

# Mars Exploration Rover Mobility and IDD Downlink Analysis Tools

Jeffrey J. Biesiadecki and Robert Liebersbach and Mark W. Maimone  
Jet Propulsion Laboratory, California Institute of Technology  
Pasadena, CA USA  
mark.maimone@jpl.nasa.gov

**Abstract**—Driving and manipulation activities on the Mars Exploration Rovers often must be planned from scratch each day, making this a fast-paced activity by typical space mission standards. Such a schedule requires rapid turnaround in understanding what happened during activities on the previous Martian Solar Day (Sol). In addition to obvious faults, subtle changes in expected behavior must be detected and understood rapidly, to prevent the next day’s plan from exacerbating any imminent problem.

We developed an advanced set of tools to help speed up the understanding of previous Sols’ activities. Plots of detailed engineering data are generated automatically and incorporated into HTML pages, allowing quick visual inspection of each activity. Data from different sources are grouped together and written into “Annotated CSV” (Column-Separated Value) files, enabling detailed analysis even in the presence of only partial data, enabling multi-Sol queries (including those spanning the whole mission lifetime), and greatly reducing the need for all operations team members to acquire a deep understanding of the many constituent pieces of data. Automated processing of newly received data informs the team of the current vehicle state via email and multimedia cell phone updates.

Our system has enabled rapid understanding of complex mobility and manipulation activities during the first four years of Mars Exploration Rover operations. This paper describes the tools developed by the downlink team that perform the bulk of this analysis.

**Index Terms**—MER, Mars rover, Ground software, Downlink Assessment, Telemetry Analysis, Partial Data, Mobility and Manipulation

## I. BACKGROUND

The goal of the Mars Exploration Rover (MER) project is to elucidate the history of water on the surface of Mars. The scientific instruments on the vehicles must be deployed at multiple surface locations to perform the *in situ* analysis needed to support this goal. Such activities are only possible through use of mobility and Instrument Deployment Device (IDD, or robotic arm) manipulation subsystems, which must be monitored as part of the daily operations process [11], [16], [17], [7].

Telemetry data generated on a spacecraft follow a long and circuitous route to Earth. Data are created by the

various flight software (FSW) subsystems [13] (e.g., the Mobility software module [2]), and are typically stored in Flash memory prior to transmission [15]. Data are usually relayed through UHF frequencies to an orbiter at Mars (e.g., Mars Odyssey) or can be transmitted directly to Earth at lower bandwidth using X-band frequencies. The Deep Space Network receives the raw data and forwards it to the Jet Propulsion Laboratory (JPL), where processes controlled by the Ground Data Systems [19] and Mission Data Operations Teams automatically archive the raw data, generate preliminary reports [18], and populate an archived filesystem with individual Data Product files. These files are stored in a format similar to that used in the onboard flash memory, but they are split into a directory structure based on the Martian Solar Day on which they were created (each *Sol* is about 24.6 hours long). The tools described in this paper deal with the data only after it has already been populated into the UNIX filesystem.

Several issues complicate the analysis of mobility and manipulation operations. Spacecraft telemetry is typically a disjoint set of independent pieces of information (text messages, scalar “channelized” values with only approximate timestamps, and individual data products each adhering to a particular fixed format), so discovering correlations among raw downlink data and the commands that created them can be difficult. The format of the generated data has evolved over time, since there have been four updates to the MER flight software between launch in June 2003 and December 2007 [9]. Most importantly, only a fraction of the engineering data generated by the rovers is typically received on the same day it was created, so we often have to describe the activities given only partial information. In addition, the distributed nature of the operations team makes it critical that results be viewable from sites outside the primary physical operations area.

Although there are many specialized interactive tools that perform specific useful downlink analysis duties (e.g., our nominal 3D visualization and planning tool RSVP [20], remote science planning tool SAP/Maestro [12], image

browser Marsviewer [6], and science team visualizer Viz [3]), they did not (or still do not) support detailed views into all the necessary data products and were not amenable to automated scripting or remote display. So we developed a system comprised of tools written in C and C++ that parse the low-level information, and scripts written in Perl that organize the results and orchestrate the automated analysis. Our system produces plain text and HTML-annotated reports either automatically or interactively, and provides interactive query tools that work effectively via remote access.

## II. RAW DATA

There are three categories of data produced by the MER flight software:

- 1) “Event Reports” (or EVRs) are timestamped non-periodic text messages with a severity level and up to six scalar parameters. One important use of EVRs is to trace progress through executing sequences, indicated by command dispatch and completion reports. Generally a handful of EVR messages are generated per minute, though it depends on the activities being performed.
- 2) Engineering, Housekeeping, and Accountability “Channelized Telemetry” (EH&A) comprise independent scalar values (commonly sensor measurements and flight software state variables) recorded roughly once every ten minutes regardless of what activities are being performed. The timestamp associated with EH&A data is only approximate; it reflects the multi-minute poll time, not the time the data was generated.
- 3) Timestamped binary files called Data Products. Each flight software subsystem generates its own data products, with unique format and information content. Although many of these products have XML descriptions, there is little consistency in how their various components are labeled. All images, and nearly all of the detailed time sampled information are reported in data products.

Each Data Product is tagged with additional meta-data including a numeric label representing its type, the name of the sequence and number of the command that created it, and the time at which it was created. Certain critical products also report the Rover Motion Counter (RMC), a set of five integer indices that uniquely identifies each spacecraft configuration. All timestamps are logged in “Spacecraft Clock” (or SCLK) units.<sup>1</sup> The vast majority

<sup>1</sup>SCLK is the number of seconds since 1 January 2000 11:58:55 UTC.

of data therefore comprises time-tagged scalar quantities, much of which is only available inside Data Products.

There is some non-temporal data as well. Scientists and Rover Drivers generate lists of 3D targets and features of interest (indexed by RMC and location rather than SCLK), either explicitly or implicitly in the sequences of commands that execute drive and manipulation activities. Also, the Rover Driver sequences of commands often contain useful predictions or annotations in command arguments and comments. These can be automatically correlated with the received data by tracking their execution time using the “Command Dispatched” EVR text messages.

How do we know whether all of the generated data has been received? The Data Product Summary report (which is itself a data product) includes a catalog of the data products currently stored onboard, including those that were created earlier during the current Sol. The combination of this report’s contents and the original command sequences with EVR annotations informs the analyst not only what data has yet to be downlinked, but also when no further information should be expected.

**Mobility Data Products** The Mobility flight software produces several types of data products which are used to track the path driven by the rover, describe characteristics of the terrain that was driven over (such as terrain slope and roughness, and wheel traction), assess and monitor mobility mechanism health, and troubleshoot mobility faults. These data products consist of the following main types:

- 1) Parameter dumps, which contain the control parameters currently in use by the rover and are generated by an explicit dump-parameters command.
- 2) Rover kinematic state, which is a snapshot of rover pose estimate and all mechanism positions (e.g., rocker/bogie suspension). These are very small in size and are generated by explicit dump-state command
- 3) Mobility summary, which contains information captured after every drive segment regardless of whether the motion was explicitly commanded or selected by autonomous navigation [2]. This includes intended segment path length and heading change, final pose estimate after maneuver, statistics about any goalward and anti-goalward terrain assessment images taken, delta pose computed by Visual Odometry [8], and average current draw and final positions of mobility mechanisms (steering, wheels, and suspension). These small products typically are assigned a high priority for the telemetry scheduler. Therefore they are often received even during low-volume communication sessions, and provide an overview of the Sol’s mobility activities, including maneuvers selected by

the autonomous navigation system.

- 4) Detailed motion history, which contains mobility mechanism state recorded up to eight times per second during commanded motion. This state includes mechanism positions, motor currents and applied voltages, as well as Inertial Measurement Unit data and rover pose estimate. These products allow higher fidelity monitoring of mechanism health and vehicle/terrain interactions. Because they can grow fairly large, they are given a lower priority for telemetry scheduling, can be downsampled to twice per second for long drives, and may be losslessly compressed prior to transmission. However, should a mobility fault occur during motion (such as excessive tilt or current draw; see [2] for a complete list) or should the flight computer reset during motion, the last thirty seconds of non-downsampled motion data will be telemetered at high priority.
- 5) Terrain assessment and Visual Odometry images. All images processed by the navigation system are kept on-board and available for transmission if there is a problem. Should a mobility fault be triggered, the most recent autonomously acquired images are given a high priority for telemetry scheduling. This increases the likelihood of being able to see the terrain now under the rover, which would otherwise not be currently visible to any camera.
- 6) Navigation diagnostic information such as individual path evaluations, single-step Visual Odometry and D\* path planning statistics, and stereo correlation statistics for up to ten image pairs processed at each step.
- 7) Navigation maps can include the Visual Odometry list of 2D tracked features, and dozens of products related to the autonomous hazard detection and avoidance system. These include elevation map, terrain goodness map, slope map, stereo disparity results, and memory map of the buffer area dedicated to the autonomous navigation system.

**Instrument Deployment Device Data Products** The IDD flight software produces a similar set of data products - compact summary products and larger detailed motion history products with arm joint angles, motor currents, and software state. Similarly, the Rock Abrasion Tool flight software also produces both summary and detailed motion history products used to characterize mechanism health and properties of the rock being ground. In fact all motors will produce similar detailed history reports when operated directly using low-level motor commands.

### III. DATABASES

The Ground Data System team provides the complete mission data archive, and provides query tools for EVRs (text messages) and EH&A (channelized telemetry). But queries can be slow, EH&A data are only small samples of vehicle state (being logged no more often than once per minute), and their database does not permit *any* searches into the contents of individual Data Products (apparently a recent innovation in spacecraft engineering operations). JPL's Planetary Data System Navigation and Ancillary Information Facility [5] provides some detailed mobility data as SPICE kernels (e.g., rover and site position SPK files), but does not include all mobility-related information and is not available in real-time. So we created additional databases which are stored in *Annotated Column-Separated Value* (ACSV) files. A plain CSV file is a spreadsheet in text format; our files include not only the spreadsheet but also the meta-information needed to describe its contents. A sample ACSV header is shown in Figure 1.

```

# MER NAV Datasets  corrected_poses 3 ($ Id $)
# 1 SCLK sclk %13s seconds
# 2 Command Start f_start_sclk %13s seconds
# 3 Command Name f_command %s
# 4 Duration f_step_duration %6.2f seconds
# 5 Hazavoid f_hazavoid %s
# 6 Tolerance f_goal_tol %2.2f meters
# 7 Step Size f_step_size %4.2f meters
# 8 Site Frame f_rmc_site %3d
# 9 Drive Index f_rmc_drive %3d
# 10 Corr NORTH f_corr_north %9f meters
# 11 Corr EAST f_corr_east %9f meters
# 12 Corr DOWN f_corr_down %9f meters
# 13 Roll f_roll %9f radians
# 14 Pitch f_pitch %9f radians
# 15 Yaw f_yaw %9f radians
# 16 Tilt f_tilt %9f radians
# .....
# 39 Mission f_mission %s
# 40 Source f_source %s

```

Fig. 1. Partial example of an auto-generated Annotated CSV File Header. Each line gives the column number, an English description of the column, the formal database name used to represent the column, the printf-style format string used to generate the values themselves, and optional units associated with those values.<sup>2</sup> Only the header is shown here, the actual spreadsheet data would follow below. Each data row corresponds to a single SCLK time (given in column 1), and each space-separated entry on the row has the current value for each field written using the named printf-style format (or the fixed string "UNDEF" if no value was defined). Multiple values at the same timestamp are separated by commas.

Our database supports over 600 different engineering variables holding scalar values.<sup>3</sup> The ACSV files comprising the database are created by automatically parsing the text views of the raw data found in the shared filesystem as described in Section II. These raw files have wildly disparate formats; EVRs differ from EH&A, and very few

<sup>2</sup>Other database field attributes include absolute min/max bounds, and whether to ignore zero-resets and use nominal min/max values in plots.

<sup>3</sup>MER provides thousands of scalar variables in telemetry, but our attention was focused on those related to mobility, manipulation, and low-level motor activities.

of the data products share a common naming scheme and data format. So each data type is typically parsed by a specialized Perl function.

Certain compromises had to be made to deal with times when only part of the generated data has been received. If only EH&A is available, then it will be plotted. But whenever more detailed data products have been received near the same time, the EH&A data will be filtered out of any plots. This is necessary because the timestamp associated with EH&A data is only approximate, and has often resulted in unnecessary concern over strange jumps in the data when plotted at high temporal resolution; most low-level motor data is reported at 2-8 samples per second, but EH&A times may be artificially delayed by as many as tens of seconds. The EH&A system and some data products also inappropriately report out of bounds values during times when motors are actually switched off. These are pruned out automatically through the use of optional absolute bounds checks before the ACSV file is written.

ACSV files have multiple benefits. Being stored in plain text, they are immediately human-readable: they display easily in any text editor, browser or spreadsheet program, simplifying human inspection and verification of the database contents. Individual ACSV files are easily processed by column-oriented command-line tools (e.g., UNIX commands `awk`, `grep`, `perl`), and we developed annotation-parsing tools that allow us to search thousands of ACSV files quickly and effectively. They contain enough information to uniquely identify the numbers within (e.g., the units associated with each column). They minimize the negative impact of making changes to the file format over time because they are self-documenting: we can add new columns (or reorder existing ones) and yet still process older files just as easily. Finally, these files provide the *only* searchable database of the detailed motor, drive and arm history that is reported only in Data Products.

Our ACSV file database also enables fast multi-Sol queries. These are needed tactically (e.g., human Rover Drivers often set limits on expected vehicle tilt or suspension angles by inspecting the past several drives) and strategically (e.g., studying the a wheel's drive motor currents over the entire mission). They complete quickly because each query is converted into an auto-generated C++ program that is auto-compiled and parses the ACSV files looking only at the values of interest. Thus we get the benefits of both fast searches and database transparency.

#### IV. AUTOMATED REPORT GENERATION

##### A. Low-level Data Product Visualization

The lowest-level insight into rover operations comes from a direct interpretation of each individual data product

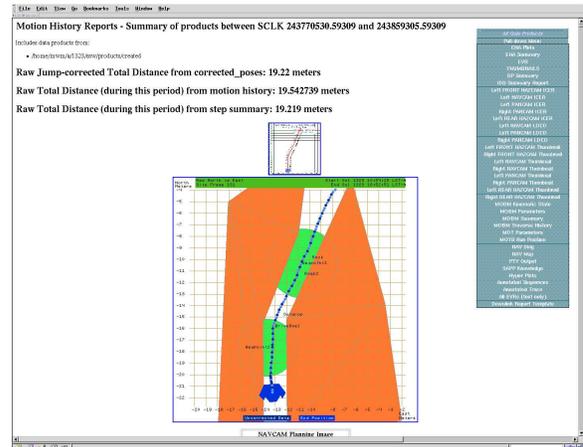


Fig. 2. Part of the Mobility/IDD HTML page summarizing Spirit's Sol 1323 Drive data. All data can be viewed at once, or a green pull-down menu can be used to limit attention to a particular class of downlink data. The course plot in the middle includes an auto-generated imagemap that provides hyperlinks to pages describing each individual plot point showing *all* data collected at that position.

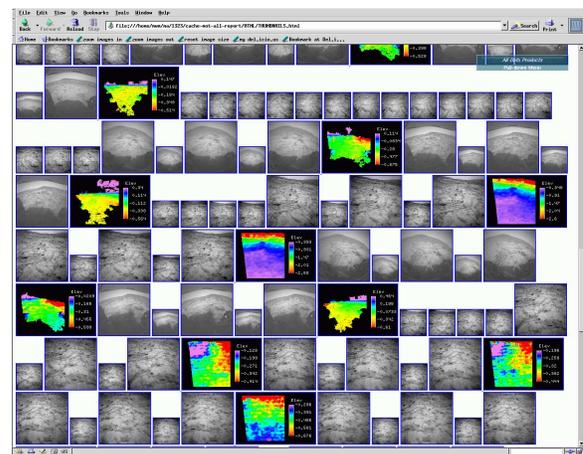


Fig. 3. Some of the A-1323 thumbnails included on the auto-generated HTML pages. Color annotations are elevation plots, computed on the ground by running stereo vision software on the 128x128 low-resolution images.

file. Each of these represents either a detailed view of a short activity or a compressed view of a set of many similar activities (e.g., multiple motions of the instrument arm). The values encoded in these data products are often most effectively viewed in graphical form. As it happened, the first data products that had to be explored in a visual way during flight software development were those related to the operation of individual motors. Once several independent tools had been written, these were merged into one tool that would create reports for any available motor

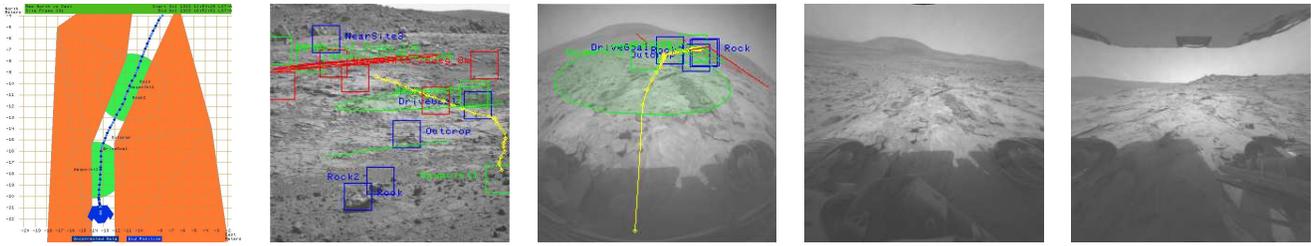


Fig. 4. Images comprising the Cell Phone Summary of Spirit's Sol 1323 Drive data: overhead Course Plot, some previous Sol's best view of the drive (obviously without tracks), the current Sol's best view of the drive (often showing tracks), distortion-corrected front Hazard-detection Camera (HAZCAM) view, and distortion-corrected rear HAZCAM view. The second and third images include auto-annotated features: actual drive telemetry showing where the center of the vehicle was (in yellow), Rover Driver-specified waypoint discs (in green) and keepout zones (in red), and labelled terrain features (in blue).

data products; this was called *mot-all-report*, and while the name remains its scope has increased dramatically. More than just motors, it now encompasses images, autonomous navigation products, sequences of commands automatically annotated with their corresponding EVR messages and more. Figure 2 shows the top of a web page generated by the current version of *mot-all-report*, and Figure 3 shows a thumbnail image summary page.

### B. High-level Visualization and Reports

The creation of ACSV databases that span a whole Sol of activities makes it possible to automatically generate many types of summary reports. A plot of the course driven by the rover is generated automatically each Sol. This includes annotations of not only those features included in the plan, but also those found by the rover (e.g., during Visual Target Tracking). Images are automatically annotated with plots of the rover's trajectory and interesting features. Textual reports summarize the total odometry driven, the amount of slip measured during a drive, the number and precision of individual arm placements, the EVR-based explanation for the first mobility fault of the Sol (if any), and both text and XML files summarize the rover's final state at the end of the Sol. *All* the activities performed that Sol are also summarized both in separate plots and also XML files which can be interactively replayed using RSVP. Short summaries of each Sol's activity (including course plots, annotated images, and fault summaries) are automatically forwarded to team members' email accounts and/or cell phones (see Figure 4 for sample imagery).

Dozens of reports are available, but due to limited space we will only describe three in detail.

**IDD/RAT Reports** The MER vehicles have exceeded their designed lifetime of 90 Sols (three months) by more than 15 times as of December 2007. As a result of their extended exposure, several components are exhibiting new failure modes, so all motors must be monitored closely.

For example, each IDD has begun to experience random current spikes that persist long enough to cause a fault condition and the immediate precluding of of the rest of the Sol's IDD and drive activity. This precaution is necessary because if a short were to occur it could damage the motor controller. Because each motor controller is used to control several motors on the rover, damage to the controller would result in the loss of several motors. A page was developed to display all critical information collected in the *idd-report* data product, data like motor current plots with the current limit displayed on the graph, joint positions and joint speeds. A similar page was built for the Rock Abrasion Tool (RAT) as it uses the same types of motors and requires comparable monitoring. With these pages downlink personnel can quickly look at the data collected during the last IDD or RAT activity and recommend any necessary precautions.

**EPEC** The geometric camera calibration on Spirit's Front Hazard Avoidance Cameras was found to be slightly out of alignment soon after landing. This was not enough to impact the 20cm resolution needed for terrain assessment, but it caused the rover to fail the 1cm placement accuracy requirement for the IDD. New camera models for IDD operations were obtained by acquiring full resolution stereo images of the IDD in different positions. At each position, the Mössbauer contact plate (MBCP) was located in the images, and the kinematic model of the IDD was used to determine its actual 3D position. This data was used to generate new ground-based camera models that are used for planning all IDD activities [14]. Since then, full resolution stereo images are typically captured every time the Microscopic Imager on the IDD is used, as this presents a clear view of the MBCP to the front HAZCAMs. *Mot-all-report* finds any such full resolution stereo front HAZCAM images with the MBCP in clear view and pointed toward the cameras and automatically runs the EPEC [1] tool to find the 3D coordinates of the MBCP, then computes and

tracks the discrepancy between the camera model derived position and the known kinematic position of the MBCP. Data is collected in such a way that new camera models can be quickly regenerated should the current ground models slip out of alignment again.

**Hyperplots** Sometimes it can be helpful to see all the data associated with a particular drive location. Each Sol's drive activities are summarized in several graphs related to the drive: three attitude plots (time vs Roll, Pitch and Yaw), and three plots projecting the 3D position history into two dimensions (East vs North, North vs Down and, East vs Down). Each point on these plots is automatically associated with a hyperlink to a web page that includes all ACSV spreadsheet data available at that point, including images; dots without associated images are a darker color than dots with images. On the East vs North graph in particular, a layered map of the waypoints, keepout zones and targets is also included (e.g., see the course plot in Figure 2). While generally this is used to display data from one Sol it can also be used to display multi-Sol drives.

## V. INTERACTIVE QUERY TOOLS

When the rovers first landed in January 2004, only a small number of fragile tools were available. For example, our primary course plotting tool depended completely on our receiving Mobility Step Summary Data Products. When those products were not received, we had to reconstruct the rover course by hand, manually extracting positions from EVRs, EH&A, image headers, and other data products. But now all those sources are automatically parsed into ACSV files, and inspection of mobility results is as easy as selecting a given rover and Sol (by changing into the corresponding flight operations directory) and running short commands from the UNIX shell.

Dozens of tools are available, but due to limited space we will only describe three in detail.

The primary interactive query tool is called *showme*. *showme* was originally developed for the tactical analysis of a single Sol's worth of data. It reads data from the current directory and matches fields named on the command line against those available in the ACSV files. For example, to see the best view of the rover's pitch throughout the entire Sol, simply type `showme pitch`. Note that the source of the data is no longer relevant, since the ACSV database combines data from many sources and accepts queries using just the field of interest. Field names are substring-matched against the available names, so `showme tilt` will display all four available tilt measurements (Tilt, Raw Tilt, Northerly Tilt, Easterly Tilt). Myriad options exist, several of which are enumerated here; see the similarly-labeled sub-plots in Figure 5 for an illustration of each of

these commands using Spirit's Sol 1323 data:

- 5a Plot one or more fields vs. time (SCLK seconds or fractional Sol number). By default the horizontal axis represents time. `showme roll`.
- 5b Plot one or more fields vs. some other field (e.g., plot East (Y) coordinate vs North (X) coordinate to see a simple overhead course plot using `showme east v north`).
- 5c Plot one or more fields vs. Sample Number instead of time (eliminating gaps during which values do not change) `showme -samples drive_current`.
- 5d Label plots with the name of the command that was executing at that time `showme -commands pitch`.
- 5e Ignore resets to zero (e.g., on Encoder counters between motor commands, on rover positions during Site Frame increments, or an RMC-based index resetting back to zero) `showme +zero f_rmc_pma`.
- 5f Restrict the time range being plotted `showme -start 12:30 -end 13:15 tilt`.

Other versions of this tool have evolved for more strategic analyses. *showmer* provides the same functionality using data spanning any number of Sols (even across the whole mission), automatically generating query-specific C++ code to quickly parse all the files. *showme-scaled* plots the detailed rover course labeled by drive mode.

The sheer number of files and Sols makes it impossible to use normal UNIX commands to search files spanning the whole mission; there are more files than shell filename expansion can handle in our environment. But a command called *sols* will execute any UNIX command over and over in a given range of Sol directories. This makes it possible to search through any of the text views (Event Reports, EH&A, text views of data products) spanning any number of Sols quickly and easily.

A list of three-dimensional features (e.g., rover locations, science features, drive waypoints, keepout zones) can be annotated in any image that has an associated camera model (described in [10], [4]). The original image serves as background, and the list of features will be plotted in the image and enumerated on the side, with each feature's range and position expressed in the Site Frame in which the image was taken. A 3D compass indicating the orientation of the original image is also added at the upper right. See the two examples in Figure 4 (the compass and list of points have been cropped in the figure).

## VI. FUTURE WORK

The tools described in this paper are contained in three interdependent systems. The MER Telemetry Extraction Reporting System (METERS) comprises the low-level



Figure 5a. showme roll

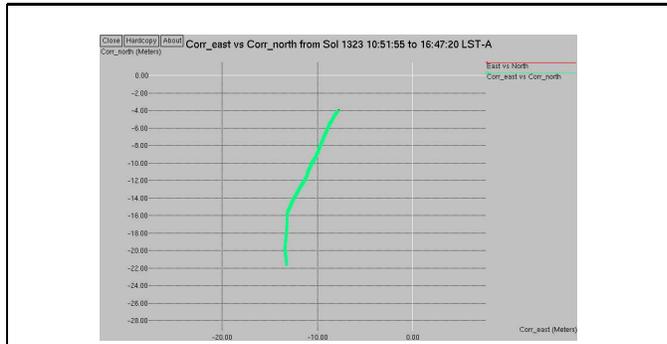


Figure 5b. showme east v north

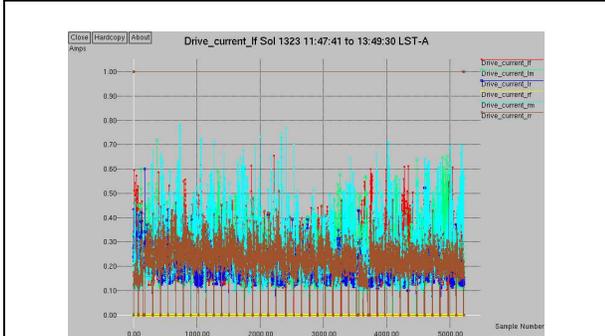


Figure 5c. showme -samples drive\_current



Figure 5d. showme -commands pitch



Figure 5e. showme +zero f\_rmc\_pma

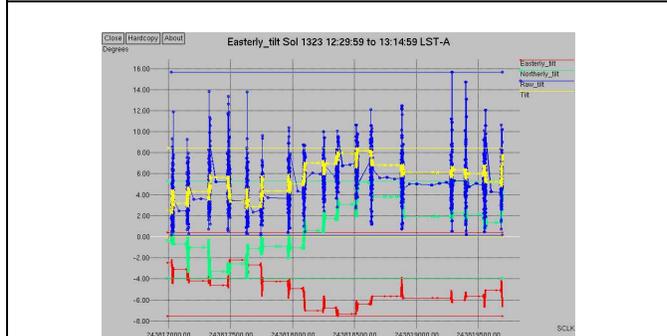


Figure 5f. showme -start 12:30 -end 13:15 tilt

Fig. 5. Screen captures of some interactive, zoomable plots generated by showme.

MER-specific data product tools. The Robotic Operations Automation: Mechanisms, Imaging, and Navigation Report Generation (ROAMING) system includes the generic ACSV database creation, maintenance and query tools. And the Scripts Providing A Cool Kit of telemetry Enhancing tools (SPACKLE) system includes tools generally useful to mission operations, not just mobility and manipulation analysis. We expect to port these tools for use in future mission FSW development and/or operations (e.g., the Mars Science Laboratory and possibly Phoenix missions).

Several important lessons were learned in the development of this software. We would have benefited greatly

from an additional RMC index to keep track of autonomously generated position and attitude updates. A reported position (e.g., in an Event Report) is not useful unless it has been tagged with RMC. Parameters and state information should have consistent labels (or at least consistent aliases) across all types of reported information. Data product viewers should be endian-independent, to enable processing on the fastest available system. Data displays are most understandable when you can tell what command generated them; so it is most appropriate to display data embedded within the original command sequence. This also implies there should be an automated

way to automatically retrieve a local copy of whatever command sequence executed onboard.

In the future we would like to enable queries grouping multiple database fields together (e.g., “find five or more contiguous steps drawing more than 0.25 Amps on average”). We would like to incorporate the resolution of timestamps and scalar values more deeply into the database to make it easier to merge values reported at different resolutions. And we would like to improve the interpretation of partial data using constraint-based reasoning. For example, we know the RMC fields are monotonically nondecreasing, so any non-unit jump or zero reset in an RMC index means data is missing; and we can also infer when all information has been received whenever the appropriate Data Product Summary report has been received.

## VII. CONCLUSION

The MER Mobility/IDD downlink tools have greatly sped up the understanding of new mobility and manipulation data, even when such data are only partially received. They implement the only database of detailed motor-based activities available to the operations team, and the primary means of visualizing detailed mobility activities and features in MER imagery. They provide tools helpful both for very detailed analysis over a small time scale, and general trend analysis over the whole mission timeframe.

## VIII. ACKNOWLEDGMENTS

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Thanks to Joseph Carsten for helpful comments and for providing the tools needed to process Field D\* data, to Chris Leger for providing the IDD Autoplace tools, and to the whole MER Mobility/IDD team for providing helpful feedback in the development of these tools (especially team lead Eddie Tunstel and overall MER Engineering Team lead Jake Matijevic).

## REFERENCES

- [1] Max Bajracharya, Matthew DiCicco, Paul Backes, and Kevin Nickels. Visual end-effector position error compensation for planetary robotics. *Journal of Field Robotics*, 24(5):399 – 420, May 2007.
- [2] Jeffrey J. Biesiadecki and Mark W. Maimone. The Mars Exploration Rover surface mobility flight software: Driving ambition. In *IEEE Aerospace Conference*, volume 5, Big Sky, Montana, USA, March 2006.
- [3] Laurence Edwards, Michael Sims, Clayton Kunz, David Lees, and Judd Bowman. Photo-realistic terrain modeling and visualization for mars exploration rover science operations. In *IEEE Conference on Systems, Man and Cybernetics*, The Big Island, Hawaii, USA, October 2005.
- [4] Donald B. Gennery. Generalized camera calibration including fish-eye lenses. *International Journal of Computer Vision*, 68(3):239 – 266, July 2006.
- [5] JPL Planetary Data Systems Navigation and Ancillary Information Facility. <http://naif.jpl.nasa.gov/naif/>.
- [6] P. C. Leger, R. G. Deen, and R. G. Bonitz. Remote image analysis for Mars Exploration Rover mobility and manipulation operations. In *IEEE Conf. on Systems, Man and Cybernetics*, Big Island, Hawaii, USA, October 2005.
- [7] Mark Maimone, Jeffrey Biesiadecki, Edward Tunstel, Yang Cheng, and Chris Leger. *Surface navigation and mobility intelligence on the Mars Exploration Rovers*, chapter 3, pages 45–69. TSI Press, San Antonio, TX, USA, 2006. <http://www.intelligentspacerobotics.com/>.
- [8] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the Mars Exploration Rovers. *Journal of Field Robotics*, (to appear), 2007.
- [9] Mark W. Maimone, P. Chris Leger, and Jeffrey J. Biesiadecki. Overview of the mars exploration rovers’ autonomous mobility and vision capabilities. In *International Conference on Robotics and Automation*, Rome, Italy, 2007.
- [10] J. N. Maki, J. F. Bell III, K. E. Herkenhoff, S. W. Squyres, A. Kiely, M. Klimesh, M. Schwochert, T. Litwin, R. Willson, A. Johnson, M. Maimone, E. Baumgartner, A. Collins, M. Wadsworth, S. T. Elliot, A. Dingizian, D. Brown, E. C. Hagerott, L. Scherr, R. Deen, D. Alexander, and J. Lorre. Mars Exploration Rover engineering cameras. *Journal of Geophysical Research*, 108(E12):12–1–12–24, December 2003. <http://www.agu.org/pubs/crossref/2003/2003JE002077.shtml>.
- [11] Andrew H. Mishkin, Daniel Limonadi, Sharon L. Laubach, and Deborah S. Bass. Working the martian night shift: The MER surface operations process. *IEEE Robotics and Automation Special Issue (MER)*, pages 46 – 53, 2006.
- [12] Jeffrey S. Norris, Mark W. Powell, Jason M. Fox, Kenneth J. Rabe, and I-Hsiang Shu. Science operations interfaces for mars surface exploration. In *IEEE Conference on Systems, Man and Cybernetics*, The Big Island, Hawaii, USA, October 2005.
- [13] Glenn E. Reeves and Joseph F. Snyder. An overview of the Mars Exploration Rovers flight software. In *IEEE Conference on Systems, Man and Cybernetics*, The Big Island, Hawaii, USA, October 2005.
- [14] Matthew L. Robinson, Eric T. Baumgartner, Kevin M. Nickels, and Todd E. Litwin. Hybrid image plane/stereo (hips) manipulation for robotic space applications. *Autonomous Robots*, 23:83–96, August 2007.
- [15] Joseph F. Snyder and David E. Smyth. Data management for mars exploration rovers. In *International Mars Conference*, Ischia Island, Italy, September 2004.
- [16] A. Trebi-Ollennu, E. Baumgartner, C. Leger, , and R. Bonitz. Robotic arm in-situ operations for the mars exploration rovers surface mission. In *IEEE Conference on Systems, Man and Cybernetics*, The Big Island, Hawaii, USA, October 2005.
- [17] Edward Tunstel, Mark Maimone, Ashitey Trebi-Ollennu, Jeng Yen, Rich Petras, and Reg Willson. Mars Exploration Rover mobility and robotic arm operational performance. In *IEEE Conference on Systems, Man and Cybernetics*, The Big Island, Hawaii, USA, October 2005.
- [18] Colette Wiklow, Jesse Wright, David Noble, Kathryn Studevant, and Joe Snyder. Reducing the cost of ground system development and mission operations using automated XML technologies. In *Fifth International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations*, Pasadena, CA, July 2003.
- [19] Colette Wiklow, Bryan Allen, Kelly Clarke, Young Lee, Jordan Lei, Kyran Owen-Mankovich, Priscilla Parrish, Gustavo Razo, Antonio Sanders, and Robert Wendlandt. The Care and Feeding of the Mars Exploration Rover (MER) Ground Data System (GDS). In *Ninth Annual Ground Systems Architecture Workshop*, Manhattan Beach, CA, March 2005.
- [20] John Wright, Frank Hartman, Brian Cooper, Scott Maxwell, Jeng Yen, and Jack Morrison. Driving on mars with rsvp. *IEEE Robotics and Automation Special Issue (MER)*, pages 37 – 45, 2006.