Real-Time EDL Navigation Performance Using Spacecraft to Spacecraft Radiometric Data

P. Daniel Burkhart*, Todd Ely* and Courtney Duncan*

Jet Propulsion Laboratory, Pasadena, CA, 91107, USA

Glenn Lightsey†, Todd Campbell‡ and Andy Mogensen‡

The University of Texas at Austin, Austin, TX, 78712, USA

A two-year task sponsored by NASA's Mars Technology Program's Advanced Entry, Descent and Landing (EDL) work area includes investigation of improvements to EDL navigation by processing spacecraft-to-spacecraft radiometric data. Spacecraft-to-spacecraft navigation will take advantage of the UHF link between two spacecraft (i.e. to an orbiter from an approaching lander for EDL telemetry relay) to build radiometric data, specifically the velocity between the two spacecraft along the radio beam, that are processed to determine position and velocity in real time. The improved onboard state knowledge provided by spacecraft-to-spacecraft navigation will improve the performance of entry guidance by providing a more accurate state estimate and ultimately reduce the landed position error. Work on the final year of this task is reported here. Topics include modifications to the previously selected algorithm set for implementation, and performance of the implemented algorithms in a stand-alone filter, on an emulator of the target processor and finally on a breadboard processing unit.

I. Introduction

The future investigation of Mars includes increased interest in placing landers at pre-selected landing locations of scientific interest. Key scientific goals for the next decade of Mars exploration, such as the search for water and characterization of aqueous process on Mars, the study of mineralogy and weathering of the Martian surface and the search for preserved biosignatures in Martian rocks, has driven the formation of the Advanced Entry, Descent and Landing (EDL) work area to enable *pinpoint landing* (defined for the purpose of this discussion as landing within 1 km of a preselected target). The National Aeronautics and Space Administration (NASA) Mars Technology Program formed this work area to investigate potential advances in critical components of EDL systems. A critical component of the closed-loop guidance, navigation and control (GN&C) system required for pinpoint landing is position and velocity estimation in real time. Spacecraft-to-spacecraft navigation will take advantage of the UHF link between two spacecraft (i.e. to an orbiter from an approaching lander for EDL telemetry relay) to build radiometric data, specifically the total count carrier phase of the Doppler shifted 2-Way coherent UHF signal, that are processed to determine position and velocity in real time. The improved onboard state knowledge provided by spacecraft-to-spacecraft navigation will reduce the landed position error and improve the performance of entry guidance.

A two-year task within the Advanced EDL work area will implement an EDL navigation system using spacecraft-to-spacecraft radiometric data. A previous paper documented the progress of the first year of this task, including a detailed description of the motivation, the spacecraft definitions, selection and documentation of the required algorithms and analysis results used to define the algorithm set.[1]

*Senior Staff Engineer, Guidance, Navigation and Control Section, 4800 Oak Grove Drive, M/S 301-125L, Senior Member AIAA.

†Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, 1 University Station C0600, and AIAA Member Grade.

‡Graduate Student, Department of Aerospace Engineering and Engineering Mechanics, 1 University Station C0600, and AIAA Member Grade.

Spacecraft-to-spacecraft navigation for EDL involves forming radiometric data with a radio link using the UHF band ($300-3000Mhz$) between the approaching lander and an orbiting spacecraft, as shown in Figure 1 for MSL. The UHF link will be made between the two spacecraft before atmospheric entry, but the focus of this effort is processing this data after cruise stage separation (roughly 10 minutes before entry). These data are available throughout EDL with the exception of periods during hypersonic flight where the ionizing plasma around the entry body makes closing the communication link difficult. Even with these outages, enough data can be collected and processed onboard in real time to significantly improve the onboard state knowledge during hypersonic flight.[2]

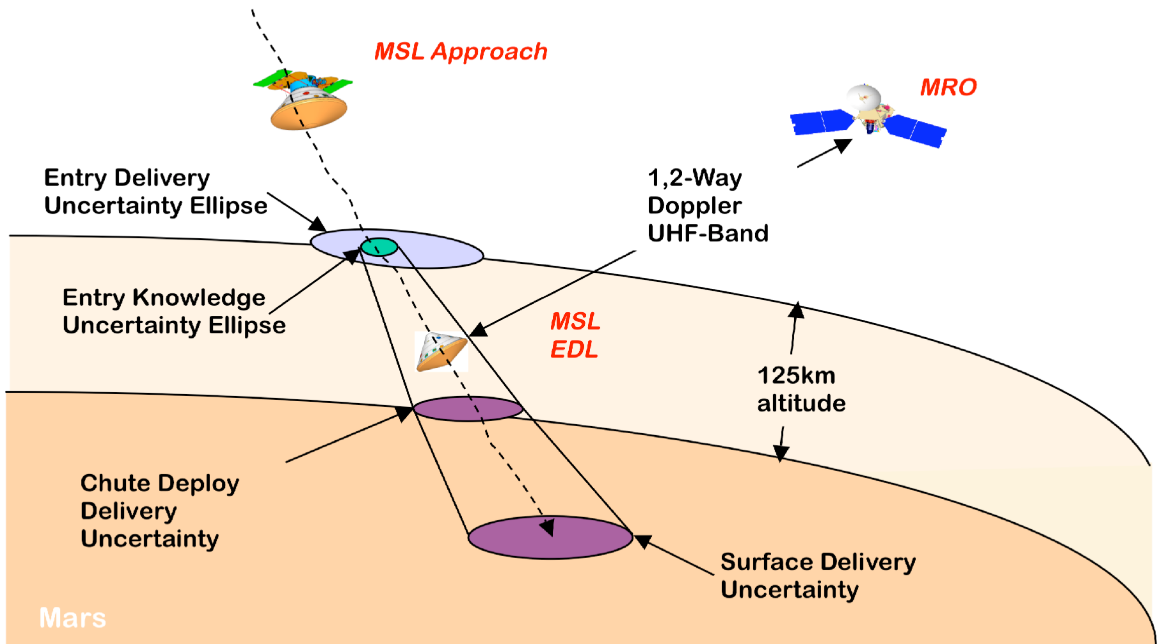

Figure 1. Schematic of spacecraft-to-spacecraft navigation as used for EDL. The entry delivery and knowledge uncertainties for the approaching lander are depicted, as is the surface positioning error. Spacecraft-to-spacecraft navigation data will be processed throughout EDL, including other external sensor data such as radar altimeter measurements.

The Electra Program at JPL has developed and produced a software defined radio (SDR) which is manifested as baseline equipment on future Mars missions beginning with Mars Reconnaissance Orbiter (MRO) scheduled for launch in August 2005. These SDRs are nominally configured for Mars-local UHF operation for data relay in the vicinity of $437MHz$ and are also capable of making high precision radiometric measurements in Doppler and range.

By design, Electra SDRs have excess computing capacity and memory sufficient for hosting an on board, real time navigation filter. The current Electra design features a space qualified Sparc V-7 processor running at $25MHz$ and several megabytes of available memory. Over half of each of these resources is available for applications such as real time navigation filtering under direction of the RTEMS (Real-Time Executive for Multiprocessor Systems) operating system. (Electra is a single processor application for RTEMS.)

Utilizing spare Electra capacity for on board navigation frees other resources, such as the main spacecraft housekeeping computer, from involvement in such a computationally intensive, time critical task. Also, the radiometric data is locally available inside Electra and need not be transferred over the spacecraft bus.

In the present work a navigation filter task has been demonstrated on prototype Electra baseband hardware with the goal of showing that Electra resources are sufficient to host a sophisticated navigation algorithm with reasonably low latency in terms of data input to updated state output times.

Topics covered here from the final year of this task include modifications to the previously selected algorithm set for implementation and performance of the implemented algorithms in a stand-alone filter, on an emulator of the target processor and finally on a breadboard processing unit.

II. Scenario Update

The approach and EDL scenarios developed in the first year of this task for the orbiting relay and the approaching lander have been modified based on program feedback. In the first year of this study, the approaching lander was assumed to be Mars Science Laboratory (MSL) and the relay orbiter used was Mars Telecommunications Orbiter (MTO). Due to the cancellation of MTO, the relay orbiter has been changed to to Mars Reconnaissance Orbiter (MRO). The trajectory for MSL was modified from that used in the first year so the trajectories for both MSL and MRO are consistent with a viable relay strategy as developed by the MSL project.[3] Although MSL will not be attempting pinpoint landing as defined above, it will have guided entry using a hypersonic guidance strategy that is a viable candidate for future pinpoint landing missions.[4] The current pinpoint landing scenario differs from the MSL baseline only after chute deploy,[5] so analysis of the MSL scenario through chute deploy applies to pinpoint landing. Although other entry guidance options are under consideration for pinpoint landing use (including hypersonic guidance approaches not derived from Apollo Earth return guidance and approaches optimized for higher L/D entry bodies), all will benefit from improved onboard knowledge.

MSL is currently scheduled to launch in 2009 with arrival at Mars in the Fall of 2010. Although the landing site for MSL has not been selected, analysis of cruise, approach and EDL has been performed by the project for various combinations of launch date, arrival date and landing site. The baseline landing site and trajectory used for this orbiting beacon navigation analysis is based on one of these cases studied by the MSL project. The entry state for the selected MSL trajectory is listed in Table 1. This entry state is the final condition of a valid Earth/Mars transfer trajectory that was used for approach navigation analysis.[6] When combined with an assumed entry body and EDL timeline, the entry state also is the initial condition for a trajectory that lands at the desired landing site, defined for this analysis as 41.45S latitude, 286.74E longitude. The details of this trajectory are not as important to the beacon navigation analysis as the fact that they represent a reasonable EDL trajectory for MSL and a reasonable trajectory for a pinpoint landing scenario from entry to parachute deploy, or to the end of the entry guidance phase.

Table 1. MSL Entry State*, Mars planet centered inertial Mars-mean-equator of epoch 26-JUL-2010 22:46:54.98

Component	Value (km)	Component	Value (km/s)
X	2160.471418	$\dot{X}$	−2.759539
Y	1334.404227	$\dot{Y}$	4.420825
Z	−2440.823929	$\dot{Z}$	2.052581

* Entry defined as a radial distance of 3522.2 km from the center of Mars.

The orbit elements assumed for MRO are shown in Table 2. These orbit elements are not ideal for telcommunication purposes, but they are consistent with the expected MRO orbit in the timeframe of MSL approach and EDL. The mean anomaly and node are parameters that were adjusted by the MSL project to achieve the required UHF relay coverage. Since there is little room to modify these parameters and still meet the relay requirements, a single relay orbiter placement was assumed for this updated scenario.[3]

Table 2. Nominal orbital elements for the relay orbiter, Mars planet centered inertial Mars-mean-equator of epoch 26-JUL-2010 22:46:54.98

Component	Value	Component	Value (deg)
Semi-Major Axis (km)	3648.6052	Longitude of Node	41.1147
Eccentricity	0.012181	Argument of Periapsis	−89.9876
Inclination (deg)	92.6428	Mean Anomaly	17.8897

III. Algorithm Selection Updates

A draft algorithm set was documented in the previous report from this task. As reported previously, the selection of algorithms for use in the embedded filter is separated into three main areas: dynamic modeling, measurement modeling and selection of the filter algorithm.[1] Updates to the draft algorithm set defined for the embedded filter will be described here.

A. Dynamic Model Updates

The first area is dynamic modeling, which includes all forces and moments acting on the spacecraft. Specifically, this includes gravity and aerodynamic forces. Data for a specific approach and EDL trajectory was used to define the truth model for this analysis. These data include time histories of the spacecraft position, velocity and attitude, along with environmental data such as atmospheric density along the trajectory and vehicle data including lift and drag coefficients. It is expected that these data will not apply in other cases, but the idea is to have a representative data set for defining the model fidelity requirements. The modeling changes described here are based on analysis of this specific data set from a single trajectory. The defined force models in the previous report are unchanged, but the fidelity has been improved by increasing the number of layers in the layered exponential atmosphere and by writing functional forms for the aerodynamic lift and drag coefficients as opposed to using constant coefficients.

A scale-height exponential density profile is used to compute the atmospheric density ρ at height relative to the surface h based on the equation

$$\rho = \rho_0 \exp\left(\frac{h0 - h}{h_s}\right)$$

where ρ_0 is the atmospheric density at surface-relative height h_0 and h_s is the scale height. A series of reference altitudes are selected with corresponding base densities and scale heights to match the trajectory-dependent density profile from a high-fidelity simulation. The previously reported density model included 10 layers, which resulted in errors relative to the density profile from the original trajectory of greater than 10%. Increasing the number of layers to 17 reduced the errors in the fit to under 5%.

For the lift and drag coefficients, analysis of the computed aerodynamic lift and drag from entry through chute deploy demonstrated large variations with velocity. For the drag coefficient, the variation with speed was under 20% but the variation with speed was well over 100% for the lift coefficient. Due to these variations, the constant aerodynamic coefficients were replaced by polynomial functions with independent variable relative speed v_{rel}. The functions used for computing lift coefficient C_L and drag coefficient C_D are

$$\begin{aligned} C_L &= C_{L_5} v_{rel}^5 + C_{L_4} v_{rel}^4 + C_{L_3} v_{rel}^3 + C_{L_2} v_{rel}^2 + C_{L_1} v_{rel} + C_{L_0} \\ C_D &= C_{D_5} v_{rel}^5 + C_{D_4} v_{rel}^4 + C_{D_3} v_{rel}^3 + C_{D_2} v_{rel}^2 + C_{D_1} v_{rel} + C_{D_0} \end{aligned}$$

With these functional forms, the errors in the lift and drag coefficients are well under 1%. Note that the simulation data used for defining these models and the independent variable does not include winds.

B. Measurement Model Updates

The second area is measurement modeling, which includes all incoming data that are to be processed with this filter. This includes inertial measurement unit (IMU) data and the Electra radio Doppler data. The IMU measurement model is unchanged from the previous report.

The Electra radiometric observable formulation has been updated and refined since the previous report. The standard for this data type has been defined and reported by the Mars Network Office, supported by the Mars Program Office of the Jet Propulsion Laboratory. This report is the basis for use of the radiometric data types for the Electra development project. The details of this data type as defined by the standard will not be repeated here but are available.[7]

Link analysis has been performed to support the measurement modeling used in the simulation analyses. These analyses demonstrate the availability of a UHF link throughout EDL based on range, geometry and antenna patterns only.[8] However, environmental effects reduce the availability of the link. Of particular interest for the EDL problem is plasma blackout caused by increased electron content around the entry body due to the extreme heating during hypersonic flight. Unfortunately plasma outages are inversely proportional to the transmission band, resulting in significant outages for UHF-band transmissions that are the focus of this analysis.

A sample analysis has been obtained and used to approximate the link outages expected during EDL. Figure 2 shows the results of that analysis for several transmitter bands including UHF.[9] This analysis was

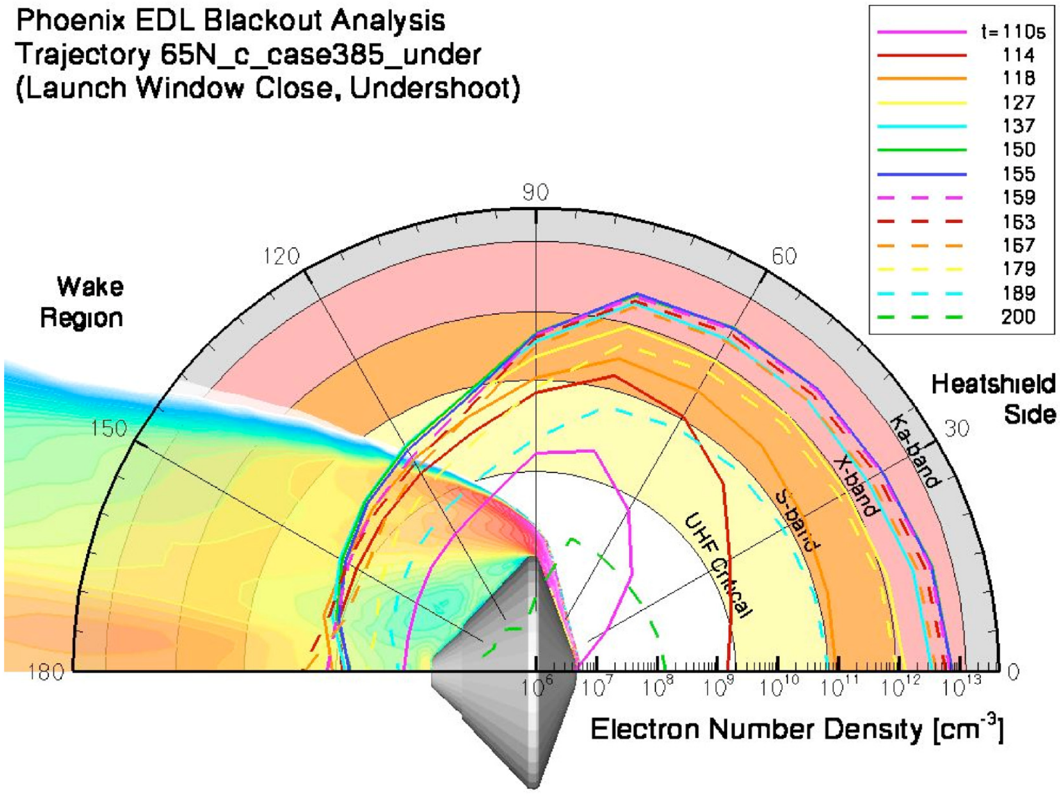

Figure 2. Electron density from plasma resulting from aerodynamic heating for an entry vehicle. For a backshell-mounted antenna and an arbitrary signal path, the outage starts before 110 seconds from entry interface and are shown to end before 200 seconds from entry

performed for the Phoenix project. The main relevant difference between Phoenix and MSL is the angle of attack of the entry body. MSL achieves lift by creating a center of mass offset relative to the center line of the entry body (as opposed to using an aerodynamic shape). For this configuration, the resulting angle of attack for a full lift-up configuration (defined as zero bank angle) results in a decrease the electron distribution above the local horizontal and an increase in the electron distribution below the local horizontal. The electron distribution is assumed to remain fixed relative to the bank angle. In other words, for a 90° positive bank, the electron distribution relative to the zero-lift case will be lower in the direction of lift and higher opposite the lift direction.

For the purposes of applying the results of the plasma study, the signal path must be specificed. For example, a signal path with an angle from aft of 60 degrees has an outage which begins approximately 114 seconds after entry interface. The electron densities drop low enough to establish a UHF link approximately 75 seconds later, or 180 seconds from entry interface. Since MSL has an antenna with a boresight aligned with the backshell, the absolute limit on the signal path is parallel to the backshell surface, or 40° in Figure 2. For a signal that can be anywhere in the span of possible signal paths, the outage has already started at 110 seconds from entry for a small set of beam angles and covers all paths after another 4 seconds have passed. A small set of beam angles are usable 57 seconds later or 167 seconds from entry. By 190 seconds, most of the range is open with all possible beam angles free of electron outage by 200 seconds from entry.

C. Filter Algorithm Updates

The third area is the selection of a filter algorithm. Two different but related filtering approaches were considered. The first is an Extended Kalman Filter (EKF),[10] considered to be the baseline for any real-time sequential estimation task with a nonlinear plant. Several factorization approaches were considered, including a simple Joseph suboptimal covariance update, a square-root implementation based on the work of Carlson,[11] and a factorized approach based on Bierman.[12] The second approach is a sigma-point filter.[13] The main advantage of the sigma-point filter over an EKF is the lack of a linearization step in computing the propagated covariance. In an EKF, a linearized transition matrix or other equation based on the first derivative of the dynamics with respect to the state vector is required to propagate the state covariance in time. This not only introduces linearization error but also requries the derivation and coding of a series of dynamic partials, which is complex and error-prone at best and may not be possible for some highly nonlinear problems. The sigma-point filter avoids this linearization by propagating a small dispersed set of states to the desired time step (the sigma points) and constructs the covariance matrix based on statistics.

Another factor in the filter algorithm selection, and the main criterion used, is the array of operational scenarios envisioned for this filter. In order to improve the overall return to the advanced EDL work area of their investment, the managers from three of the work area elements agreed to share development cost of this filter. Each element involved navigation using the Electra radio as a platform and real-time processing. The three applications for this filter being considered include UHF radio navigation on approach to Mars,[14] the use of UHF Doppler during EDL and the use of the Electra radio as a platform for adaptive estimation during EDL[15] Although each application has a different solution to optimize the usage of the limited processing resources available on the Electra hardware, a single filter that can be applied to all three problems was desired. For example, a factorized approach is desired for the approach problem but not required for the EDL problem. However, the impact of using a slightly less optimal solution for one particular problem is more than offset by the savings in effort to build and validate a single filter on the hardware as opposed to three separate filters. For the EDL navigation task, an EKF was judged the best option. The equations of motion for the Doppler-based filters are well understood by this team and are highly accurate. The impact of this decision will be discussed later.

With the selection of the EKF for the approach and EDL tasks, the question remains as to which formulation to choose. The EDL problem has the best observability due to the high dynamics but is much more constrained by processing time constraints than the approach problem. In contrast, the approach problem is much more numerically problematic, requiring factorization beyond the optimal EKF formulation and the Joseph update form. Based on the experience of the team and external resources available for the development, the Bierman factorization was selected. While providing the numerical stability required for the approach problem with its large variation in signal path distance, this approach does not impose additional computational burden above the standard EKF.[12]

IV. Covariance Analysis

Approach navigation covariance analysis for MSL was performed using the baseline navigation assumptions. This analysis includes Deep Space Network (DSN) radiometric data collected starting 45 days before entry and ending 6 hours before entry, which is the assumed data cutoff for the final onboard state update before entry. The resulting uncertainties from the approach navigation process are mapped to cruise stage separation, defined for the purposes of this study to be 10 minutes before entry body atmosphere interface[a] The covariance matrix for the spacecraft position and velocity at entry interface were supplied as inputs to the beacon navigation analysis initial conditions.[6] The use of spacecraft-to-spacecraft navigation data to improve approach navigation is being investigated separately.[14]

The assumption used for this study is that spacecraft-to-spacecraft radiometric data collection starts at cruise stage separation. Results with simulated UHF data were created, one with a diagonal *a-priori* covariance and one with a full 6×6 covariance matrix from the approach navigation analysis with DSN data. Other parameters include uncertainty in the orbiter ephemeris and spacecraft clock errors. The results for a diagonal *a-priori* covariance are significantly degraded relative to the full covariance initialization cases. If the UHF data arc length is increased to 30 minutes, the results are similar to the full covariance results. Since the focus of this effort is on UHF data collection after cruise stage separation, these results are not discussed

[a] Atmosphere interface is defined to be a radial distance from the center of Mars of $3522.2\,km$

here. The results for the full initial covariance case are shown below both for position and velocity errors as a function of time relative to atmospheric entry. The uncertainties are reported in radial-transverse-normal coordinates, defined here to be

$$\begin{aligned}\text{Radial} &= \boldsymbol{R} = \text{radially away from Mars}\\ \text{Transverse} &= \boldsymbol{T} = \boldsymbol{N} \times \boldsymbol{R} \text{ (roughly along inertial velocity)}\\ \text{Normal} &= \boldsymbol{N} = \text{normal to inertial position and velocity}\end{aligned}$$

where current-state position and velocity in inertial coordinates are used to define the axes of the frame.

The current-state residual and uncertainty plots for the position components and an RSS of all the elements is shown in Figure 3. Each time history starts shortly before the start of UHF tracking and extends for 2 minutes past entry interface. It is expected that the link will be interrupted by plasma between 1 and 2 minutes after entry, based on the EDL trajectory and telecom link analysis,[8] so the filter results are only presented to 2 minutes after entry. The most interesting aspect of the results is the rapid reduction in the uncertainties when the UHF data collection begins. The RSS error drops from nearly 800m to under 200m in one minute, with a nearly linear decrease to well under 100m by entry interface (defined as $t = 0$ in the figure) and steady error performance after entry interface. These results clearly show the benefits of collecting spacecraft-to-spacecraft UHF Doppler and processing with a fit covariance from DSN radiometric data processing.

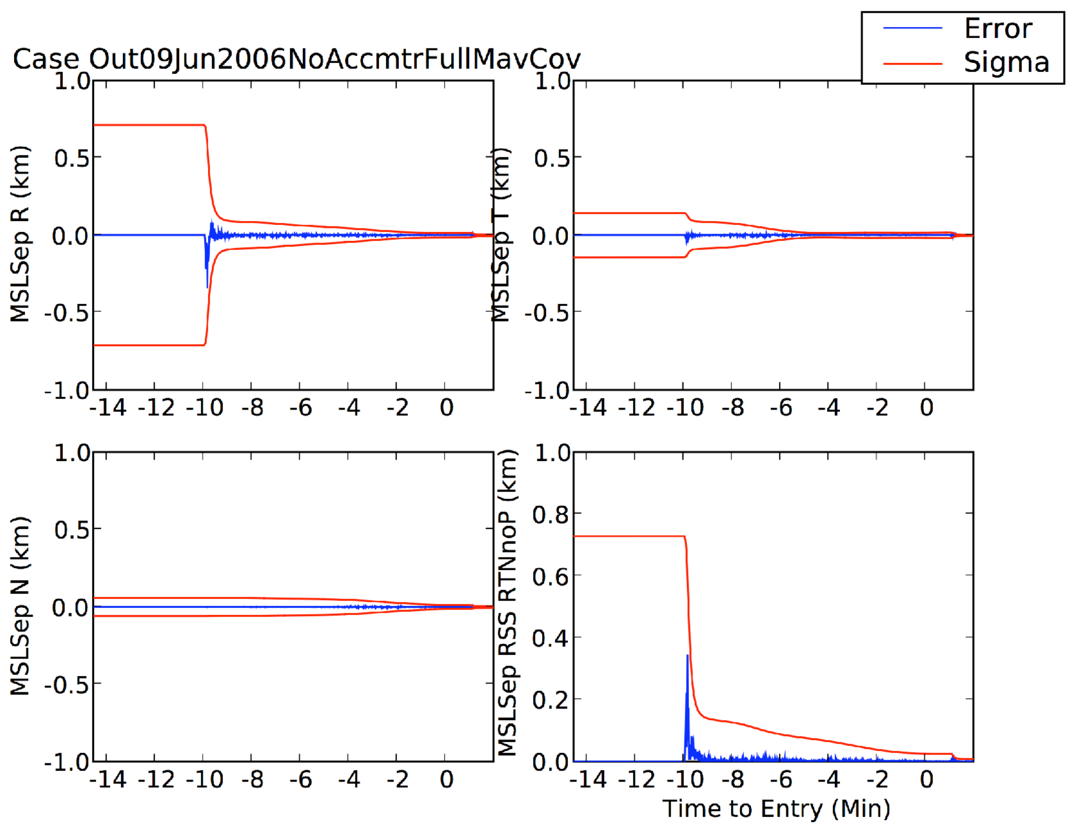

Figure 3. Position results for simulated UHF data processing with DSN tracking to 6 hours before entry, focusing on the last 15 minutes before entry. Within only a few minutes of collecting and processing UHF Doppler the uncertainties drop below 200 m (1σ) and continue to improve as more data are collected.

The current-state residual and uncertainty plots for the velocity components and an RSS of all three elements is shown in Figure 4. The velocity shows a similar trend as the position, with large improvements to the uncertainties immediately after UHF data processing begins. After less than a minute of data processing, the velocity RSS error drops nearly a factor of three, with smaller continued improvement as compared with the position error but continued improvement after entry interface.

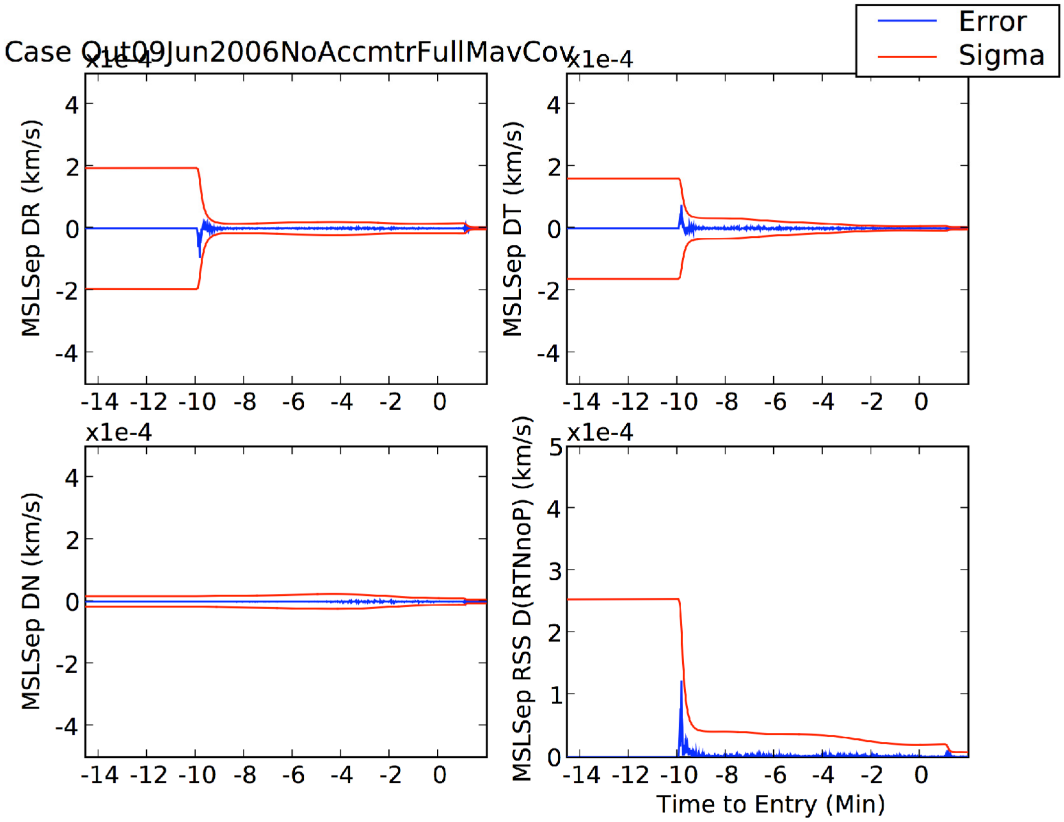

Figure 4. Velocity results for simulated UHF data processing with DSN tracking to 6 hours before entry, focusing on the last 15 minutes before entry. With only a few minutes of collecting UHF Doppler the uncertainties drop below $0.1\frac{mm}{s}$ (1σ) and continue to improve slightly as more data are collected.

V. Algorithm Implementation and Performance

The final goal of this task is to demonstrate the implemented algorithms processing data on the Electra SDU. In order to reach this goal, a three-step implementation process was utilized. The simulation results discussed earlier were precursors to the implementation process, but the results generated served as the baseline for the navigation performance.

The first step was to implement the algorithms in a stand-alone filter. While this step was free of any hardware constraints, coding practices consistent with the limitations of the final platform were enforced. These include language and processor speed limitations that require implementation of small and fast algorithms. While enforcing the above restrictions on the stand-alone resulted in some development issues on commodity hardware, the result was an implementation that was similar to the the final software but utilized a more diverse suite of optimization and debugging tools than were available for the target processor. With a standard desktop computer, the implemented algorithms execute many times faster than real time, allowing more iterations and testing.

The second step was to execute the above filter using an emulator of the Electra processor. The emulator provided a platform that was nearly the same as the final hardware target while having more diagnostic output options than the hardware. The emulator also can be run much faster than real time and gives estimates of hardware performance that are reasonable estimates of the performance on the target hardware. Use of an emulator allowed the analysts more executions of the code and further insight into the execution of the implemented algorithms than was possible with the hardware in addition to providing an alternative to using the hardware for every test. This allowed a majority of the testing to be performed in the emulator with less frequent but more productive testing with the hardware. The implementation required for the navigation filter on the emulator included not only the filter itself, but the input interface, data buffering, output interface and housekeeping functions of the Electra SDU. The goal was to be able to build an executable that runs in the emulator and required no modification to run on the hardware. Navigation performance was evaluated here and compared with the stand-alone results. In addition, the run time of the navigation algorithms was determined and any speed issues were addressed at this stage.

The final step was execution on the Electra SDU. Care was taken to minimize the amount of modification to go from step two to step three: in reality, there were minor differences between the emulator and Electra SDU that required modification of the navigation and associated software. Once these modifications were made and the filter is operational on the Electra SDU, similar navigation performance and processing speed analysis were performed.

While the first two steps above were useful for implementation of the final filter, the results of interest were performance on the Electra SDU. Intermediate results are not presented, specifically since these results must match the results of the hardware execution. The parameter of interest here is the execution speed of the algorithms in the Electra SDU relative to real time. This will be the focus of the remainder of the paper.

A. Electra Hardware

Software verification and performance testing are performed on an actual Electra Baseband Processor Module (BPM) in order to demonstrate compatibility with and suitability for the Electra computing environment. The Electra BPM is a rack-mount unit constructed from commercial parts and includes the Sparc, memory, I/O with all support logic, and the signal processing Field Programmable Gate Array (FGPA). No UHF downconverter is included, but the unit does support signal processing at the Electra 71 MHz intermediate frequency (IF), transmit and receive.

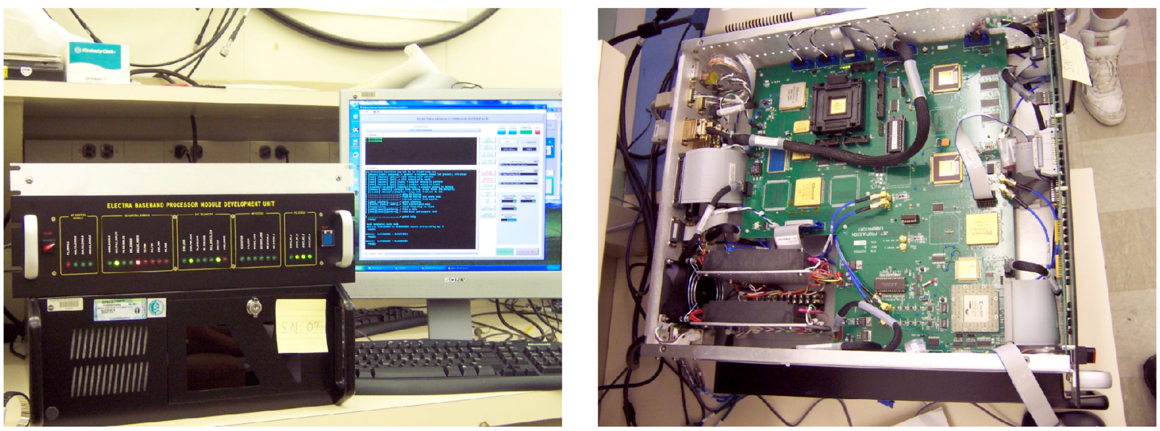

Figure 5. Electra Baseband Processor Module and control computer showing software interface (left) and the BPM showing the commercial parts used for construction (right).

The signal processing circuitry provides timing signals and interrupts to the BPM and must be supplied with a 76 MHz reference signal for the unit to operate properly. The precision and stability of the 76 MHz reference signal are important for IF testing but not for software testing.

The primary user interface to Electra is a local PC running a custom Labview package. This package is used to load and retrieve software and data, to interact with that software, and to interact with the Electra monitor program residing in Boot PROM.

The BPM Sparc V7 CPU runs at 24 MHz and is supported by a 1 Mbit Boot Prom, 16 Mbits of EEPROM, 16 MBits of SRAM and 256 Mbits of EDAC (Error Detection And Correction) SDRAM of which 170.5 Mbits (67%) is available for mass data storage. Interfaces supported include serial (RS-232), IEEE-1553, and LVDS (Low Voltage Differential Signalling) for high speed DMA (Direct Memory Access) transfers.

B. Electra Navigation Software

Electra is a software defined radio (SDR) intended for proximity UHF data relay operation in the vicinity of Mars. Signal processing is performed in FPGA while host software running on the Sparc manages operation, encodes and decodes protocols, performs radiometric measurements, handles data storage, and provides the command and data handling (C&DH) interface to the host spacecraft computer.

The Sparc software consists of several tasks that run under RTEMS (Real Time Executive for Multiprocessor Systems) in a single-processor mode. A public domain Sparc RTEMS development environment that includes a linux hosted software simulator is used for development. This product, circa 1998, uses GNU development tools (g++ version egcs-2.91.66, gdb 4.17.gnat.3.11p, etc.) and an ERC32 Gnu Cross-Compiler. RTEMS is version 4.0.0. In order to maintain both compatibility with the Electra Software Radio work and a target of a joint radio and navigation software load for an operational Electra, this package was used for the Electra Navigation demonstration work. (Electra personnel have upgraded these tools and ElectraNav plans to follow suit.)

Table 3. Comparison of Electra analysis elements that are part of the advanced EDL work area. All tasks target the same hardware, but two of the three tasks also have a common emulation and filtering approach

	Approach	EDL	Adaptive
Measurement Models	Doppler	Doppler, IMU	IMU, Doppler
Filter Strategy*	Factorized EKF using common libraries		EKF, Sigma Point Filter using common libraries
Emulation	Linux-based RTEMS/Sparc CPU with C++ s/w. MATLAB-based emulation of Electra radio		Linux-based RTEMS/Sparc CPU with C++ s/w
Electra	Commercial parts version of Electra digital baseband processing module		

* Final filter will be a common superset for use with all three problems.

As discussed earlier, a combined effort involving approach navigation, EDL navigation and adaptive estimation during EDL was arranged to share development resources for the embedded filter. Some characteristics of each work area are described in Table 3, specifically relevant areas of similarity and difference. From this comparison, the approach and EDL tasks that use UHF Doppler are the most similar, while the adaptive estimation task is arguably the most nonlinear. The equations of motion for the Doppler-based filters are well understood by this team and are highly accurate, while the adaptive task models have larger uncertainties and are more dependent on unknown environmental factors. For these reasons, the UHF Doppler tasks developed an EKF and the adaptive estimation task chose a sigma-point filter. Note that either filter will work for all areas of investigation, but the team decided on this breakdown as the most reasonable.

Navigation algorithms are computationally intensive so the purpose of this demonstration is to show that meaningful navigation results can be obtained within the Sparc V7 24 MHz computing environment, ultimately with enough spare resources to allow co-existent operation with the Electra Radio software.

The ElectraNav software resides in three RTEMS tasks utilizing two data queues, as shown in Figure 6. The data-source is responsible for all input interfaces to ElectraNav which includes receiving radiometric records from the Electra Radio software or other types of measurements from outside of Electra altogether, time-tagging and ordering them in an appropriate way, formatting them, and queueing them for the filter task on the data queue. At this point in development, the data source simply reads test data from internal memory arrays.

The filter task implements a data-driven Extended Kalman Filter. As data are received from the queue the filter state and covariance are advanced and updated with the measurement information. The features of this filter are described in more detail below. Filter output is queued for the results sink on the results queue.

The results sink is responsible for all output interfaces to the outside. In a mission operational environment this would mean building an Electra data file in SDRAM for later return to earth or providing feedback to a guidance control system. At this point in development, the results sink writes appropriate values to the console that are then captured via the Labview operator interface for evaluation.

In order to separate the RTEMS task issues from the filtering algorithm, a single-process main routine

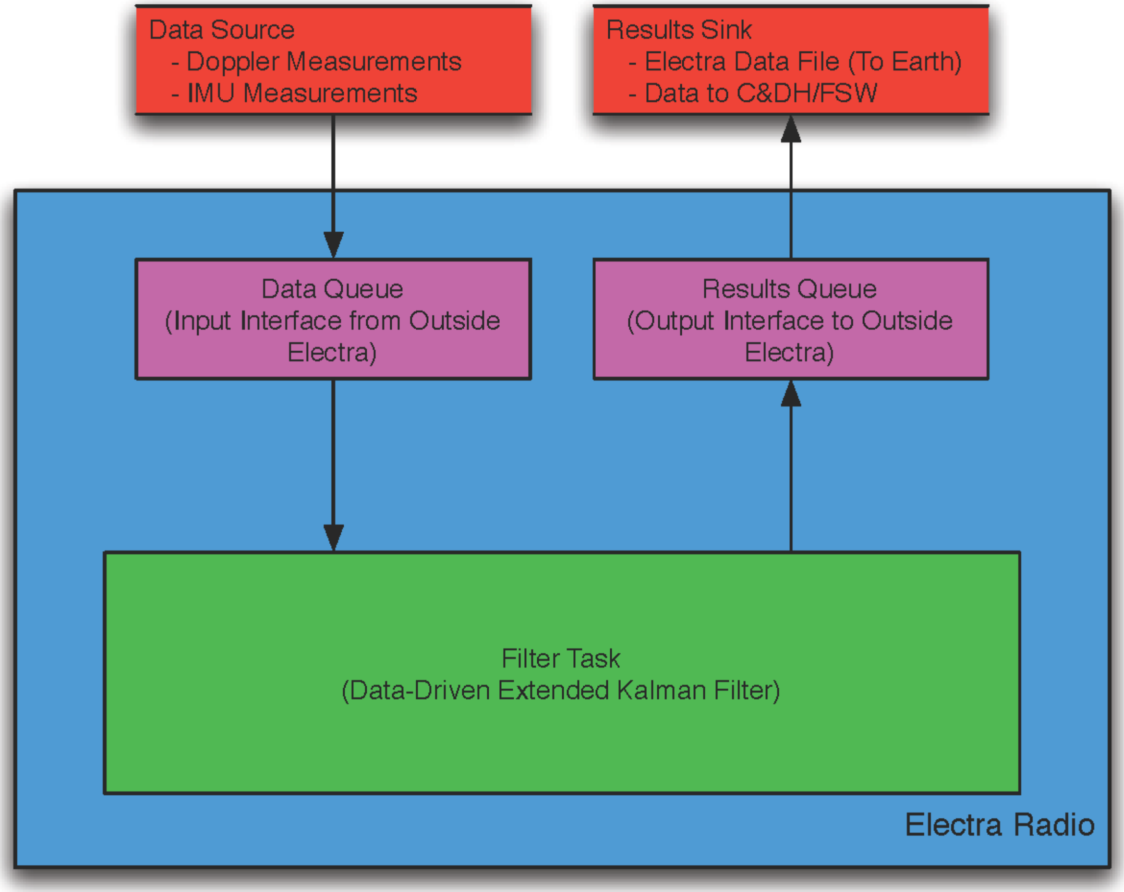

Figure 6. Electra data flow diagram. Measurement data are formatted and stored in the data queue for processing in the EKF. Results are formatted and stored in the results queue for transmission to Earth or use onboard.

was written which preserves the data transfer and ordering of the three tasks while running as the sole occupant of the Sparc resource on the Electra BPM. Reintegration with RTEMS and ultimately integration with the radio software is planned.

The code set is maintained in two forms, hardware/simulator and linux. The source is identical except for a few system specific compile time switches (such as code to exercise the hardware watchdog timer). Algorithm debugging and verification is much simpler in the linux development environment. Identical code images run on the software simulator and the hardware. Outputs are captured and a set of python tools are used to verify that the two builds give identical results on the three platforms (linux, software simulator, and hardware), to within machine precision.

An overall make-based system manages and builds the libraries and executables for these targets. The software modules are:

- Apps The navigation filtering application, such as the Furies Filter, described below.
- Framework The three task - two queue structure in the form of C++ base classes.
- Library Calculation algorithms from Monte and Ipanema (JPL navigation software).
- Platform Main or RTEMS based execution.
- Test Output data and python utilities for evaluation.
- Utility Functions to work around lack of C++ templates and exception handling.

A typical development cycle involves getting code to work properly as a linux build, then recompiling it for Sparc and running it on the linux-hosted software simulator to verify that it will run similarly (or at all) in the hardware environment, then transferring it to the hardware via the Labview operator interface and running it there for characterization, performance metrics, and validation.

The Furies Filter, Version 3.0, is the current navigation filter application under test in this environment. It has simulated inputs in the form of Electra Integrated Doppler and Inertial Measurement Unit (IMU) measurements. Integrated Doppler is a two-way measurement between an Electra on a Mars Approach Vehicle (MAV) and an Electra on a Mars Orbiting Vehicle (MOV). The approach vehicle transmits a carrier that is received and coherently returned by the MOV Electra in a transponding mode. The return signal received back at the MAV Electra is then processed in the filter to improve knowledge of position and velocity (state). IMU measurements are collected from the MAV IMU and include integrated sensed acceleration and integrated angular rate over a defined count interval. The Electra filter can process these data as they arrive or an intelligent accumulation can be performed (correcting the accelerometer output based on the computed vehicle attitude) to reduce the number of data processing steps. The rates under investigation vary from $10\,Hz$ to $0.1\,Hz$ or 10 measurements per second to one measurement every 10 seconds. The $0.1\,Hz$ rate is consistent with the expected Doppler rate.

The nominal state is propagated by a Runge-Kutta fourth order integrator with adjustable step size (RK4-5). Integration is evaluated using the adjustable step size, then the step size is fixed at a reasonable value for the test regime to save integration time. Both MAV and MOV are integrated in the modeling process. During integration, partial derivatives are also computed using the chain rule in parallel with the calculations used to model physical reality. This makes partial derivatives with appropriate time references available for filter time and measurement updates. The integrator is to be extended with a splining algorithm to save processor time for interpolated state and partial derivative values.

Filter covariance is processed in UDU factorized form for greater numerical stability. While the use of a factorized filter is not required for the EDL navigation problem, the computational impact of using a factorized filter only appears at initialization and for output, where conversion to standard metric units from a factorized form is required. All filter processing is done with the factorized matrices.[12] Integrated Doppler measurements are corrected for both vehicles travel during the light-time propagation. The Mars GM (gravity-mass) constant, planetary rotation, and an exponential atmosphere are included in the models currently under test. These models are all extensible. For example, the atmosphere model may be increased to multiple exponential layers, the gravity model may be increased in degree, and relativity may be added to the light-time and other models.

A high fidelity model of MAV approach is used at truth for algorithmic validation and test comparison and for the generation of test observables. As work progresses, the intent is to refine models in ElectraNav to give acceptable agreement with this truth without exceeding resource limits, particularly CPU usage.

The implemented navigation filter has slightly over $14,000$ lines of code with an image size of $1.328\,MB$, both well within the capacity of the hardware. The performance for a twelve-state model (position and velocity for MAV and MOV) with Integrated Doppler Measurements every ten seconds and IMU measurements at various rates is shown in Table 4.

Table 4. Electra navigation filter performance for varying IMU data rates with a 12-state model processing 10-second Doppler measurements for 75 seconds and varying IMU data rates.

Case*	SW Emulator	Hardware	% Real Time	Notes
1 IMU	29 sec	70 sec	93	No IMU data
70 IMU	32 sec	80 sec	107	One IMU point per second
790 IMU	67 sec	170 sec	227	10 IMU points per second

* All tests were for a $75s$ integration with 8 Doppler measurements.

VI. Future Work

The task objectives of demonstrating a navigation filter that can run in the Electra radio has been completed. The next steps, to be carried out in a follow-on task to this effort, include additional optimization and some additional model improvements, most notably adding clock parameters to the state and estimate lists. While thought was given to performance in the current implementation, the focus to date has been computational correctness. This balance will shift more to meeting the computational performance targets as the code matures. For example, direct integration of the orbiter trajectory and partials can be replaced by Keplerian elements, which saves processor time and retains precision.

The expectation is that, if the navigation software of suitable algorithmic fidelity and stability can operate in the single process mode in 30-50% of real time, it will be able to coexist with the radio software under RTEMS in real time. This level of performance appears to be within reach.

VII. Acknowledgments

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

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

References

[1] Burkhart, P. D., Ely, T., Duncan, C., Lightsey, E., Campbell, T., and Mogensen, A., "Expected EDL Navigation Performance with Spacecraft to Spacecraft Radiometric Data," *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA-2005-5949, San Francisco, California, Aug. 15-18, 2005.

[2] Crain, T. P. and Bishop, R. H., "Mars Entry Navigation: Atmospheric Interface through Parachute Deploy," *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA-2002-4501, Monterey, California, Aug. 5-8, 2002.

[3] Bowes, A., Burkhart, P. D., Kerridge, S., Neelon, J., Strauss, W., Watkins, M., Whetsel, C., and Wolf, A., "Ramifications of Communication During the Entry, Descent, and Landing Phase of the 2009 Mars Science Laboratory Mission," *Proceedings of the AAS/AIAA Astrodynamics Specialists Conference*, AAS 05-268, Lake Tahoe, California, Aug. 7-11, 2005.

[4] Mendeck, G. F. and Karman, G. L., "Guidance Design for Mars Smart Landers Using The Entry Terminal Point Controller," *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA-2002-4502, Monterey, California, Aug. 5-8, 2002.

[5] Wolf, A., Tooley, J., Ploen, S., Gromov, K., Ivanov, M., and Acikmese, B., "Performance Trades for Mars Pinpoint Landing," *Proceedings of the IEEE Aerospace Conference*, Paper 1661, Big Sky, Montana, March 4-11, 2006.

[6] Burkhart, P. D., "Approach Navigation for the 2009 Mars Large Lander," *Advances in the Astronautical Sciences, Volume 114*, Univelt, San Diego, CA, 2003, pp. 2183–2198, also AAS Paper 03-250, Jan. 2003.

[7] Noreen, G. K., "Mars Network User's Guide Section 203: Navigation Services," http://mrn.jpl.nasa.gov/handbook.html, 2006.

[8] Lightsey, E., Campbell, T., Mogensen, A., Burkhart, P. D., Ely, T., and Duncan, C., "Expected Performance of the Electra Transceiver for Mars Missions," *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA-2005-5948, San Francisco, California, Aug. 15-18, 2005.

[9] Kallemeyn, P. H., private communication, Lockheed Martin Space Systems, Littleton, CO, Sept. 2005.

[10] Gelb, A., *Applied Optimal Estimation*, The MIT Press, 1992.

[11] Carlson, N. A., "Federated Square Root Filter for Decentralized Parallel Processes," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 26, No. 3, 1990, pp. 517–525.

[12] Bierman, G. J., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, 1977.

[13] Julier, S. J. and Uhlmann, J. K., "A New Extension of the Kalman Filter to Nonlinear Systems," *Proceedings of SPIE - The International Society for Optical Engineering*, Vol. 3068, 1997, pp. 182–193, April 21-14, 1997, Orlando, FL.

[14] Ely, T. A. and Guinn, J. R., "Mars Approach Navigation using Mars Network based Doppler Tracking," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, AIAA-2002-4816, Monterey, California, Aug. 5-8, 2002.

[15] Heyne, M. C. and Bishop, R. H., "Spacecraft Entry Navigation using Sigma Point Kalman Filtering," *Proceedings of the IEEE/ION Position, Location and Navigation Symposium*, IEEE, April 25-27, 2006, pp. 71–79.