Path Planning Challenges for Planetary Robots

Yoshiaki Kuwata, Alberto Elfes, Mark Maimone, Andrew Howard, Mihail Pivtoraiko, Thomas M. Howard, and Adrian Stoica

Abstract— **This paper discusses several path planning challenges for planetary robots and the on-going efforts to address them. The considered challenges include limited onboard computing resources, degraded/failed mechanical parts, and specific operating conditions such as the influence of atmospheric currents on aerobots with significant dynamics. These are examples of real-world challenges identified in the current and potential future missions. For each of the challenges, the paper also illustrates potential future solutions from the current research efforts.**

I. INTRODUCTION

This paper focuses on path planning and navigation aspects of planetary robots. The term "planetary robots" is used here to include not only surface rovers, as the ones already deployed or planned for Mars and the Moon, but underwater and aerial robots as well. Some technologies needed for such robots are currently under development and would have dual use, both in space-based and terrestrial applications.

The Jet Propulsion Laboratory (JPL) is NASA's lead center for robotic space exploration. Several potential future planetary robots are being designed at JPL for surface operations, including Mars Science Laboratory (MSL), targeted in scientific exploration of Mars, and various rovers for the Moon to support human exploration. Potential undersea robots for Europa and aerobots for Venus are also under design and concept development, because the unique planetary environments, such as potential planetary oceans and dense atmosphere, might benefit from different types of mobility. Figure 1 illustrates these missions in artist's view.

Planetary environments pose unique challenges to robot operations [4], from many operational perspectives, and these reflect in path planning constraints and difficulties of solutions. General challenges relate, for example, to

- Limited bandwidth for communication/control because the distance to planets precludes real-time teleoperation (except to the Moon); thus requiring a certain degree of autonomy.
- Harsh environments leading to more rapid degradation of components/systems, and aging due to longer missions. Overall, robots face degraded/faulty components and subsystems, from sensors to actuators, and the computing platforms in between. While designing robust and fault-tolerant hardware might be conceived (and would be used in certain cases, for example, in triple modular redundancy solutions on the critical subsystems), it would add significantly to the costs, both directly and indirectly (e.g. increased weight affecting launch costs, etc.).
- Less-than-state-of-the-art technologies, due to selecting technologies mature enough far prior to missions, often in excess of several years prior to launch, also due to difficult/expensive process of flight validation.

Y. Kuwata, A. Elfes, M. Maimone, M. Pivtoraiko, A. Howard, and A. Stoica are with Mobility and Robotics Systems, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA, USA `Yoshiaki.Kuwata@jpl.nasa.gov`

T.M. Howard is with Robotic Institute, Carnegie Melon University, 5000 Forbes Avenue, Pittsburgh, PA, USA.

Fig. 1. Artist's conceptions of future planetary robots: (top, left) Mars Science Laboratory [1]; (top, right) Lunar exploration with robots and humans [2]; (bottom, left) Picture of a possible undersea robot that would explore Europa's oceans for life: "Hydrobot" [3]; (bottom, right) Venus altitude-cycling balloon based on phase-change buoyancy fluids.

There are also specific challenges, which might depend on a particular scenario, such as aerial exploration with balloons subjected to various currents. In addition to Earth, seven other bodies in the Solar System have enough atmosphere to allow aerial exploration. The NASA 2003 Solar System Exploration Roadmap identified aerial vehicles as a strategic new technology for Solar System exploration [5], and emphasized the development of advanced autonomy technologies as a high priority area for the operation of aerial probes. NASA's 2006 Solar System Exploration Roadmap [6] confirms and extends this vision, listing future Titan and Venus in situ missions using lighter-than-air vehicles as two of its top three flagship mission priorities. In situ platforms would be essential because of the dense clouds that cover Titan and Venus, limiting orbital surveys. However, the dense atmospheres at Titan and Venus enable the use of buoyant robotic vehicles (aerobots) that could be either self-propelled (airships) or wind-driven (balloons). The primary challenge associated with their operation would be that they must address currents constantly changing in the dynamic aerial environments during navigation.

The rest of the paper is organized as follows. Section II elaborates the specific challenges that planetary robots face. Section III describes some of the ongoing efforts conducted at JPL to address these challenges. Finally, Section IV discusses future directions.

II. Challenges in current and future planetary robot missions

A. Resource Constraints

JPL has operated two Mars Exploration Rovers (MERs) called "Spirit" and "Opportunity" since January 2004. In 2009, the third generation rover called Mars Science Laboratory (MSL) is planned to be launched. These current and future Mars rovers are subject to limited computation and power resources constraints.

For example, the onboard computer of the MER vehicles uses a radiation hardened processor, with a 20 MHz CPU and 128 MB of RAM. Furthermore, each CPU runs more than 90 concurrent tasks such as data communication, command handling, and health monitoring. In addition, interfaces with instruments are not optimized for speed; simply taking an image and storing it into RAM can take a minimum of 5 seconds.

MSL is expected to have an average power generation of about 100 W. Considering the power consumption of driving motors and electronics, the navigational autonomy system will not be able keep all the sensors powered on and/or keep high CPU utilization for very long.

B. Hardware Degradation

Another challenge for planetary robots is that repairing their hardware is infeasible once they leave Earth. During a long mission, hardware failures will limit the capability of the robot. In such cases, the navigational autonomy must be able to work even with these failed hardware components.

The mission life of MERs was designed to be three months. However, with successful operation in the Martian environment, the rover missions have been extended significantly (both rovers landed on Mars in January, 2004, and both of them are still operational as of June 2008). In 2006, Spirit's right front drive motor stopped working, creating significant drag on one side. Figure 2 shows the trench developed by dragging this frozen wheel along the ground. Spirit was able to continue driving successfully using its onboard Visual Odometry software to compensate for the drag induced by the stuck wheel.

C. Spatio-Temporal Planning

The challenges presented above are associated mainly with deep space missions, where sending astronauts would be infeasible. In future space missions in a closer range,

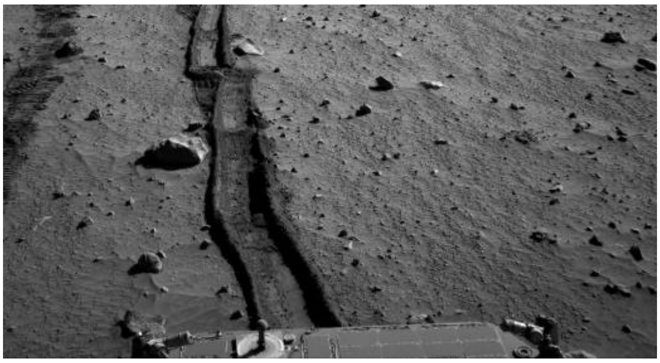


Fig. 2. Trench developed by dragging broken drive motor on the MER Spirit planetary rover.

such as moon, robots would work closely with astronauts and/or other groups of robots. How to deal with moving agents (called movers) in the environment is becoming an important challenge to be addressed. In addition to spatial path planning, to avoid potential collisions, the robot must take into account how other agents would move. In such a case, the temporal component would play a key role, and the path planning would become more complicated compared to the current Mars rovers which are able to plan assuming a static environment.

What makes the problem more challenging is the uncertainties associated with the movers. The future motion of astronauts would hardly be deterministic, and the future motion of the robots operated on an unknown terrain could not be known exactly. Thus, the planning algorithm must accurately infer and predict their future trajectories while allowing for the inherent uncertainties.

The planning with movers is becoming an active research area also outside of the context of space exploration. For example, the DARPA Urban Challenge (DUC), in which 89 teams from universities and industry teams competed, demonstrated the urban driving capabilities of the robotic cars that interacted with numerous manned and unmanned traffic vehicles.

D. Airborne Navigation

Trajectory planning for an aerobot would pose very different problems from those faced by ground robots. These would include the dynamics of the vehicle, and the dynamics of the atmosphere. Planetary rovers move at very low speeds, and from the point of view of vehicle dynamics, path planning could be considered quasi-static systems. Aerobots, by contrast, would have significant vehicle dynamics (which include virtual inertias). This is true both for self-propelled vehicles (airships or Montgolfiere balloons with propellers) and for standard unpropelled Montgolfieres (which would have altitude control, but no horizontal control capability).

Atmosphere dynamics expresses itself in weather (storms and turbulence) and in the planetary wind field. For safety reasons, an aerobot would need to avoid turbulent weather, which can be considered a dynamic, spatio-temporal "obstacle". At the same time, a planetary aerial vehicle should

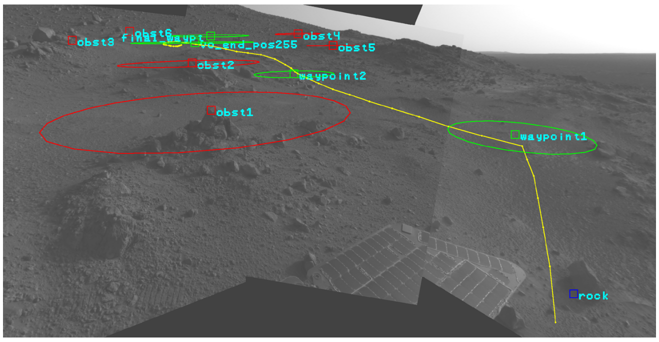


Fig. 3. Pre-drive annotation on the image from the MER onboard camera

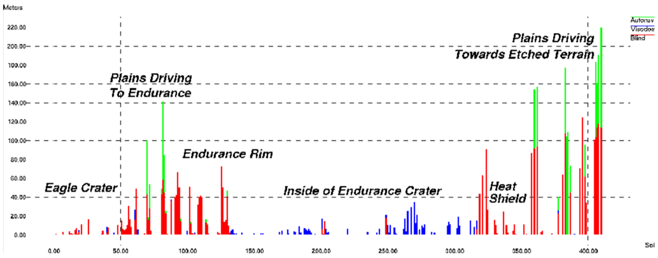


Fig. 4. Opportunity Drive Modes in first 410 Sols [9]

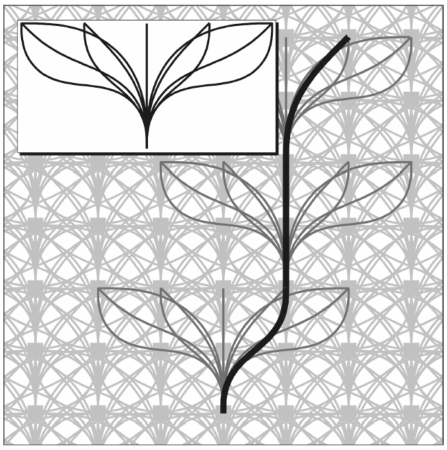


Fig. 5. A 2D projection of a simple example of the state lattice, a repeated and regular pattern of sampling robot state and motions (inset in the figure). Discrete values of state (gray squares in the figure) are vertices of the graph. The feasible motions that take the system from one state to another are edges between the corresponding vertices (gray lines). The solution to the motion planning problem is a path in this graph, a concatenation of edges (thick black line). Since each edge is kinematically feasible by construction, the solution is also feasible and can be used by the robot without any post-processing such as path smoothing.

take advantage of local and global winds to reach a desired destination. This would be important for self-propelled aerobots, due to onboard power limitations, and would be crucial for unpropelled aerobots, where "sailing" the three-dimensional wind field might be the only way to reach a general area of scientific interest (such as the methane lakes region in the northern hemisphere of Titan) [7]. While initial estimates of the global wind field might be derived from a Global Circulation Model (GCM), it would be essential for an aerobot to estimate the local wind field frequently to take advantage of it.

Trajectory planning for aerobots bears similarities to the challenges faced by Unmanned Sea Surface Vehicles (USSV) or Autonomous Underwater Vehicles (AUV). Here, the ocean dynamics (currents and waves) act as a disturbance on the vehicle path, and have to be compensated accordingly. JPL is also addressing this problem in the context of USSV projects, such as the OASIS boats [8].

III. On-going Research in Planetary Mobile Robot Path Planning

This section describes several on-going efforts being conducted at the Robotics section of JPL to overcome the challenges presented in Section II.

A. MER Activity Planning with Limited Resources

Due to the resource constraints, Mars rovers must be selective about what resources to use at when. MER has three drive modes. The so-called "Blind" mode simply drives to the next waypoint without onboard terrain analysis or obstacle avoidance. The "VisOdom" mode performs visual odometry [10] that can measure the actual wheel slip by comparing images taken before and after each drive step (normally less than 75cm). This is useful when driving on steep/rough/slippery terrain. The "AutoNav" mode uses Stereo Vision to detect geometric hazards, assesses terrain traversability and then autonomously plans the rover's drive to avoid any hazards.

If the rovers had enough computation power, VisOdom and AutoNav could be always enabled for safe driving. With limited resources, however, turning on these modes slows down the rover quite a bit (e.g., Visual Odometry drives are typically 8-9 times slower than blind drives). Therefore, if the human assessment of nearby terrain shows a large flat open field of bedrock, Blind mode is likely to be used [11]. Humans optimize the rover's time by operators determining which mode to use in different terrains, relying on images taken the previous day. Figure 3 shows an example of the pre-drive assessment made by human rover drivers.

The style of driving changes depending on the terrain. Figure 4 shows what modes were used during the first 410 Sols of the Opportunity driving. Note that in the area labeled "Inside of Endurance Crater," VisOdom is mostly used, while Blind driving was enough for other areas. That is because the rover experienced slopes between 17 and 31 degrees while in the crater.

B. Motion Planning under Computational Constraints

In environments with rough terrain, a robot's motion planner must consider a detailed model of the mobility constraints and capabilities of the vehicle. The only available path may be a complicated maneuver, and it is important for the planner to generate it in order to prevent failure of traversing the environment. At the same time, the motion planner must operate under significant computational constraints. A planner based on search in a specially designed search space, a state lattice, was developed and successfully applied in this setting. The state lattice is a sampling pattern of robot state and motions. By representing the state lattice as a graph, search algorithms (e.g. A*, D*, etc.) can compute the motion plan by finding the shortest path in this graph. Furthermore, by discretizing the state space, in contrast to control space, the state lattice simplifies reducing dispersion of sampling, thereby allowing a more uniform distribution of samples in the state space. This is beneficial to exploring the state space efficiently, as the search attempts to find a path from the specified initial state to the final state.

A simplified example of the state lattice graph is shown in Figure 5, where the squares are the vertices, and the lines are the edges. The example shows a lattice of three dimensions (translational coordinates (x, y) and heading) projected on 2D plane for illustration. A feasible motion between two robot states is a concatenation of the edges (thick black line). This approach was successfully tested in field experiments using a FIDO rover prototype [12], [13]. The rover was capable of traversing an unknown rough terrain environment while replanning, on average at 10 Hz, due to incoming perception information, using a single CPU shared among all software processes of the rover.

C. Predictive Model Compensation for Impaired Hardware

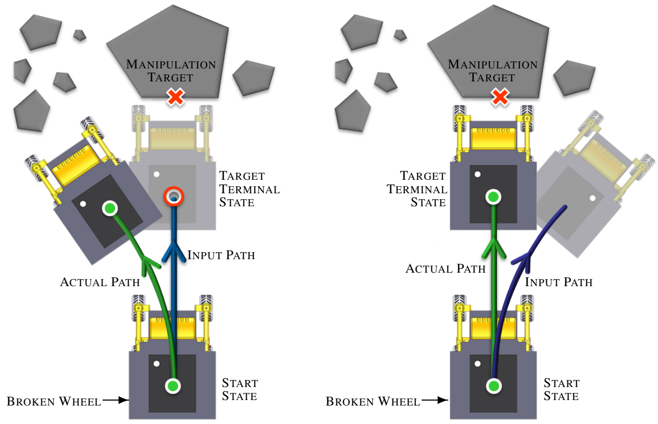
When mobility systems degrade, a standard technique is to compensate for the path following error reactively. Spirit deals with deviations from the intended path caused by the broken wheel by generating a correction factor proportional to the measured crosstrack error from Visual Odometry. In situations with more critical hardware degradation, a mobile robot could wander significantly from the desired course, leading to inefficient navigation behaviors that could endanger the platform. Consider the problem posed in Figure 6, where a mobile robot must move from a start state to a goal state to deploy instruments on a science target. To model the effects of degraded hardware, the robot operates with a disabled rear left drive wheel. If purely reactive approaches fail to move the base in a position where the science target is within the workspace of the manipulator, it would be difficult to generate the proper corrective action given the vehicle's nonholonomic constraints (Figure 6a).

Alternatively, predictive motion planning and control methods could be used to reduce the path following error. A more informed controller that knows that it must steer to the right to compensate for the dragging wheel over the course of the path would potentially be more effective. A model-predictive control and trajectory generation technique in [14] can be used to compensate for disturbances predictively, rather than reactively (Figure 6b). This method was validated in field experiments using the Rocky8 rover prototype [12] in the JPL Mars Yard.

D. Spatio-Temporal Planning in Dynamic Environment

1) Decoupled Search: A* search is a widely-used search method applied over a grid, but extensions are required to apply it to a robot with kinematic constraints in dynamic environments. In Team Caltech's entry to DUC, we have developed a decoupled spatial/temporal planner. It first finds an obstacle-free path to the goal (spatial planning) that ignores other movers. It then finds a velocity profile that minimizes the risk of collisions with movers (temporal planning). This decoupling allows the planner to be as efficient as the static planner. Although this is effective for driving on structured roads, the resulting path could be highly suboptimal in a more general setting.

2) Search in Action Space with Pruning: Another approach being developed that uses A* search is a spatio-temporal planner that can simultaneously find collision-free trajectories with both static obstacles and movers. The search



(a) w/o predictive model comp. (b) w/ predictive model comp.

Fig. 6. Predictive model compensation of impaired mobility and wheel slip. In (a), a plan is generated assuming ideal mobility and no wheel slip. The result of applying this action with impaired mobility (broken rear left drive wheel) is shown. In (b), given the model of impaired mobility, the model-predictive trajectory generator predictively compensates for this effect.

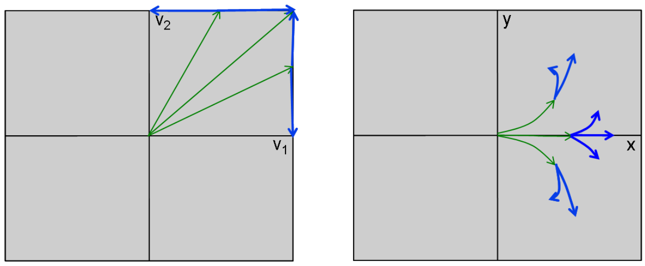


Fig. 7. Left: Input to the vehicle over two time steps. Right: Resulting trajectories. The number of branches grows exponentially.

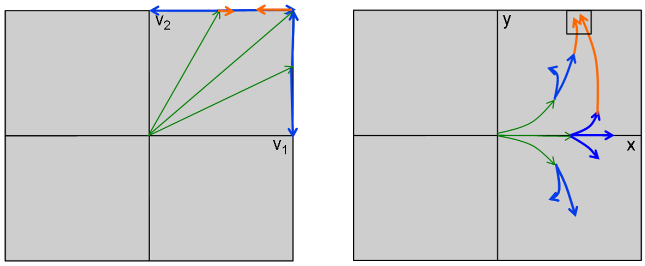


Fig. 8. Search over three time steps. If two different input sequences (shown in orange lines) result in the same states, one of them is pruned.

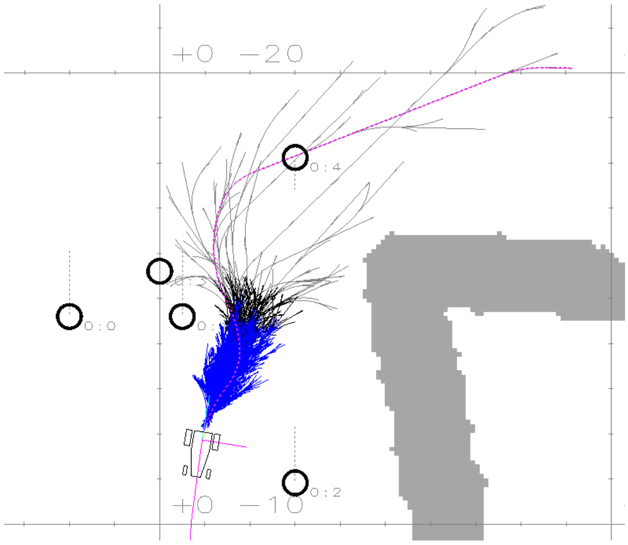


Fig. 9. Output of the spatio-temporal planner. The robot is at the bottom, going to a target in the upper right corner. Movers are shown with black circles. Note that the best trajectory goes through a mover. By the time the robot reaches there, the mover will have moved away.

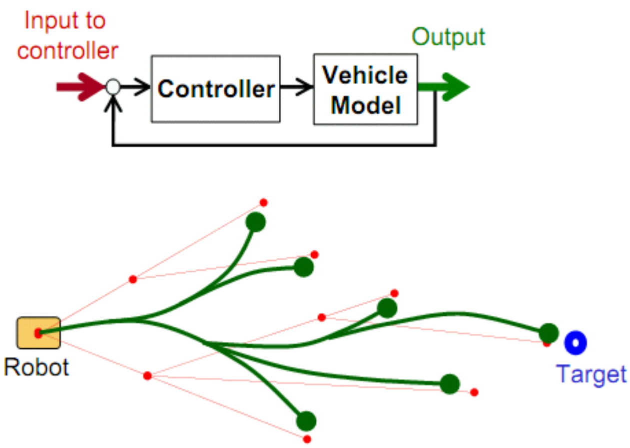


Fig. 10. RRT with closed-loop simulation. Input to the vehicle (shown in red) is sampled, and using the stabilizing controller and the model of the vehicle, the predicted states of the vehicle (shown in green) are obtained.

is over the input velocity command to the vehicle and its duration, and the vehicle states are obtained by feeding in the velocity command to the vehicle model. Because of the large decision space, the full dimensional search is limited to short look-ahead times (on the order of a few seconds). To avoid the exponential growth of the search space, some pruning heuristics are employed, as shown in Figures 7 and 8. Figure 9 shows a representative result of an optimal trajectory with movers. This algorithm has been also demonstrated on a mobile robot platform in an environment with pedestrians.

3) Randomized Planning: For problems with a high dimensional search space, sampling-based approaches have been employed successfully. Recently, Rapidly-exploring Random Tree (RRT) has been extended to include closed-loop prediction of the full vehicle dynamics [15]. Unlike

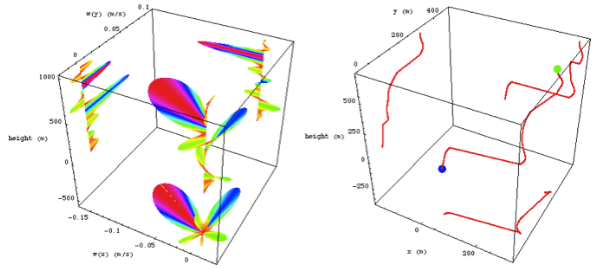


Fig. 11. Navigation in a wind field. A 3D wind field is shown on the left, and an optimal wind-based trajectory going from the start (blue dot) to the goal (green dot) is shown on the right.

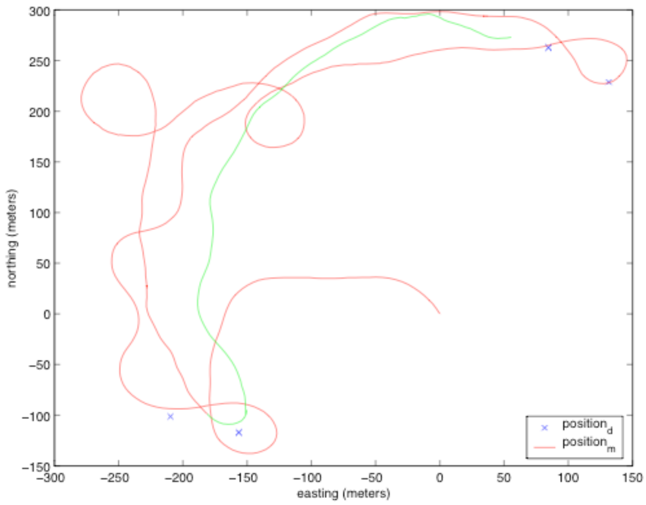


Fig. 12. Waypoint flight control under severe wind disturbances. Desired positions are marked with $\times$, and the red lines show the actual flight paths. [16]

the standard RRT, an input to the controller is sampled, which consists of a 2D point and a speed command profile. Then, it performs forward simulation using the vehicle model and the controller to compute the predicted trajectory, as shown in Figure 10. This enables the planner to generate smooth trajectories much more efficiently, while the randomization allows the planner to explore cluttered and dynamic environment. The predicted trajectory is checked with static and dynamic obstacles for feasibility and risk evaluation. The main advantages are that the simulation can easily incorporate any nonlinear control law and nonlinear vehicle dynamics, and the resulting trajectory is dynamically feasible. This planner was successfully demonstrated at the final race of DUC.

E. *Aerobot Planning with Combined Dynamics*

Planning an optimal aerobot trajectory that takes into account vehicle and atmospheric dynamics requires us to model the problem in terms of a continuous dynamic system. An optimal control problem (OCP) can be formulated by combining the vehicle dynamics and the atmospheric dynamics in the state equation, and defining some cost function to minimize.

In practice, trajectory optimization problems such as OCP often are converted to a multistage (discrete time) dynamic system model, and the solution is obtained using dynamic programming techniques (of which A* is a special case). Also, if there is significant uncertainty in the wind field model, planning should be done over a limited horizon and the wind field model should be updated with data obtained from the aerobot. Figure 11 shows a minimum-time trajectory in a wind field. The generated trajectory can be followed using a simple feedback control, and waypoint-based navigation results are shown in Figure 12. In this field experiment, the maximum speed of the aerobot was 13 m/s, with wind gusts up to 15 m/s. Even under significant wind disturbances, the aerobot reached all waypoints (shown with blue $\times$).

IV. DISCUSSION

This paper presented the challenges related to path planning for planetary robots. The following challenges have been reviewed: resource-constrained autonomy; dealing with degraded hardware; planning with movers; and significant vehicle dynamics under significant wind flows. The paper illustrated how these challenges are addressed by discussing on-going efforts conducted at JPL.

Despite numerous on-going efforts, several issues remain open for research.

First, to efficiently use the resource, the MERs had to be manually configured for each terrain type, even within a single drive. *Optimal resource allocation* algorithm should be developed, and in this regard, a coarse but long-range terrain analysis is a key to the automatic planning of a drive mode. This requires better adaptation to novel terrain, and a new adaptive terrain analysis technology needs to be also developed that can integrate various types of data such as slip measurements, terrain geometry, terrain texture, possibly onboard science analysis.

To develop a planning architecture with resource constraints, a more detailed modeling of overall system resource use is needed, such as a prediction of CPU resource use as a function of sensed data size (e.g., image resolution).

Feedforward methods using model-predictive trajectory generation techniques can be effective for mobile robot navigation when disturbances are regular and reasonably predictable. In the face of disturbances that are unpredictable or difficult to predict (e.g. wind, wheel slip), model-predictive techniques can be effectively used in combination with feedback control methods. This is important for planetary mobile robot applications because feedback control rates are inherently slow from Visual Odometry being so expensive. Online system identification will further improve the performance of model-predictive techniques for planning and control by regularly tuning the predictive motion model. Another direction for degraded hardware compensation includes reconfiguration of the hardware itself. Self-reconfigurable hardware is becoming an active research area although further efforts are needed to bring the technology to the robotics field.

Initial results with dealing with movers showed the importance of handling uncertainties. Even with perfect sensing, the movers' future trajectories are uncertain, and more stochastic representation of movers and how the planner plans on it are important areas of further research. In addition to the drivable/non-drivable representation of the 2D static path planning, several new representations such as an object list of movers, a map of occluded region, and flows of potential mover direction in a currently open space, are required, which would complicate the planning problem.

ACKNOWLEDGMENTS

Parts of this work described here were performed at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the National Aeronautics and Space Administration (NASA). Additional research was conducted at the Robotics Institute of Carnegie Mellon University under contract to NASA/JPL as part of the Mars Technology Program.

REFERENCES

[1] http://mars.jpl.nasa.gov/msl/gallery/
[2] http://spaceflight.nasa.gov/gallery/
[3] http://www2.jpl.nasa.gov/files/images/browse/p48326.gif
[4] M. Maimone, "Challenges for Planetary Rover Navigation," presented at International Conference on Robotics and Automation, 2008.
[5] NASA, "Solar System Exploration", NASA, Washington, DC, 2003.
[6] NASA, "2006 Solar System Exploration Strategic Roadmap". NASA, Washington, DC, 2006.
[7] T. Kämpke and A. Elfes. "Optimal Wind-Assisted Flight Planning for Planetary Aerobots". In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, IEEE. New Orleans, LS, April 2004.
[8] A. Elfes et al. "The Telesupervised Adaptive Ocean Sensor Fleet (TAOSF) Architecture: Coordination of Multiple Oceanic Robot Boats". In *Proceedings of the 2008 IEEE Aerospace Conference*, Big Sky, MT, March 2008.
[9] J. Biesiadecki and M. Maimone, "The Mars Exploration Rover surface mobility flight software: driving ambition," in *IEEE Aerospace conference proceedings*, 2006.
[10] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the Mars Exploration Rovers," Journal of Field Robotics, Volume 24, No. 3, special issue on Space Robotics, 2007, p.169-186.
[11] J. Biesiadecki, C. Leger, and M. Maimone, "Tradeoffs Between Directed and Autonomous Driving on the Mars Exploration Rovers," The International Journal of Robotics Research, Volume 26, No. 1, 2007, p.91-104.
[12] M. Pivtoraiko, T.M. Howard, I. Nesnas, and A. Kelly, "Field Experiments in Rover Navigation via Model-Based Trajectory Generation and Nonholonomic Motion Planning in State Lattices," in the Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS '08), 2008.
[13] M. Pivtoraiko and A. Kelly, "Differentially Constrained Motion Replanning Using State Lattices with Graduated Fidelity," in the Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008.
[14] T.M. Howard and A. Kelly, "Optimal Rough Terrain Trajectory Generation for Wheeled Mobile Robots," The International Journal of Robotics Research, Volume 26, No. 2, 2007, p. 141-166.
[15] Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. P. How, "Motion planning in complex environments using closed-loop prediction," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008, to appear.
[16] A. Elfes et al. "An Autonomy Architecture for Aerobot Exploration of Titan", in *Proceedings of the 2008 IEEE Aerospace Conference*, Big Sky, MT, 2008.