Real Time Mars Approach Navigation aided by the Mars Network

Todd A. Ely[*] and Courtney Duncan[†]
Jet Propulsion Laboratory, Pasadena, CA, 91109-8099

E. Glenn Lightsey[‡] and Andreas Mogensen[§]
University of Texas at Austin, Austin, TX, 78712

A NASA Mars technology project is described that is building a prototype embedded real time Mars approach navigation capability which can be hosted on the Mars Network's Electra transceiver. The paper motivates the reason for doing real time Mars approach navigation via a set of analyses demonstrating its utility for enabling Mars pin-point landing (< 1-km landing error). The development approach, software design, and test results are discussed. Finally, the way forward towards a flight demonstration on the Mars Science Laboratory is presented

I.   Introduction

Achieving key scientific goals of the Mars Exploration Program, including the search for water and life, will require placing landers at locations of the greatest scientific interest.[1] The capability to land within 1 km of a pre-determined site will enable landing and roving to this site while avoiding potential hazards that might lie near its vicinity.

It stands to reason that in order for a guidance system to succeed with a pinpoint landing that precise trajectory knowledge will be required. In particular, this is true during the mission's final phases when the vehicle is actively guiding itself – which include the final approach phase and entry/descent/landing (EDL) phase. For the purposes of this technology effort, *final approach* is defined to be the period from ½ day out to just before entry at the top of the atmosphere. An illustration showing these mission phases is exhibited in Figure 1. Prior to final approach is the initial approach phase – here accurate trajectory knowledge is useful for minimizing Mars targeting errors, but is mostly an Earth ground based activity because there is sufficient time to relay telemetry and uplink commands. However, it is the final, and most critical, mission phases that precise trajectory information provided to an on-board guidance system can be most useful for enabling pinpoint landing.

These final mission phases are characterized as brief, and, because of light time delays, proceeding without ground-based Earth support. The implication is that an approach vehicle's trajectory knowledge needs to be obtained *in-situ* and processed *on-board*. Table 1 shows the performance of several navigation and guidance strategies for Mars landing, including the current baseline tracking strategy that uses only Earth based radiometric data (Row 1), and an approach using Mars Network (MN) based spacecraft-to-spacecraft radiometric data (Rows 2 & 3).[2] First, a few notes about the columns:

1. Entry *knowledge* uncertainty represents the trajectory uncertainty at the top of the atmosphere given the proposed tracking strategy in stated each row.
2. Entry *delivery* uncertainty represents the trajectory uncertainty at the top of the atmosphere when the knowledge (up to a certain data cutoff time) is used with guidance.
3. *Ballistic* surface delivery represents an unguided entry, descent, landing (such as with the Mars Exploration Rovers)
4. *Hypersonic guidance* represents guidance in the upper atmosphere.

---

[*] Senior Engineer, Guidance, Navigation, and Control Section, 4800 Oak Grove Drive/MS 301-125L, Senior Member.
[†] Senior Engineer, Guidance, Navigation, and Control Section, 4800 Oak Grove, Drive
[‡] Associate Professor, Aerospace Engineering and Engineering Mechanics, Room 412C, Mail Code: C0600, Senior Member.
[§] Graduate Student, Aerospace Engineering and Engineering Mechanics.
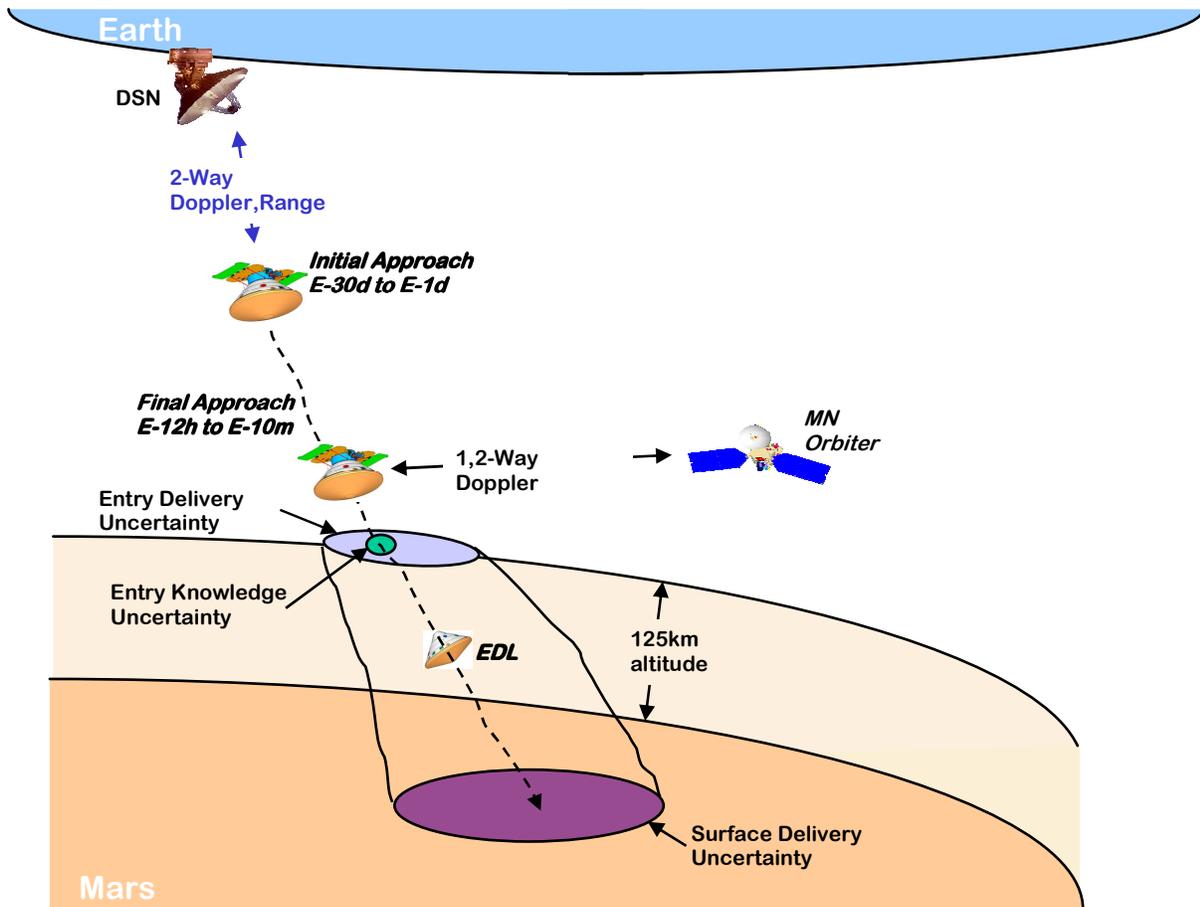
American Institute of Aeronautics and Astronautics

Figure 1: Illustration of a Mars lander during initial approach, final approach (the subject of this task), and entry, descent, and landing (EDL).

5. *Hypersonic entry + chute guidance* adds guidance while on the parachute.

The state of the art landing system are the Mars Exploration Rovers (MER) (shown in Table 1 as the three boxes outlined with the wavy border), which yields final delivery errors to the top of the Mars atmosphere of 9 km. Since MER's entry is ballistic these errors grow to 80 km by the time it reaches the surface. Consider that even with active guidance during entry (as with the Mars Science Laboratory), the surface delivery errors (~ 10 km) do not decrease to less than the entry errors without further tracking data. Indeed, it is an accurate statement to say, that in order to even consider pinpoint landing accuracies of less than 1 km requires that an approaching/entering vehicle's guidance system have real time trajectory knowledge updates at this same level of accuracy during final approach. Pinpoint landing that is aided with Mars Network navigation during both final approach and EDL and integrated with active guidance is represented in Table 1 as the last row (#3). This case illustrates that final approach navigation is enabling for pinpoint landing, without it the best that a lander could hope to achieve is the MSL baseline result of a 10 km surface delivery error.

In this technology task we built a preliminary autonomous final approach navigation system capable of on-board processing radiometric tracking data between a MN orbiter and an approach vehicle to achieve 300 m or better atmosphere entry knowledge error (as highlighted in gray in row 2). The resulting technology is enabling for pinpoint landing (i.e., row 3). Ultimately, this navigation technology should be integrated with a Mars approach vehicle's onboard guidance system for complete closed-loop guidance and navigation (GN). Doing so will achieve a 300m or better atmosphere delivery error. However, in the interest of keeping the size of the task modest, this effort addresses only the navigation portion of a complete GN system.

The Mars Network is ideally situated to provide autonomous navigation support using a version of Electra (the MN's next generation software UHF transceiver) that has been programmed to do so during a mission's final approach and terminal phases.[3,4] A key service of the Mars Network is to provide communications using Electra during mission critical events. Indeed future relay orbiters that will make up the MN, such as the Mars

American Institute of Aeronautics and Astronautics

Table 1: Atmosphere entry and surface delivery errors of a lander using DSN only or DSN + Mars Network radiometric tracking for various guidance strategies. Wavy line = state of the art, Gray box = this technology effort

Radio Navigation Capability	3σ Entry Uncertainty (km)		3σ Surface Delivery Uncertainty (km)			Comments
	Knowledge	Delivery	Ballistic (MER)	Hypersonic Guided Entry (MSL)	Hypersonic + Chute Guided Entry	
1) Ground Based X-Band DSN Radio Nav. (Doppler, Range, ΔDOR), E-18 hr data cutoff, E-6 hr maneuver, trajectory update at E-4 hrs	1.5 x 1.5	9 x 1.5	80 x 12	10 x 5	10 x 5	Baseline tracking for MER and MSL. Chute guidance of no value without additional tracking
2) 1 + S/C to S/C UHF-Band Doppler using the MN, autonomous processing begins at E-10 hrs, maneuver at E -1 hr	0.3 x 0.3	0.3 x 0.3	38 x 5	3 x 3	3 x 3	Improved entry knowledge improves MER and MSL case.
3) 2 + additional UHF data through EDL	0.3 x 0.3	0.3 x 0.3	38 x 5	3 x 3	0.5 x 0.5	Improved entry knowledge with EDL beacon nav enabling for pinpoint landing.

Reconnaissance Orbiter (MRO) or another combined science/relay orbiter being considered for launch in the 2011/2013 timeframe, will have budgeted maneuvering capability to ensure coverage for a Mars mission during its critical event.[5] By design, Electra is also capable of collecting Doppler data concurrent with data transmission while the link is active. Furthermore, Electra has been designed with spare processing and memory capabilities that can be utilized for higher level processing.[6,7] Electra has a Sparc V7 RISC based processor with a clock speed of 24MHz and about 256 MBits of storage. It is estimated that about 2/3 of this processing and memory is available for use. Given a baseline scenario, where radiometric data between a MN orbiter and a user vehicle (also carrying a version of Electra) are available, this paper describes a technology task currently building prototype real-time embedded Mars approach navigation software that is being built for the Electra transceiver.

II.   Project Overview and Results

Our end goal is to research and develop algorithms and prototype software to be hosted on Electra that can process Electra based radiometric data and determine trajectories during final approach in real-time. Operationally, we expect the navigation to be an autonomous process that can be monitored at Earth in the first few hours after the final trajectory correction maneuver, however, the end phase of EDL necessarily occurs in a fully autonomous mode. Initial monitoring will be used as a checkout period by the Earth based ground team to validate the on-board processing. We developed an initial capability that is readily positioned for development and demonstration of this navigation technology on the Mars Science Laboratory (MSL) which will have a version of Electra on-board. Some key challenges that have presented themselves include:

1. Determining navigation algorithms that yield sufficiently precise solutions; yet are robust and efficient.
2. Characterization of the performance of these algorithms with realistic/detailed scenarios.
3. Characterization of the Electra sensor acquisition and tracking performance in the presence of weak signals.
4. Embedding these 'right-sized' algorithms in an emulated Electra and testing in a realistic simulated environment, and then hosting the software on a version of Electra made with commercial parts.
5. Integrating autonomous real-time approach navigation with its counterpart that is active during the EDL mission phase, integrating the navigation strategy with other sensor data (i.e., accelerometers).

Our plan to meet these challenges has been based on an analysis and simulation approach with ever increasing levels of fidelity, and a development approach that yields prototype autonomous approach navigation software that is eminently ready for hosting on a flight version of Electra. We have completed a majority of the above tasks raising the technology's TRL from 3 to, almost, 5 (integration with Electra's real time operating system remains). The final product is a *unique* application of Mars Network services that can provide an autonomous final approach navigation capability at Mars. This task was a two-year effort culminating in an initialapproach navigation system that can be hosted on Electra and will be proposed for development and demonstration on MSL.

A. Accomplishments
In the first year ourobjective was on conceptual development and simulation. We achieved a number of specific objectives including:
1. Development of a representative detailed Mars approach scenario using a nominal MSL approach trajectory and Mars Network tracking from the, now defunct, Mars Telecom Orbiter (MTO). On the cancellation of MTO we transitioned our scenario to using MRO.
2. Development of a precise model for Electra's 1-Way and 2-Way integrated Doppler radiometric data, and navigation algorithms.
3. Characterized MSL and MTO/MRO scenario link dynamics and strength using realistic assumptions about the relative dynamics between the spacecraft, antenna gain patterns, and other link budget factors affecting the signal strength.
4. Development of an Electra processor software/hardware emulation capability that allows us to develop and test software performance in a realistic environment that emulates the real time performance of the Electra processor and memory.
5. Acquired an Electra baseband processing module (BPM) engineering development unit (EDU) that consists of the digital portion of the Electra transceiver containing all of the digital signal processing and uses commercial grade (non-space qualified) parts.

In the second year our objective was to develop, host and test the approach navigation software in a realistic hardware environment. Specifically we:
1. Performed trades to optimize navigation software for hosting on an Electra processor, including selection of necessary propagation algorithms, a sufficient modeling set to exercise the software, and a robust, factorized sequential filter.
2. In tandem with the above trades, we hosted the software on both the emulated Electra processor, and on the hardware the Electra EDU.
3. Unit tested the prototype software in emulation and on the Electra EDU for self-consistency; comparison to the high fidelity simulation and processing with real data is an activity for the next fiscal year.

A follow on project has been funded for next year to collect orbiter to orbiter data at Mars and process it in the Electra EDU. This will validate the software using actual flight data, hence significantly retiring the risk associated with collection and processing of in-situ Mars radiometric data. The follow-on project will be discussed in more detail later.

B. High Fidelity Simulation Results with MRO
The first step in this technology development effort was to validate the claim that Mars approach navigation could be significantly improved with use Mars Network derived Doppler data. To this end we developed a representative detailed Mars approach scenario using a nominal MSL approach trajectory and Mars Network tracking from the, now defunct, MTO. On the cancellation of MTO we transitioned our scenario to using MRO. Analysis has shown that MRO can acquire the signal continuously at ~100,000 km or about 10 hrs prior to entry.[8] It should be noted that these acquisition distances are based on signal powers that are slightly above the demonstrated acquisition performance of Electra in the lab with signal power near -150 dBm. The scenario work provides a basis for simulating expected performance of the navigation algorithms and providing a truth model for testing the performance of the algorithms on the target hardware. Some example results of this simulation capability are shown in Figure 2. The two plots depict the 3-sigma semi-major axis of the current state position uncertainty for a simulated MSL approach trajectory in the final day prior to entry (left plot), and from separation to landing (right plot). The plots depict 3 tracking scenarios:

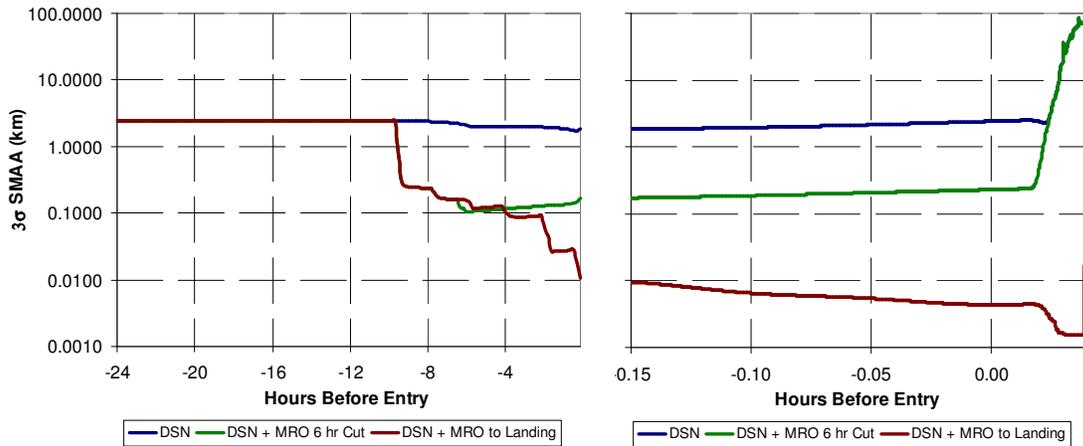American Institute of Aeronautics and Astronautics

Figure 2: Illustration of trajectory knowledge produced by MN orbiter tracking a Mars approach vehicle showing the (left pane) final day prior to entry, and (right pane) separation to landing.

1. DSN with a blue line: 2-Way Doppler, Range, and Delta-DOR from DSN stations located at Goldstone, Madrid, and Canberra. This level of DSN support provides 24 hour coverage. The data cut off is at 6 hours prior to entry which is nominal for MSL. For this case, after data cutoff the trajectory error simply maps forward in time until landing. The entry trajectory is simulated as a ballistic entry (note that MSL is baselining a hypersonic guided flight phase). Because the entry is ballistic the landing errors are more representative of a MER-class EDL system – in this case a 75 km (3-sigma) semi-major axis error at landing. The error growth during EDL is due almost entirely to the stochastic drag, which has been modeled at a level of 35% uncertainty (1-sigma) applied every 5 seconds. The other significant result is the error at the atmospheric interface (i.e., at 125 km altitude) which is ~ 2.4 km (3-sigma) differs slightly from the anticipated MSL delivery error of ~ 1.5 km (3-sigma). The difference between the result in Figure 2and the MSL estimate occurs because our simulation uses only 30 days of tracking data, whereas MSL simulations routinely use 45 day arcs. Before moving on to the next result, it should be noted that MSL's hypersonic guidance will fly to a nominal drag profile using its dead-reckoned IMU data to maintain trajectory knowledge. MSL analysis has suggested that the IMUs can maintain trajectory knowledge at the level of the entry error with a small growth to the order of a few hundred meters.[2] Indeed, MSL's expected 3-sigma knowledge error at parachute deploy is expected to be around 2.1 km. Hence, even though Figure 2 illustrates an error growth, it is expected that MSL would maintain trajectory knowledge in practice using its accelerometers with error growth on the order of hundreds of meters.

2. DSN + MRO with a 6 hr data cut off: This sceanrio builds on Scenario (1) except now there is 2-Way Doppler data being collected between MSL and MRO. The data starts at about 10 hours prior to entry, and is cut off at 6 hours prior to entry. In the event that MRO has to point its nadir deck to get the data it can do so for 30 minutes at a time, before batteries get low and it needs to re-orient to collect solar energy. This 4 hour period results in 3 passes of data. This case represents a scenario where the Mars Network data could be telemetered back to Earth and processed with the DSN data to provide a knowledge update to the guidance system prior to atmospheric entry. Even this small amount of tracking indicates the value of the Mars Network based Doppler tracking data. Indeed, the uncertainty at the top of the atmosphere has decreased to less than 230 m (3-sigma). This result displays the utility of using MN based tracking data at improving Mars-relative trajectory knowledge even if it is not processed on-board and in real time.

3. DSN+MRO tracking to landing: This case differs from Scenario (2) in that tracking persists past the 6 hour period and is continuous in the last thirty minutes including the EDL phase. This results in 6 tracking passes between MRO and MSL. In this case it is assumed at some point the in-situ data would be processed on-board (in the present case by Electra) and in real time to provide navigation updates. The results vividly display the benefit of the in-situ Doppler even during EDL. At atmospheric interface the solution is at better than 10 m (3-sigma) and only grows slightly during atmospheric flight to about 15 m (3-sigma) at landing. It should be emphasized that key error sources not included in this result is the map tie error relating knowledge of the inertial frame to the Mars body fixed frame. Current estimates place the
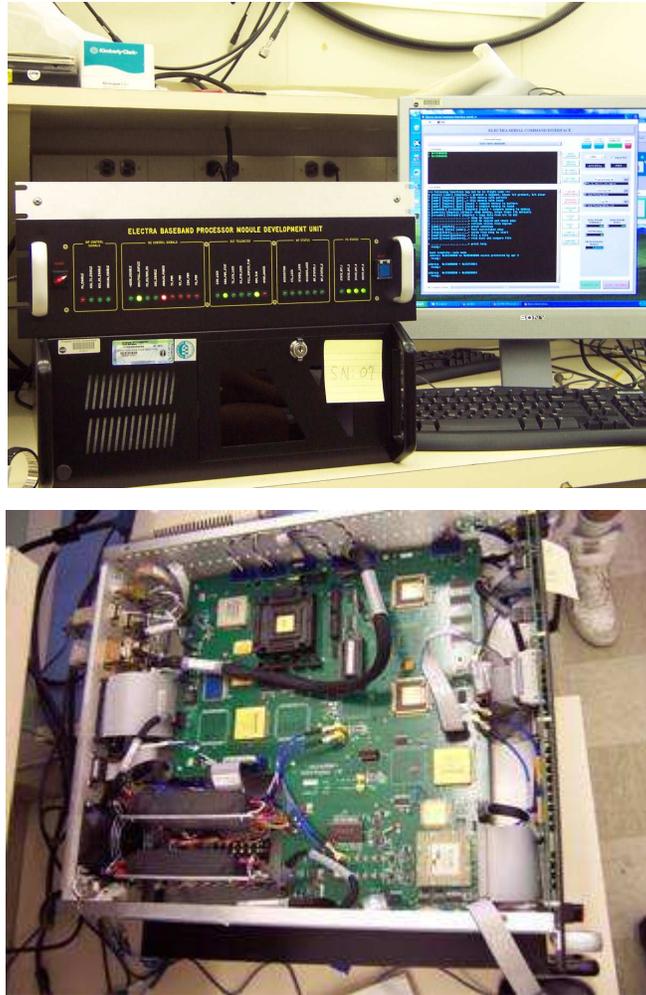
American Institute of Aeronautics and Astronautics

Figure 3: Electra EDU in the lab (upper) and exposed breadboard (below)

map tie error at around 450 m 3-sigma.[9] Hence, the landed error in this example related to the Mars surface map is at this level. Additionally, dynamic flight with a lifting trajectory and parachutes are not included in the simulation – these will affect the performance of navigation with the in-situ Doppler data.

The preceding simulation results provide a basis for comparison with the real-time navigation software that we developed for the Electra EDU. Ultimately, the high fidelity simulation will generate the truth data that will drive the Electra prototype flight software.

C. Electra Hardware

Software verification and performance testing are performed on an actual Electra BPM EDU in order to demonstrate compatibility with and suitability for the Electra computing environment. The hardware configuration is illustrated in Figure 3. The Electra BPM EDU is a rack-mount unit constructed from commercial parts and includes a Sparc V7 central processor, memory, I/O with all support logic, and the signal processing field programmable gate array (FGPA). The analog UHF down converter is not included, but the unit does support transmit and receive signal processing at the Electra 71 MHz intermediate frequency (IF).

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.

American Institute of Aeronautics and Astronautics

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 Error Detection And Correction (EDAC) SDRAM of which 170.5 Mbits (67%) is available for mass data storage. Interfaces supported include serial (RS-232), IEEE-1553, and Low Voltage Differential Signaling (LVDS) for high speed Direct Memory Access (DMA) transfers.

D. Electra Navigation Software

Electra is a software defined radio intended for proximity UHF data relay operation in the vicinity of Mars. Signal processing is performed in the 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 the Real Time Executive for Multiprocessor Systems (RTEMS) 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 the 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 compatibility with the flight version of the Electra radio software, with an eventual target of a joint radio and navigation software load to an operational Electra, the Electra navigation (ElectraNav) software package has been developed RTEMS environment and in Linux. However, because of limitations with version 2.91.66 of the g++ compiler, the navigation software currently only builds and runs with Linux hosted on Electra. Electra personnel have upgraded RTEMS to support a more recent g++ compiler; we intend to rebuild the ElectraNav software in the new RTEMS environment next year.

Navigation algorithms are computationally intensive so the purpose of this demonstration is to show 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 (data-source, filter-task, and result-sink) utilizing two data queues (data-queue and results-queue). This is illustrated in the notional flow below:

$$data\text{-}source \rightarrow \text{data-queue} \rightarrow filter\text{-}task \rightarrow \text{results-queue} \rightarrow result\text{-}sink$$

These tasks conduct the perform the following functions:
1. 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, time-tagging and ordering them in an appropriate way, formatting them, and queuing 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.
2. The *filter-task* implements a data-driven Extended Kalman Filter (EKF) that is in a UD-factorized form for numerical stability (named the Furies 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 result-sink on the result-queue.
3. The *result-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 result-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 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:

1. Apps – The navigation filtering application, such as the Furies Filter, described below.
2. Framework – The three task and two queue structures in the form of C++ base classes.
3. Library – Calculation algorithms from Monte and IPANEMA (JPL navigation software).
4. Platform – Main or RTEMS based execution.
5. Test – Output data and python utilities for evaluation.
6. 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. Nominally, the integrated Doppler is collected by the Mars approach vehicle's Electra radio and the IMU data is measured and transferred to Electra via the interface to the MAV's CD&H. This is the notional concept, for this technology demo these interfaces do not exist yet. 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, and then the step size is fixed at a reasonable value for the test regime to save integration time. Both the MAV and the Mars Network orbiter states are integrated using appropriate acceleration models. Currently these include the Mars GM (gravity-mass) constant and an exponential atmosphere. These 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. The use of an analytic ephemeris based on osculating Keplerian elements is being investigated for the calculation of MNO trajectory and partials in order to save processor time. During integration, partial derivatives are also computed. The integrator is to be extended with a splining algorithm to save processor time for interpolated state and partial derivative values. Integrated Doppler measurements are corrected for both vehicles' travel during the light-time propagation. The Furies Filter covariance is processed in UDU factorized form for greater numerical stability. Currently the Furies Filter includes 14,381 lines of C++ code, and builds to an image size of 1.328 Mbytes. In addition to integration model fidelity upgrades, numerical enhancements with analytic an MN Orbiter (MNO) trajectory, and splinning we will be adding clock parameters to the filter state and stochastic process modeling.

A high fidelity model of the Mars Approach Vehicle (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 following performance, shown is for a twelve-state model (position and velocity for MAV and MNO) with integrated Doppler Measurements every ten seconds and IMU measurements at various rates.

Table 2: Performance Comparison of the Furies Filters

Case	Electra SW Emulator	Electra BPM EDU	% Real Time
75 sec integration, 8 Doppler, 1 IMU	29 sec	70 sec	93
75 sec integration, 8 Doppler, 70 IMU	32 sec	80 sec	107
75 sec integration, 8 Doppler, 790 IMU	67 sec	170 sec	227

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. Given the performance enhancements we have identified we expect this level of performance to be within reach.

American Institute of Aeronautics and Astronautics

E. The Way Forward

Our work next year will be to update the software to include the enhancements that have been indicated including the reintegration back with RTEMS, and verify the software against the simulated data sets. More importantly, there exists the possibility of collecting Odyssey to MRO 1-Way Doppler data using the Electra on MRO. We will advocate for getting this data and processing it in the Furies Filter. This will be a significant step towards validating the processing and modeling with a real data set. This is an important step in the path towards developing a flight demonstration for MSL.

III.   Conclusion

Existing missions such as MSL are being designed with landing errors that are consistent with existing technologies. For instance, MSL is designing for a 10 km (3-sigma) error on landing. This is characteristic of what has been termed a "Generation 2" lander. However, to achieve "Generation 3" pinpoint landing accuracies of 10's of meters to 1 km requires advances in navigation and guidance technology. The technology proposed here is relevant to any Generation 3 type landings, or any Mars mission requiring precision approach trajectory information. Demonstration of precision approach using the Mars Network will pave the way to achieving sub-1 km pinpoint landing accuracies. Without improving final approach trajectory knowledge (using whatever means) it simply will not be possible to achieve this objective. That is, a guidance system can not correct for trajectory errors that it has no "knowledge" of. The Mars Network brings a capable and available sensor (namely Electra) to help solve this problem; this technology effort will use this critical resource in a novel and efficient way.

IV.   Acknowledgements

References

[1]*Solar System Exploration*, NASA Office of Space Science, JPL 400-1077, May 2003.

[2]Mars Program Pinpoint Landing System Study, 2003.

[3]Bell, D. J., Cesarone, R., Ely, T. A., Edwards, C., Townes, S. "Mars Network: A Mars Orbiting Communications & Navigation Satellite Constellation." Paper #252, 2000 IEEE Aerospace Conference, Big Sky, MT, March 18-25, 2000.

[4]Hastrup, R. C., Bell, D. J., Cesarone, R. J., Edwards, C. D., Ely, T. A., Guinn, J. R., Rosell, S. N., Srinivasan, J. M., Townes, S. A., "Mars Network for Enabling Low-Cost Missions." Paper # IAA-L-0509, Fourth IAA International Conference on Low-Cost Planetary Missions, The Johns Hopkins University, Applied Physics Laboratory, Laurel, MD, May 2-5, 2000.

[5]Edward, C. D., Adams, J. T., Bell, D. J., Cesarone, R., Depaula, R., Durning, J. F., Ely, T. A., Leung, R. Y., McGraw, C. A., Rosell, S. N., "Strategies for Telecommunications and Navigation in Support of Mars Exploration." Paper No. IAF-00-Q.3.05, 51st International Astronautical Congress, Rio de Janeiro, Brazil, Oct 2 – 6, 2000.

[6]Ely, T. A., Guinn, J., "Mars Approach Navigation using Mars Network Doppler Data." Paper No. AIAA 2002-4816, AIAA/AAS Astrodynamics Specialist Conference, Monterey, CA, August 5 – 8, 2002.

[7]Ely, T. A., Guinn, J., Quintanilla, E., "Navigation Services of the Mars Network." ION 59th Annual Meeting, Albuquerque, June 23 – 25, 2003.

[8]Mogensen, A., Campbell, T., and Lightsey, E. G., "Performance Analysis of the Tracking Loop Design of the Electra Transceiver", Proceedings of the Institute of Navigation National Technical Meeting 2006, Monterey, California, Jan. 18-20, 2006

[9]Li, R., et. al., "Initial Results of Rover Localization and Topographic Mapping for the 2003 Mars Exploration Rover Mission." Photgrammatic Engineering & Remote Sensing, Vol 71, No 10, October 2005.