Software

Aerobraking Maneuver (ABM) Report Generator

abmREPORT Version 3.1 is a Perl script that extracts vital summarization information from the Mars Reconnaissance Orbiter (MRO) aerobraking ABM build process. This information facilitates sequence reviews, and provides a high-level summarization of the sequence for mission management.

The script extracts information from the ENV, SSF, FRF, SCM FMax, and OPTG files and burn magnitude configuration files and presents them in a single, easy-to-check report that provides the majority of the parameters necessary for cross check and verification during the sequence review process. This means that needed information, formerly spread across a number of different files and each in a different format, is all available in this one application. This program is built on the capabilities developed in dragReport and then the scripts evolved as the two tools continued to be developed in parallel.

This program was written by Forest Fisher, Roy Gladden, and Teerapat Khanampan of Caltech for NASA’s Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-44883.

ABM Drag_Pass Report Generator
dragREPORT software was developed in parallel with abmREPORT, which is described in the preceding article. Both programs were built on the capabilities created during that process. This tool generates a drag_pass report that summarizes vital information from the MRO aerobraking drag_pass build process to facilitate both sequence reviews and provide a high-level summarization of the sequence for mission management. The script extracts information from the ENV, SSF, FRF, SCM FMax, and OPTG files, presenting them in a single, easy-to-check report providing the majority of parameters needed for cross check and verification as part of the sequence review process.

Prior to dragReport, all the needed information was spread across a number of different files, each in a different format. This software is a Perl script that extracts vital summarization information and build-process details from a number of source files into a single, concise report format used to aid the MPST sequence review process and to provide a high-level summarization of the sequence for mission management reference. This software could be adapted for future aerobraking missions to provide similar reports, review and summarization information.

This program was written by Forest Fisher, Roy Gladden, and Teerapat Khanampan of Caltech for NASA’s Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-44884.

Visualization Component of Vehicle Health Decision Support System

The visualization front-end of a Decision Support System (DSS) also includes an analysis engine linked to vehicle telemetry, and a database of learned models for known behaviors. Because the display is graphical rather than text-based, the summarization it provides has a greater information density on one screen for evaluation by a flight controller. This tool provides a system-level visualization of the state of a vehicle, and “drill-down” capability for more details and interfaces to separate analysis algorithms and sensor data streams.

The system-level view is a 3D rendering of the vehicle, with sensors represented as icons, tied to appropriate positions within the vehicle body and colored to indicate sensor state (e.g., normal, warning, anomalous state, etc.). The sensor data is received via an Information Sharing Protocol (ISP) client that connects to an external server for real-time telemetry. Users can interactively pan, zoom, and rotate this 3D view, as well as select sensors for a detail plot of the associated time series data. Subsets of the plotted data can be selected and sent to an external analysis engine to either search for a similar time series in an historical database, or to detect anomalous events.

The system overview and plotting capabilities are completely general in that they can be applied to any vehicle instrumented with a collection of sensors. This visualization component can interface with the ISP for data streams used by NASA’s Mission Control Center at Johnson Space Center. In addition, it can connect to, and display results from, separate analysis engine com-
ponents that identify anomalies or that search for past instances of similar behavior.

This software supports NASA's Software, Intelligent Systems, and Modeling element in the Exploration Systems Research and Technology Program by augmenting the capability of human flight controllers to make correct decisions, thus increasing safety and reliability. It was designed specifically as a tool for NASA's flight controllers to monitor the International Space Station and a future Crew Exploration Vehicle.

This program was written by Joseph Jacob, Michael Turmon, Timothy Stough, and Herbert Siegel of Caltech and Patrick Walter and Cindy Kurt of United Space Alliance for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-43952.

### Problem Reporting System

The Problem Reporting System (PRS) is a Web application, running on two Web servers (load-balanced) and two database servers (RAID-5), which establishes a system for submission, editing, and sharing of reports to manage risk assessment of anomalies identified in NASA's flight projects. PRS consolidates diverse anomaly-reporting systems, maintains a rich database set, and incorporates a robust engine, which allows tracking of any hardware, software, or paper process by configuring an appropriate life cycle. Global and specific project administration and setup tools allow lifecycle tailoring, along with customizable controls for user, e-mail, notifications, and more. PRS is accessible via the World Wide Web for authorized user at most any location.

Upon successful log-in, the user receives a customizable window, which displays time-critical “To Do” items (anomalies requiring the user’s input before the system moves the anomaly to the next phase of the lifecycle), anomalies originated by the user, anomalies the user has addressed, and custom queries that can be saved for future use. Access controls exist depending on a user’s role as system administrator, project administrator, user, or developer, and then, further by association with user, project, subsystem, company, or item with provisions for business-to-business exclusions, limitations on access according to the covert or overt nature of a given project, all with multiple layers of filtration, as needed. Reporting of metrics is built in. There is a provision for proxy access (in which the user may choose to grant one or more other users to view screens and perform actions as though they were the user, during any part of a tracking life cycle — especially useful during tight build schedules and vacations to keep things moving). The system also provides the ability to have an anomaly link to or notify other systems, including QA Inspection Reports, Safety, GIDEP (Government-Industry Data Exchange Program) Alert, Corrective Actions, and Lessons Learned.

The PRS tracking engine was designed as a very extensible and scalable system, able to support additional applications, with future development possibilities already discussed, including Incident Surprise Anomalies (for anomalies occurring during Operations phases of NASA Flight projects), GIDEP and NASA Alerts, and others.

This work was done by Don Potter, Charles Serian, Robert Swet, Babah Sapir, Enrique Gamez, and David Mays of Caltech for NASA's Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-44222.

### G-Guidance Interface Design for Small Body Mission Simulation

The G-Guidance software implements a guidance and control (G&C) algorithm for small-body, autonomous proximity operations, developed under the Small Body GN&C task at JPL. The software is written in Matlab and interfaces with GOPT, a JPL-developed optimization package written in C that provides G-Guidance with guaranteed convergence to a solution in a finite computation time with a prescribed accuracy. The resulting program is computationally efficient and is a prototype of an onboard, real-time algorithm for autonomous guidance and control.

Two thruster firing schemes are available in G-Guidance, allowing tailoring of the software for specific mission maneuvers. For example, descent, landing, or rendezvous benefit from a thruster firing at the maneuver termination to mitigate velocity errors. Conversely, ascent or separation maneuvers benefit from an immediate firing to avoid potential drift toward a second body. The guidance portion of this software explicitly enforces user-defined control constraints and thruster silence times while minimizing total fuel usage.