Path Planning and Execution Monitoring for a Planetary Rover

Erann Gat Marc G. Slack David P. Miller R.James Firby

Jet Propulsion Laboratory/California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109

ABSTRACT

In order to navigate through natural terrain an autonomous planetary rover must be able to sense its environment, plan and traverse a course through that environment, and react appropriately to unexpected situations as they appear. All this must be done while guiding the vehicle towards the goals that have been given to it by its operators on the Earth. This paper describes research at the Jet Propulsion Laboratory which concentrates on the planning and execution monitoring that must be carried out by the rover to ensure that a safe and efficient path is found and traversed correctly.

1. Introduction

In the next two decades, NASA is planning several missions involving an autonomous rover moving across the surface of a planet and collecting samples. Due to communication delays of up to tens of minutes the rover must be able to navigate autonomously to sites of scientific interest and back to the return vehicle. To do this the rover must be able to plan and execute paths over dozens of kilometers through natural terrain.

The major problem in traversing natural terrain autonomously is that the information available to the vehicle about its environment is necessarily noisy and incomplete. Therefore, the path planning algorithm must be as robust as possible in the face of incomplete and inaccurate data. Furthermore, the vehicle must react appropriately and in real time to unexpected situations which will inevitably arise.

Because of the variety of possible missions (e.g., Lunar site survey, Mars sample return, Mars survey, etc.) the actual rover hardware will probably be different from mission to mission. Our approach tries to standardize the interactions between the planning systems, the rover's sensors, and the rover's mobility system and other subsystems. We are using a general architecture for autonomous rovers consisting of a sensing and perception system, a path planner, an execution monitoring planner, and a plan execution and monitoring system. The system is being tested on two robot testbeds with very different mobility characteristics.

This paper concentrates on the path planner and execution monitoring planner that will enable the rover to safely and correctly navigate to its various destinations while detecting and avoiding hazards. The remainder of this paper is divided into five sections. Section 2 gives an overview of the complete architecture. Section 3 describes the path planner. Section 4 describes the execution monitoring system. Implementation and testbeds are described in section 5. In the final section we summarize the unique aspects of this research.

2. Architecture Overview

In order to support a wide variety of mission scenarios we have developed a general architecture for a semiautonomous navigation (SAN) system. In the SAN approach the vehicle navigates autonomously to destinations which are given to it by human operators. The rover uses a coarse-resolution global terrain map together with local sensor information to plan and execute paths.

The system consists of four major components: a sensing and perception system, a path planner, an execution-monitoring planner, and a vehicle control system. (See figure 1.) In addition there is a system executive which coordinates the operations of the various subsystems, and a vehicle simulator which is used by the path planner and the execution-monitoring planner.

The system operates in a modified sense-plan-execute cycle. At the start of a cycle, the system executive instructs the vehicle's sensing and perception system to construct a model of the terrain surrounding the vehicle. The sensing and perception system takes raw data from the vehicle's sensors and constructs a representation of the local environment [Wilcox87]. It also correlates this local data with a global terrain map obtained from orbital data to more accurately determine the rover's current position. Information from the local and global maps is integrated to construct the final local terrain model which is passed to the path planner.

The local terrain model consists of arrays of equally-spaced data points at various resolutions. Each data point contains the average height and slope of the

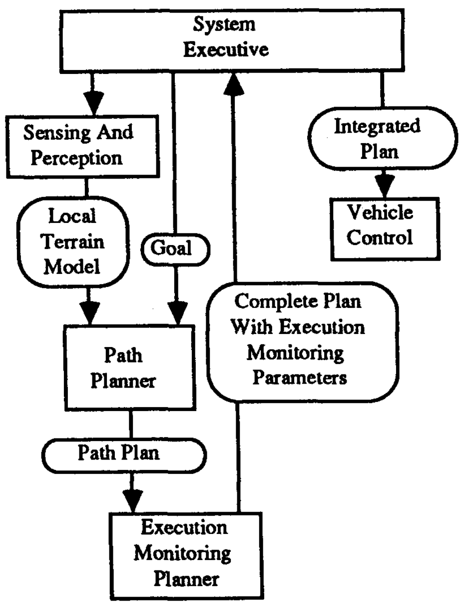

Figure 1. A block diagram of the semi-autonomous navigation system.

corresponding terrain area, as well as a 3x3 covariance matrix which indicates the uncertainties in the height and the slope and provides a measure of how rough the terrain is. The system currently produces three separate maps: a 1 meter resolution map covering a 64x64 meter square, a 0.25 meter map covering a 16x16 meter square, and a 0.125 meter map covering an 8x8 meter square. The arrays are centered at the rover's current position and are aligned to the coordinate frame of the global map.

The local terrain model is then passed to the path planner which plans a local path between five and ten meters long. This path is passed to a vehicle simulator which performs a detailed kinematic simulation of the vehicle as it traverses the planned path, and computes expected values for the rover's sensors over the path. These expected values are passed to the execution monitoring planner which produces execution monitoring profiles. These profiles tell the run-time execution monitoring system which sensors to monitor and when to monitor them. They also include recovery procedures to be executed in the event that something unexpected should occur.

The planned path and the execution monitoring parameters and recovery procedures are integrated by the execution monitoring planner and passed back to the system executive. The system executive checks that the plan conforms to any global constraints that the rover has. If the plan is acceptable, the system executive passes that plan to the vehicle control system which moves the vehicle along the planned path. During the traverse, if an execution monitoring parameter is violated, the vehicle immediately executes the recovery procedure associated with that violation.

When one path segment is complete, the system executive then begins a new cycle with the construction of a fresh local terrain model. The system architecture allows the new terrain model to be constructed during the traverse of a previous segment to allow interleaved operation of the various subsystems and continuous movement of the vehicle, though interleaved operation is not currently implemented.

3. Path Planning

Much work on path planning has been done for mobile robots [Linden87, Brooks82, Chatila85, Chubb87, Crowley85, Elfes87, McDermott84, Miller87, Nguyen84, Slack87, and Thorpe84]. However, most of this work has been done for vehicles designed to travel in structured environments where an area is either traversable or not. For a planetary rover in rough natural terrain, traversability is a continuous quantity. A path plan must take into account what kind of terrain the rover is currently on, what the transitions are like to the adjoining types of terrain, the size of the rocks, and other factors. Additionally the route planner must model non-geometric hazards that cannot be directly detected by the sensing and perception system, but can only be inferred by the geologic context, or the behavior of the rover as it makes its traverse.

Our path planner operates in two phases. In the first phase the rover is given a global terrain map and a goal. The goal is a description not only of the rover's desired location, but also its orientation and articulation state, together with acceptable error bounds. This information collectively is called an objective posture.

The planner begins by generating a global path gradient similar to [Payton88] from the global terrain map, using a spreading activation algorithm like that in [Slack87] and [Thorpe84] which propagates information from the goal to all other locations of the global map. Our algorithm takes into account the kinematic constraints of the vehicle as well as the traversability of the terrain. This is achieved by computing the gradient in two steps. In the first step, a gradient called the "quality gradient" is computed which takes into account only the locations of obstacles and terrain which is difficult to traverse. This gradient is then used as a cost function for computing the actual path gradient by spreading activation from the goal. The propagation of the gradient is constrained according to the vehicle kinematics. The result is a global gradient which conforms to the vehicle kinematic constraints as well as environmental constraints.

The global gradient is used to guide the second phase of planning which is a repeated incremental heuristic search through the local map that will move the robot closer to its goal. The approach exploits the hierarchical structure of the local terrain maps in a way that allows the robot to find local paths very quickly where the terrain is relatively free of obstacles, while retaining the ability to navigate cluttered areas through the use of progressively more refined planning techniques.

The first step in the local path planning process is to find a set of locations along the edge of the low resolution local map that will bring the robot closer to its goal as measured by the magnitude of the global gradient. These locations are called exit zones. A local gradient is then computed for the local terrain map as it was for the global map. The exit zones are then ranked according to the magnitude of the local gradient. A path from the current vehicle position to the top-rated exit zone is then determined by a simple spline-fitting algorithm. This "quick-path" is sent to the vehicle simulator which determines if this is an acceptable path to traverse. The exit zones are checked until an acceptable path is found, or until some fixed number have been checked without finding an acceptable path.

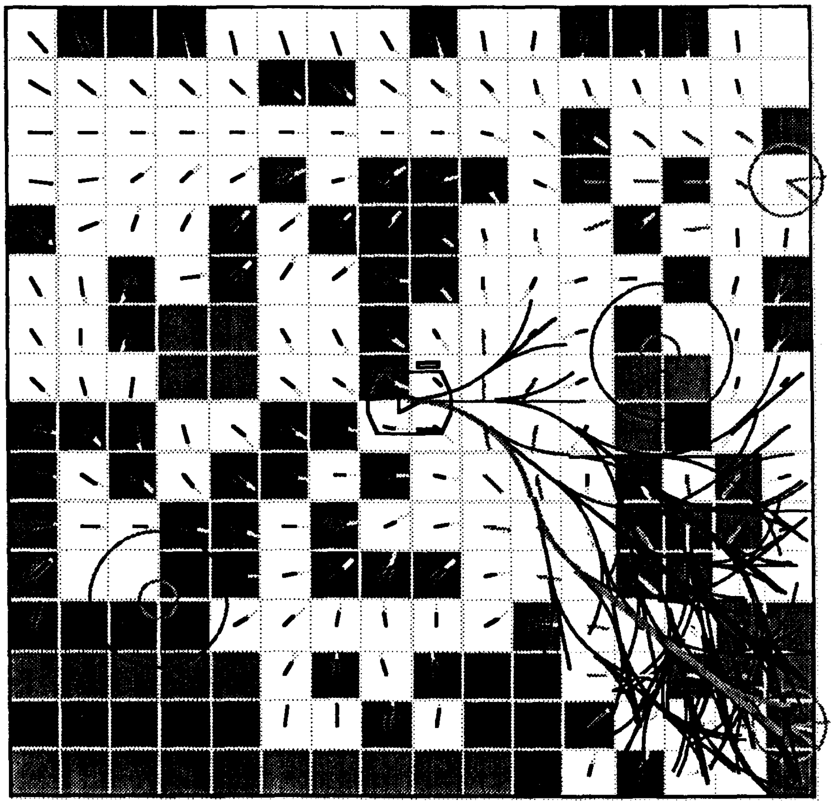

Figure 2: A segment search.

This procedure is very efficient, and will usually yield a path when the terrain is relatively free of obstacles. However, if the terrain is cluttered then this procedure will fail. In this case, a detailed search through a space of short path segment sequences is performed. An example of a segment search is shown in figure 2. The grey areas are obstacles of varying heights. The short line segments indicate the direction of the local gradient. Note how the gradient tends to keep the rover moving in the white clear areas and away from the grey cluttered areas, especially along the top third of the map. The concentric circles are previously encountered non-geometric hazards (i.e. obstacles which were not detected by the remote sensors). The circles with the wedges along the right border are exit zones. The search has just finished finding a path to the lower exit zone, shown in light grey. The dark grey curves show the rest of the search tree.

If the segment search fails, the entire procedure is repeated at the medium-resolution map. In order to set up the exit-zones for the medium-resolution layer the low-resolution layer generates a path gradient (similar to the global path gradient) using the low-resolution exit-zones as its goals. The exit-zones of the medium resolution layer are then computed and the process of finding a path is repeated. If planning is not successful in the medium-resolution layer then the process is again repeated at the high resolution layer. If the search fails at the highest resolution layer then a skid-turn to align the robot to the local path gradient is returned.

If the region is extremely cluttered or full of non-geometric hazards, then the strategy of last resort is to back up some distance along the previously traversed path, recompute the global gradient, and start again from the beginning. This situation is extremely rare, resulting only when the global map is very inaccurate (such as when there are many non-geometric hazards), or when no traversable paths to the goal exist.

Our approach has two interesting features. The first is that the robot's kinematic constraints are taken into account implicitly through the path gradient and explicitly by the segment search. Second, the hierarchical search algorithms result in extremely fast path planning through open terrain. More expensive search techniques must be employed only in areas where there are many geographic features to help constrain the search. Thus the system is able to plan efficiently in almost any terrain.

4. Execution Monitoring and Planning

The execution monitoring system [Gat89] consists of two main components, the execution monitoring planner and the execution monitoring run-time system. The execution monitoring planner uses information generated by the traverse simulator as well as the local terrain model to produce a set of execution monitoring profiles. An execution monitoring profile defines an envelope of acceptable values for one sensor as a function of another sensor. Each envelope consists of a set of minimum and maximum values for a given sensor. These limits may be functions of the value of another sensor.

An execution monitoring profile is shown in figure 3. The expected value for the sensor is the dark line near the center, and the envelope is shown by the lighter lines above and below. This particular profile is for a path segment whose first part traverses terrain with high uncertainty in the local map. Therefore, the range of the expected values is higher for this part of the path.

Associated with each minimum and maximum value is a reflex action which is to be performed in the event that the value of the dependent sensor should violate one of these limits. The reflex action is simply an index into a table of precomputed reflex actions. Thus, the invocation of a reflex action once a parameter violation is detected is virtually instantaneous. The most common reflex is to stop the vehicle and back up far enough that the remote sensors can see the area where the expectation violation occurred. A new local terrain map is then generated and the traverse is attempted anew. If the expectation violation recurs, then the vehicle backs up again, and the area is considered a non-geometric hazard and marked untraversable in the global map.

There are three types of sensors that the execution monitoring system must deal with. First, there are physical sensors which do not require resource scheduling, such as wheel encoders and inclinometers. Their values are available continuously to any subsystem which needs them. Second, there are physical sensors which require resource scheduling such as cameras which must be aimed in the right direction at the right time and which require significant processing before useful information is available from them. Finally, there are virtual sensors which are mathematical functions defined over the values of the physical sensors. For example, there are virtual sensors for the vehicle's absolute spatial location in Cartesian coordinates. These values do not correspond to any physical sensor, but are computed using the values of many different sensors. A virtual sensor may require resource scheduling.

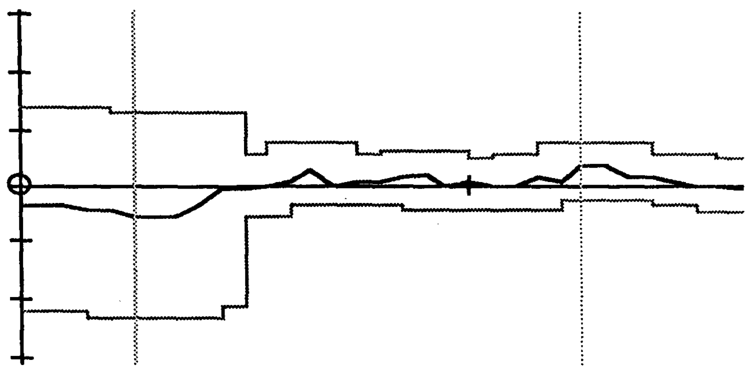

Figure 3: An execution monitoring profile.

The execution monitoring planner uses expected values from the vehicle traverse simulator in order to generate execution monitoring profiles. The traverse simulator takes into account the uncertainty in the local terrain data in order to directly produce expected value ranges for all of the vehicle's physical non-scheduled sensors for every few centimeters along the path. These values are analyzed by the execution monitoring planner in order to construct a set of execution monitoring parameters. The planner selects segments of the path where the expected sensor values are more or less constant and sets the limits on that sensor to a value close to the expected deviations predicted by the simulator. The planner attempts to achieve maximum sensor coverage with a minimum of execution monitoring parameters since the performance of the runtime system can be impaired as the number of parameters grows large.

The execution monitoring run-time system is responsible for the actual monitoring of the vehicle sensors during a traversal. From the point of view of the execution monitoring runtime system, no distinction is made between a physical sensor and a virtual sensor. Thus, very complex interactions among physical sensors may be monitored by simply setting bounds on a single virtual sensor, which allows the runtime system to be very simple and efficient. This is essential in order to achieve the necessary real-time performance.

5. Implementation and Testing

The navigation system described in the previous sections is being implemented and tested on two testbed rovers at the Jet Propulsion Laboratory. For ease of prototyping and testing, parts of the navigation system have been implemented on a modified Cybermation K2A industrial mobile robot. This robot is designed for indoor use, and uses a simulated sensing and perception system. The sensing and perception simulator is given an accurate map of the robot's surroundings. However, the planner has no access to this map. Instead, the sensing and perception system simulator uses this map to construct local terrain maps which are passed to the planner. The simulator adds gaussian noise to the data to make it more realistic. The resulting simulated terrain maps thus have quite similar characteristics to those generated by the actual sensing and perception system.

The indoor testbed has been successfully run using the path planning and execution planning/monitoring software over "rough indoor terrain." This terrain consists of a large open tiled area which is cluttered with obstacles. Some of these obstacles are traversable, e.g. carpeted areas or small books, while other obstacles are impassable, e.g. display cases, walls, and stairways. The testbed has motor encoders, contact sensors, and inclinometers so that the rover can know its position accurately from dead reckoning, its attitude from direct sensing, and whether it has run into anything from its contact sensors. Because of the absence of a vision system, physical objects not modeled in the vision simulator may be used to simulate non-geometric hazards, since the rover cannot sense the obstacle until it can be detected by the inclinometers or the contact sensors.

For experiments in actual rough outdoor terrain an outdoor full-scale navigation testbed has been built that has mobility characteristics similar to that of an actual planetary rover [Miller89]. The outdoor testbed uses a stereo vision system [Gennery77] [Gennery80] for generating actual terrain maps from natural terrain. The outdoor testbed is currently in the final stages of system integration. The computational hardware on both testbeds is compatible.

A trace of an actual run of the indoor testbed is shown in figure 4. The path traversed by the rover starts in the upper left and ends at the lower right. Each small circle is the endpoint of a local path. The irregular section approximately two-thirds of the way along the path is where a non-geometric hazard was encountered. The grey areas are traversable obstacles of various heights, most of which are the result of sensor noise. The black areas are non-traversable obstacles. (In real life these are a number of display cases and a staircase.) The two small hollow rectangles just below the center of the picture are two obstacles which were not included in the simulated terrain maps which serve as non-geometric hazards. The short line segments indicate the direction and magnitude of the

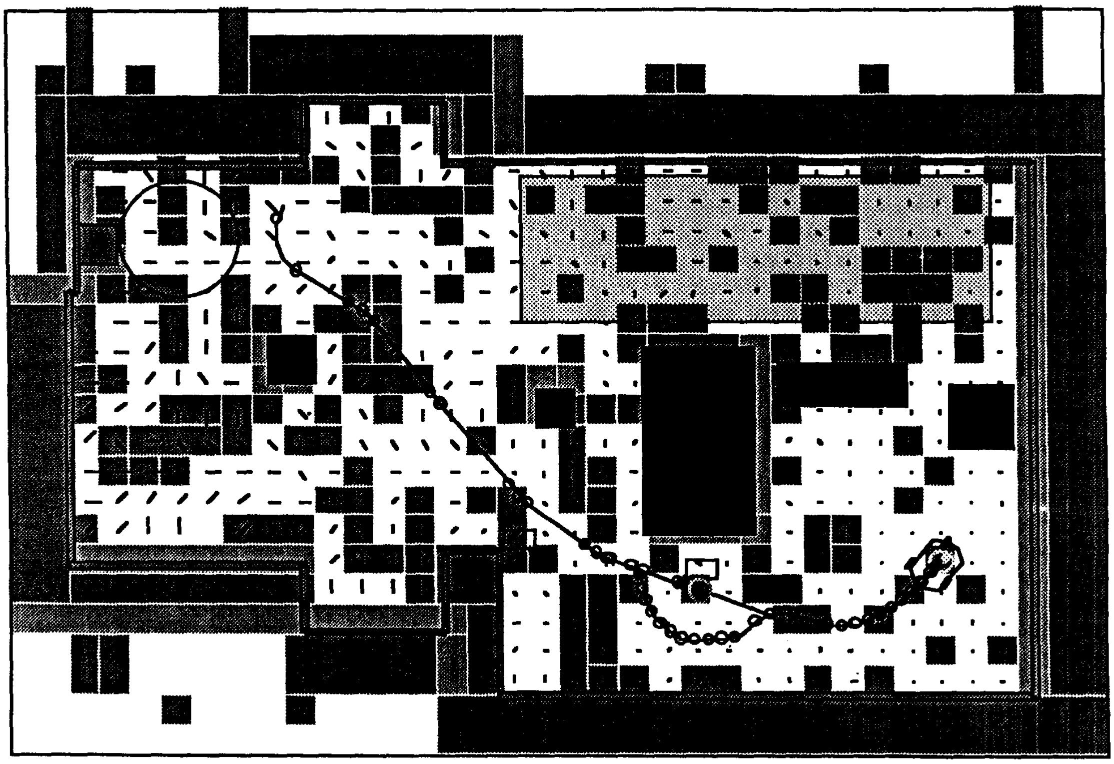

Figure 4: A sample run through an indoor test course.

global gradient. The objective posture (covered by the robot icon) is in the lower right hand corner facing northeast.

6. Conclusions and Future Research

The path planning and execution monitoring software has been successfully demonstrated on the indoor testbed. The robot can plan and execute a path through complex terrain using sensor and actuator interfaces which are essentially identical to those on the outdoor testbed. The robot can detect unforseen obstacles and take appropriate action. This includes having the rover back away from the hazard and mark the area as untraversable in the rover's internal map. Our experiments have consisted of paths roughly twenty meters in length.

Our architecture provides a number of capabilities not demonstrated by other mobile vehicles. It will work with a large variety of rover configurations with different kinematic constraints. The planner finds acceptable paths through natural terrain very efficiently, and the execution monitoring planner allows monitoring limits to be dynamically assigned, providing tight execution monitoring.

Over the next year we hope to add several capabilities to the navigation system. The path planner should be expanded to take into consideration such factors as the vehicle's power budget and the anticipated load bearing strength of the terrain. The execution monitoring system should be expanded to allow it to schedule pointable sensors in order to do landmark-based navigation [Miller86].

We also wish to explore the use of the execution monitoring system for doing accurate sensor-based positioning and reacting. For example, to position the rover a given distance from a rock in order to collect a sample, the execution monitoring planner could point a range sensor at the rock and set up a reflex action to stop the rover when the rock was within the proper range.

Acknowledgements: 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. The authors wish to thank John Loch, Brian Wilcox, Brian Cooper and Andy Mishkin for help in porting the system to the testbeds.

References

[Brooks82] R. A. Brooks, "Solving the find path problem by a good representation of free space," *Proceedings of AAAI 82*, pp. 381-386, 1982.

[Chatila85] R. Chatila, "Position referencing and consistent world modeling for mobile robots," *Proceedings of the International Conference on Robotics and Automation*, pp. 138-145, 1985.

[Chubb87] D. W. Chubb, "An introduction and analysis of a straight line path algorithm for use in binary domains," *Proceedings of the Workshop on Spatial Representation and Multi-Sensor Fusion*, St. Charles, Illinois, pp. 220-229, October 1987.[Crowley85] J. L. Crowley, "Dynamic world modeling for an intelligent mobile robot using rotating ultrasonic ranging device," *Proceedings of the International Conference on Robotics and Automation*, pp 128-135, 1985.

[Elfes87] A. Elfes, "Sonar based real world mapping and navigation," *IEEE Journal of Robotics & Automation*, vol. RA-3, pp. 249-265, 1987.

[Gat89] E. Gat, *et al.*,, "Planning for Execution Monitoring on a Planetary Rover", *Proceedings of the NASA Conference on Spacecraft Operations Autonomy and Roboics*, Houston, Texas., July 1989.

[Gennery80] D. B. Gennery, "Modelling the Environment of an Exploring Vehicle by Means of Stereo Vision," AIM-339 (Computer Science Dept. Report STAN-CS-80-805), Stanford University, (Ph.D. dissertation), 1980.

[Gennery77] D. B. Gennery, "A Stereo Vision System for an Autonomous Vehicle," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pp. 576-582, 1977.

[Linden87] T. Linden, J. Glicksman, "Contingency planning for an autonomous land vehicle," *Proceeding of the 10th International Joint Conference on Artificial Intelligence*, pp 1047-1054, 1987.

[Khatib85] O. Khatib, "Real-time obsticle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, pp. 90-98, 1986.

[McDermott84] D. V. McDermott and E. Davis, "Planning routes through uncertain territory," *Artificial intelligence*, vol. 22, pp. 107-156, 1984.

[Miller86] D. P. Miller, "Scheduling robot sensors for multisensory tasks," *Proceedings of the 1986 Robots West Conference*, Long Beach, California, September 1986.

[Miller87] D. P. Miller and M. G. Slack, "Efficient navigation through dynamic domains," *Proceedings of Workshop on Spatial Representation and Multi-Sensor Fusion*, St. Charles, Illinois, October 1987.

[Miller89] D. P. Miller, *et al.*, "Autonomous navigation and mobility for a planetary rover, " 27th Aerospace Sciences Meeting, AIAA, paper #89-0859, Reno, NV, January 1989.

[Nguyen84] V. Nguyen, "The find-path problem in the plane," MIT AI Technical Report 760, 1984.

[Payton88] D. W. Payton, "Internalized plans: a representation for action resources," *Proceedings of the Workshop on Representation and Learning in an Autonomous Agent*, Lagos, Portugal, November 1988.

[Slack87] M. G. Slack and D. P. Miller, "Path planning through time and space in dynamic domains," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pp 1067-1070, 1987.

[Thorpe84] C. E. Thorpe, "Path relaxation: path planning for a mobile robot," *Proceedings of AAAI*, pp. 318-321, 1984.

[Wilcox87] W. H. Wilcox, et al., "A vision system for a mars rover," *Procedings of SPIE Mobile Robots II*, vol. 852, November 1987.