N90-25548

Planning for Execution Monitoring on a Planetary Rover

Erann Gat R. James Firby David P. Miller

Jet Propulsion Laboratory/California Institute of Technology
M.S. 301-440
4800 Oak Grove Drive
Pasadena, California 91109
USA

Abstract

A planetary rover will be traversing largely unknown and often unknowable terrain. In addition to geometric obstacles such as cliffs, rocks, and holes, it may also have to deal with non-geometric hazards such as soft soil and surface breakthroughs which often cannot be detected until rover is in imminent danger. Therefore, the rover must monitor its progress throughout a traverse, making sure to stay on course and to detect and act on any previously unseen hazards. Its onboard planning system must decide what sensors to monitor, what landmarks to take position readings from, and what actions to take if something should go wrong. This paper describes the planning systems being developed for the Pathfinder Planetary Rover to perform these execution monitoring tasks. This system includes a network of planners to perform path planning, expectation generation, path analysis, sensor and reaction selection, and resource allocation.

1. Introduction

Efforts are currently underway to develop an autonomous mobile robot for the unmanned exploration of planetary surfaces. Such a robot must be able to plan its actions based on sensor data which is inexact and incomplete. Furthermore, there are non-geometric hazards such as dust pits and unstable slopes which cannot be detected reliably with remote sensors. Therefore the robot must possess a robust execution monitoring system which will allow it to detect and recover from unexpected occurrences in real time during path execution. The execution monitoring system described in this paper consists of an integrated architecture that includes a number of different planning systems working together.

There are several issues which must be addressed when designing an execution monitoring system. First, the computational resources available at run time may not be sufficient to constantly monitor all of the vehicle sensors at once. Therefore the system must choose judiciously which sensors to monitor and with what duty cycle to monitor them. The system also must schedule the operation of sensors such as cameras or rangefinders which may require significant amounts of time for aiming and data processing. Ideally, when an unexpected sensor reading is encountered, the system should be able to diagnose the source of the problem and take appropriate corrective action. This must occur in real time as sensor violations could indicate that the vehicle is in imminent danger. The rover must not compute for an hour to decide to back out of a dust pit into which it is sinking. Finally, the use of shared resources during execution monitoring must be coordinated with the other subsystems that use those resources (*e.g.*, cameras might be used by the science subsystem as well as for navigation).

This paper describes an execution monitoring system currently under development which addresses many of these issues. The system is integrated into an autonomous path-planning and execution system which controls a six-wheeled vehicle traversing rough outdoor terrain. Section 2 gives an overview of the entire vehicle control system. Section 3 describes the execution monitoring runtime system which monitors the vehicle during a path traverse. Section 4 describes the execution monitoring planner which produces the execution monitoring profiles that control the runtime system. Section 5 presents an example. Section 6 summarizes.

2. System Overview

In the semiautonomous navigation (SAN) approach which we are investigating, local paths (five to ten meters in length) are planned autonomously using local sensor data obtained by the vehicle. This local path planning is guided by a global route which is planned off-line using a low-resolution topographic map. The global route takes the form of a potential field defined over a region between the rover's starting location and goal [Payton88, Arkin89]. After the local path is generated, it is simulated to generate sensor expectations and appropriate reflexes are set up for execution when a sensor expectation is violated. Finally, the path plan, including expectations and reflexes is made available for execution. The various steps in this process are coordinated by a system executive. A block diagram of the overall system is shown in figure 1.

The system operates in cycles. At the beginning of a cycle, the system executive instructs the vehicle's sensing and perception system to construct a model of the terrain surrounding the vehicle. This model is based on information from stereo cameras, laser rangefinders, and a low-resolution database provided by an orbiting spacecraft. The final local model includes height, slope and roughness information at varying resolutions, and is in a form that is independent of the particular physical sensors used to collect the data [Gennery77, Gennery80, Wilcox87].

The local terrain model is passed to the path planning subsystem along with a goal location from the system executive. The path planner constructs a local path between five and ten meters in length [Miller87, Slack87]. This path is passed to a vehicle simulator which performs a detailed kinematic simulation of the vehicle traversing the planned path. This simulation serves two functions. First, the resulting information can be used by the planner to perform local optimization of the path. This is done by making small changes to the original path and sending it to the simulator again to determine if a more efficient path results [Thorpe84]. Energy to power the rover's motors and computers is a scarce resource so the local optimization continues as long as the energy saved by optimizing the path is more than the energy required to compute the optimizations [Miller89].

The simulator's second function is to produce expected values for all of the physical sensors on the vehicle as it traverses the path. These expected values are used by the execution monitoring planner to construct execution monitoring profiles. These profiles tell the run-time execution monitoring system which sensors to monitor and when to monitor them.

The execution monitoring planner also contains a predictive monitoring system which attempts to identify specific problems which may arise during path execution. When it identifies a potential problem, it inserts a set of monitoring parameters and recovery procedures to detect and deal with the problem should it arise. For example, large areas devoid of rocks may be dust pits. If the rover is about to traverse such an area, the predictive monitor may insert specialized sensor operations into the plan to look for deep dust in that area of the traverse [Linden87, Doyle89].

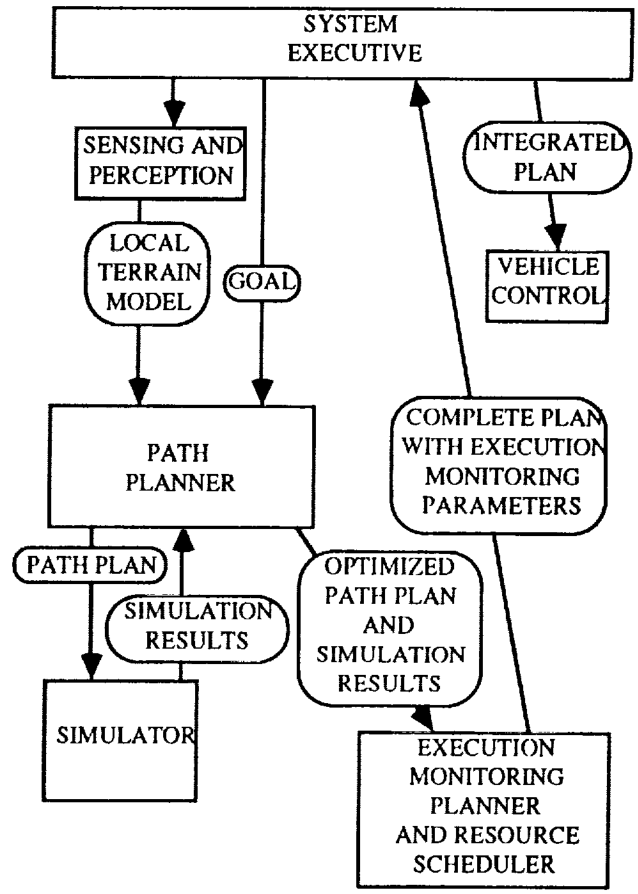

Figure 1: The Execution Monitoring System

The planned path, execution monitoring parameters, and recovery procedures are integrated by the execution monitoring planner into a locally consistent plan using a simple resource scheduler and the result is passed back to the system executive. The system executive checks that this plan conforms to any global constraints that the rover has, such as power limits, shared resource constraints or temporal deadlines. If the plan is acceptable, the system executive passes that plan to the vehicle control system to actually move the vehicle along the planned path. During the traverse, if a sensor reading falls outside of its profile (*i.e.*, an expectation is violated), the vehicle immediately aborts execution of the remainder of the path and executes the recovery procedure associated with that violation (if any).

The system executive then begins a new cycle with the construction of a fresh local terrain model. This may be done during the traverse of a previous path in order to allow interleaved operation of the various subsystems and continuous movement of the vehicle.

3 The Execution Monitoring Runtime System

Vehicle sensors come in three varieties. 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.

From the point of view of the execution monitoring runtime system, no distinction is made between a physical sensor and a virtual sensor. Complex interactions among physical sensors are monitored by setting bounds on a single, specially coded virtual sensor function. Virtual sensors allow the runtime system to be simple and efficient which is essential to achieve real-time performance.

The behavior of the runtime system is defined by a set of execution monitoring profiles computed by the execution monitoring planner. An execution monitoring profile defines an envelope of acceptable values for one sensor, called the dependent sensor, as a function of another, the independent sensor. The envelope is defined by a set of ranges and associated minimum and maximum values for the dependent sensor. The minimum and maximum values specify the limits on the dependent sensor whenever the value of the independent sensor falls in the associated range.

Assigned to each minimum and maximum value is a reflex action to be performed if the value of the dependent sensor should violate one of its limits. The reflex action is simply an index into a table of precomputed reflex actions which can be augmented by the execution monitoring planner. Thus, at runtime, the invocation of a reflex action once a sensor violation is detected can be virtually instantaneous.

By far the most common reflex action is simply to stop the vehicle. However, there are times when this is not appropriate. For example, if the front wheels suddenly start spinning free, and the suspension encoders indicate that those wheels have suddenly dropped, then the front of the rover has probably broken through the surface. If this was not expected, the rover should immediately stop and backup to avoid getting completely mired.

The runtime system can also be used to accurately position the vehicle relative to certain physical landmarks. Suppose the rover needs to position itself one meter from a certain rock in order to collect a sample. This can be accomplished by aiming the rover's rangefinder at the rock and setting up a reflex action to stop the vehicle when the range is one meter. Positioning accuracy can often be significantly improved over simple dead reckoning using such techniques.

4 The Execution Monitoring Planner

The execution monitoring planner uses the local terrain model and information generated by the traverse simulator to produce a set of execution monitoring profiles. These profiles define acceptable ranges for the values of vehicle sensors during the traverse. Whenever the value of a vehicle sensor goes out of the bounds specified by an execution monitoring profile the vehicle immediately executes the reflex action associated with that profile.

The traverse simulator uses the local terrain data, and its uncertainty, to produce expected value ranges for all of the vehicle's physical non-scheduled sensors at points every few centimeters along the path. These values are analyzed by the execution monitoring planner in order to construct a first 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 becomes impaired as the number of parameters grows large.

This initial set of parameters is almost certain to detect a deviation from expected behavior should one occur. However, at runtime, it is very difficult to quickly determine the cause of a problem and decide on an appropriate reflex action using raw physical sensor data alone. Thus, the execution monitoring planner includes a second level of processing to examine the local terrain model and attempt to predict potential problems in the plan. This predictive monitoring system uses a rule-based model of the domain physics which includes information about the likely locations of dust bowls, loose gravel, and other non-geometric hazards. Once the system has identified a potential problem, it finds (or constructs) a virtual sensor, or a set of virtual sensors, to detect that problem specifically and assigns reflexes to handle the problem should it occur.

The predictive monitor also examines the local terrain model for geometric features that it can use as landmarks if special positioning accuracy is required during a traverse. When such landmarks are used, the system generates an execution monitoring profile to check the landmark at strategic points in the traverse, taking into account such things as visibility of the landmark and possibility of confusion with similar nearby landmarks [Chatila85].

All of the execution monitoring parameters generated by these mechanisms are passed to a simple resource scheduler which removes temporal conflicts among shared resources. For example, if many landmarks are to be monitored the traverse plan may have to include delays to allow sensors to be pointed, or there might be more subtle conflicts involving power usage, setup or computation time. In addition, the resource scheduler takes into account some constraints which it may be given by the system executive (*e.g.*, power or time limitations) [Miller86].

Finally, the path description, annotated with the self-consistent execution monitoring profiles, is passed back to the system executive, which then passes it on to the vehicle control subsystem for execution.

5 Example

As an example of the operation of the execution monitoring system, consider the situation depicted in figure 2. The rover path planning subsystem has planned a 10 meter long path that goes between a large rock outcropping to the left of the vehicle and a group of four boulders to the right. The traverse route is mostly flat, with a large open area around the second half of the path. This is passed to the vehicle simulator which generates expected values for the vehicle sensors along the path.

For simplicity we consider only five vehicle sensors in this example, an odometer, an inclinometer, a compass, an elapsed-time clock, and a pointable range finder. From the expected values generated by the simulator, the following execution monitoring parameters could be derived:

Dependent Sensor	Independent Sensor	Range	Min	Max
Inclinometer	Odometer	0 m	-10^{o}	10^{o}
Compass	Odometer	0 m	-45^{o}	10^{o}
Compass	Odometer	2 m	-50^{o}	-40^{o}
Compass	Odometer	4 m	-50^{o}	10^{o}
Compass	Odometer	6 m	-20^{o}	20^{o}
Odometer	Clock	30 sec	9 m	11 m

The first parameter checks the vehicle tilt along the entire path. Since the entire traverse area is fairly flat, all of the inclinometer monitoring is accomplished by a single parameter.

Monitoring the vehicle heading is somewhat more complex. The path is segmented into four pieces. Between 0 and 2 meters the vehicle is turning towards the southeast and so the acceptable range for the heading is quite large. Between 2 and 4 meters the rover travels in more or less a straight line, and so the acceptable range is narrower. There is another transition segment between 4 and 6 meters, and another straight segment between 6 and 10 meters.

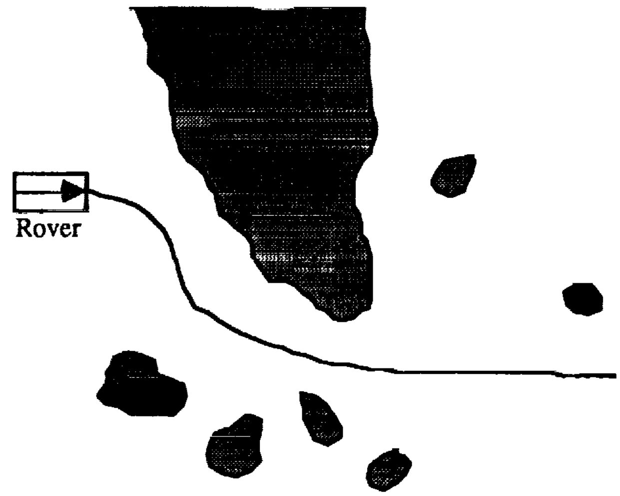

Figure 2: A 10 meter path.

The final execution monitoring parameter states that the path must be nearing completion before 30 seconds have elapsed. On the actual vehicle there would be wheel slip sensors which could detect lack of progress long before the end of the path.

These parameters represent the simplest sort of analysis that can be performed on the simulation data: the sensor values are simply analyzed for segments where the values all fall within a certain range. This sort of analysis works well when sensor values are constants, but often creates transition regions where sensor values are not closely monitored, such as the path segments where the vehicle heading is changing. In these cases, the execution monitoring planner could construct a virtual sensor which compared, say, the vehicle heading to the odometer reading (normalized to the start of the transition region) and set up an execution monitoring parameter which monitored the ratio of these two values. Similar correlations can allow nearly every sort of sensor value transition to be monitored as closely as necessary.

Finally, the predictive monitor could insert a number of execution monitoring parameters in this situation. It might, for example, schedule a range reading off the rock outcropping on the left of the vehicle just before the vehicle entered the area between the rocks. This would ensure that the vehicle was not in danger of colliding with a rock as a result of dead reckoning errors. The system might also notice that the large open area towards the end of the path could be a dust bowl, and insert more checks on vehicle articulation. The operation of the predictive monitor is highly heuristic and is based strongly on domain-dependent issues which will be the subject of future research.

6 Summary

An autonomous planetary rover needs a robust execution monitoring system to detect and recover from unexpected occurrences in real time. A system which addresses these goals is currently under development and the Jet Propulsion Laboratory.

The system has two major components, an execution monitoring planner and a runtime system. The runtime system is very simple, which allows it to respond to situations in real time. All of the complex computations are done by the execution monitoring planner before execution begins.

The execution monitoring planner produces execution monitoring profiles which describe acceptable limits on the values of the vehicle's sensors at various stages during the traverse. Vehicle sensors may be actual physical sensors, or they may be virtual sensors which are simply mathematical functions defined over the values of the physical sensors. This allows complex aspects of the vehicle's performance to be monitored efficiently.

The execution monitoring planner derives profiles from two sources. The first is a vehicle traverse simulator which computes the expected values and variances for all of the vehicle's physical sensors at a series of points throughout the traverse. The second is a predictive monitoring system which anticipates potential problems and inserts explicit checks and recovery procedures for those problems.

All of the execution monitoring parameters are passed through a task scheduler to remove conflicts among shared resources. The final, self-consistent traverse plan is sent to the rover's system executive which fits the plan into the vehicle's global plan. If the plan is acceptable, it is sent to the vehicle control subsystem for execution.

During execution the runtime system checks the values of the vehicle sensors against the limits imposed by the execution monitoring profiles. If these limits are violated, the remainder of the path traverse is aborted and a reflex action associated with the violated profile is executed.

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.

References

[Arkin89] Arkin, R., C., Three-Dimensional Motor Schema Based Navigation, in the *Proceedings of the NASA Conference on Space Telerobotics*, Pasadena, California, Jan31-Feb2, 1989.

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

[Doyle89] Doyle, R., S.Sellers and D.Atkinson, Enhancing Aerospace Systems Autonomy Through Predictive Monitoring, *27th Aerospace Sciences Meeting*, AIAA, paper #89-0107, Reno, NV, January 1989.

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

[Gennery80] Gennery, D.B., *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.

[Linden87] Linden, T. & Glicksman, J., Contingency Planning for an Autonomous Land vehicle, in *Proceeding of the 10th IJCAI, AAAI, IJCAI*, Milano, Italy, pp 1047-1054, 1987.

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

[Miller87] Miller, D., P., Slack, M., G., Efficient Navigation Through Dynamic Domains, in the *Proceedings of Workshop on Spatial Representation and Multi-Sensor Fusion*, St. Charles Illinois, October 1987.

[Miller89] Miller,D.P., A.Mishkin, K.Lambert, D.Bickler, and D.Bernard, Autonomous Navigation and Mobility for a Planetary Rover, *27th Aerospace Sciences Meeting*, AIAA, paper #89-0859, Reno, NV, January 1989.

[Payton88] Payton, D., W., Internalized Plans: A Representation for Action Resources, in the *Proceedings of the Workshop on Representation and Learning in an Autonomous Agent*, Lagos, Portugal, November, 1988.

[Slack87] Slack, M. G., Miller D. P., Path Planning Through Time and Space in Dynamic Domains, in *Proceeding of the 10th IJCAI, AAAI, IJCAI*, Milano, Italy, pp 1067-1070, 1987.

[Thorpe84] Thorpe, C. E., Path Relaxation: Path Planning for a Mobile Robot, in *Proceedings of AAAI*, AAAI, pp. 318-321, 1984.

[Wilcox87] Wilcox, W.H., D.B. Gennery, A.H. Mishkin, B.C. Cooper, T.B. Lawton, N.K. Lay, and S.P. Katzmann, A Vision System for a Mars Rover, *Proc. SPIE Mobile Robots II*, vol. 852, 1987.