Enhanced Mars Rover Navigation Techniques

Richard Volpe Tara Estlin Sharon Laubach
Clark Olson J. (Bob) Balaram
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

Abstract

Robust navigation through rocky terrain by small mobile robots is important for maximizing science return from upcoming missions to Mars. We are addressing this problem at multiple levels through the development of intelligent sequencing, sensor constrained path planning, natural terrain visual localization, and real-time state estimation. Each of these techniques will be described, and their complementary aspects discussed. Experimental results are provided from implementation on our Mars rover prototype operating in realistic scenarios.

1 Introduction

Launching in 2003 and again in 2005, NASA's Mars Sample Return (MSR) spacecraft will place two science rovers on the surface of the planet. Each rover will be the primary sample acquisition system for the mission, venturing out from the lander to core rocks and place instruments against them. Mission constraints will restrict these surface operations to less than 90 days, with communication only twice per day. Therefore, autonomous operations are required to image, navigate, sample, and return to the lander. Obviously, the complexity and accuracy of these autonomous operations directly influence the amount of science operations that will be performed.

For this reason, it has been the research objective of the Long Range Science Rover research project to introduce new functionality and features into Mars rovers to enable greater science return from upcoming missions. This paper reports on advances in four areas of rover systems: *dynamic sequence generation, autonomous path planning, visual localization, and state estimation.*

On-board planning with dynamic sequence generation allows much higher level commands to be provided by ground controllers, while increasing the optimality and robustness of rover operations on the surface. For instance, during the Pathfinder Mission, the Sojourner rover [7] was provided extremely detailed sequences on a daily basis, fatiguing operators while disallowing contingency operations when the flow of execution was non-nominal. Contrary to this, we have been experimenting with on-board replanning which can change the execution of daily activities based on unanticipated variations in quantities such as position, terrain, power, and time.

For the longer traverses required of upcoming missions, autonomous path planning is desirable since operators will not be able to see three-dimensional terrain features out to the more distant goal locations. Whereas Sojourner drove a total of 84 meters during its entire mission, MSR rovers will be capable of driving this distance in a single day. But stereo imagery of the terrain provided to operators will only have an envelope of 20 meters at best resolution. Therefore, the path planning advances described here, will allow the rover to be its own operator. It can image the terrain from periscopic cameras, select a path through the terrain to the edge of the effective stereo range, and repeat the process until the goal is achieved.

Visual localization is a technique for using the changing view imaged by the rover to accurately determine its change in position in the environment. It utilizes the same terrain imagery as path planning, but for the purpose of monitoring the apparent motion of three dimensional ground features after the rover has completed a move. In this way, the on-board position estimate of the rover can be updated to compensate for errors caused by wheel slippage or rock bumping. On Pathfinder, this localization functionality was performed manually by operators viewing Sojourner from the fixed position lander cameras, restricting the update to once a day and permitting operations only within the stereo envelope of the lander. In contrast, the terrain-based localization described here can be applied to many forms of landerless operations: incremental long traverses, local operations within range of a prior stereo panorama, localization in descent imagery, and closed chain rover moves with estimate smoothing.

In addition to periodic visual localization of the rover, we have developed real-time position and heading estimation using all other sensors on the vehicle: angular rate, accelerometer, sun sensor, and mobility system linkage (rocker-bogey) configuration. This technique moves far beyond the simple dead-reckoning of Sojourner, and improves upon our previous advances in position estimation with sun sensing [11]. The results aid navigation during path execution, provide better input to localization, and replace both in visually featureless terrain such as sand dunes.

In the next section we describe a typical scenario in which these techniques are employed for a rover long traverse. This is followed by sections describing each of the techniques in greater detail. All have been implemented in our research system, Rocky 7 [12], and experimental validation is

Figure 1: Aerial view of the Arroyo Seco test site.

in progress.

2 Long Range Traverse Scenario

To better understand the utility and sequence of operation for these techniques, this section describes a typical long range traverse scenario. Figure 1 shows an aerial view of the Arroyo Seco next to JPL, where we conduct tests requiring traverses longer than our MarsYard outdoor test facility. This view simulates imagery that will be obtained during lander descent in future Mars missions. It is assumed that the actual landing location is contained in the scene (which is 100 meters, top to bottom). Therefore, from this view operators and scientists may select goal locations that are outside of the stereo ranging envelope of the rover cameras.

We have conducted previous tests in the upper portion of the image, where the goal location, though distant, was visible from the start position of the rover. Ongoing tests use the large hill in the center of the image to obscure the goal from a start location in the lower left corner of the image.

As an aid to the rover, an intermediate waypoint may be selected above and to the left of the hill. In addition, science targets may be specified along the way. When such a list is provided to the sequence planner, it will generate a set of subgoals for the path planner.

Before moving toward each subgoal, mast stereo imagery is obtained and processed to form a partial panorama and elevation map of the terrain in the desired direction, as shown in Figure 2. The elevation map is used to select a path through the local terrain, out to a distance of approximately five meters in the direction of the subgoal. After a path is determined, the rover moves along it using hazard cameras to avoid previously unseen obstacles, while position and heading estimation are used to maintain its course.

Once the envelop of the previous panorama is reached, the rover raises the mast again for more imaging. First, it looks back from its new vantage point to perform a localization operation. Then, it images forward in the direction of the current subgoal, repeating the path planning process. If panoramic imagery cannot discern a path around a large obstacle, the rover will move to the obstacle and begin to follow its boundary until a straight line path is possible.

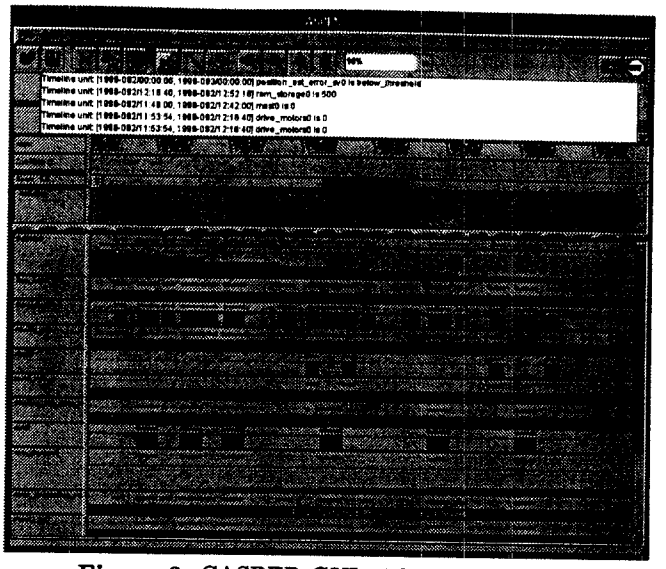

Figure 3: CASPER GUI with example plan.

After reaching the subgoal, the sequence planner provides the rover with the next subgoal. Modification of this subgoal is possible, and some targets may be discarded if constraints dictate. Based on the sequencer's decision, path planning proceeds to the next subgoal it is provided. Details of this sequencer are provided next.

3 Planning and Sequencing

To sequence and prioritize activities during the rover traverse, we have employed a dynamic on-board planning system named CASPER (Continuous Activity Scheduling, Planning, Execution, and Replanning) [2, 3]. CASPER uses techniques from Artificial Intelligence planning and scheduling to generate rover command sequences and to dynamically modify those sequences in response to changing operating context. Generated sequences satisfy the input goals while obeying each of the rover's resource constraints and operations rules. Through dynamic planning, CASPER can autonomously modify planned activities when unexpected or unpredictable events occur. Figure 3 shows an example of a rover plan, displayed in the CASPER graphical user interface.

Plans are produced for the rover by utilizing an iterative repair algorithm [13] which classifies plan conflicts and resolves them by performing one or more plan modifications. Conflicts occur when a plan constraint has been violated, where this constraint could be temporal (e.g., a science activity must occur at a certain time) or involve a rover resource (e.g., the rover has a limited amount of power) or state (e.g., the rover must be at a certain position). If orbital or descent imagery is available, CASPER interacts with a tangent graph path planner [4] to estimate traversal lengths and to determine intermediate waypoints that are needed to navigate around any known obstacles. In addition, optimization heuristics can be utilized within CASPER to improve plan quality. For the scenario discussed below, heuris-

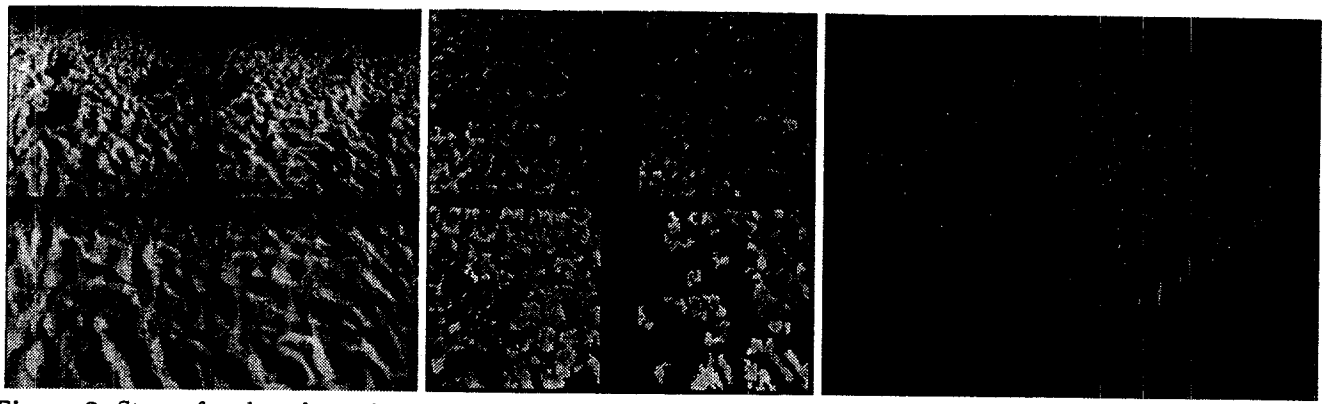

Figure 2: Steps of on-board terrain sensing: panoramic mosaic view from rover mast stereo imager, composite range map extracted from stereo views, and elevation map created from range data.

tics based on the Traveling Salesman Problem were used to reduce overall traversal distance and execution time.

As the rover moves sequentially through the plan, CASPER monitors feedback from the rover sensors and control systems. Examples of this information are: command execution status, current position estimate and its uncertainty, and current power reserves. From these updates, new conflicts and/or opportunities may arise, requiring the planner to re-plan in order to accommodate the unexpected events. For example, an update may cause an immediate plan change. If the wheel slippage has caused the position estimate uncertainty to grow too large, the planner can command the rover to stop and perform localization earlier than originally scheduled. An update may also affect activities scheduled much later in the plan. If a particular traversal has used more battery power than expected, the planner may need to discard one of the remaining science goals. Plan updates can also cause a re-ordering of plan activities.

Figure 4 shows an example scenario in a map, where dark shapes represent obstacles known a priori (e.g. from lander descent imagery). In this case, the initial plan for the traverse will bring the rover to an unexpected obstacle near the first goal, represented as a lightly shaded shape. Circumnavigation around this obstacle will move the rover closer to other goals, triggering CASPER to recognize the situation and re-plan to visit the closest goal first. We are currently evaluating this and similar scenarios experimentally.

4 Path Planning

The on-board path planner for Rocky 7 is the sensor-based *RoverBug* algorithm [5, 6]. This algorithm was developed for vehicles with a limited field-of-view (FOV), and constraints on computational resources. It comprises two operational modes, *motion-to-goal* and *boundary-following*, which interact to ensure global convergence. In addition, a "virtual" submode of *boundary-following* improves efficiency and handles the limited FOV of the vehicle's sensors: stereo camera pairs, mounted on the chassis and on a 1.2 m mast. The visible region ranged by each stereo pair is roughly a wedge of the ground plane, with limited downrange radius R and half-angle α. For the chassis $(R, \alpha) = (1.5m, 45°)$, and for the mast $(7m, 15°)$.

The RoverBug motion planner identifies the minimal number of sensor scans needed to proceed at each step, while specifying the area to scan and avoiding unnecessary rover motion. The planner uses a streamlined local model of convex polygons enveloping detected obstacles. This is renewed at every step to avoid global map maintenance issues. Other than recorded points and parameters, no information is passed between steps. However, the algorithm does require good localization to track the goal position and to determine whether the rover has executed a loop around an obstacle. Therefore, the planner has been paired with an on-board localization algorithm described in Section 5.

The RoverBug algorithm relies upon the construction of a local tangent graph within the visible wedge. The tangent graph consists of all line segments in freespace connecting

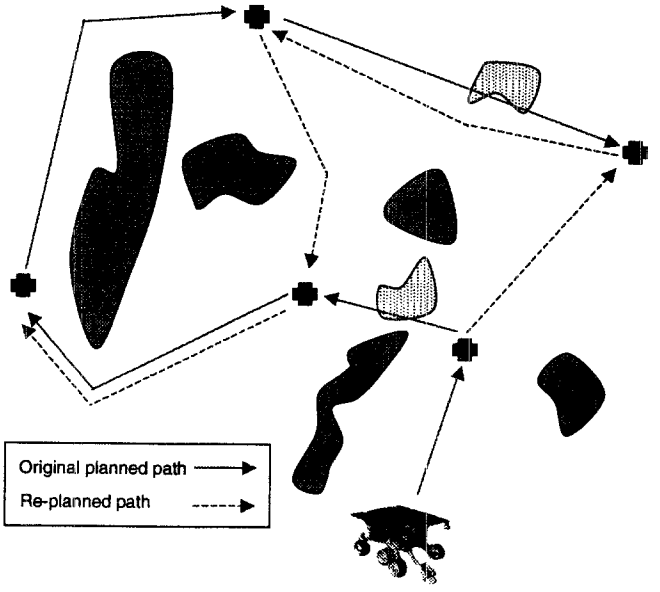

Figure 4: After encountering a previously unknown obstacle shown in light grey, CASPER replans the sequence of targets.

the initial position, the goal, and all obstacle vertices, such that the segments are tangent to any obstacles they encounter. A line l is *tangent* to an obstacle O at a vertex x iff in a neighborhood of x, the interior of O lies entirely on one side of l [4]. The *local tangent graph* (LTG) is the tangent graph restricted to the visible region.

Typically, *motion-to-goal* is the dominant behavior. Its objective is monotonic progress toward the goal, T, along the LTG. After executing the resultant subpath from a given wedge, *motion-to-goal* begins anew. This cycle repeats until either the rover reaches the goal, or no clear path to T exists within the visible region. If the planner detects that the rover cannot make forward progress through the current wedge, the rover must skirt an obstacle to reach the goal. In this case, RoverBug then switches to its *boundary-following* mode.

Upon detecting the obstacle O, it is clear that the rover must circumvent it in order to resume progress toward T. The objective of the *boundary-following* mode is to skirt the boundary of the obstacle, finding shortcuts where possible. Upon first detecting the obstacle, and on each subsequent imaging step along the obstacle boundary, the algorithm "virtually slides" along the boundary using "gaze control", avoiding unnecessary motion toward the obstacle while progressing as far as is locally possible around the border. *Boundary-following* continues until the rover either completes a loop, in which case T is unreachable and the algorithm halts, or the locally visible region contains a new subpath toward the goal. In the latter case, the mode switches back to *motion-to-goal*.

It can be shown that with the two simple operational modes of *motion-to-goal* and *boundary-following*, the RoverBug algorithm is guaranteed to reach the goal or halt if the goal is unreachable [5]. The algorithm completes in finite time, is correct, and produces locally optimal paths (that is, shortest length paths given local knowledge of obstacles). Furthermore, RoverBug deals with the limited FOV of typical all-terrain rovers in a manner which is made efficient by the use of autonomous gaze control to reduce the need to sense and store data.

Experimental data from the use of RoverBug on Rocky7 is shown in Figure 5. The path begins in the lower right corner of the image, toward a goal approximately 21 m distant, in the upper left. Each wedge depicts a rangemap produced from mast imagery, and extends roughly 5m from the imaging position. The obstacles are marked by a black convex hull, and a grey silhouette. Each subpath ends with an apparent side-step in the path, which is a position correction. At these spots, rover localization is performed and the result is used to update the current estimate of the rover position. This localization technique is described next.

5 Localization

Visual localization is performed by Rocky 7 in order to correct errors that have accumulated in the rover position during traverses, as described in Section 6. Localization is accomplished by imaging the terrain nearby the rover and

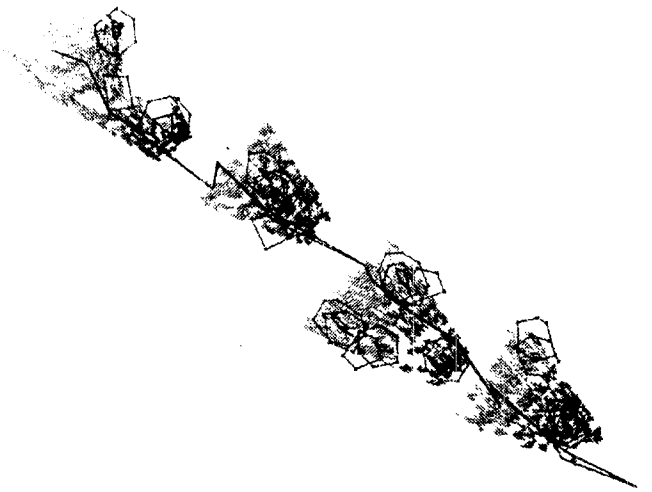

Figure 5: Experimental results from a multi-step run using Roverbug in the JPL MarsYard.

comparing it to a previously generated elevation map. Both terrain maps are generated using stereo vision on-board the rover, as shown in Figure 2.

The problem is formulated in terms of maximum-likelihood estimation, using the distances from the occupied cells in the local map to their closest occupied cells in the global map as measurements. In order to accurately model the sensor uncertainty, we use a probability density function for the measurements that is the weighted sum of two terms representing the cases where the measurement is an inlier (in the sense that the terrain position under consideration in the local map also exists in the global map) or an outlier [9].

The robot position is determined by a multi-resolution search strategy which is guaranteed to locate the optimal position in the discretized search space [10]. The pose space is first discretized at the same resolution as the occupancy grids so that neighboring positions in the pose space move the relative positions of the grids by one grid cell. We then test the nominal position of the robot given by dead-reckoning so that we have an initial position and likelihood to compare against. Next, the pose space is divided into rectilinear cells. Each cell is tested to determine whether it could contain a position that is better than the best position found so far. Cells that cannot be pruned are divided into smaller cells, which are examined recursively. When a cell is reached that contains a single position in the discretized pose space, then this position is tested explicitly.

The uncertainty in the localization is estimated in terms of both the variance of the estimated positions and the probability that a qualitative failure has occurred. Since the likelihood function measures the probability that each position in the pose space is the actual robot position, the uncertainty in the localization is measured by the rate at which the likelihood function falls off from the peak. In addition, subpixel localization is performed in the discretized pose space by fitting a surface to the peak which occurs at the most likely robot position.

Prior to performing localization, the rover analyzes the

Figure 6: Example of target selection for localization.

terrain in the map generated at the initial rover position in order to select a *localization target*. This target is the position in the terrain that the rover images from its final position in order to generate a match against the global map. It is important to select a location that has very distinctive terrain to enable the localization to be performed with the smallest uncertainty.

The localization target is determined by estimating the amount of error present in the map computed at the initial rover position as well as the amount of error that would be generated by imaging the terrain from the final rover position. These errors are encoded in a probability map of the terrain expected to be seen from the final rover position. Each cell in this map contains an estimate of the probability that the cell will be seen as occupied by the rover. By treating this probability map as a terrain map and comparing it to the map generated at the initial rover position, we can predict the uncertainty that will occur in the localization for any target that the rover may view for use in terrain matching. The location with the lowest predicted uncertainty is selected as the localization target.

Figure 6 shows an example of a target selected using Rocky 7. Four stereo pairs were used to build an elevation map of the terrain in front of the rover, toward the goal position which is 5 meters from the rover. A localization target between the current position and goal position was then selected automatically. When viewed from the goal location, localization results with one-percent position error are typical.

6 State Estimation

In between visual localization operations, non-visual methods of state estimation are used. Such methods are important in their own right since they serve as backup to vision-based methods in regions of low visual content and also allow better initialization of the vision methods. This section describes a method to improve the precision of the odometry estimate by using the full kinematics of the rocker-bogey mechanisms of the rover as it traverses undulating, bumpy terrain. The Kalman filtering framework adopted also provides a natural Bayesian means of combining any visually based motion estimates into the full state estimate.

6.1 Rover Model For Estimation

The process model used in the filter is chosen so that the sensor data is used as an input to drive the process equation. This avoids the difficulty of modeling the detailed process dynamics. Since details are discussed elsewhere [1], we only outline some features of the rover contact and kinematics models here.

The contact point vector is modeled very simply as a set of one parameter contacts about the equator of each wheel. In reality there is an additional off-equatorial coordinate for the contact point at each wheel, a contact rotation angle, and two parameters that describe the point on the ground [8]. However the one parameter model suffices to capture and couple the rotational and translational velocities.

Instead of attempting to formulate an explicit inverse kinematics relation for the articulated rover, we choose to embed the easily established forward kinematics within a *constraint* that is treated as a *measurement* in the filter. This exploits the ability of the Kalman filter to perform the appropriate least-squares averaging of the action of each kinematic chain in the rover. Each such forward kinematic chain has a component defined by sequence of links joining the rover frame to each wheel contact point, and a component given by the slip between the wheel and the ground.

We introduce the notion of a slip measurement or constraint, that defines the relative 6-DOF motion of the contact frame on the wheel with respect to the ground. This slip is a function of the vehicle configuration, the 6-DOF vehicle velocity, the wheel-to-ground contact point location, and the joint rates associated with the kinematic chain emanating from the rover frame to the contact point.

The slip constraint measurement can be decomposed into a known deterministic component and a component that is only known in a statistical sense. The deterministic component of the slip, indicated by a non-zero nominal value of the slip, is used to capture the effects of a known steering action or a known average slip rate over different kinds of terrain. The statistically modeled component is due to wheel-ground interaction at each individual wheel. This is a function of the controller (compliant or stiff) being used, and the terrain curvature. We have selected a simple uncorrelated slip model for our early implementations, with the covariance strengths determined by experiments.

6.2 Estimation Experiments

Figure 7 shows the results of a simulation for a rover moving over undulating terrain. Such simulation experiments are valuable since ground-truth is easily established. Note that the tracking of the contact points is quite accurate, although with some lag. By comparing the performance of the kinematic estimator with one based upon dead-reckoning, the improved performance is apparent. Specifically, the averaged wheel odometry obtained by integrating the speed (as indicated at each wheel) results in a 2% error over the actual distance traveled (even when compensated for the instantaneous vehicle pitch angle). Using the kinematic estimator,

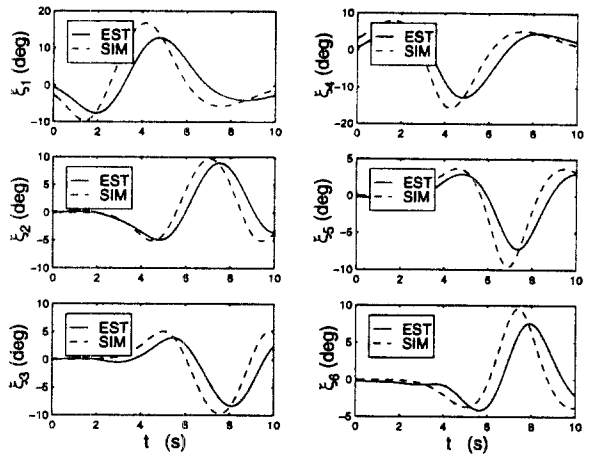

Figure 7: Simulation results showing true values and estimates for wheel contact positions.

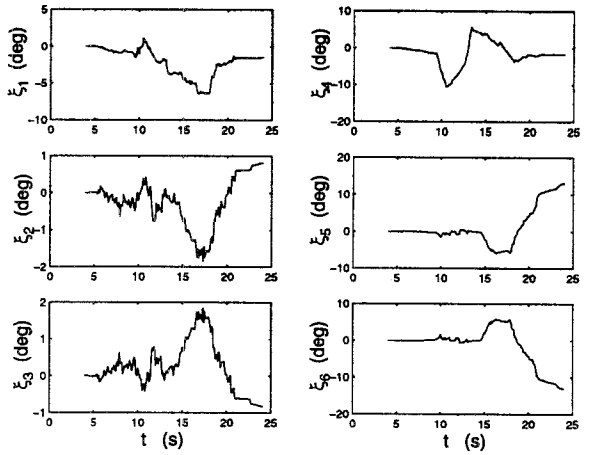

Figure 8: Experimental results from Rocky 7 showing the estimate for wheels contact angles.

the error in distance traveled is then estimated to within the much improved value of 0.3%. This is due to the corrections for the effects of the nonlinear internal kinematics and the variations in contact angles at all of the wheels.

We have also conducted experiments with Rocky 7 traversing an obstacle on the right-side of the vehicle. Estimated contact states are shown in Figure 8. Note that the contact angle variations are quite large under the right wheels as would be expected by the traversal of those wheels over the obstacle. Since the final configuration of the rover is such that the right-side bogey wheels are in the middle of traversing the obstacle, the corresponding contact points are significantly displaced from zero at the end of the motion. However, the contact point for the right front wheel returns to near zero as it proceeds on level ground after climbing over the obstacle. As expected the wheels on the left side of the vehicle experience very little change in contact angles. The estimated position values are within 1 cm of the ground-truth data and the estimated contact angles are within 5 degrees.

7 Summary

This paper has provided an overview of four new techniques developed to enhance the functionality of autonomous long range Mars rover navigation: intelligent sequencing, sensor constrained path planning, natural terrain visual localization, and Kalman Filter state estimation. While integration and tests continue, initial results indicate a dramatic improvement in both knowledge of the rover's position in the terrain and navigation decision making based on that knowledge.

8 Acknowledgments

Large systems like ours require the support of many people beyond the authors. We would like to recognize the following individuals for their invaluable contribution to the work described in this paper: Richard Petras, Darren Mutz, Greg Rabideau, Mark Maimone, and Samad Hayati.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

References

[1] J. Balaram. Kinematic State Estimation for a Mars Rover. *Robotica, Special Issue on Intelligent Autonomous Vehicles*, Accepted for publication, 1999.

[2] S. Chien, R. Knight, R. Sherwood, and G. Rabideau. Integrated Planning and Execution for Autonomous Spacecraft. In *IEEE Aerospace Conference*, Aspen CO, March 1999.

[3] T. Estlin, G. Rabideau, D. Mutz, and S. Chien. Using Continuous Planning Techniques to Coordinate Multiple Rovers. In *IJCAI Workshop on Scheduling and Planning*, Stockholm, Sweden, August 1999.

[4] J. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1991.

[5] S. Laubach. *Theory and Experiments in Autonomous Sensor-Based Motion Planning with Applications for Flight Planetary Microrovers.* PhD thesis, California Institute of Technology, May 1999.

[6] S. Laubach and J. Burdick. An Autonomous Sensor-Based Path-Planner for Planetary Microrovers. In *IEEE International Conference on Robotics and Automation*, Detroit MI, 1999.

[7] J. Matijevic et al. Characterization of the Martian Surface Deposits by the Mars Pathfinder Rover, Sojourner. *Science*, 278:1765–1768, December 5 1997.

[8] D. Montana. The kinematics of contact and grasp. *International Journal of Robotics Research*, 7(3):17–32, 1988.

[9] C. Olson. Subpixel Localization and Uncertainty Estimation Using Occupancy Grids. In *IEEE International Conference on Robotics and Automation*, pages 1987–1992, Detroit, Michigan, May 1999.

[10] C. Olson and L. Matthies. Maximum-likelihood Rover Localization by Matching Range Maps. In *IEEE International Conference on Robotics and Automation*, pages 272–277, Leuven, Belgium, May 1998.

[11] R. Volpe. Navigation Results from Desert Field Tests of the Rocky 7 Mars Rover Prototype. *International Journal of Robotics Research*, 18(7), 1999.

[12] R. Volpe et al. Rocky 7: A Next Generation Mars Rover Prototype. *Journal of Advanced Robotics*, 11(4):341–358, 1997.

[13] M. Zweben, B. Daun, E. Davis, and M. Deale. Scheduling and Rescheduling with Iterative Repair. In J. Zweben and M. Fox, editors, *Intelligent Scheduling*, pages 241–256. Morgan Kaufman, 1994.