High Speed Lunar Navigation for Crewed and Remotely Piloted Vehicles

L. Pedersen*, M. Allan*, V. To*, H. Utz*, W. Wojcikiewicz**, C. Chautems***

*NASA Ames Research Center / Carnegie Mellon University, Moffett Field, California, USA
email: Liam.Pedersen@nasa.gov, Mark.Allan@nasa.gov, Vinh.To@nasa.gov, Hans.Utz@nasa.gov
**Humboldt University, Germany
***ETH, Zurich, Switzerland

Abstract

Increased navigation speed is desirable for lunar rovers, whether autonomous, crewed or remotely operated, but is hampered by the low gravity, high contrast lighting and rough terrain. We describe lidar based navigation system deployed on NASA's K10 autonomous rover and to increase the terrain hazard situational awareness of the Lunar Electric Rover crew.

1 Introduction

High speed mobility (by planetary rover standards) is desirable for more efficient exploration of the lunar surface, particularly if human crews are present there. NASA's Lunar Electric Rover (LER, Figure 1) can drive at 10km/h (3m/s) on rough terrain, faster on known smooth surfaces. By comparison, the MER vehicles moves at a few cm/s while stopping regularly.

Driving on the Moon is complicated by the low gravity, rough terrain, high contrast lighting and alien environment. Under the $\frac{1}{6}g$ lunar gravity a vehicle becomes airborne at relatively modest speeds (driving the Apollo era lunar rover at 7 mph was reportedly like being aboard a rowing boat for this reason). The reduced gravity also lessens vehicle-ground friction. Together these effects increase maneuvering distances.

Astronauts complained of difficulty discerning surface features on the Moon under certain lighting conditions, such as with the sun behind them. The lack of terrestrial depth cues compounds the difficulties of driving.

The 3 sec signal round trip time delay to the Moon permits tele-operating a vehicle from Earth (e.g. Lunakhod) with difficulty (increasing with speed).

This paper describes our robotic rover navigation systems, currently running on NASA's autonomous or tele-operated K10 rover (Figure 2) and as a crew aid on the LER for increasing pilot situational awareness of the surrounding terrain. We address terrain sensing for 4 different classes of rovers: K10, the Lunar All Terrain Utility Vehicle (LATUV, Figure 3), the LER and the LER operating under lunar gravity and time-delayed tele-operation. Subsequent sections describe the terrain mapping and haz-

Figure 1. : NASA's Lunar Electric Vehicle (LER), designed to sustain a 2 person crew and drive at 3m/s on rough terrain

Figure 2. : NASA's K10 rover, an autonomous or tele-operated robot designed for robotic site survey [3] at speeds up to 0.6 m/s

Figure 3. : The Lunar All Terrain Utlity Vehicle (LATUV), developed by ProtoInnovations LLC under a NASA SBIR as a research platform for lunar navigation and capable of speeds up to 2m/s.

ard detection software, pose estimation, the rover software infrastructure for command and data handling, the graphical user interface and performance figures.

2 Problem Statement

Table 1. : Rover specifications. LER Moon refers to the LER operating under lunar gravity and time delays.

	K10	LATUV	LER	LER Moon
Speed (m/s)	0.6	2.0	3.0	2.0
Max pitch rate (°/s)	10	10	10	10
Max deceleration (m/s^2)	2.0	2.0	2.0	1.0
Reaction time (s)	2.0	2.0	2.0	5.0
Min obstacle size (cm)	10	15	15	15
Min hole size (cm)	30	45	50	50
Max slope (°)	30	30	35	35

We address the problems of terrain mapping and hazard detection for navigating a set of lunar rover prototypes under on lunar like terrain at relatively high speeds, summarized in Table 1. The maximum pitch rates are for travel on level (but bumpy) ground, and may be exceeded approximately 10% of the time. Quoted maximum deceleration rates are chosen to limit loads on the vehicles and occupants, and are less than the maximum possible decelerations used in an emergency stop situation permitted by the friction between the wheel and the ground. The 2 sec reaction times are estimates for a human operator. An additional 3 sec round trip time delay is added to the LER Moon.

Lunar terrain hazards [7] addressed are slopes and discrete obstacles, such as ejecta blocks associated with crater rims (Figure 9). The maximum obstacle height that a wheel can surmount is typically given by 0.7 times the wheel radius. We assume more conservative values (Table 1) given to us to limit shock impact to the vehicle suspension. From these we deduce a maximum traversable negative obstacle (hole) size.

Additional requirements are to operate in dark or shadowed regions, limit the onboard computational resources to a single laptop (a 2.33GHz dual pentium with 2GB memory), and use common sensors, avionics and software to the greatest extent possible across all rovers.

3 System Overview

Our approach to high speed navigation uses multiple scanning laser rangefinders (lidars) in a push-broom configuration to sense the terrain geometry as the rover advances. Terrain points (specified in a site centered coordinate frame) are deduced from range measurements combined with vehicle pose (interpolated from pose estimates before and after each scan), and used to generate local traversability maps centered around the rover. Maps, 3D points, and rovers state are sent to an (off-board or onboard) immersive 3D GUI for tele-operation and situational awareness. Maps and rover state are also sent to the onboard navigator software to autonomously drive the rover (if desired). The navigator is based on the classic Ranger/Morphin architecture first described in [9], and in use by many vehicles today including MER, incorporating a local path planner combined with a global planner (such as D* [10]) operating on a larger but lower resolution map build up from successive local map updates.

4 Relevant Work

The Stanford Darpa Grand Challenge vehicle Stanley [12] established an efficient statistically grounded method for detecting obstacles with single line scanning lidars based on the height differences within a neighborhood [12]. This proved effective for obstacle detection on relatively flat terrain (roads), but to traverse extreme terrain it is desirable to check potential paths against a full 3D terrain reconstruction such as [17] which uses a reduced triangulated mesh terrain representation to navigate a vehicle at slow speeds (1 cm/s) with regular stops.

As vehicle speeds increase, non-holonomic motion constraints become pronounced with the result global path planners such as D* produce unfeasible trajectories. [8]

Figure 4. : Ballet Crater (Apollo 17)

seeks to improve the performance of the Ranger architecture using more realistic motion primitives, while [4] demonstrates combined local and global planning in a higher dimension lattice that guarantees feasible paths at the expense of more computation than available to us.

5 Sensors

Terrain and hazard detection sensors consist of scanning laser rangefinders (the SICK LMS291 and Hokuyo UTM-30LX) in a forward looking push-broom configuration, arranged with various look ahead distances. Forward looking stereo cameras (PGRFlea's, synchronized over the firewire bus) provide stereo ranging with a 100° field of view to complement the lidars. Pose sensing is accomplished with an Xsens MTi-G INS, a Honeywell HMR3000 compass/inclinometer, Novatel carrier phase DGPS and vehicle odometry (except on the LER, where we rely on the vehicle's own pose estimation system provided by it's makers).

The look ahead distance to detect discrete obstacles is given by the maneuvering distance, which we choose as the maximum of stopping distance and swerving distance [15], plus a safety margin accounting for map cell sizes and distance from sensors to the vehicle front.

$$d_{stop} = vT_{react} + v^2/2a \tag{1}$$

$$d_{swerve} = vT_{react} + \sqrt{\frac{2s_0v^2}{a} - s_0^2} \tag{2}$$

where v is rover speed, T_{react} is the reaction time, s_0 the swerving distance, and a the maximum permitted acceleration or deceleration (limited by μg, the wheel/ground coefficient of friction times the acceleration due to gravity).

Figure 5. : Terrain mapping, hazard detection and pose estimation sensors on the LER

The look ahead distance for slope detection is given by the obstacle maneuvering distance above plus half the vehicle wheel base.

The maximum traversable obstacle height imposes the requirement that the range sensors be able to resolve height differences half that at the obstacle detection look ahead. The required horizontal along-track resolution is similarly determined by the maximum traversable hole size. We *assume* that obstacles are at least as wide as tall, and therefore require a cross track resolution equal to the vertical resolution. Table 1 shows derived detection distances and resolutions.

A forward looking scanning laser rangefinder (lidar) vertical resolution is given by

$$\delta_{vert} = v\sin(\theta) + \frac{r\dot{\theta}}{f_{scan}\cos(\theta)} \tag{3}$$

where *theta* is the angle between the horizontal plane and the lidar beam (in radians), r is the slant range, and f_{scan} the scanning frequency (in Hz). The horizontal along-track resolution is given by

$$\delta_{along\ track} = \frac{1}{f_{scan}}(v + \frac{r\dot{\theta}}{\sin(\theta)}) \tag{4}$$

Table 3 summarizes the lidar performance for each rover at the obstacle detection distance (the slope detection requirements are less restrictive). For further details on lidar and stereo requirements see [15]

6 Terrain Mapping and Hazard Detection

The terrain hazards we consider are slopes, discrete obsacles (e.g. ejecta blocks) and voids. Slopes hazards are mapped using the *Time-Windowed* variant of Morphin

Table 2. : Rover maneuvering distances and resultant lidar requirements.

	K10	LATUV	LER	LER Moon
Stopping distance (m)	1.3	5.0	8.3	12.0
Swerving distance (m)	1.2	6.0	9.9	13.6
Margin (m)	3.0	3.0	4.9	6.8
Positive obstacle look ahead (m)	4.3	9.0	14.8	20.4
Slope look ahead (m)	4.8	9.9	17.0	22.6
Required vertical resolution (cm)	5	4.5	7.5	7.5
Required cross-track resolution (cm)	5	4.5	7.5	7.5
Required along-track resolution (cm)	15	23	25	25

Table 3. : Rover lidar performance figures at obstacle detection distance from sensors.

	K10	LATUV	LER	LER Moon
Lidar	UTM	LMS 291	LMS 291	(tbd)
Inter-point angle (°)	0.25	0.25	0.25	0.25
Scan rate (Hz)	40	75	75	75
Vertical resolution (cm)	2	2	4	5
Cross-track resolution directly in front (cm)	1	3	6	8
Along-track resolution (cm)	3	12	18	28

[7](Figure 6) and obstacles with a variant of the *Probabilistic Terrain Analysis*[13](Figure 7). Void detection is currently still under investigation but not yet part of our system.

Time Windowed Morphin (TWM) is based on the observation that relative pose errors increase over time so that terrain points measured close in time are more likely to be consistent. TWM by fitting planes to the most *recent* points within a time window at each map grid cell, followed by the standard Morphin traversability analysis [9] of the plane slopes and residuals over a rover sized patch.

Rover pose noise induces terrain reconstruction errors often exceeding discrete obstacle sizes. Because of this, and an $O(n^4)$ (for an n x n map), TWM is ill suited for detecting small obstacles, especially as greater map resolution is needed to localize these.

Probabilistic Terrain Analysis (PTA) marks a cell as untraversable if two measured terrain points within a distance ϵ of each other differ in height by more than δ plus a margin that increases with the time difference between when the points were measured, taking into account sensor geometries and pose estimation error characteristics in a principled manner. Only points measured within a specific time interval of each other are used in our variant of PTA (since, unlike in the Darpa Grand Challenge for which PTA was developed, our rovers have multiple opportunities to probe the same terrain). PTA is computationally very efficient ($O(n^2)$ for an n x n map) and robust to most pose estimate noise (in fact, noise robustness can be traded against obstacle size detection limits.)

The TWM and PTA maps are pessimistically combined (that is, we take the worst case traversability assessment above a given confidence threshold), and the resultant map sent to the navigator module and/or the user interface.

7 Pose Estimation

High rate (25Hz) locally accurate pose information, particularly attitude, is required to transform lidar points into a common reference frame for mapping. For K10 (and eventually LATUV) we rely on the Xsens MTi-G INS, which uses MEMS accelerometers, gyros and a magnetometer inputs to produce roll/pitch estimates with 0.5 °, and yaw estimates with 2.0 ° quoted static accuracies. In practice, a slowly varying attitude offset up to ± 5.0 ° is sometimes observed, and accuracy is significantly degraded by vibrations due to the rover traversing rough terrain or maneuvering.

Position estimates are obtained from odometry, corrected by differential GPS updates in a standard EKF. The EKF improves the INS attitude estimates by estimating the INS attitude error (in roll, pitch and yaw), modeled as

Figure 6. : Time Windowed Morphin traversability map. Green areas have slopes < 13 °, red areas have slopes > 23 °.

Figure 7. : Probabilistic Terrain Analysis (PTA).The red areas have confirmed obstacles at least 10cm high. Areas with sufficient measurements but no obstacles are labelled traversable (green).

a first order damped Gauss-Markov process:

$$\delta\hat{q}(t + \Delta t) = \Phi\delta\hat{q}(t) + \sigma\sqrt{1 - \Phi^2}\eta \tag{5}$$

where $\Phi = e^{-\Delta t/\tau}$, σ and τ are process noise standard deviation and time constant respectively, and η is zero mean, unit variance white Gaussian noise. The estimated attitude is then given as

$$\hat{q} = q' + \delta\hat{q} \tag{6}$$

where q' is the INS roll/pitch/yaw output, and $\delta\hat{q}$ is the EKF estimate of the INS error. This indirect filter approach combines the rapid response time of the INS to attitude changes with the improved *static* accuracy available from the HMR3000 compass/inclinometer.

In environments where the magnetic field is unreliable for yaw estimates (such as high magnetic latitudes) the INS is operated in a mode that ignores the internal magnetometer, providing yaw estimates by integrating the gyros only. In this case, we use a random walk process model for the yaw error estimate:

$$\delta\hat{q}(t + \Delta t) = \delta\hat{q}(t) + \alpha\sqrt{\Delta t}\eta \tag{7}$$

where α is the *angular random walk* ($^\circ/\sqrt{s}$), and external yaw corrections are provided by a suntracker [1].

On the LER we rely on a separate pose estimation system provided by its makers, using a Hemisphere Vector GPS, Microstrain 3DMGX1 inclinometer, Crossbow VG700 fiber optic vertical gyro and vehicle odometry. The VG700 quotes 2.0 ° roll and pitch accuracies, but the overall system accuracy is unknown.

8 Software Infrastructure

Increased navigation speed poses challenges to the software infrastructure. The data volume of the navigation sensors is significant and needs to be delivered to the relevant processing modules. Push models ensure lower latencies but carry the risk of clogging, if parts of the processing pipeline don't keep up with the load. But soft real-time requirements now apply to entire software system, so the individual modules need to become insulated from latency spikes in other sub-system components.

Controlling a dynamic vehicle in this distributed environment also poses unique challenges in terms of engineering support. For adequately assessing analyzing system performance, it is crucial to enable the analysis and replay of data traces of the entire distributed system.

We addressed these challenges in the design of IRG's Service Oriented Robotics Architecture (SORA)[2]. For managing the high data volume in this challenging distributed environment, we deploy proven middleware technologies, namely CORBA (for commanding) and DDS (for data distribution). The middleware-supported publish-subscribe infrastructure provides a push-model for sensor data delivery while providing extensive quality of service management, that facilitates a central capability for load management. Modelling the individual subsystems as insulated services with high-level interfaces provides an adequate insulation of the modules. Latency peaks can be more easily located and addressed within the individual subsystem. The software infrastructure also provides generic logging capabilities provided by Miro (Middleware for Robots)[16]. It allows distributed real-

time logging of data streams as well as the synchronized replay of such acquired log-files.

A recurring challenge in software-intensive systems is the integration of outside software components. In this particular case, a target platform, LER, is developed at another NASA center. Sensor data provided by the LER vehicle is crucial input for the navigation system. For this purpose we are using the Robot API Delegate (RAPID), an intercenter robot API, which is also based on DDS middleware. This significantly reduced the integration effort and enabled the integration of LER provided telemetry into the hazard-analysis.

9 User Interfaces

The best tele-operation user interface [5] displays a 3D visualization of the rover situated within a terrain reconstruction, colorized either by surface albedo (and appropriately shaded to indicate slope) or in false colors according to traversability. VERVE (Visual Environment for Remote and Virtual Exploration) is our high performance 3D user interface used for real time visualization of robots' activities. VERVE is geared toward visualizing highly dynamic environments and monitoring multiple robots simultaneously. VERVE (Figure 8) allows operators to visualize robot pose and configuration, raw sensor data, and derived data such as navigation maps and path plans (at present we are unable to obtain terrain albedo from our sensors).

VERVE is component based software which is comprised of many Java plugins, or bundles, that are assembled into an end user application. Specifically, it is built on top of the Eclipse RCP (Rich Client Platform) which is an implementation of the OSGi Framework[18]. The OSGi programming model encourages modularity and code reuse and permits new capabilities to be easily inserted into existing applications. The flexibility of the component based development approach allows VERVE to be easily retargeted to different end user applications[14][11].

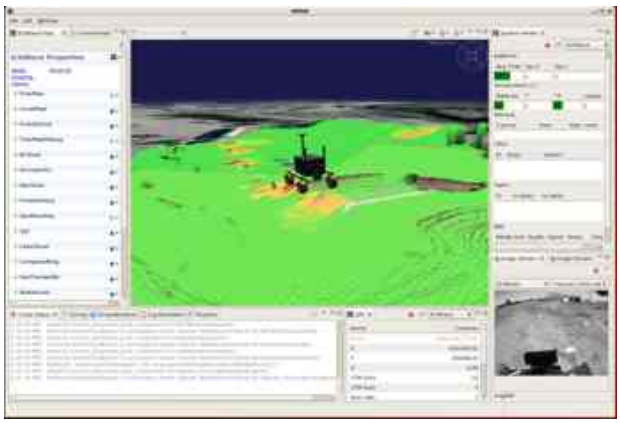

Figure 8. : VERVE user interface, showing the K10 rover on a 3D terrain reconstruction (from lidar measurements) with traversability analysis overlaid.

In order to monitor the activities of multiple robots which may have different communication mechanisms, VERVE decouples telemetry data transport from the graphical display. The most commonly used telemetry components in VERVE are the COBRA Notification Service and OMG Data Distribution Service (DDS), but new networking middleware can be easily supported by adding additional event collector components. Each event collector is responsible for gathering telemetry and distributing it internally to data listeners. Data delivery is asynchronous and event throttling is handled within the event collector.

In order to represent the robots in the 3D virtual world, each avatar is modelled as a collection of loosely coupled parts. Each part binds data received though event collection to an action on the scene graph. In the simple case of joint angle telemetry, a data event would update transform nodes in the graphical 3D model, whereas a navigation map data event would trigger reconstruction of the height field mesh. The 3D scene is updated at a fixed rate (typically 30Hz), and telemetry is processed asynchronously upon arrival.

Parts may be freely mixed and matched and are often reused between robots. Because data transport is decoupled from the rest of the system in the telemetry components, a single, logical robot avatar can consist of subsystems which use different communication protocols.

10 Performance

Table 4 summarizes worst case algorithm performance rates on a representative test problem. K10, LATUV and LER lidars produce 58, 223 and 223 thousand points/sec respectively, well within the processing capacity of the laptop navigation computer. The limiting factor is the rate at which the Time Windowed Morphin map can be updated.

11 Conclusions and Further Work

We have described a flexible navigation system designed to fit the needs of several distinct lunar rover prototypes, spanning a range of sizes, speeds and operational modes, using a common set of sensors and software. The system is currently autonomously navigating the K10 rover, and tested as a pilot aid for the LER on Earth. The sensor analysis suggests that sensors with similar performance could be used to remotely operate the LER on the

Table 4. : Execution speed

	Dell Precision	MacBook Pro
Clock speed	2.33GHz	2.8GHz
No. processors	2	2
Memory	2GB	4GB
OS	RH5 (linux)	MacOs 10.5
Coordinate transforms (million points/sec)	15.4	5.77
Adding points to PTA map (million points/sec)	5.57	5.77
Adding points to TwMorphin map (million points/sec)	2.34	2.36
PTA Map Update rate	59Hz	79Hz
TwMorphin Map Update rate	0.44Hz	0.55Hz

Moon with a 3 sec round trip signal delay plus 2 sec operator reaction time.

The major weakness of the current system is a susceptibility to un-modeled pose estimate noise and the limited opportunities to sense the terrain as the rover drives forward. For example, when cresting a hill in the LER (Figure 1) the forward looking lidars were unable to see the ground, or did not see far ahead when driving down the hill. Both problems are reduced, if not eliminated, with more terrain sensors, covering both a greater vertical field of view and in a smaller time interval. While full frame range sensing (such as with stereo imaging or flash lidar) is ideal, an analysis of the geometry shows it to be unnecessary. Multiple closely space scanning lidars, if in a suitably dense configuration (such as a Velodyne sensor) are sufficient.

The use of GPS continues to be an irritant in our system. Besides being unavailable on the Moon, GPS estimates contain discontinuities and drift that negatively affect map construction. The ideal pose estimate system would be locally accurate, compensate for slip and discontinuity free.

The grazing lidar-ground angles required for any reasonable look ahead distance result in the occlusion of many areas, including *potential negative obstacles* (PNO's or voids). It behooves us to explicitly detect occluded regions and distinguish them from unexplored areas. We have obtained promising results using a linear filter to predict the location of terrain measurements in the absence of occlusion based on previous measurements. Areas with predicted measurements but few real measurements are labelled as occluded (Figure 9);

Figure 9. : Occlusion map of a crater generated as the K10 rover drives towards it on the indicated path (occluded areas are red).

Computational constraints drove our decision to use grid based 2D C-space traversability maps for navigation, justified by our performance numbers (Table 4). It is unclear how well this approach will handle autonomous navigation on extremely rugged terrain wherein hazards include high centering, direction dependent tip-overs, and dynamic effects (such as becoming airborne after hitting a bump in low gravity). Promising approaches include generating reduced 3D triangulated meshes [17] to represent terrain, and dynamical simulations and lattice space path planning [4] for more complex maneuvers. Unfortunately, the terrain mesh approach does not take pose error induced reconstruction noise into account and would therefore miss small obstacles using the current sensor suite (it requires full frame 3D scans) and at present requires too much computation.

It is desirable to modulate vehicle speed in response to terrain properties, for example when approaching an occluded area (eg cresting a hill) that may contain an obstacle. In this situation an appropriate response is to slow down until the occluded area is in full view (and confirmed safe) but not necessarily detour around it. To date, velocity planning is done at the goal level. We are investigating local, variable velocity, trajectory planning (Figure 10).

References

[1] M. C. Deans, D. Wettergreen, and D. Villa, "A sun tracker for planetary analog rovers," in *ISAIRAS*.

[2] L. Flueckiger, V. To, and H. Utz, "Service oriented robotic architecture supporting a lunar analog test," in *Proceedings of the 9th Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS 2008)*, (Los Angeles, California), February 2008.

[3] T. W. Fong, M. Bualat, M. Deans, M. Allan, X. Bouyssounouse, M. Broxton, L. Edwards, R. Elphic, L. Fluckiger, J. Frank, L. Keely, L. Kobayashi, P. Lee, S. Y. Lee, D. Lees, E. Pacis, E. Park, L. Peder-

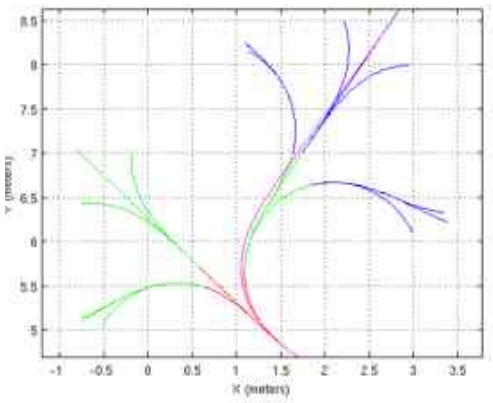

Figure 10. : Variable velocity trajectory expansion for local path planning

sen, D. Schreckenghost, T. Smith, V. To, and H. Utz, "Field testing of utility robots for lunar surface operations," in *AIAA Space*, 2008.

[4] T. Howard, C. Green, D. Ferguson, and A. Kelly, "State space sampling of feasible motions for high-performance mobile robot navigation in complex environments," *Journal of Field Robotics*, vol. 25, pp. 325–345, June 2008 .

[5] D. Huber, H. Herman, A. Kelly, P. Rander, and J. Ziglar, "Real-time photo-realistic visualization of 3d environments for enhanced tele-operation of vehicles," in *Proceedings of the International Conference on 3-D Digital Imaging and Modeling (3DIM)*, October 2009.

[6] K. Husmann and L. Pedersen, "Strobe lit high dynamic range stereo imagery for dark navigation," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, 2008.

[7] L. Pedersen, C. S. Han, and M. Vitus, "Dark navigation : Sensing and rover navigation in permanently shadowed lunar craters," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, 2008.

[8] M. Pivtoraiko, A. Kelly, and I. Nesnas, "Autonomous robot navigation using advanced motion primitives," in *IEEE Aerospace*, pp. 1–7, March 2009.

[9] R. Simmons, L. Henriksen, L. Chrisman, and G. Whelan, "Obstacle avoidance and safeguarding for a lunar rover," in *AIAA Forum on Advanced Developments in Space Robotics*, (Madison, WI), 1996.

[10] A. Stentz, "The Focussed D* Algorithm for Real-Time Replanning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995.

[11] V. SunSpiral, D. Chavez-Clemente, M. Broxton, L. Keely, P. Mihelich, D. Mittman, and C. Collins, "FootFall: A ground based operations toolset enabling walking for the ATHLETE rover," in *Proceedings of AIAA Space Conference*, 2008.

[12] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Winning the darpa grand challenge," *Journal of Field Robotics*, 2006. accepted for publication.

[13] S. Thrun, M. Montemerlo, and A. Aron, "Probabilistic terrain analysis for high speed desert driving," in *Robotics Science and Systems Conference* (G. Sukhatme, S. Schaal, W. Burgard, and D. Fox, eds.), (Philadelphia, PA), 2006.

[14] R. Torres, M. Allan, M. Stop, R. Hirsh, N. Space, and M. Wallick, "RAPID: Collaboration Results from Three NASA Centers in Commanding/Monitoring Lunar Assets,"

[15] C. Urmson, "Driving beyond stopping distance constraints," in *IROS*, pp. 1189–1194, IEEE, 2006.

[16] H. Utz, S. Sablatnög, S. Enderle, and G. K. Kraetzschmar, "Miro – middleware for mobile robot applications," *IEEE Transactions on Robotics and Automation, Special Issue on Object-Oriented Distributed Control Architectures*, vol. 18, pp. 493–497, August 2002.

[17] D. Wettergreen, D. Jonak, D. Kohanbash, S. J. Moreland, S. Spiker, J. Teza, and W. R. L. Whittaker, "Design and experimentation of a rover concept for lunar crater resource survey," in *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, January 2009.

[18] http://www.osgi.org