MARS SCIENCE LABORATORY ENTRY DESCENT AND LANDING SIMULATION USING DSENDS

P. Daniel Burkhart, Jordi Casoliva, and Bob Balaram

The most recent planetary science mission to Mars was Mars Science Laboratory (MSL) with the Curiosity rover, launched November 26, 2011 and landed at Gale Crater on August 6, 2012. This spacecraft was the first use at Mars of a complete closed-loop Guidance Navigation and Control (GN&C) system, including guided entry with a lifting body that greatly reduces dispersions during the Entry, Descent and Landing (EDL) phase to achieve a $25 \, \text{km} \times 20 \, \text{km}$ landing error relative to the selected Gale Crater landing target. In order to confirm meeting the above landing criteria, high-fidelity simulation of the EDL phase is required. The tool used for 6DOF EDL trajectory verification analysis is Dynamics Simulator for Entry, Descent and Surface landing (DSENDS), which is a high-fidelity simulation tool from JPL's Dynamics and Real-Time Simulation Laboratory for the development, test and operations of aero-flight vehicles. DSENDS inherent capability is augmented for MSL with project-specific models of atmosphere, aerodynamics, sensors and thrusters along with GN&C flight software to enable high-fidelity trajectory simulation. This paper will present the model integration and independent verification experience of the JPL EDL trajectory analysis team.

INTRODUCTION

The most recent planetary science mission to Mars was Mars Science Laboratory (MSL) with the Curiosity rover, which launched November 26, 2011 and landed at Gale Crater on August 6, 2012. Curiosity is a rover with a significantly larger and more advanced landing payload than any previous Mars lander mission. In addition, MSL was the first use at Mars of a complete closed-loop entry Guidance Navigation and Control (GN&C) system, including guided entry with a lifting body (via center of gravity offset) to greatly reduce dispersions during the Entry, Descent and Landing (EDL) phase. The hypersonic entry guidance enables the entry body to fly out the remnant delivery error from the final Trajectory Correction Maneuver (TCM) and other sources, resulting in less than a $25 \, \text{km} \times 20 \, \text{km}$ landing error relative to the selected Gale Crater landing target.

In order to achieve landing within the above landing target uncertainty, a high-fidelity trajectory model of the EDL phase was required. A tool developed by JPL's Dynamics and Real-Time Simulation Laboratory was selected as the environment for high-fidelity simulation of EDL. This tool,
Dynamics Simulator for Entry, Descent and Surface landing (DSENDS), served as a tool for reference trajectory simulation used for cruise trajectory correction maneuver (TCM) design and for independent verification of the primary performance simulation used for MSL. While it was necessary to add MSL-specific capability to model atmosphere, aerodynamics, sensors, thrusters and GN&C flight software, the capability to model a multi-body spacecraft with gravity, aerodynamics, sensors and thrusters was available at selection. An overview of the DSENDS tool is presented first, both to define the framework of the tool and to describe the inherent capabilities in detail. This includes setup of mass properties and body tree for the MSL components, finite state machine configuration to interface the capsule configuration with flight software and a parachute model framework to accept user-supplied tabular parachute aerodynamics data.

Several MSL-specific models were required to complete the simulation. Environment models for atmosphere and winds based on mesoscale atmosphere simulations and Mars digital elevation maps were developed based on the arrival season and Gale landing site. Aerodynamics models for the entry capsule configuration and the supersonic parachute were constructed based on the MSL hardware. Onboard device models for the EDL inertial measurement unit (IMU) and entry attitude control thrusters were delivered. Finally the MSL EDL flight software was included to ensure execution of the EDL sequence as it was flown at Mars. Each of these models is briefly described. A detailed checkout of each model was performed as part of the model implementation to verify the models perform as expected in the simulation, which will be described. Also included here are models that were not implemented due to budgetary constraints, specifically the landing radar and landing engines, along with the reduced-fidelity powered flight model implemented to extend the simulation all the way to rover touchdown and the resulting impact to the analysis.

In addition to the simulation models, user input and output must be defined. The interfaces for trajectory and attitude initial conditions are detailed, along with configuration control and updating of other simulation setup parameters. Simulation output includes time history data and output of selected parameters at EDL events of interest, along with additional post-processed output for specific purposes, such as Navigation Ancillary Information Facility (NAIF) trajectory and attitude profiles. Additional data interfaces are required for dispersed trajectory or Monte Carlo analysis, which are multiple full-fidelity EDL simulations executed with randomly dispersed inputs. The framework and specification of dispersed input parameters is discussed, along with the mechanism for running Monte Carlo analyses for MSL. Monte Carlo output and post-processing for EDL performance evaluation is outlined. The results of these simulations are used to define statistics about the expected trajectory, timing and parameters of interest.

Finally, the performance of DSENDS as an independent verification and validation of the primary performance simulation will be discussed. The focus is mostly on the results of EDL simulation through parachute deploy, with modeling differences for the remainder of EDL discussed and results at landing also discussed.

MARS SCIENCE LABORATORY ENTRY, DESCENT AND LANDING OVERVIEW

A graphic showing the series of EDL events for MSL is shown in Figure 1. In the minutes before atmospheric entry interface, the cruise stage with the cruise support hardware and cruise balance masses added to move the capsule center of mass to the axis of symmetry are separated. The resulting offset center of mass results in aerodynamic lift, which is utilized during the entry phase by a modified Apollo entry guidance algorithm. Entry guidance commands the capsule bank angle to control the range flown by modulating the vertical lift component to deliver the capsule
to the desired parachute deploy location and velocity. The onboard state required for closed-loop guidance is computed via inertial propagation of the ground-provided initial conditions using an inertial measurement unit (IMU). Shortly before parachute deploy, MSL performs the Straighten Up and Flight Right (SUFR) maneuver in which entry balance masses are separated to re-align the capsule center of mass with the axis of symmetry and the parachute line of force. Deployment of the 21.5 m Viking-heritage disk-gap-band (DGB) parachute is commanded near Mach 2 based on navigated velocity. Once the system decelerates to around Mach 0.7 the heat shield is separated and a Ka-band narrow-beam Doppler altimeter/velocimeter begins collecting data. Once the onboard navigation filter converges to a solution using radar data, the eight throttle-able Viking-derived Mars Landing Engines (MLEs) are primed. The powered descent vehicle with the Curiosity rover attached separates from the backshell based on estimated altitude and velocity and performs an out-of-plane divert maneuver to avoid recontact with the backshell and then flies to a constant vertical velocity condition. The system has altitude margin built in to accommodate variations in both the terrain below the rover and measurement error. At a commanded altitude the rover is separated from the powered descent vehicle on a tether and the rover mobility is deployed for touchdown. The powered descent vehicle descends at a constant velocity until a persistent rover offloading is detected by GN&C. Once this has been detected, the tether is cut and the descent stage flies away to a safe distance.
**DSENDS DESCRIPTION**

The JPL MSL EDL trajectory analysis team had two major simulation actions for operations. The first was high-fidelity trajectory and simulation verification analysis of the POST2 simulation from the Langley Research Center, the primary performance simulation tool for MSL EDL analysis. This verification analysis required including the full suite of EDL simulation models including dispersed inputs for Monte Carlo analysis of EDL. The second was EDL trajectory simulation to support cruise TCM targeting and maneuver design, which required modeling of the EDL dynamics without closed-loop GNC. Both are covered in the following section. The simulation chosen for MSL was DSENDS (Dynamics Simulator for Entry, Descent and Surface landing). DSENDS provides the framework for modeling various aero-assisted simulations (EDL, aerobraking and aerocapture) with varying complexity from simple systems with single bodies and 3 degrees-of-freedom (DOF) dynamics to multi-body, flexible systems. In addition to the framework and as-delivered models, several MSL-specific models were added to complete the simulation. All are covered below.

The simulation toolkit contains an extensive library of models for sensors (e.g. inertial measurement units, cameras, encoders), actuators (e.g. motors, thrusters, propellers), environments (e.g. atmospheric models, 2.5- and 3-dimensional terrain models), environment interaction (e.g wheel slip, vehicle flight), and avionics elements (e.g. communication devices). A number of common control and navigation functions are available as well to close the loop in the absence of a user-supplied model, while providing a framework for the easy linking of user-supplied code such as control, estimation and sensor routines. A state-machine orchestrates mode and event-related functions including the activation and de-activation of elements in the data-flow. The simulation also provides an automated Monte Carlo, parametric simulation capability allowing automated dispatch of jobs onto computing clusters. In addition to the core simulation capabilities, a multi-stream continuous and event data logging facility is provided to facilitate data gathering. For the MSL EDL simulation, several project-specific modules were added, as described below.

**MSL SPECIFIC MODELS**

To have a complete MSL EDL simulation, the generic simulation capabilities of DSENDS were augmented with MSL-specific models developed for high-fidelity simulation and integrated into DSENDS. The table shown in Figure 2 was a reporting tool used to track the implementation progress of the models and is a list of the added models. The table shows the model version number if applicable, the first DSENDS version with the model, checkout progress and date of final checkout. The process of developing and integrating the models involved the model developers, the simulation teams that received the models from the developers and integrated them into the high-fidelity simulations, and the EDL systems engineering team who managed the effort. The models were of various types, some being complete callable functions that were compiled and called by the main simulation, others were data that were used by the built-in functionality of the simulations to represent the dynamics of interest, and yet others were mathematical models that were coded by the simulation team and integrated. For all models, checkout involved running a series of test cases and verifying with the model developers that the model is performing as expected and with EDL systems engineering that the simulation with the included models provides the simulation required for operations.

Each MSL-specific model is discussed in the following sections. Emphasis here is on the integration and checkout in the main simulation, as opposed to discussion of the model development and validation.
<table>
<thead>
<tr>
<th>GNC Models</th>
<th>Rev</th>
<th>Code</th>
<th>Test</th>
<th>Checkout</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCS Device Model</td>
<td>7</td>
<td>R4-01</td>
<td>100%</td>
<td>5/24/2011</td>
</tr>
<tr>
<td>DIMU Device Models</td>
<td>6</td>
<td>R4-01</td>
<td>100%</td>
<td>9/16/2011</td>
</tr>
<tr>
<td>Flight Software</td>
<td>R9.4.6</td>
<td>R4-03e-6</td>
<td>100%</td>
<td>5/1/2012</td>
</tr>
<tr>
<td>Other Selected Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parachute Model Updates</td>
<td>N/A</td>
<td>R4-03b</td>
<td>100%</td>
<td>11/30/2011</td>
</tr>
<tr>
<td>MSL Powered Flight Profile</td>
<td>N/A</td>
<td>R4-01</td>
<td>100%</td>
<td>9/9/2011</td>
</tr>
<tr>
<td>Aerodatabase</td>
<td>2.1</td>
<td>R4-01h</td>
<td>100%</td>
<td>6/15/2011</td>
</tr>
<tr>
<td>Terrain</td>
<td>N/A</td>
<td>R4-01d</td>
<td>100%</td>
<td>6/6/2011</td>
</tr>
<tr>
<td>Telecom Module</td>
<td>N/A</td>
<td>R4-01h+</td>
<td>100%</td>
<td>9/9/2011</td>
</tr>
</tbody>
</table>

Figure 2. DSEND MSL Model Checkout Progress table.

**MSL Atmosphere Model**

The Martian atmosphere, including atmospheric density, temperature, winds and dispersion of these quantities, was modeled based on mesoscale-derived atmospheric simulations of the Gale crater landing site at the defined landing season, or $L_s$. Data from the mesoscale simulations on a range of dates centered around the MSL arrival date and selected for a Gale local time for a range of hours around the expected arrival time of day. This range was selected to represent the mean atmosphere and the expected dispersions around that mean. From the raw mesoscale model data tables of the atmospheric and wind parameters of interest were generated as a function of altitude and longitude for use in the simulations. These tables were read by a version of MarsGRAM 2005 modified to read the tabular data and the dispersed output was generated using the MarsGRAM 2005 dispersion engine.\(^3\)

The first model check was to take detailed model output from the simulation for the atmosphere parameters and query the mesoscale tables by hand with the same inputs: as expected the output matches. These parameters include the MarsGRAM dispersion model parameters and the tabular atmosphere and wind data.\(^4\) Further, a set of dispersed simulations were run with the user random parameters (density scale factor and two wind parameters, all dispersed uniformly) varied discretely and independently across the range of the uniform random variables to show the output spans the expected range of wind and density values. Values for East-West and North-South wind magnitude, along with density, are compared with the tabular data reports and show agreement. As an example, the results for the East/West wind as a function of altitude is shown in Figure 3.

**MSL Aerodynamics Model**

To model the blunt-body aerodynamics of the entry capsule configuration, a software subroutine requiring capsule center of mass, wind-relative attitude, velocity and speed of sound that provides static and dynamic aerodynamic coefficients was provided and integrated with the simulation.\(^5\) To verify the model is working properly, the EDL simulation was executed and data for the simulation
Figure 3. East-West wind as a function of altitude, dispersed case.

inputs and outputs along with atmospheric and other environmental data were passed to the model developer. This is a different paradigm than for the other models, with the model developer performing the checkout. Output data both for a nominal case and a case with dispersed inputs were checked by the model developer.

Terrain Model

To properly model altitude above the Martian surface and the terrain of the landing site, a high-fidelity digital elevation map (DEM) was developed for the Gale Crater landing site. This map, coupled with the built-in highest-resolution MOLA Mars terrain (at 1/128° per pixel), supplied data for the Mars surface flown over during EDL. The DSENDLS terrain engine includes the capability to smoothly transition from one data source to another, greatly simplifying the work required by the MSL analysts to integrate the DEM.

The analysis performed for MSL began with the collection and processing of imaging from the High Resolution Imaging Science Experiment (HiRISE) on MRO. HiRISE images allowed for characterization of the local surface topography at length scales smaller than the Curiosity rover. Combined with the smaller landing ellipse due to guided entry, the project was able to characterize the entire expected landing terrain for the first time in the history of planetary exploration. The product of this work included a DEM that covered the entire MSL landing ellipse and extending beyond to
Verification of the complete terrain engine including both the MOLA and DEM products involved comparing stand-alone queries of the terrain data compared with output from the DSEND5 simulation. First, using the terrain model function calls found in the main EDL simulation, sets of longitude, latitude and surface altitude were generated from the MOLA data. The latitude and longitude output were then used as input to a stand-alone MOLA terrain query tool that performs a bilinear interpolation of the MOLA terrain for surface altitude. These two surface altitude profiles were compared, with agreement to numerical precision. This test was repeated with the DEM with similar results. A final test was performed using the EDL simulation with only the MOLA terrain and re-run with the MOLA terrain including the DEM. The plot of surface altitude versus longitude is shown in Figure 4, with curves of the 1/2° per pixel coarse global terrain, along with the highest-resolution MOLA terrain with and without the DEM. The coarse global terrain is clearly not accurate enough for high-fidelity simulation so was replaced over the Gale region with the MOLA data. The transition to use of the DEM happens near a longitude of 137.1° and the plot shows a smooth transition to the higher-fidelity data.

Figure 4. Surface altitude for an MSL EDL trajectory using the coarse global terrain (blue), using the high-resolution MOLA terrain only (red) and the combined MOLA and DEM data (green).
MSL Parachute Model

MSL used a supersonic Disk-Gap-Band (DGB) parachute during its EDL sequence. Thus far, every robotic mission to Mars has used heritage from the deceleration technologies developed during the Viking era in the 1960s. The MSL DGB parachute had a 33% larger diameter than the Viking parachute due to the heavier weight of the MSL spacecraft during EDL. The MSL parachute model includes tabular aerodynamic properties both as a function of parachute angle of attack and Mach number, a set of mass properties for the mechanical framework of the parachute and an area oscillation model that is active above Mach 1.4.

The supersonic MSL parachute consists of a parachute canopy, riser, confluence fitting and triple bridle legs as shown in Figure 5. The canopy and riser are modeled as a single rigid body with composite mass properties. The triple bridle attached to the capsule backshell is modeled as a single line rigidly attached to the backshell. The bridle and riser are connected by a ball joint, with all other joints shown in Figure 5 modeled as rigid links.

![Figure 5. MSL DSEND simulation body diagram for the parachute phase.](image)

The MSL parachute model consists of six components as listed below

1. Mach efficiency curve
2. Area ratio curve
3. Parachute aerodynamic coefficients $C_T$, $C_N$ and $C_M$
4. Parachute apparent mass
5. Parachute area oscillation
6. Parachute force
All the above components, along with the parachute body geometry relative to the capsule body, were validated individually to complete the model checkout required for use of the model in operations. The methodology used to verify the Mach efficiency, area ratio and aerodynamic coefficients involved comparing the simulation output with a hand-calculated tabular interpolation using appropriate simulation truth data as the independent variables in the interpolation (parachute Mach number and angle of attack, as appropriate). In addition to a nominal case, two dispersed cases were analyzed to ensure the dispersed values are properly computed. An example for the $C_T$ dispersed quantity is shown in Figure 6 for quantities relative to Mach number and angle of attack. In both plots, there is excellent agreement between the simulation and hand-interpolated table values. The apparent differences near the angle of attack peak in the left plot are an artifact of connecting the data points with straight lines. All the computed data points are on the hand-computed curve.

**Figure 6. Dispersed case #1 tangential parachute aerodynamic coefficient $C_T$**

Similar comparisons are made for apparent mass, area oscillations and the force computations, with the implemented model output compared with hand calculations with the appropriate input values.

**MSL Device Models**

MSL-specific device models for the Inertial Measurement Unit (IMU) and RCS thrusters were integrated into DSEND. Both models were built and maintained by the EDL GN&C team and source code was delivered to the simulation teams for integration. The GN&C team was responsible for unit and performance testing of the models. The GN&C team worked both with the hardware developers to verify the models performed as expected and with the DSEND team to ensure the inputs supplied to the models were properly defined and that the implemented model operated correctly within the simulation. Input and sign-off from all three groups was required to certify the IMU and RCS thruster models for operations use.

**Inertial Measurement Unit Model.** The checkout for the implemented IMU model was roughly split into internal and external tests, where internal checks refer to confirmation that the model input provide the expected output and external checks refer to ensuring the input and output interfaces to the model match the expected spacecraft model. Internal checks include measurement and data
packet generation along with the noise calculations. These are verified by running the simulation with the models active and recording both the telemetry and event records from the IMU model and flight software to verify the data are computed and passed properly. Noise is verified both by turning the noise on and off for subsequent runs to show a change, plus changing the random number generator seeds to ensure the output profiles are different. Other One-Variable-At-a-Time (OVAT) parameter checks are done to ensure all parameters can be changed by the user input interface, and that all parameters change as expected (the input parameters and internal variables match).

External includes nominal and dispersed location and orientation parameters. Timing of calls to the routine in the simulation framework are also checked. Both are verified as before with the model and flight software telemetry and event records. Input and output are checked, including dynamic states input and telemetry output. One check here is a full EDL run with all models active but all noise turned off in the IMU model and comparing the IMU-derived trajectory and attitude profiles with the simulation truth values. The output for this case is shown in Figure 7. Attitude is directly integrated using IMU attitude rate data and compared. For translation, the IMU acceleration is combined with a supplied gravitational acceleration to get the total acceleration that is integrated to get position and velocity. Gravity is modeled both in the flight software and the simulation truth and are compared here. Based on the level of agreement shown in these comparisons, the IMU model was certified for operations use.

Finally Monte Carlo dispersions, including internal noise seeds are checked by computing dispersed inputs for a full 8000 run Monte Carlo and computing statistics on the input. This check includes verifying the computed distributions have the desired type and specification and that the random samples for each parameter are independent distributions. This check was successfully performed for all dispersed parameters.

Reaction Control System Model. As for the IMU above, the Reaction Control System (RCS) model checkout was split into internal and external checks. Internal checks includes the electronics, valve, and combustion response models. Errors such as scale factor and thruster noise are verified, along with the pressurization force change (change from unpressurized to pressurized propulsion system). An example of the internal checks is analyzing a 1s pulse from a single thruster. The start and end of the pulse is shown in Figure 8. Included here are verification of the command transport delay, valve current profile, valve opening time and thrust profile.

External checks include location and orientation parameters, command delays and scheduling, I/O and telemetry and Monte Carlo dispersions, including noise seeds and dispersed model parameters. An example external test is firing a single thrust pulse with the thruster attached to the composite capsule body in the proper location with the proper orientation and comparing the capsule dynamics with a hand-computed force and moment calculation of the expected values. A minimum duration pulse is used to closely approximate an impulsive force. This check was completed successfully for all eight RCS thrusters. Finally, as for the IMU model, all dispersed parameters were checked to confirm they were dispersed and the profiles are independent.

Terminal Descent Sensor and Mars Lander Engines Two more device models were supplied for high-fidelity simulation of EDL but not implemented in DSEND due to budgetary constraints. These were models for the Terminal Descent Sensor (TDS) used to acquire surface-relative altitude and velocity measurements after heat shield separation and the Mars Lander Engine (MLE) model used for completing the final deceleration after release from the parachute to rover touchdown and fly-away of the descent stage.
To complete the end-to-end simulation, the trajectory profile used for powered flight was modeled separately as a new module and used to integrate the descent stage/rover trajectory after backshell separation using the truth trajectory and attitude states, which removed the need for the TDS model. To model the MLEs, a simple lumped thruster model that is part of the as-delivered DSENDS package was used to decelerate the powered flight vehicle and accurately estimate the propellant consumed. The resulting powered flight trajectory was compared to a stand-alone powered flight simulation used for the flight software module development with excellent agreement between the output.

**MSL Flight Software**

The MSL EDL GN&C flight software includes the core EDL GN&C functionality along with the EDL timeline engine to model the actual mode changes of the spacecraft. The inputs to the flight software are initialization data and sensor data from the IMU during operation. The TDS model, which is the other source of sensor data, is not modeled in this simulation and these sensor data inputs are not created. The initial conditions supplied included the time, position and velocity of the spacecraft at GNC start. The passing of DIMU sensor data was verified in the IMU device model checkout, discussed earlier. Also shown with the DIMU checkout is the performance of the state
propagation relative to simulation truth (see Figure 7).

Outputs from flight software are messages reporting data of interest (EVent Records, or EVRs), commands to fire the RCS thrusters and commands to fire various pyrotechnic devices (pyros) for separations, ejections and propulsion system pressurization. Pyros are used to separate components from the entry body (two cruise balance masses, 6 entry balance masses, deploy the parachute, separate the heat shield, and separate the powered descent vehicle from the backshell/parachute combination) and change the configuration of the propulsion system (activate and pressurize the RCS thrusters). Each pyro command was individually checked to ensure its timing is correct and that the simulation reacts properly to the pyro effect (i.e. separate the appropriate mass or make the correct change to the propulsion system). Commanding to fire the RCS thrusters was verified in the RCS model checkout, as shown earlier in Figure 8. Working with the GN&C team, the DSENDS team verified the proper operation of flight software in the simulations, correct input of sensor data, and expected simulation responses to all flight software output.

**MSL Telecom Module**

A final module was implemented to model the radio link performance between MSL and both relay orbiters at UHF bank and Earth at X band. This is different from the previously covered models as it does not impact the EDL trajectory but requires an accurate EDL trajectory along with the positions and orientations of the relay orbiters UHF antennae and the location of Earth. This module includes all the mathematics, antenna models, and other effects required to compute the received power. The module also reports geometric quantities of interest such as line-of-sight range, range rate and acceleration, along with antenna angles for the various links both for the modeled antenna and relative to the capsule anti-velocity direction.

The telecom engineers delivered to the simulation teams both a set of equations to compute the link budget along with test cases to verify the computations, with the DSENDS implementation of a telecom link module performed by the simulation team. The test cases consist of a series of single calculations at various points in the trajectory to cover both the UHF and X-band link calculations. The final telecom module output was checked by executing the test case in the simulation module.
with the output compared to the delivered results, both for UHF and X-band DTE test cases. All test cases were executed successfully. An example plot showing the UHF link performance with Mars Reconnaissance Orbiter is shown in Figure 9.

![Figure 9. Telecom link performance for UHF data link to Mars Reconnaissance Orbiter](image)

**MSL HIGH-FIDELITY SIMULATION**

With the MSL-specific models tested and integrated, the complete end-to-end simulation of MSL EDL is nearly complete. The final step is connecting these models together with the inherited DSENDS capabilities. A representation of the required EDL modeling for the complete simulation is shown graphically in an EDL timeline format in Figure 10 for models and events through parachute deploy and for EDL events after parachute deploy in Figure 11.

The final simulation requires the correct inputs before use, so a discussion of initialization follows. Once the simulation is properly configured, two usage cases are discussed. The first is generation of trajectory data products for project use with simulation output from cases with nominal input. The second is an extension of this, taking the nominal set of inputs, varying a subset of these, and executing the simulation multiple times to assess the system performance. Both are discussed below.
Simulation Initialization

The high-fidelity EDL simulation has over 30,000 parameters that must be specified for proper operation. A large subset of these parameters, over 90% of the total, were either managed by an EDL configuration control spreadsheet or were compiled into the software. The EDL configuration control spreadsheet values were mostly parameters related to models that did not have MSL-specific code provided, such as body geometry, thruster and IMU locations and orientations and antenna location and boresight values. Values coded into the MSL-delivered RCS, IMU and flight software modules were both data from model performance verification but also configuration choices.

The remaining 10% of the parameters can be divided into two groups. The first group are flight software parameters that had a defined update mechanism based on binary files that could be supplied to the flight vehicle and ground simulation tools to maintain a consistent setup between flight and ground. Updates to these values were managed by the flight operations process and configuration management of the active parameter files, with the ground simulation inputs updated as required when the flight values were modified. The second group are the set of parameters that are most likely to change or are planned to be updated during late approach. This group includes the EDL GN&C trajectory initial conditions (position, velocity, attitude and time), landing site latitude and longitude coordinates, entry guidance initial commanded bank angle, expected range flown from parachute deploy to landing and post-landing descent stage flyaway direction. This set of parameters was captured in the EDL Parameter Update File (EPUF).

The EPUF was used both to update the flight vehicle state and to initialize the ground simulation parameters listed above. The file was generated using an automated process that took as input
the best estimated approach trajectory of MSL and a handoff configuration file. Since the best estimated approach trajectory changes every time new DSN tracking data are processed by the navigation team, a series of distinct update opportunities were defined based on the expected navigation performance and available opportunities to command the spacecraft. The handoff configuration file was the source for the remaining planned update parameters. This file was reviewed and updated as needed within the flight operations process for evaluating changes to the onboard parameters, with the values captured in this file as the source for the uplink products that were created and the source for the ground simulation inputs.

**EDL trajectory nominal runs**

Nominal EDL runout trajectories were simulated from the Entry interface minus 9 minutes (E-9min) to touchdown each time a new orbit determination (OD) solution was computed. As MSL was cruising towards Mars, better state estimates were obtained and the propagated state of the spacecraft at E-9min varied. Thus, nominal runout simulations were used to quickly evaluate the sensitivity of state variation at E-9min with respect to the rover position at touchdown.

The EDL simulation output was used to create trajectory data products that were used for subsequent analysis. Each trajectory data product included the EDL simulated trajectory along with an approach trajectory segment until E-9min and a 10 minute segment that contained the touchdown position coordinates representing the static rover on the surface of Mars. The added segments were included on both ends of the EDL runout trajectory to allow the orbiter phasing and communication analysis teams\textsuperscript{9} to query trajectory state and attitude values before, during and after the EDL runout trajectory.
For each navigation solution, three EDL runout nominal trajectory files were generated with varying simulation fidelity. The main trajectory output was from the high-fidelity 6-DOF EDL simulation, used to compute the mapping between the spacecraft state at E-9min to touchdown. This output is used both for landing hazard analysis, discussed later, and to evaluate the need to update the onboard EDL flight software parameters. In addition, time histories of position, velocity and attitude were used to analyze the phasing of Mars Reconnaissance Orbiter, Mars Odyssey, and Mars Express with respect to MSL.

In addition to the full-fidelity 6-DOF EDL trajectory, two 3-DOF EDL trajectory runouts were also created using the targeting simulation, one with a complete sequence of EDL events and another that does not include parachute deploy or events after parachute deploy. The full-fidelity output was used to evaluate potential shifts on the touchdown ellipse so that hazard analysis could be performed and the probability of mission success could be computed. The 3-DOF output was used as the input trajectory models for EDL radiometric Doppler event monitoring during EDL.

One cross-check that was done for open-loop targeting cases was to compare the guidance design runouts with the targeting runouts, which compare the POST open-loop trajectory with the equivalent DSEND5 trajectory. While this was done as part of the targeting process for verification of inputs and the handoff of data, these cases were also useful to check the performance of the simulations. A set of trajectory data was defined for use in comparing the output of the simulations and plots comparing data for both sims was done as part of the process. An example is the plot shown in Figure 12 that compares the lift projected to the vertical plane. Note that differences before guidance start between the simulation output are differences in the modeling of the attitude before aerodynamics are significant and, thus, do not affect the trajectory. The differences at SUFR are due to slightly different balance mass ejection times, which are modeled differently in the open-loop simulations (note that the full simulation models these the same way). The other parameters show equivalent or better agreement between the simulations. In addition to aiding the targeting process, this also provides some independent V&V of several models, including mass properties, atmosphere and aerodynamics.

EDL Monte Carlo runs

An extension of the nominal high-fidelity EDL simulation was analysis of the EDL trajectory where the inputs are not nominal. The simulation is executed multiple times with randomly chosen values for a defined set of input parameters and the aggregate results analyzed to bound the performance of the system for the expected range of input parameters. This analysis is commonly referred to as Monte Carlo analysis.

The DSEND5 toolkit includes the capability to compute randomly dispersed inputs for a user-defined set of input parameters for a variety of random distributions. This includes arranging the inputs for a single sample of the random samples of each parameter into a file that can be read by the EDL simulation and applied as overrides to the nominal parameters. This is further organized to allow multiple EDL simulations with different parameters to be executed in parallel as independent processes, enabling the use of hundreds (or more) of processors to execute these individual dispersed cases. This ”embarrassingly parallel” process is well suited to execution using supercomputing resources and standard off-the-shelf job execution programs. For MSL, two JPL Supercomputing clusters were used, each with 512 processors available, for executing the 8001 run Monte Carlo simulations used for EDL performance analysis.
In order to minimize the execution time, much of the required processing to generate products used for analysis is moved out of the simulation execution itself and done as a post process. Most of the computation and report generation is automated as well to further increase throughput. In addition to plots, tabular data and summary comparisons referred to a scorecards were generated automatically for each Monte Carlo. An example output of the post processing are plots showing the navigated and true entry guidance delivery performance at parachute deploy, along with parachute deploy Mach versus dynamic pressure and altitude versus velocity, as shown in Figure 13.

**DSEND INDEPENDENT V&V**

One role of the DSEND simulation for MSL was an independent verification and validation (V&V) of POST, the primary performance simulation. Due to budgetary constraints, the original plan of using DSEND for V&V of the complete MSL EDL was reduced to comparisons at parachute deploy. This was a natural end point for a descoped cross-check as the full-fidelity modeling through parachute deploy was mostly the same functionality required for other roles such as cruise maneuver targeting. Figures 10 and 11 show the DSEND simulation components and their fidelity. As is shown in Figure 10, all the models through parachute deploy are full-fidelity models so should provide equivalent functionality. A parachute deploy comparison of a Monte Carlo solution for both simulations is shown in Figure 14. The result shows good agreement in the size and...
Figure 13. Example Monte Carlo post-processing output of entry guidance and parachute deploy performance.

Orientation of the dispersed points, with a slight downrange shift in the DSENDS derived ellipse of 400m relative to the POST ellipse. One factor contributing to the difference was attitude dispersion, which was done by each simulation at run time. Since the differences were acceptable to the project, no further work was done to quantify them. The MSL project concluded that the independent V&V was successful based on these results.

Extending beyond parachute deploy, the modeling begins to differ between the DSENDS and POST simulations. One difference was the parachute model. Both simulations used the same aerodynamics and parachute geometry inputs but the DSENDS simulation components, as shown in Figure 5, are all rigid components that were grouped into two rigid bodies connected by a pure ball joint. This purely mechanical model is different from the POST parachute implementation, which results in different dynamics on the parachute. The other major difference is the modeling of powered flight (see Figure 11). With the project choosing a reduced-fidelity powered flight model for DSENDS, this removed the need to implement the MLE model to control the lander through flight software and the use of truth data with an approximate thruster model for powered flight became an option. Without the need for commanding from flight software, the need to model the TDS radar also was eliminated. With these two major differences, the results in Figure 15 showing a slightly smaller DSENDS ellipse and maintaining the small downtrack difference from before can be explained. The main cause is likely the parachute model differences, as the trajectory dispersions during the powered flight phase do not change the landed ellipse enough to account for more than a small part of the difference. Despite these differences, the project was again satisfied with the successful independent DSENDS V&V.
SUMMARY AND CONCLUSIONS

Analyzing the performance of Entry, Descent and Landing (EDL) systems is an intricate, complex and multi-disciplinary task. All the EDL sub-systems operate synergistically, are very intertwined and it is generally difficult to make decisions based purely upon subjective judgement. The objective of developing a high-fidelity EDL simulation is to provide a tool that can quantify the EDL performance and risks associated to a tailored mission configuration. Furthermore, the EDL simulation tool can accommodate our latest knowledge of Mars and spacecraft conditions to obtain the best EDL performance predictions at a given time while approaching Mars. Scientists were continuously predicting Mars weather conditions near the Gale landing site, guidance engineers performed IMU calibrations, navigation engineers were constantly estimating the spacecraft state and four Trajectory Correction Maneuvers (TCM) were executed during the cruise phase. All of the above updates and calibrations had to be modeled in this high-fidelity simulation to be able to predict mission performance as accurate as possible at a given time. This high-fidelity tool has proven to be very effective for spacecraft operations, especially since the authors were able to evaluate the performance of different potential scenarios which equipped the decision-making process with quantitative predictions of mission performance and risk for each scenario.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contributions of Tomas Martin-Mur, the MSL navigation team chief, to the processing of hazard map files and the plotting tools used for the landed graphics shown in this paper. We also thank the DSEbDS simulation developers Christopher Lim, Jonathan Cameron and Kam Tso for their support both for software and for user support. Thanks also to Jeremy Shidner of the NASA Langley Research Center for providing the POST2 simulation.
comparison data shown in the figures and Gavin Mendeck of the NASA Johnson Space Center for the open-loop comparison data.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES


