control algorithm development; parachute sizing trades; terminal descent engine trades; project Monte Carlo statistics (precision landing, touch down velocity, etc.). Current plans are to continue to supply mission and vehicle trade studies, as well as provide inputs to the design process (e.g., heat-shield thermal protection system sizing). The POST-based simulation will also be used in the operations phase to support day-of-entry preparation and analyses. The POST-based simulation is also providing validation and verification support to the real-time, hardware-in-the-loop (DSENDS-based) simulation; further discussion on this support is provided in the third part of this paper (POST-DSENDS Validation and Verification).

Gravity Model

The gravity model in the POST-based simulation uses zonal, sectoral, and tesseral harmonic terms to determine the acceleration due to gravity. This model is based on the one used in the Artificial Satellite Analysis Program (ASAP). [Ref. 20] The 50-by-50 Mars gravity field used in the simulation is the MARS50C model established to support the Mars Polar Lander mission. [Ref. 21] This gravity model will be updated as higher order data (such as the 75 x 75 gravity model used to support the Mars Odyssey mission) becomes available.

Planet Model

An oblate spheroid Mars model is also used in the POST-based simulation. This planet model defines the physical dimensions (e.g., equatorial radius, polar radius) and characteristics (e.g., rotation rate) of Mars. This model is not only used for latitude and longitude determinations, but is also necessary to determine Mars

relative velocity used by the guidance algorithms and other simulation models.

The local altitude is determined using Mars Observer Laser Altimetry (MOLA) topography data and a reference areoid model. The recent availability of electronic topographic data provided by the MOLA project [Ref. 22] has enhanced POST simulations by allowing the calculation of vehicle height above local features at Mars. There are three primary surface references for measuring altitude: the areoid, the reference ellipsoid, and the surface. The areoid (or Mars geoid) is a gravitational equipotential surface, analogous to the theoretical mean sea-level surface on Earth. A “plum bob” would hang perpendicular to its surface at every point. The angle between this vertical and the equator defines the astronomical latitude. The MOLA areoid is defined to be a surface with the same gravitational potential as the mean equatorial radius (3396 km) and is determined from an 80 by 80 coefficient representation of the gravitational field. The geocentric radius to the areoid is provided as a dataset at 1/16 degree resolution. The reference ellipsoid, used within MarsGRAM and POST, is an engineering approximation of the areoid. The surface of the ellipsoid is completely defined by an equatorial radius of 3396 km and a polar radius of 3378.32 km. The normal vector to the ellipsoid is the direction that a plumb bob would hang if it were not for local gravitational anomalies. The angle between this normal vector and the equator defines the geodetic latitude, which is the basis for most maps and charts. The MOLA dataset provides a planet-wide model of the Mars surface topography at 1/32 degree resolution, expressed relative to the areoid.

The problem of determining the vehicle’s altitude above the surface, in the geodetic sense, requires an iterative solution. A declination to the surface (the angle between the radius vector and equatorial plane, *decln_star*) is first guessed which defines a point beneath the spacecraft, measured in the geodetic sense. (See Fig. 4.) The length of the vector measured geocentrically to this point from the center of the planet is calculated geometrically using the law of sines:

$$r_{calc} = g_{rad} \frac{\sin(\beta)}{\sin(\theta)}$$

where

r_{calc} = calculated radius to surface

g_{rad} = geocentric radius to spacecraft

β = center-spacecraft-subsurface angle (angle between *g_{rad}* and *h_{gtagl}*)

θ = center-surface-spacecraft angle (angle between *r_{calc}* and *h_{gtagl}*)

This calculated radius is compared to the sum of the radius to the areoid plus the topographic altitude, determined from the MOLA dataset at the geocentric latitude equal to the guessed declination. A bisection root finding method is used to drive the error between the calculated and actual radii to zero. Once the surface declination is known, the altitude of the spacecraft, measured geodetically, is calculated from the law of cosines

$$h_{gtagl}^2 = g_{rad}^2 + r_{calc}^2 - 2g_{rad} \cdot r_{calc} \cos(\alpha)$$

where

h_{gtagl} = height above ground level (AGL)

g_{rad} = geocentric radius to spacecraft

r_{calc} = calculated radius to surface

α = spacecraft-center-surface angle (angle between *g_{rad}* and *r_{calc}*)

Atmosphere Model

The Mars Global Reference Atmosphere Model [Ref. 23] version 2001 (Mars-GRAM 2001) has been included in the simulations (as FORTRAN subroutines in POST). Mars-GRAM provides all of the atmospheric data (temperature, density, pressure, and wind velocity) as well as random perturbations to certain atmospheric quantities (e.g., density) while including seasonal, diurnal, latitudinal, and longitudinal variations. The atmospheric data is a function of the spacecraft location (latitude, longitude, and altitude) as well as other user supplied inputs. [Ref. 24] These inputs include the date of Mars arrival, the minimum update distance for dispersion calculations, a scale factor on the atmospheric dispersions, interpolation option for the upper atmosphere, and the f10.7-cm solar flux value. In addition, the capability to model the effect of dust opacity or dust storms is included. The Mars arrival date and f10.7-cm solar flux values reflect the period during the solar cycle in which the entry occurs.

The Mars-GRAM subroutine is a parameterization of the atmospheric properties, so that the model runs relatively quickly and the overall simulation speed is not hampered by the atmospheric subroutine. Recent versions of Mars-GRAM include density profile data from more detailed simulations using global circulation models (GCM) being developed at the NASA Ames Research Center (by Robert Haberle and James Murphy) and at the University of Arizona (by Steve Bougher); that is, Mars-GRAM can reproduce the more realistic densities from the GCM for a specific entry profile in the simulation but in a fraction of the time. These recent versions also include atmospheric wave models which incorporate wave effects on atmospheric density. The latest Mars-GRAM version includes the 1/2 degree resolution topographic data for Mars from the Mars Observer Laser Altimetry (MOLA) instrument onboard the Mars Global Surveyor spacecraft in orbit about Mars. The dust opacity parameter is used to define the amount of airborne dust particles so that Mars-GRAM can simulate their affect on the atmospheric properties.

Two significant adjustments have been made to support Mars-GRAM inclusion in the high fidelity engineering MSL EDL end-to-end simulation. First, a wrapper subroutine was developed to provide a software interface between the Mars-GRAM program, developed by Jere Justus (through the NASA Marshall Space Flight Center) and the POST-based simulation. The wrapper converts between the double precision variables used in the flight simulation and the single precision variables used by Mars-GRAM, and it prohibits Mars-GRAM from being called too frequently while dispersed density atmospheres are generated during Monte Carlo analyses. Second, a higher resolution MOLA topography data (1/16 degree resolution) was added to the Mars-GRAM 2001 software. Jere Justus suggested the necessary subroutine adjustments that were implemented to include this newer data. The 1/16 degree resolution MOLA data includes most of the surface features (e.g., craters) found in higher resolution data (such as 1/32 degree resolution), while requiring a manageable amount of computer memory.

Aerodynamic Model

A FORTRAN subroutine supplies aerodynamics data to the POST-based simulation. [Ref. 25] Whereas different subroutines are supplied depending on the configuration to be simulated, the basic difference between routines is the data included for the specific configuration. The routine uses first derivative, or C(1),

continuous interpolations between a database of discrete solutions. This interpolation scheme is applied to free molecular solutions for the rarefied region of the atmosphere, and computational fluid dynamic (CFD) solutions combined with wind tunnel test results for the continuum regime. A modified Lockheed bridging function [Ref. 26] is used in the transitional region between rarefied and continuum regimes. The various flow regimes are delineated according to Knudsen number.

Entry capsules for robotic missions tend to spend a significant amount of time in rarefied and transitional flow regimes. Therefore, free molecular values are included in the aerodynamic databases. The aerodynamic data in the rarified regime are a function of vehicle attitude. In the transitional regime, the aerodynamic data are a function of both vehicle attitude and Knudsen number.

For the continuum region, static aerodynamic data were obtained from CFD solutions using the Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA) [Ref. 27-29] and tests conducted in the NASA Langley's Unitary-Plan Wind Tunnel (UPWT) [Ref. 30]. LAURA was used to generate aerodynamic databases for the Mars Pathfinder [Ref. 31], Mars Microprobe [Ref. 26], and Stardust [Ref. 32] entry capsules. Confidence in the LAURA solutions comes from validations with Viking data, wind tunnel data, and Mars Pathfinder mission results. [Ref. 33] Dynamic aerodynamic quantities were included from the data generated for the Viking missions.

Parachute aerodynamic data is taken from Viking and Mars Excursion Rover (MER) mission data. Supersonic parachute data is taken from existing disk-gap-band parachutes from the Mars Pathfinder mission [Ref. 34] and planned for the MER mission in 2003. The subsonic parachute is a ringsail parachute of the type used in Apollo, Gemini, and Mercury [Ref. 35] with parachute area scaled for the mass of the MSL entry system.

Terminal descent phase aerodynamics were taken from Viking mission data. Prior to heatshield separation, the entry phase aerodynamics is used, After heatshield separation, the data from wind tunnel tests conducted on a Viking lander inside a backshell was incorporated into the aforementioned aerodynamics subroutine format. [Ref. 36]

Control System

Reference 37 describes the 6 DOF entry control system under development, whereas reference 38 discusses the 6 DOF terminal descent controller being designed. While higher fidelity 6-Degree-of-Freedom (DOF) simulations with entry and terminal descent control systems are being developed, a lower fidelity 3-DOF simulation that executes faster while providing similar results is desired. That is, a model that simulates the behavior of a 6-DOF controller in a 3-DOF simulation of the EDL phase of a planetary entry vehicle is wanted. For the entry phase, a pseudo-controller of bank angle is employed. Whereas, the terminal descent controller in 3-DOF simulations controls attitude of all three axes. Both of these controller models were developed and integrated within POST to respond to vehicle guidance commands in a 3-DOF POST simulation.

During the entry phase, vehicle attitude in the 3-DOF simulation is determined by balancing the aerodynamic moments acting on the vehicle (i.e., flying in an aerodynamic trim attitude) for angle of attack and sideslip angle while a pseudo-controller is employed for the commanded bank angle. Work with previous 6 DOF simulations has shown that aerodynamic trim conditions generally occur at about the mean attitude when rotational dynamics are included. Therefore, results from 3 DOF simulations using aerodynamic trim translate better to the 6 DOF simulation than constant attitude 3 DOF runs. The single axis controller determines the appropriate bank angle and bank rate change for the input maximum acceleration and bank rate (5 deg/s² and 20 deg/s for MSL) using an Euler integration scheme. The maximum acceleration is assumed until either maximum rate is achieved or the controller determines that maximum deceleration must begin to reach the commanded bank angle. This pseudo-controller model was also used in the MSP'01 Lander simulation. [Ref. 14]

A terminal descent controller was developed and integrated within POST that models the 6-DOF rotational dynamics of a vehicle in a 3-DOF simulation. This terminal descent controller may operate in either of two modes: acceleration control or throttle control. In the first mode, the controller solves for only the angular acceleration vector needed to obtain the commanded attitude, within prescribed angular velocity and acceleration limits. This acceleration vector is then used to update the vehicle's attitude. This mode is advantageous when only limited information is known about the vehicle's propulsion system, attitude control system,

and moments of inertia. In the second mode, the controller first determines the necessary angular acceleration and then solves for the actual terminal descent engine throttle settings that would provide the correct angular acceleration and commanded thrust, within prescribed minimum and maximum throttle limits. These throttle settings are used to determine the actual angular acceleration, which is then used to update the vehicle's attitude. This mode requires detailed knowledge of the magnitude and direction of thrust and moments provided by each engine that is being manipulated by the controller.

In either acceleration control or throttle control, the terminal descent controller must first determine the desired angular velocity vector necessary to achieve the commanded attitude. The direction of the angular acceleration vector is chosen such that the resultant angular velocity vector lies along the single axis-of-rotation between the current attitude and the commanded attitude. The single axis-of-rotation is found from the vector component of the quaternion that, when multiplied by the current attitude quaternion, produces the commanded attitude quaternion. The magnitude of the acceleration vector is determined from the angular error between the commanded and current attitudes, a controller gain, and the maximum allowable angular velocity. The strategy employed is to complete a certain percentage of the desired angular rotation, controlled by the gain, within the current time step. However, if the maximum angular velocity would be exceeded, the angular rotation is limited to the product of the maximum angular velocity and the time step. This acceleration vector is finally scaled such that the maximum component of acceleration along each axis is not exceeded.

Guidance Algorithms

The guidance algorithm for the entry phase (known as the Entry Terminal Point Controller, or ETPC) determines if modifications to the current atmospheric flight path are required and directs the control system to make attitude adjustments based on the navigation system input and the desired target location. As illustrated in Fig. 5, this system modulates the vehicle bank angle (direction of the lift vector, ϕ) such that the vehicle adjusts its atmospheric trajectory. In this manner, the vehicle can accommodate off-nominal entry-state or atmospheric-flight conditions and achieve a significant reduction in landed footprint over non-lifting (ballistic) or constant bank angle (Viking-type) entries. Maximum control authority occurs when the vehicle is traveling at hypersonic speeds through the peak dynamic pressure

(and peak deceleration) portion of the atmospheric entry. The ETPC algorithm is derived from the final phase logic of the Apollo command module entry guidance. Bank angle commands for terminal point range control are derived with a linear perturbation algorithm using influence coefficients of drag acceleration and altitude rate errors with respect to a fixed nominal reference trajectory as a function of relative velocity. Crossrange control is accomplished with bank reversals at target out-of-plane corridor limits; however, a final heading alignment phase is used to null terminal crossrange errors. Additionally, the guidance initiates the supersonic parachute deployment to achieve minimum target range within supersonic parachute deploy constraints (Mach number and dynamic pressure constraints are implemented implicitly as relative velocity and drag acceleration corridors). Further detail of the ETPC is given in reference 39.

For the terminal descent phase, the guidance algorithm is not only used to ensure a successful touchdown, but also provides a capability to divert away from detected hazards. The guidance commands an acceleration profile based on navigation estimates of position and velocity. This desired acceleration is implemented via appropriate throttle settings on the six main terminal descent engines. The control system is assumed to align the thrust vector to the commanded acceleration direction via the appropriate vehicle attitude. A commanded divert or change in the descent profile to avoid a hazard is reflected in the acceleration profile commanded by the guidance. In the final five meters, a constant velocity descent is commanded until the engines are shut off at one meter. Further detail on the terminal descent guidance is given in reference 38.

Navigation System

A model of the onboard inertial measurement unit (IMU) and navigation system is included in the simulation. This model uses an estimate of the initial states that would be determined as the spacecraft approached Mars, whereas the simulation uses an actual or delivery state provided by the interplanetary trajectory analysis. A model of the IMU provides adjustments to simulation generated quantities to account for sensor errors. The output from the IMU model is used by the navigation system model to produce an estimate of the vehicle state for use by other onboard system models (such as guidance algorithms, control systems, etc.). More detailed information about IMU/navigation system models can be found in reference 40.

Monte Carlo Dispersions

A Monte Carlo dispersion analysis is used to quantify the acceptability and robustness of a given vehicle configuration, as well as determine areas of risk associated with certain designs and mission phases. These dispersion analyses are obtained by randomly varying key parameters and characteristics of the environment as well as the vehicle assuming a normal or uniform distribution of these quantities. The engineers responsible for the subsystem models identify the 3-sigma or maximum/minimum values of the uncertainties for these key parameters. These inputs are then used in the MSL end-to-end EDL engineering simulation to determine various outputs of the trajectory. The outputs are compared with given metrics for each; thus, the suitability of the vehicle and mission can be assessed. A similar approach has been applied to the entry phase of several previous missions. [Refs. 5-8, 11, 14]

Table 4 indicates the parameters currently varied in the POST-based simulation during the Monte Carlo analyses. This table also shows the nominal value, type and limits of variation (either minimum/maximum or 3-sigma) for each. These quantities are varied randomly over 2000 simulation runs. Various mission and vehicle parameters are recorded at certain events throughout the simulations. These quantities are evaluated relative to MSL project metrics to assess vehicle performance, mission risk, and system robustness. Characteristics of Monte Carlo cases that consistently fail are identified for further investigation by system, vehicle, and mission designers. During mission operations as day-of-entry approaches and occurs, the POST-based Monte Carlo capability can be used to rapidly assess many off-nominal conditions to identify several challenging scenarios to be further analyzed using the real-time, hardware-in-the-loop (DSENDS-based) EDL testbed simulation. This rapid assessment using the POST-based simulation to support detailed subsystem hardware analyses using the DSENDS-based testbed permits quick, but very detailed analysis of any anomaly that occurs as entry is approached.

Sample Monte Carlo results of 2000 runs for the 70 deg trim shelf configuration are shown in figures 6, 7, and 8. The results at supersonic parachute deploy (see Fig. 6) indicate that the parachute deploy constraints on Mach and dynamic pressure were met, and the guidance delivered the entry system right on its target (note that the guidance only acts on the NAV or knowledge state, actual states differ due to knowledge error and IMU/Navigation error buildup). Figure 7 shows the

actual footprint at various EDL events (note that the Challenge crater's edge occurs at about 41.45 S, 286.28E and 41.45S, 286.75E) indicating that the supersonic deploy footprint basically defines the touchdown footprint size, but not location. The last figure provides histogram information for the touchdown conditions of the lander. These histograms (see Fig. 8) indicate that all the cases met the project metric of vertical touchdown velocity less than 4 m/s and horizontal velocity below 2 m/s while maintaining a near zero orientation relative to vertical. These figures show only a few of the key output parameters generated during a Monte Carlo run. A much larger set of data is generated with various subsystem design and assessment teams interested in different subsets of the data. Using this information, overall mission and vehicle statistics as well as risk assessments are provided to the MSL project leaders. Further discussion of Monte Carlo results can be found in references 1 and 39.

Validation and Verification

Each model or dataset that is included into or used by the POST-based high fidelity MSL EDL end-to-end engineering simulation must complete the validation and verification process described below. In this process (summarized in Table 3), both the model developer/data provider and the model/data implementer must concur before the process is complete. The developer is responsible for model formulation and verifying that the model and data are correct for the system it is supposed to reflect. The developer also is responsible for providing computer code of the model formulated and verifying that the code produces expected results when used in a standalone mode. As such, the developer is responsible for providing a set of test data and results from the standalone runs. The developer is also required to provide expected ranges of key input parameters associated with their system and model for use in Monte Carlo dispersion analyses.

The implementer of the model into the POST-based simulation must properly include the data or software into the simulation and ensure that all the appropriate interface quantities are provided to the model. The model must produce the same output from within the POST-based simulation as was produced in the standalone test case. Both implementer and developer provide their expertise to resolve any discrepancies in the output. The implementer then provides sample input and output from a typical nominal and off-nominal case that can be checked by the developer using their standalone capability. When only data is provided, the

provider is responsible for confirming with the implementer that results are reasonable for the system that the data is provided.

Some model developers are using their own specific subphase simulation for elements such as the entry phase only for control system development or parachute phase for sizing and dynamics modeling. Validation of the results from these subphase simulations with the POST-based simulation provides a verification of both. Additional verification of the POST-based simulation with the real-time, hardware-in-the-loop DSENDS-based simulation is discussed in the third part below (POST-DSENDS Validation and Verification).

Future Work

Development of the POST-based simulation supporting the MSL mission is continuing. References 37, 38, 40, 41, and 42 describe various models that either have become or are becoming available soon. These models (which will be implemented in the POST-based simulation in the near future) include a multibody parachute model, surface terrain model, hazard avoidance logic, 6 DOF entry and terminal descent control systems, reaction control system data and firing logic, a navigation filter and associated sensor models, as well as LIDAR and RADAR models. The simulation is also updated as newer, higher fidelity models of various systems and the environment are developed and validated.

PART 2. END-TO-END EDL REAL-TIME SIMULATION TESTBED (DSENDS-BASED)

The Smart Lander system uses extensive sensor-based real-time control and decision making for precision landing and hazard avoidance during the entry, descent and landing phases. Testing and validating such a system requires the use of a high-fidelity, real-time spacecraft simulator. The Jet Propulsion Laboratory (JPL) is in the midst of adapting its EDL simulator, DSENDS (Dynamics Simulator for Entry, Descent and Surface landing) [Ref. 4] for use by the Smart Lander. DSENDS is an EDL specific extension of the JPL Darts/Dshell multi-mission spacecraft dynamics and devices simulation toolkit [Ref. 43, 44] used by missions such as Cassini, Galileo, etc. [Ref. 45].

DSENDS provides for the modeling of the dynamics of tree-topology multi-body systems with flexible modes within a real-time simulation. It also provides the capability to simulate, in real-time, various spacecraft devices such as actuators (e.g. thrusters) and

sensors (e.g. IMU's). A variety of EDL related environment models (e.g. gravity, terrain digital elevation maps, atmospheric models), are adapted for real-time use and support modeling of EDL flight system elements such as parachutes, landers, and terrain interacting instrument simulations (e.g. altimeter, LIDAR). Together these capabilities allow the modeling of the flight-train dynamics and sensor-based control during the Smart Lander EDL sequence. A block diagram of the DSENDS architecture and associated model libraries are shown in Figure 9 and 10. The recent capability to include the aerodynamic libraries from POST allows high-fidelity aerodynamics modeling, especially during the entry phase of flight leading to parachute deployment. Planned extensions for landing kinematics and dynamics will allow the modeling of contact and impact forces associated with touchdown. Nominal as well as fault behaviors are incorporated into the device models. A state-machine driven model switching capability within DSENDS handles spacecraft separations and re-configurations such as the example in Figure 11. Stub guidance/navigation controller modules for hypersonic steering, parachute activation, hazard avoidance, and powered descent guidance/control allow standalone simulation as shown in Figure 12 and 13.

Some of the system engineering issues related to the DSENDS system are presented in a companion paper [Ref. 46]. Here, we focus on a system overview as they relate to the real-time architecture of the simulation. We also briefly discuss the verification of these capabilities, including comparisons with off-line simulators (e.g. POST), mission data (e.g. Mars Pathfinder), as well as experimental data (e.g. Smart Lander Rocket Sled tests).

Real-Time Multi-Body Dynamics.

DSENDS utilizes the Darts multi-body dynamics engine developed at JPL. This dynamics engine provides for extremely fast computations of rigid and flexible body dynamics of a tree-topology multi-body system. The underlying computational algorithms for Darts are based upon the Spatial Operator Algebra system [Ref. 47] and result in the numerical complexity of the dynamics algorithm growing only linearly with the number of bodies. Such $O(n)$ algorithms allow high-fidelity modeling of spacecraft dynamics without compromising fidelity to meet real-time constraint. Constraining forces and torques between each connected body in the multi-body system are transmitted through joints that can be of a variety of types. Each body in the multi-body system can also be acted upon by external

as well as by additional inter-body forces and torques. In the EDL simulation context, these forces and torques represent the actions of gravity, aerodynamic forces, and non-linear spring elements between the bodies. The underlying dynamics engine also support the notion of prescribed motions where forces and torques are derived from a kinematic specification of the trajectory. This allows certain simulation elements to be driven by trajectory profiles rather than force/torque applications and is useful for modeling elements where the trajectories are well known (e.g. from test data) but the force/torque relations are not. The rigid-body modeling capability allows models for the entry capsule, heat-shield, lumped approximations to parachutes, and tether/bridle link elements. The flexible-body modeling capability allows modeling of lightweight members such as landing gear and sensor mounts. The prescribed motion capability is potentially useful for certain EDL parachute reefing and bridle-lowering models.

Real-Time Aerodynamics

DSENDS provides a number of aerodynamics models at various levels of fidelity. The highest fidelity models are encapsulated subroutine libraries from the POST program. These libraries are C routines compiled for the Solaris operating system and embed calls to determine aero-coefficients (as a function of Mach and Knudsen number and aerodynamics angles) as well as atmospheric models (e.g. MarsGram [Ref. 48]). Other lower fidelity models available for use in DSENDS include analytical linearized as well as table-interpolated models for aerodynamics coefficients, stand-alone encapsulations of the MarsGram atmospheric database, and several table-driven models of atmospheric density and temperature profiles. Within the MSL simulation project high-fidelity models from POST are the primary models used for the entry phases of the flight. These models preserve the high-fidelity performance of the original aerodynamics databases within POST. During the parachute and later descent phases either POST derived aerodynamics or the lower fidelity models within DSENDS may be used, with the choice determined by availability and computational burdens.

In order to use the libraries obtained from POST within a real-time simulation testbed, two options are possible. To maintain maximum fidelity it is desirable to execute the libraries on the same processor as that used for POST execution. The other option is to cross-compile the code to the typical processor and operating system environment used in real-time testbeds. The first option requires the utilization of a Sparc® processor

with the Solaris® operating system (from Sun Corp). The second option would require cross-compiling to a VxWorks® operating system (from WindRiver Corp), on a PowerPC® or other similar target system. We have chosen to use the first option where the code libraries from POST can be received in binary object form and source code deliveries are not necessary. We have successfully verified the real-time performance of the POST libraries using the real-time operating system features within the Solaris® operating system.

Embedded Real-Time Architecture

The Darts/Dshell toolkit operates in standard workstation based environments such as Solaris®, Irix® (from SGI Corp), and Linux® as well as in a real-time embedded environment such as VxWorks®. In addition, Darts/Dshell supports the VxSim® emulation of the VxWorks® system on a workstation. As DSENDS is implemented in Darts/Dshell, all of the execution environments supported by Darts/Dshell are also supported by DSENDS. The real-time Darts/Dshell execution tool supports the loading of core C/C++ library modules cross-compiled to the appropriate target hardware platform. Models of the spacecraft dynamics and devices are instantiated at run-time and sorted into an execution order defined by the partial ordering derived from the data dependencies established during the definition of model input, output and connectivity. The Darts/Dshell architecture also provides a mechanism to utilize multiple embedded target CPU's to provide computational speedup. This allows inherent parallelism in the data-flow computations within the simulation system to be exploited. Simulation execution time is only constrained by the longest path through the graph representing the partial ordering constraints dictated by the data flow.

A user interface built upon the Tcl [Ref. 49] interpreter provides for convenient model definition, loading, simulation scripting and run-time interaction. This interface is typically only executed in the initialization phase of the simulation so as to not impact the real-time performance. An interface to the real-time data graphing tool Stethoscope® from RTI Inc., and an optional message passing interface to a workstation-based 3D visualization tool built upon the Open Inventor® graphics standard, provides the real-time engineering instrumentation into the simulation tool. The simulation time may be advanced by means of an interface to a real-time clock provided by the embedded system. All command inputs and data outputs to the flight-software

component of the embedded testbed system are provided as time-tagged simulation data.

Real-Time Terrain Access and High-Speed Instrument Simulations

Terrain products are required within the DSENDS real-time simulation to support a number of applications such instrument simulations (e.g. a terrain scanning Lidar), data monitoring modules (e.g. a monitor of the spacecraft height over the ground), and 3-D visualization of the simulation. The location, extent and spatial resolution of the terrain segments required to support these applications varies and is a function of the bore-sight, field-of-view, and the fidelity desired. For example, a Lidar with a steering mirror could require terrain anywhere within the field-of-regard provided by the mirror at a resolution that is a function of the instantaneous field-of-view (IFOV) of each pixel in the Lidar detector.

The requirements of real-time operation require that terrain Digital Elevation Maps (DEMs) be provided in a timely manner to the various EDL device models. Instrument responses must be generated in synchronization with a real-time clock with no possibility of cycle-slips and consequent data loss. One option would be to have all of the terrain resident in memory for immediate access by the requesting EDL model. This is not feasible because of the sheer size of the data set required. For example, a 10 km x 10 km site at 10 cm resolution would require storage of 10^{10} pixels! Instead, a process of terrain generation (or enhancement in the case of synthetically augmented natural terrain) must be combined with terrain segment transport to the simulator, followed by upload to the simulator's memory. DSENDS implements a real-time interface to a Terrain Server database system to support these functions. The Terrain Server uses multiple fast processors to generate the terrain. Timely transport of data to DSENDS is made possible by using fast network hardware and protocols. Finally, real-time buffers and shared-memory segment are managed within the EDL simulation to achieve real-time terrain access. Note that the terrain generation operation can take many seconds, transport usually takes a fraction of a second, and buffer management/swapping is done at simulation rates e.g. 50 ms. As new terrain segments are needed by the simulation, successive terrain segments must be generated as needed, uploaded to the simulator, and placed into memory in a timely and seamless fashion. DSENDS has a number of real-time shared memory buffers that contain overlapping terrain segments. As

the model requests terrain in the overlapping areas, buffers are switched in real-time to allow the application to access terrain in the new segment in a seamless fashion. The simulator also uses a predictive model of terrain usage to predict the extent, resolution and extent of terrain segments required by the application. These predictive models are usually based upon a nominal EDL scenario and the current location and velocity of the ground "footprint" of the instrument/viewer field-of-view. These predictions, together with knowledge of terrain generation times, data transport times, and buffer sizes are used to sequence the generation, transport and upload of appropriately overlapping segments of terrain into the EDL simulator. An example scenario indicating successive terrain generation and use requests is illustrated in Figure 14. DSENDS manages the use (and reuse) of the real-time buffers, the extent of overlap, and provides a level of cache management (e.g. keep adjacent terrain segments in memory in case they are needed) to relieve the simulator from frequent interactions with the terrain generation/transport process. In addition the design provides for backup terrain (with lower resolution and larger spatial extent) in case the generation/transport process fails to achieve the times predicted by its model, or if the predictions of anticipated application terrain request turn out to be wrong.

Verification

Verification of DSENDS real-time system occurs in two phases. The first phase compares data from the real-time simulation with a workstation based Darts/Dshell simulation executing the same simulation and model configuration. Then the workstation data is compared against external data sources. These include comparing the DSENDS aerodynamics model output data with POST simulator data, as well as published data on various Mars mission data sets [Ref. 51, 33]. The multibody and flex-body dynamics is verified as part of the overall Darts/Dshell tool verification.

Specific device model verification is performed by comparing DSENDS model output with test data from various Smart Lander Test programs such as Rocket Sled Lidar tests [Ref. 50] and future MSL Drop/Descent tests. The approach here is to develop simulations that model the test configuration and devices. Comparison of test data and the simulated test data provides for verification of model performance.

PART 3. POST-DSENDS VALIDATION AND VERIFICATION

In addition to the validation and verification procedures outlined above, the POST-based high fidelity engineering and the DSENDS-based real-time simulations will be used to cross validate the engineering models common to both simulations as well as provide a verification check of the entire EDL trajectory. The 6-DOF, POST-based engineering simulation with the highest fidelity models of lander systems and the Mars environment will be used to compare various test case results with those produced by the DSENDS-based real-time simulation for the same tests. The exact test and validation plan is being developed, however the basic approach has been identified.

Three levels of testing will be made using these simulations. Unit tests of specific subsystem models will occur first. Next, portions of the EDL will be used with certain models simplified while others are tested. Finally, the full end-to-end simulations will be used for nominal and off-nominal runs.

The unit tests will focus on particular models that can be easily isolated. Models such as the control system, aerodynamics, and guidance algorithms can be tested while using very few additional models that can be very simplified. For example, the entry control system can be tested for an exoatmospheric case using a spherical gravity model and a simple open-loop square wave command about a single axis. This test would focus on the control system response to the given inputs in both simulations. Results from tests such as these are expected to match very closely.

The next level of testing will continue to involve simple models, but will focus on more Lander specific models from a given segment of the EDL trajectory. For instance, the entry phase can be tested using the high fidelity guidance and control system, while maintaining simple vehicle, atmosphere, and gravity models. After obtaining a satisfactory comparison of results, the fidelity of the other models will be increased until an entire phase is simulated to the highest fidelity that the 6-DOF, POST-based simulation can support. Some tests using off-nominal values of key parameters will also be included. Further increases in the model fidelity for the DSENDS-based simulation (such as including tank slosh effects), will result in a divergence of results, but the difference should be small and within an expected range.

Finally, the validated segments will be brought together until the full end-to-end EDL simulation is verified using both simulations. Tests will involve nominal and off-nominal cases. Once again, the initial tests will be to the highest fidelity the 6-DOF, POST-based simulation can support, and then the DSEENDS-based simulation will increase model fidelity to ensure there is no significant change in the validated results. This testing will also serve to provide a methodology by which future off-nominal cases will be identified from the POST-based Monte Carlo runs for more detailed analyses using the DSEENDS-based real-time simulation.

To summarize, testing will involve unit tests of specific model using simplified versions of other models. Testing will increase in number of models and fidelity until entire EDL segments are included. Then, full end-to-end simulation comparisons will be made using the highest fidelity the POST-based simulations can support. Initially, simulation result comparisons for the unit tests are expected to match very closely. As the tests and the models become more complex, some divergence in results is expected especially as the model fidelity in the DSEENDS-based simulation exceeds that of the 6-DOF, POST-based simulation.

CONCLUSION

The development of DSEENDS-based real-time, hardware-in-the-loop EDL simulation is complementary to the utilization of the POST-based high fidelity engineering simulation for the Smart Lander project. Using both simulations allows the comprehensive testing of EDL systems and flight software as well as vehicle performance and mission risk assessments in a unified manner across the Smart Lander design, development and operations life-cycle. Additional project risk reduction is obtained by using the overlap in capability between the simulations to validate them and their models against each other. The availability of such simulators will significantly reduce the risk associated with EDL development of the current and next generation of Mars Smart Lander and Sample Return.

ACKNOWLEDGEMENTS

The authors would like to thank the simulation and model development efforts of the entire Mars Smart Lander team from all the participating organizations. Each member brings unique contributions that make each of these simulations work. Without their tireless efforts at model development, validation, verification, and trouble shooting, these simulations would not have

near the level of fidelity they exhibit today. The DSEENDS work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The POST program is maintained and developed at NASA's Langley Research Center, with development support from Lockheed-Martin Astronautics (Scientific Software Group). The authors also wish to thank the DSEENDS development team and the POST support team for their contributions to this work.

REFERENCES

1. Lockwood, M.K., Sutton, K., Prabhu, R., Powell, R., Graves, C., and Epp, C., "Entry Configurations and Performance Comparisons for the Mars Smart Lander," AIAA Paper 2002-4407, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
2. Bauer, G.L., Cornick, D.E., and Stevenson, R., "Capabilities and Applications of the Program to Optimize Simulated Trajectories (POST)," NASA CR-2770, February 1977.
3. Powell, R.W.; Striepe, S.A.; Desai, P.N.; Queen, E.M.; Tartabini, P.V.; Brauer, G.L.; Cornick, D.E.; Olson, D.W.; Petersen, F.M.; Stevenson, R.; Engel, M.C.; Marsh, S.M.: "Program to Optimize Simulated Trajectories (POST II), Vol. II Utilization Manual." Version 1.1.1.G, May 2000, NASA Langley Research Center, Hampton, VA.
4. J. Balaram, R. Austin, P. Banerjee, T. Bentley, D. Henriquez, B. Martin, E. McMahon, G. Sohl, "DSEENDS - A High-Fidelity Dynamics and Spacecraft Simulator for Entry, Descent and Surface Landing", IEEE 2002 Aerospace Conf., Big Sky, Montana, March 9-16, 2002.
5. Spencer, D. A. and Braun, R. D., "Mars Pathfinder Atmospheric Entry: Trajectory Design and Dispersion Analysis." *Journal of Spacecraft & Rockets*, Vol. 33, No. 5, pp. 670-676, Sept.-Oct. 1996.
6. Desai, P., Mitcheltree, R., and Cheatwood, F., "Entry Dispersion Analysis for the Stardust Comet Sample Return Capsule," AIAA 97-3812, Aug. 1997.
7. Desai, P.N.; Braun, R. D.; Powell, R. W.; Engelund, W.C.; Tartabini, P. V.: "Six-Degree-of-Freedom

- Entry Dispersion Analysis for the METEOR Recovery Module" Journal of Spacecraft and Rockets, Vol. 34, No. 3, May-June 1997.
8. Braun, R., Mitcheltree, R., and Cheatwood, F., "Mars Microprobe Entry Analysis," Proceedings of the 1997 IEEE Aerospace Conference, pp. 247-262, Feb. 1997.
 9. Lockwood, M.K.; Powell, R.W.; et al, "Entry System Design Considerations for Mars Landers." AAS 01-023. Presented at the 24th Annual AAS Guidance and Control Conference, Breckenridge, CO, Jan 31-Feb 4, 2001.
 10. Powell, R.W.: "Numerical Roll Reversal Predictor-Corrector Aerocapture and Precision landing Guidance Algorithms for the Mars Surveyor Program 2001 Missions." AIAA 98-4574, Presented at the 1998 AIAA Atmospheric Flight Mechanics Conference, Boston, MA. August 10-12, 1998.
 11. Fraysse, H.; Powell, R.; Rousseau, Striepe, S.: "CNES-NASA Studies of the Mars Sample Return Orbiter Aerocapture Phase." IAF-00-A.6.05. Presented at the 51st International Astronautical Congress, Rio de Janeiro, Brazil Oct 2-6, 2000.
 12. Pastor, R., S. Striepe and R. Bishop, "Evaluation of Mars Entry Reconstructed Trajectories Based on Hypothetical 'Quick-Look' Entry Navigation Data", AAS Paper 00-197, AAS/AIAA Space Flight Mechanics Meeting, 23-26 January 2000, Clearwater, Florida.
 13. Queen, E.M.; Striepe, S.A.; and Powell, R.W.; "An Approach to Simulation of Extreme Conditions for a Planetary Lander" NASA TM -2001-211246
 14. Striepe, S. A.; Queen, E.M.; Powell, R.W.; Braun, R.D.; Cheatwood, F.M.; Aguirre, J.T.; Sachi, L.A.; Lyons, D.T; "An Atmospheric Guidance Algorithm Testbed for the Mars Surveyor Program 2001 Orbiter and Lander," AIAA 98-4569, AIAA Atmospheric Flight Mechanics Conference, Boston, MA, Aug. 1998.
 15. Ro, T., and Queen, E., "Study of Martian Aerocapture Terminal Point Guidance," AIAA Paper 98-4571, AIAA Atmospheric Flight Mechanics Conference, Boston Ma, August 10-12, 1998.
 16. Spencer, D.A., Blanchard, R.C., Braun, R.D., Kallameyn, P.H., and Thurman, S.W., "Mars Pathfinder Entry, Descent, and Landing Reconstruction," Journal of Spacecraft and Rockets, Vol.36, No. 3, 1999, pp. 357-366.
 17. Queen, E.M., Cheatwood, F.M., Powell, R.W., Braun, R.D., Edquist, C.T.," Mars Polar Lander Aerothermodynamic and Entry Dispersion Analysis," Journal of Spacecraft and Rockets, Vol. 36, No. 3, pp 421-428, May-June, 1999.
 18. Tartabini, P.V., Munk, M.M., Powell, R.W., "The Development and Evaluation of an Operational Aerobraking Strategy for the Mars 2001 Odyssey Orbiter," AIAA Paper 2002-4537, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
 19. Hanna, J.L., Chavis, Z.Q., and Wilmoth, R.G., "Modeling Reaction Control System Effects on Mars Odyssey," AIAA Paper 2002-4934, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
 20. Kwok, J.H. "The Artificial Satellite Analysis Program (ASAP) Version 2.0", JPL EM 312/87-153, 20 April 1987.
 21. Konopliv, A.S., and Sjogren, W.L., "The JPL Mars gravity field, Mars50c, based upon Viking and Mariner 9 Doppler tracking data," NASA CR-198881, NASA Jet Propulsion Laboratory, February 1995.
 22. Smith, D.E., Zuber, M.T., et al.: "The Global Topography of Mars and Implications for Surface Evolution." Science, Vol. 284, pp. 1495-1503, 1999.
 23. Justus, C.G., "Mars Global Reference Atmospheric Model for Mission Planning and Analysis", Journal of Spacecraft and Rockets, Vol. 28, No. 2, pp. 216-221, April-June 1991.
 24. Justus, C.G., James, B.F., and Johnson, D.L., "Recent and Planned Improvements in the Mars Global Reference Atmospheric Model (Mars-GRAM)", Advances in Space Research, Vol. 19, No. 8, pp. 1223-1231, 1997.
 25. Bobskill, G.J., Parikh, P.C., Prabhu, R.K., Tyler, E.D., "Aerodynamic Database Development for

- Mars Smart Lander Vehicle Configurations," AIAA Paper 2002-4411, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
26. Mitcheltree, R.A., Wilmoth, R.G., Cheatwood, F.M., Brauckmann, G.J., Greene, F.A., "Aerodynamics of Stardust Sample Return Capsule," AIAA Paper 97-2304, 15th Applied Aerodynamics Conference, Atlanta, GA, June 23-25, 1997.
 27. Gnoffo, P.A., Gupta, R.A., and Shinn, J.L., "Conservation Equations and Physical Models for Hypersonic Air Flows in Thermal and Chemical Non-equilibrium," NASA TP-2867, Feb. 1989.
 28. Cheatwood, F.M. and Gnoffo, P.A., "User's Manual for the Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA)," NASA TM 4674, April 1996.
 29. Riley, C.J. and Cheatwood, F.M., "Distributed-Memory Computing with the Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA)," Proceedings of the 4th NASA National Symposium on Large-Scale Analysis and Design on High-Performance Computers and Workstations, Williamsburg, VA, October 15-17, 1997.
 30. Murphy, K.J., Horvath, T.J., Erickson, G.E., and Green, J.M., "Supersonic Aerodynamic Characteristics of Proposed Mars '07 Smart Lander Configurations," AIAA Paper 2002-4409, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
 31. Gnoffo, P.A., Weilmuenster, K.J., Braun, R.D., and Cruz, C.I., "Influence of Sonic-Line Location on Mars Pathfinder Probe Aerothermodynamics," *Journal of Spacecraft and Rockets*, Vol. 33, No. 2, March-April 1996.
 32. Mitcheltree, R.A., Moss, J.N., Cheatwood, F.M., Greene, F.A., and Braun, R.D., "Aerodynamics of the Mars Microprobe Entry Vehicles," AIAA Paper 97-3658, Atmospheric Flight Mechanics Conference, New Orleans, LA, August 11-13, 1997.
 33. Gnoffo, P.A., Braun, R.D., Weilmuenster, K.J., Powell, R.W., Mitcheltree, R.A., and Engelund, W.C., "Prediction and Validation of the Mars Pathfinder Aerodynamic Database," AIAA Paper 98-2445, June 1998.
 34. Witkowski, A.: Mars Pathfinder parachute system performance, AIAA Paper 99-1701, 1999.
 35. Ewing, E. G., "Ringsail Parachute Design," AFFDL-TR-72-3, 1972.
 36. Flaherty, T., "Aerodynamics Data Book, VER-10," Viking Project Document TR-3709014, Martin Marietta Corp., June 1972.
 37. Calhoun, P.C., and Queen, E.M., "Entry Vehicle Control System Design for the Mars Smart Lander," AIAA Paper 2002-4504, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
 38. Wong, E., Masciarelli, J., and Singh, G., "Autonomous Guidance and Control Design for Hazard Avoidance and Safe Landing on Mars," AIAA Paper 2002-4502, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
 39. Mendeck, G.F., and Carman, G.L., "Guidance Design for Mars Smart Landers Using the Entry Terminal Point Controller," AIAA Paper 2002-4502, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
 40. Crain, T., and Bishop, R., "Mars Entry Navigation: Atmospheric Interface Through Parachute Deploy," AIAA Paper 2002-4501, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
 41. Queen, E.M., and Raiszadeh, B., "Mars Smart Lander Parachute Simulation Model," AIAA Paper 2002-4617, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
 42. Johnson, A., Klumpp, A., Collier, J., and Wolf, A., "LIDAR-based Hazard Avoidance for Safe Landing on Mars," AAS Paper 01-120, AAS/AIAA Space Flight Mechanics Meeting, Santa Barbara, CA, February 2001.
 43. Biesiadecki, J., Jain, A., and James, M.L., "Advanced Simulation Environment for Autonomous Spacecraft," in International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'97), (Tokyo, Japan), Jul. 1997
 44. Biesiadecki, J., Henriquez, D., and Jain, A., "A Reusable, Real-Time Spacecraft Dynamics Simu-

lator,” in 6th Digital Avionics Systems Conference, (Irvine, CA), Oct 1997.

45. JPL Solar Systems Homepage. URL: http://www.jpl.nasa.gov/solar_system/solar_system_index.html
46. Martin, B.J., Henriquez, D.A., Balaram, J., Sohl, G.A., Pomerantz, M.I., “System Engineering Challenges of Real-Time Simulation for Mars Smart Lander Entry, Descent and Landing”, AIAA Paper 2002-4413, Atmospheric Flight Mechanics Conference, Monterey, CA, August 2002.
47. Jain, A., and Rodriguez, G., “Recursive Flexible Multibody System Dynamics Using Spatial Operators,” Journal of Guidance, Control and Dynamics, vol. 15, pp. 1453-1466, Nov. 1992.
48. Justus, C.G., et al. “Mars Global Reference Atmospheric Model 2000 Version (Mars-GRAM 2000): Users Guide,” NASA TM 2000-210279, May 2000.
49. Ousterhout, J. K., Tcl and the Tk Toolkit, Addison-Wesley, 1994.
50. Skulsky, E.D., Johnson, A.E., et al, “Rocket Sled Testing of a Prototype Terrain-relative Navigation System,” AAS Paper 01-026.
51. Braun, R. D. et al, “Six Degree-of-Freedom Atmospheric Entry Analysis for the Mars Pathfinder Mission,” AIAA 95-0456, 1995.

Table 1. Current EDL Simulation Sequence and Data

- Configurations: 70° Trim Shell (2196 kg, 4.05 m dia, L/D ~0.24), CG Offset (1883 kg, 3.75 m dia, L/D ~0.18)
- Initial delivery and knowledge covariances (radiometric, optical) for October 2010 entry provided 60 min prior to atmospheric interface (JSC IMU/NAV model)
 - Nominal entry FPA of -14.5 deg
 - Atmospheric interface
 - Flight path control by bank angle commands from entry guidance
 - Mars-GRAM 2001 and 2010 entry date (nominal dust TAU=0.45)
 - Chia-Yen dispersed wind model (zero mean wind) scaled up to 30 m/s max winds
 - Supersonic chute deploy
 - Guidance initiated event
 - Nominal $C_D = 0.61$, diameter= 16.15m, MPF Mach efficiency and inflation model
 - Minimize miss distance and achieve Mach dynamic pressure MER-type Viking qualification box (Dynamic Pressure 239-850 Pa, Mach Number 1.13-2.2)
 - Supersonic chute and back shell separation, subsonic chute deploy
 - Mach= 0.8, nominal $C_D = 0.85$, diameter= 30.5m, MPF inflation model
 - Heatshield separation
 - 10 sec after subsonic chute deploy
 - Radar altitude and velocity lock-on 3 sec after heatshield separation
 - Pollard radar model
 - Track altitude AGL - check against 10 km limit
 - Lidar lock-on 1.5km actual altitude AGL (currently track event time only)
 - Terminal Descent Engine start
 - Guidance initiated
 - Engines start to 20% for 2 sec on subsonic parachute
 - 6 - 3047 N, 15° canted, throttle (in groups of 2, 20-100% capability), Isp variable
 - Subsonic chute separation (2 sec after engine start)
 - Touchdown target defined as analytic impact point
 - Target re-designated at 300m above surface to 100m from analytic impact point in uniform distributed direction (for Monte Carlo dispersion analyses only)
 - Radar Shut-off at 10 m above surface (no further updates)
 - Constant velocity phase
 - Start 5 m above surface
 - All engines cut off 1 m above surface
 - Landing
 - EDL Challenge site (41.45° S, 286.5° E), ~2500 m above MOLA areoid

Table 2. High Fidelity Simulation Inputs & Responsible Organization

Model/Data	Responsible Organization
Event Sequences	JPL
Mass Properties	JPL
Atmospheric Density Models	JPL
Atmosphere Wind (steady state and shears) models	JPL
Initial States corresponding to candidate landing sites	JPL
NAV Filter model	JPL, ISC
IMU model	JSC
RUS Prop	JPL
Entry Aerodynamics	LaRC
Entry Guidance	JSC
Entry Control System including thruster timing logic	LaRC
Supersonic Parachute	LaRC
Subsonic Parachute	LaRC
Terminal Descent Propulsion	JPL
Terminal Descent Guidance	JSC
TD Control System (engine thruster timing logic)	JPL
Terminal Descent Aerodynamics	LaRC
Landing Radar	JPL
Terrain Radar	JPL
Lidar	JPL
Hazard Avoidance Logic	JPL
Terrain model	JPL
Monte Carlo Executive	LaRC

Table 3. Validation and Verification Process for POST-based Engineering Simulation

Model Developer

- 1) Formulates and Validates model
- 2) Develops and Validates computer code representation
- 3) Develops standalone test case
- 4) Provides Monte Carlo variables (types and variation)

Simulation Integrator

- 1) Implements model code into simulation
- 2) Demonstrates that test case can be duplicated
- 3) Provides test case results to model developer for confirmation
- 4) Develops typical simulation case – provides results to model developer for confirmation

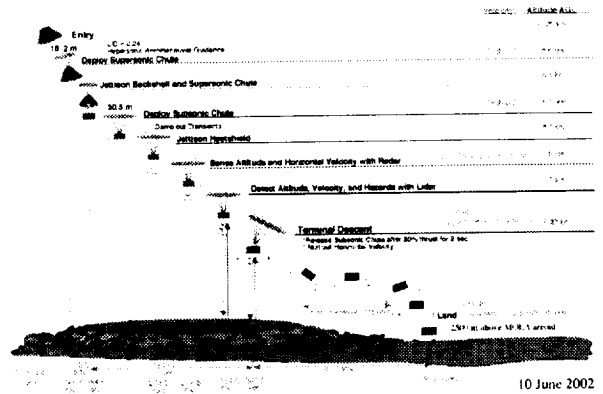


Figure 1. MSL Lander Entry, Descent and Landing Sequence.

Table 4. Monte Carlo Parameters and Variations

Parameter	Nominal	Distribution	3-σ or min/max
Entry EPA, deg	-14.0	Gaussian	0.60
Initial State Knowledge Error	0.0	Gaussian	-1/5 delivery error
Entry mass, kg (cg offset config)	2196 (1883)	Gaussian	2.0
Entry X CG, m (cg offset config)	1.296 (1.125)	Uniform	+ 0.0318
Entry Y CG, m (cg offset config)	0.0 (0.0)	Uniform	+ 0.0069
Entry Z CG, m (cg offset config)	0.0 (0.04936)	Uniform	+ 0.0069
Terminal Descent X CG, m	0.0	Uniform	+ 0.0318
Terminal Descent Y CG, m	0.0	Uniform	+ 0.0318
Terminal Descent Z CG, m	0.0	Uniform	+ 0.0318
Atmosphere Dispersion seed	0	Uniform	1 / 29999
Atmosphere Update Distance, km	0.5	Uniform	0.5 / 5.0
Atmosphere Dust Opacity	0.45	Uniform	0.1 / 0.9
Entry Maximum Bank Acceleration, deg/sec	5.0	Gaussian	10 %
IMU Initial Seed	0	Uniform	1 / 29999
IMU Initial Angular Misalignment	0.0	Gaussian	200
IMU Gyro bias drift, deg/hr	0.0	Gaussian	0.05
IMU Gyro scale factor, ppm	0.0	Gaussian	100
IMU Gyro nonorthogonality, ppm	0.0	Gaussian	60
IMU Gyro random walk (PSD), deg/hr	0.0	Gaussian	0.025
IMU Accelerometer bias, milligees	0.0	Gaussian	0.1
IMU Accelerometer scale factor, ppm	0.0	Gaussian	240
IMU Accelerometer nonorthogonality, ppm	0.0	Gaussian	210

Parameter	Nominal	Distribution	3-σ or min/max
Axial Force Coefficient Multiplier (Knudsen < 0.1)	1.0	Gaussian	5 %
Normal Force Coefficient Multiplier (Knudsen < 0.1)	1.0	Gaussian	10 %
Axial Force Coefficient Multiplier (Mach > 10)	1.0	Gaussian	3 %
Normal Force Coefficient Multiplier (Mach > 10)	1.0	Gaussian	5 %
Axial Force Coefficient Multiplier (Mach < 5)	1.0	Gaussian	10 %
Normal Force Coefficient Multiplier (Mach < 5)	1.0	Gaussian	8 %
Trim Angle of Attack Increment, deg	0.0	Gaussian	2.0
Supersonic Parachute Drag Coefficient	0.61	Uniform	+ 10 %
Subsonic Parachute Drag Coefficient	0.85	Uniform	+ 5 %
Axial Force Coefficient Multiplier (Mach < 0.8)	1.0	Uniform	+ 20 %
Normal Force Coefficient Increment (Mach < 0.8)	0.0	Uniform	+ 0.03
Pitching Moment Coefficient Increment (Mach < 0.8)	0.0	Uniform	+ 0.03
Parachute Separation Altitude of Lander (per axis), deg	0.0	Uniform	+ 30
Parachute Separation Altitude Rates of Lander (per axis), deg/s	0.0	Uniform	+ 20
Terminal Descent Engine Thrust Variation, N	3047 / 609	Uniform	+ 5 %
Terminal Descent Engine Specific Impulse, sec	194 / 210	Uniform	+ 0.67 %
Radar Altimeter Altitude Bias	0.0	Gaussian	3 %
Radar Altimeter Altitude Noise	0.0	Gaussian	1 %
Radar Velocimeter Velocity Noise	0.0	Gaussian	4 %

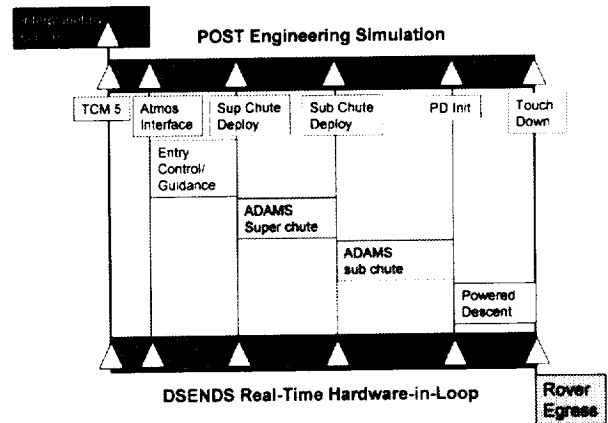


Figure 2. MSL EDL Simulation Strategy Overview.

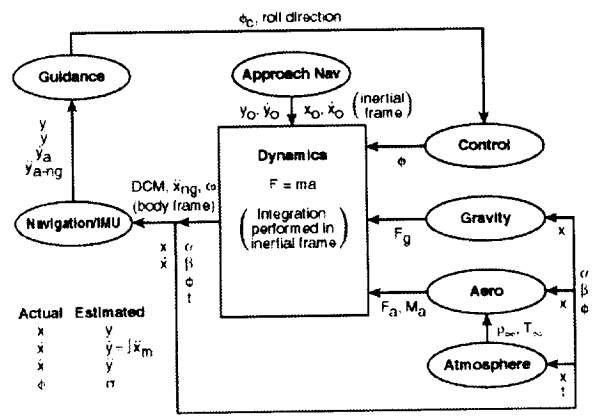


Figure 3. POST-based Simulation Models and Data Flow.

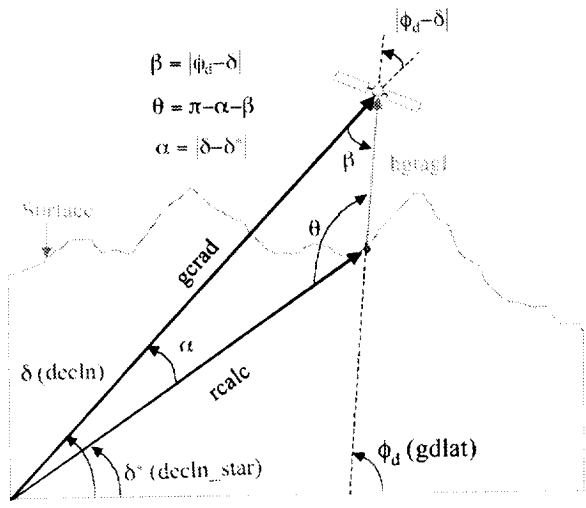


Figure 4. Altitude Calculation Parameter Definitions.

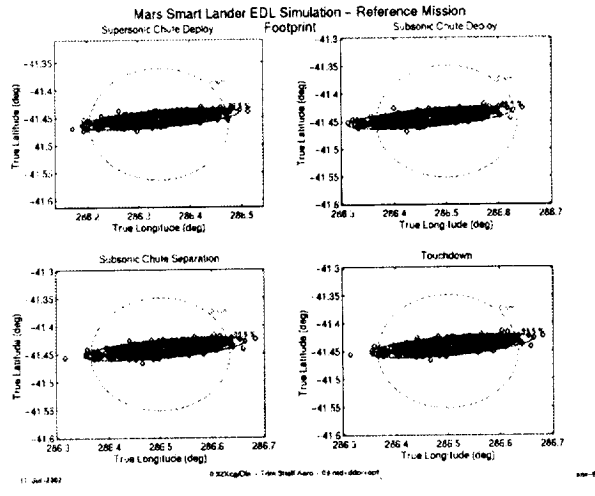


Figure 7. Sample Monte Carlo Results of Footprints Throughout EDL (Trim Shelf Configuration).

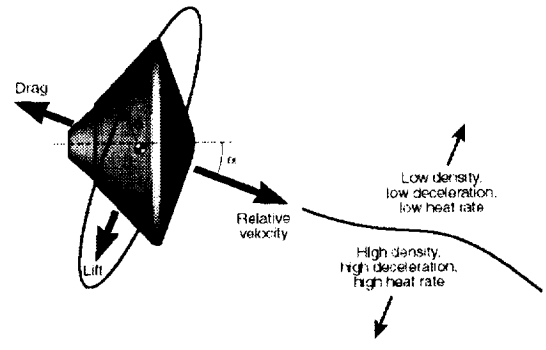


Figure 5. Hypersonic Aeromaneuvering Through Bank Angle Modulation.

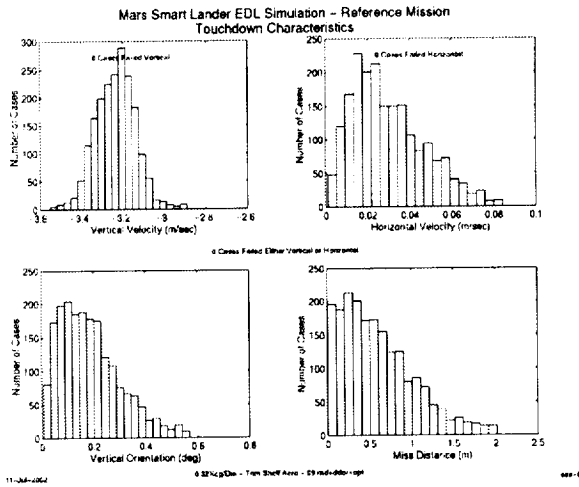


Figure 8. Sample Monte Carlo Results at Touchdown (Trim Shelf Configuration).

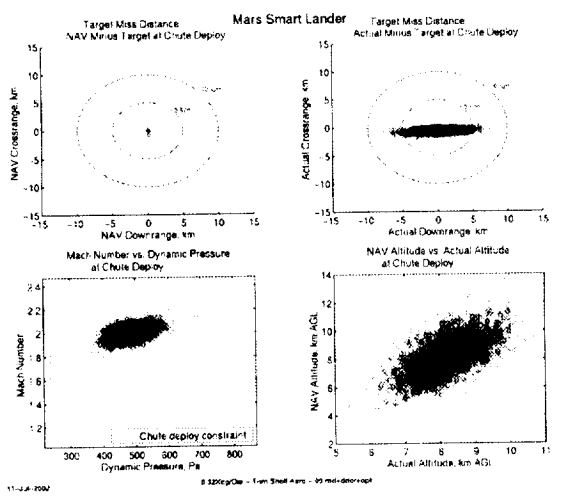


Figure 6. Sample Monte Carlo Results at Supersonic Parachute Deploy (Trim Shelf Configuration).

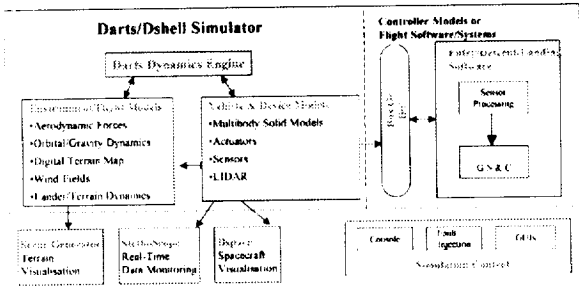


Figure 9. DSEDS Block Diagram.

EDL MODELS	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 Material Properties, Terrain-Vehicle Geometry, Elevation Maps,	LANDING SYSTEMS Hazard Sensor - Lidar, Radar, Orbital Beacons, Telecom Link Geometry, Airbags, Landing Snakes
	Albedo, Thermal Inertia	BUOYANCY SYSTEM Envelope, Inflation

Figure 10. DSENDs usage of the Darts/Dshell Model Library.

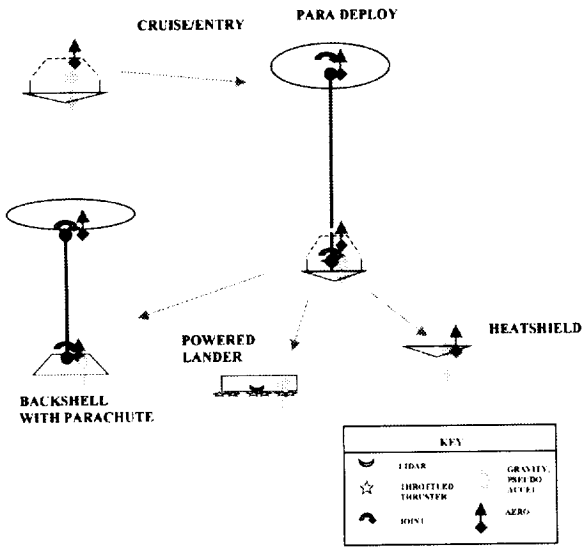


Figure 11. DSENDs Multibody Models for Flight Elements.

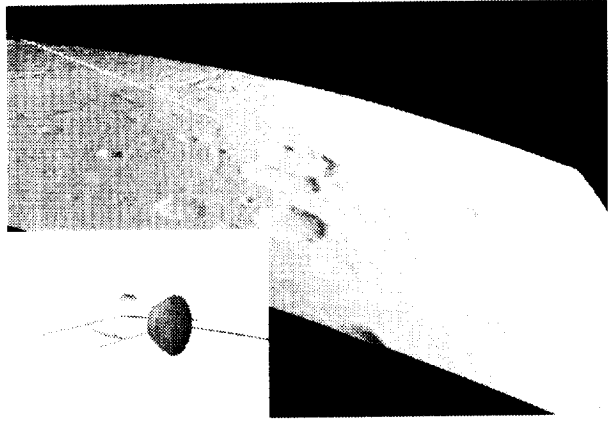


Figure 12. Screenshots from DSENDs Simulation Console.

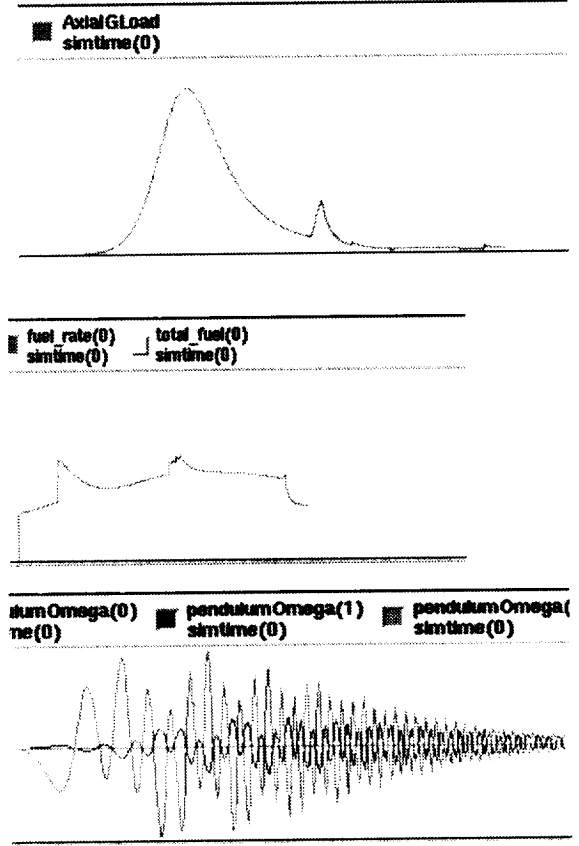


Figure 13. DSENDs Simulation Data Examples for G-Load, Fuel-Consumption, and Parachute Angular Motion.

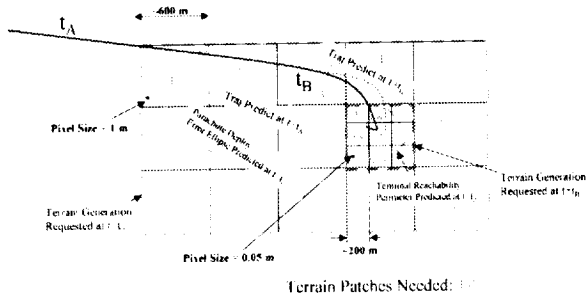


Fig 14. DSEDS Example Usage of Terrain During Spacecraft EDL.