THE "MITY" MICRO-ROVER: SENSING, CONTROL, AND OPERATION

Eric Malafeew* and William Kaliardos†
The Massachusetts Institute of Technology and C. S. Draper Laboratory
Cambridge, Massachusetts

Abstract

The paper discusses the sensory, control, and operation systems of the "MITy" Mars micro-rover. Its compact and low-power sensor suite, with customized sun-tracker and laser rangefinder, provides dead reckoning and hazard detection in unstructured environments without aid from external sources. A high-level task architecture supports mapping, obstacle avoidance, GN&C, mission monitoring, and ground telemetry with high processing efficiency. For cluttered environments, reactive obstacle avoidance chooses the clearest trajectories with non-holonomic steering constraints and passing margin tradeoffs. Wireless operator interactions range from troubleshooting and reprogramming to graphical monitoring and supervisory control of the robot. The micro-rover system has been simulated in Monte-Carlo trials and field tested in various environments. Continuing work focuses on space qualification of the sensors and control software and further implementation of the ground station.

1. Background

1.1 The MITy Mission

MIT and Draper Lab-sponsored development of a low-cost Mars micro-rover began in 1990 and has since involved seventeen graduate and undergraduate students. This project supports the NASA MESUR objective of landing a micro-rover on Mars to scout and perform experiments on the environment. The "MITy" project goal has been to develop prototypes for this mission, which imposes strict constraints on size and mass, aid from external sources, and modes of operator interaction. The proposed operation scenario was to receive a set of destination commands from Earth daily, arrive at each to perform and record an experiment, then transmit the results back to Earth on the next cycle. Destinations would be chosen from rover video images and satellite maps. The current design scenario also allows supervisory monitoring and control for terrestrial and inner-space applications, in which communications are less limited.

Present contributions of the MITy project include studies on system requirements, design for mobility, soil-wheel interactions in low gravity, and most recently the sensory, control, and operation systems in this report [1,2,3,4,5]. Current team developments include system qualification procedures with JPL, vision for navigation and hazard detection, refinements of mobility and packaging, 3D simulation and animation, and robotic manipulator construction.

1.2 System Components

The second of three prototype platforms, MITy-2, is depicted in Figure 1. Its sensor subsystem is described in detail in Section 2.

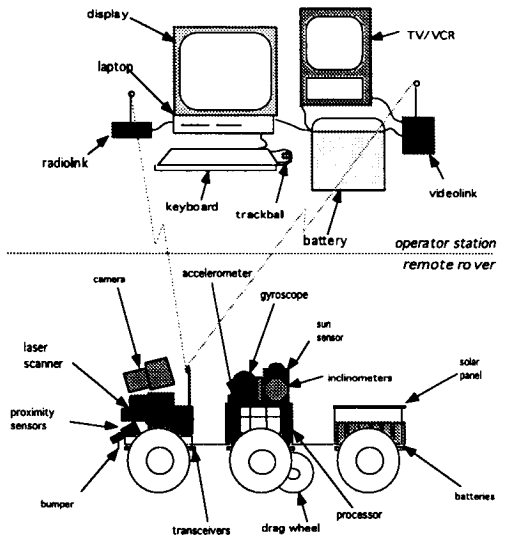


Figure 1: MITy System

Rover Breakdown

The micro-rover structure consists of three 15 x 15 cm articulated platforms, connected by a dual-spring suspension, that support sensing, processing, and payload subsystems. Six independently driven wheels enable the rover to climb steps up to 15 cm high and drive at speeds up to 30 cm/s on flat ground. The front and rear wheels are independently steered to permit a 63 cm minimum turn radius. Power is provided by a 30 amp-hr capacity battery, which can be recharged by a solar panel at a maximum rate of 6 W. The overall dimensions of the robot are 46 x 75 x 28 cm, and its total weight is about 9 kg.

*Graduate Student, Mechanical Engineering
†Graduate Student, Aerospace Engineering

The processing subsystem consists of a Z-180 based micro-controller, with a throughput of about 12 Kflops/s and 512 Kb for code and data storage. Customized drivers generate motor and communications signals for a high level control interface. Software development supports "C" and remote debugging.

The current payload on the rover is a CCD camera, which sends images over a video transmitter to the operator station. Data communication occurs over a 9600 baud radiomodem with a LOS range of 2 miles. An optional LCD/keypad, not displayed, provides much of the operator station functionality for testing purposes.

Operator Station Breakdown

The main component of the operator station is a 486/66 PC-clone laptop, which runs the graphical user interface and programming environment. Operator devices are the computer display, TV/VCR, keyboard and trackball; communication devices are the radiomodem and video receiver. The small TV/VCR is useful for real-time supervisory control. These components are powered a 12 V car battery, which makes the entire operation station portable for testing and relocation.

1.3 Related Systems

Other notable Mars micro-rovers in development include JPL's Rocky, RATLER from Sandia, and Genghis from ISX Robotics, which differ largely in terms of system capabilities and realization. For instance, MITy is geared toward the extended MESUR mission beyond 10 m of the lander, while Rocky has been designed for operation within 10 m [6]. Larger rovers include CMU's Ambler and the Mars Rover from JPL.

2. Sensory System

2.1 Sensory Objectives

The sensor suite on the rover is required to provide sufficient navigation and hazard information to the control system for autonomous transit [1]. In particular, the subsystem requirements state that the rover position and attitude $\{x,y,z,\theta,\varphi,\zeta\}$ from the landing location should be determinable to within 10% of traveled distance. The hazard requirement states that the rover should be able to detect and localize potential hazards between the range of 1-10 ft, and within 1 ft be able to sense all types of accountable hazards. At the micro-rover scale these hazards include rocks, craters, and steep grades. Sensor selection, development, and placement are features of this problem.

Constraints that apply to the sensory system are to minimize power consumption, size and mass, and prototype cost. The sensory system should not rely on external sources and should be operable in the negligible atmosphere and magnetic field of space. Redundant sensors are also desirable in case of failures.

2.2 Navigation Sensors

By keeping within the 10% navigational error bound, the video image of the landscape can be compared with that from the previous day, reducing cumulative position error over multiple days. The video image plays an important role in navigation, since selection of the daily goals as well as position calibrations are based entirely on this data, as analyzed by the ground station [7]. The result is that the performance requirements of the rover navigation sensors are greatly relaxed, reducing cost, power, size, and complexity.

Dead reckoning is used to navigate to the stated accuracy, since this does not require an existing infrastructure such as land beacons or GPS satellites. Rather than using a costly inertial navigation package, the dead reckoning sensors are divided into three types: longitudinal translation, heading angle, and inclination.

Translation

Longitudinal translation is measured with the powered drive wheels of the rover, as well as an unpowered drag wheel. Drive wheel rotation is sensed with motor tachometers, which are supplied for speed control. An optical encoder is used to sense the drag wheel rotation. Both drag and drive wheel sensing is used to accommodate the potentially large variety of terrain. A drag wheel is beneficial when the powered wheels slip on soft surfaces, or when the vehicle wheels becomes partially unloaded due to rocks underneath its belly. However, since the drag wheel is smaller for packaging reasons it is less accurate than the drive wheels over rough terrains.

Heading Angle

Heading angle sensors include a sun sensor and an inertial angular rate sensor, hereafter called the gyro. The gyro is useful regardless of environment conditions, but at a cost of unbounded error growth with time. The sun sensor is used for dynamic calibration when shading is not an issue.

The selected rate gyro is a low-cost silicon vibrating beam sensor made primarily for automotive applications. The rate signal is integrated with a low-leakage analog circuit to provide a voltage signal that is

proportional to heading angle. Bias error in the gyro and integrating circuit is measured at stopping points for substraction in software.

The sun sensor was designed and constructed at CSDL, due to its unique requirements of a hemispherical field of view, small size, non-scanning, and low cost. To achieve these characteristics, a two-axis position sensitive detector (PSD) measures the position of a spot of light focused from a small fisheye lens. The position of this light spot is a function of the sun elevation (ε) and azimuth (β), as well as the inclination of the rover, as shown in Figure 2. Simple electronics are required to obtain light spot position from the PSD currents, making the entire sun sensor assembly a small rugged device that provides heading calibrations to within 0.5°.

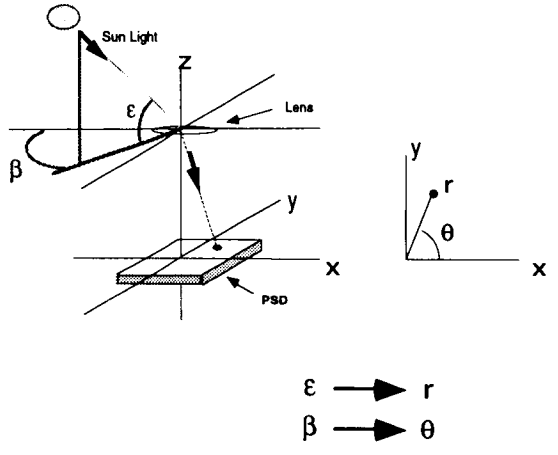


Figure 2: Sun Angle Sensor

Inclination

Two accelerometers provide tilt information for sun sensor calibration and propagation translation measurements out of the ground plane. These sensors are described in more detail below.

2.3 Hazard Sensors

One of the most difficult challenges of a micro-rover is sensing mobility hazards in an unstructured environment. The primary sensing problem is detecting objects for collision avoidance, although other mobility hazards exist.

Adequate sensing for a mobile robot typically requires a detailed depth map of the local terrain in all directions. The combination of high resolution and large field of view results in a large amount of data that requires much processing. Even if such processing were available, collecting the data with a small sensor is difficult. The MITy design presents a simplified hazard-avoidance sensor arrangement that meets its packaging requirements, and provides limited but sufficient sensing capability for autonomy. It is composed of a single axis scanning laser range finder, short distance proximity sensors, bump switches, and inclinometers.

Range Finder

Like the sun sensor, the range finder was designed and developed at CSDL in order to meet the unique needs of the micro-rover. For size and simplicity reasons, it works on active triangulation principles, illustrated in Figure 3. A solid-state laser produces a collimated beam in the near infrared. Light is diffusely reflected from the target, and a portion of this weak signal is collected by a small lens and focused to a spot on a 1-axis position sensitive detector (PSD), similar to that used on the sun sensor. Since the receiving lens is located a small distance from the laser, the angle of incidence of reflected light varies with range. Through triangulation, range can be calculated based on the reflected light angle, which is measured in the form of light spot displacement at the PSD.

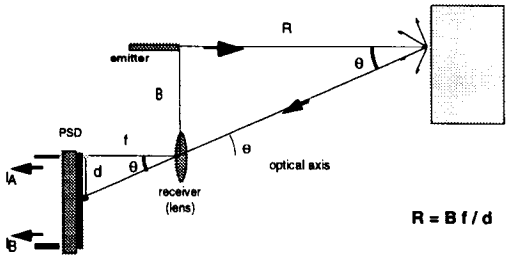


Figure 3: Triangulation Rangefinding

With simple electronics for transmitting and detecting, the complete range finder remained small, and yet ranges out to 3 m with 10% accuracy, which was specified from control system simulations [4]. The range finder is mounted to the front platform, and scans 180° in front of the rover at a height of 15 cm on flat ground. This plane-of-view approach is limited but provides adequate collision avoidance capability when used with other sensors.

Inclinometers

Sensing the tilt of a given rover platform is important for both navigation and hazard-avoidance reasons. The latter requires inclinometer data to calculate the orientation of the range finder beam, and to estimate the terrain geometry.

Accelerometers are used to sense the component of gravity when tilted, rather than the bulkier bubble level sensors. Miniature non-inertial grade accelerometers are

relatively small and faster, allowing pairs for pitch and roll on the front and middle platforms.

Proximity Sensors

Because of the planar range finding, vertical drops are sensed through two short-range proximity sensors. These are used on the front platform along with inclinometer data to estimate when the rover is partially over the edge of a cliff. The rover can then travel in reverse, using the high traction of the four rearward wheels.

Collision Sensors

Contact switches are located at the front and rear edge of the rover to detect collisions above a certain force. In addition, the odometry system can provide collision detection by observing front wheel speeds in response to commanded torques.

2.4 Implementation

The locations of individual sensors on the micro-rover are shown in Figure 1. All of these sensors are relatively low-performance, which allows them to be small, light, low-cost, and rugged. The total volume, mass, and power for the sensing system are about 2000 cm^3, 5 W, and 0.5 kg respectively, with a prototype cost of under $5 K. Rather than concentrating on fewer sensors that are high performance, this arrangement accomplishes similar tasks through sensor fusion in software, and additionally provides a certain degree of redundancy, increasing the system reliability.

3. Control System

3.1 Control Objectives

The semi-autonomous control problem is transit between a sequence of coordinate locations, without position or environment information from *a priori* or external sources. In addition, with sufficient communication rates, supervisory control should permit real-time operator interaction with sub-tasks, such as guidance and mapping. The robot should recognize traps that confuse its control logic, and be able to be recovered by the operator under supervisory mode.

Mobility, sensing, and processing constraints affect control system design. Mobility constraints include the minimum turn radius, overall dimensions, and climbing ability. Range, accuracy, and throughput of sensors must be recognized by the design, as well as processing throughput and memory.

3.2 Functional Breakdown

This control objective precipitates a variety of concurrent tasks, including mapping and obstacle avoidance, GN&C, mission monitoring, and ground telemetry. A functional breakdown of the autonomous transit mission is depicted in Figure 4. Boxed *tasks* perform an atomic function, which improves modularity and lessens code redundancy, utilizing other tasks and circled *resources*. *Map* and *Avoidance* tasks are related to trajectory generation. GN&C encompasses the *Guidance*, *Nav*, *Speed*, and *Steer* tasks. Lastly, *Sequence*, *Collision*, *Failure*, and *User* tasks comprise mission monitoring. The diagram also shows information transfer between tasks and resources. Task groups and their implementations are described below.

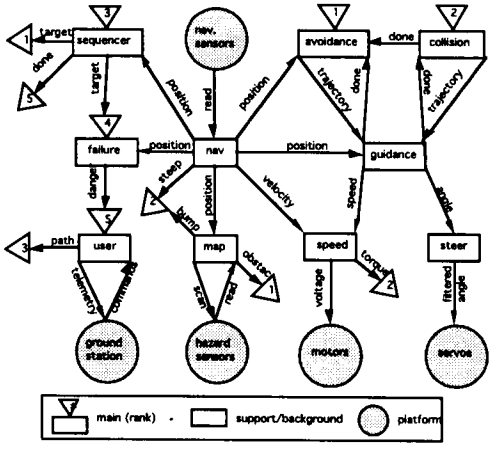


Figure 4: Functional Architecture

3.3 Task Architecture

The task architecture determines the scheduling of tasks and communications between them. The MITy architecture was mostly influenced by Payton's reflexive control ideas, which were implemented on DARPA's ALV [8]. Tasks are divided functionally and by cycle rates. Ideally each task would run concurrently, but this requires customized low-level scheduling and interprocess communications on a single processor. Instead, tasks are broken down into fast executing steps, which are interleaved by the *task planner* according to their intervals and priorities. The three types of tasks are *main*, *support*, and *background*; the type determines how a task is handled by the planner. Main tasks compete for motion control, support tasks aid main tasks, and background tasks aid all tasks. Information is communicated between tasks through a global variable pool.

All tasks are divided into *perceptions* and *reactions*. The planner decides which reactions to run based on the

"truth" of their perceptions. Any condition may exist in a perception, but every task has a *flag* and *interval*, which are set by the user to determine if and how often a reaction should be called. Background tasks have only these two conditions; support tasks will run only if their associated main task is running. Only one main task may run at a given time, which determines its *rank* as it executes. Main task perceptions have exclusive and positive *priorities* which are compared to the current reaction rank when true. A higher priority will subsume a lower rank, else the current reaction will continue without interruption. When a main task reaction ends without interruption, its rank is reduced to zero to allow the next highest priority main task to run.

This high level method is more efficient for the particular task designs than common real-time schedulers, which interleave tasks at the operating system level. The MITy task planner supports both pre-emption and concurrent execution at step resolutions, and does so without the overhead required for low-level context switching. Also, schedulers are not standardized and often unavailable for many processing platforms, which would lessen portability of the control code to future robots and simulations.

3.4 Trajectory Generation

Trajectory generation tasks produce heading and speed commands that maneuver the robot around obstacles toward the current target.

Mapping

The mapping function is a support task for obstacle avoidance. It sweeps the laser scanner 180° in 10° intervals, sampling the rangefinder at each stop. A reading is considered valid if its intensity is sufficient and the laser is not aimed at the ground. The last two sweeps of data are stored in a circular list, whose elements correspond to particular scanning directions. Elements consist of the coordinate endpoint of the last valid laser reading in that direction. This storage method constantly refreshes the obstacle list while maintaining complete coverage for asynchronous obstacle avoidance.

Obstacle Avoidance

The obstacle avoidance routine makes intelligent use of robot-centered laser information to maneuver through cluttered environments. It is the lowest priority main task--it is active in the absence of emergencies and consumes available processing time between other tasks. The general philosophy is borrowed from the VFH method [9], which is reactive in nature in that it does not plan ahead and must cycle quickly relative to robot motion. It has been shown more stable and predictable than methods based on artificial potential fields.

In the MITy approach, represented in Figure 5, polar map creation and trajectory selection are quite different from VFH. The independent axis of the polar map represents trajectory headings tangent to the minimum turn radius of the robot, while the dependent axis shows the "free distance," which is initially calculated as the distance to the first collision in all directions. The robot model is a circle that includes both its physical dimensions and the obstacle uncertainty due to laser and scanner accuracy. Free distance is then linearly traded off with safety, target heading error, and momentum heading error to obtain the weighted polar map. Safety is defined as the minimum distance to an obstacle in passing, which tends to guide the robot through the centers of clearances rather than narrowly on a side. The best trajectory is chosen in the direction of the highest free distance after weighting.

In constrast, the VFH method considers vehicle size at only one range, does not contend with steering constraints, and does not trade off safety and target error with free distance. It is therefore better suited to omnidirectional vehicles operating with a short range of concern.

The routine uses prediction and error correction to keep the robot in motion between trajectory updates. The nominal radius of concern and trajectory speed are respectively 3 m and 8 cm/s. If the robot sees no way out of a situation by moving forward, it produces a dead-end signal for collision recovery. In proximity to the target, the search radius is limited to prevent avoidance of obstacles behind it.

3.5 GN&C Module

Guidance, navigation, and control tasks concertedly command the drive and steering motors to follow a given trajectory.

Guidance

The guidance routine commands steering angles to meet a desired heading, using a proportional filter on heading error. It also stops the robot if it has not received a trajectory update after a fixed distance. Guidance supports both obstacle avoidance and collision recovery, depending on which produces trajectory commands.

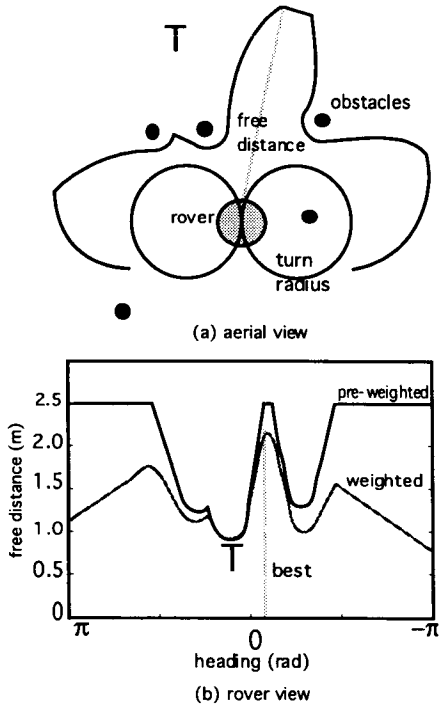


Figure 5: Obstacle Avoidance

Navigation

Navigation integrates heading, rate, and inclination information to update $\{x,y,z,\theta,\varphi,\zeta\}$ of the rover. Redundant sensors are arbitrated rather than averaged; the primary sensor is used unless disqualified, either by its own or other sensor readings. The sun sensor outweighs the gyroscope with a non-zenith sun, and the drag wheel outweighs tachometers over smooth terrain.

Control

Drive and steering control loops are separate but cooperative background tasks. Speed control uses tachometer readings and an anti-windup integral filter to generate motor voltages. Voltage commands to individual wheels are offset in accordance with steering rates and angles, to help aid steering and minimize slippage. Prohibitive torques, calculated from applied voltage and measured speed, are reported as collisions. The servo loop fixes steering and scanning rates and bounds to prevent power surges and motor damage.

3.6 Mission Monitoring

Mission monitoring tasks react to hazard collisions, monitor targets for completion or failure, and report progress to the ground station.

Collision Recovery

Rough terrain and reflective obstacles can cause the laser rangefinder to fail and may result in obstacle collisions. Collisions types are actual bumps, steep grades, deep craters, stuck wheels, and dead ends. When these occur, the collision task overrides obstacle avoidance to back the rover along its entry path, using a "look-ahead" path following method based on [10]. It then notes the collision location in the obstacle map for continued obstacle avoidance. If a collision is detected in reverse, the rover stops without updating the map and continues forward immediately.

Target Monitoring

Two functions are performed by target monitoring: sequencing and failure detection. When the rover reaches a target, determined by an estimated passing distance less than the minimum turn radius, the sequencing task overrides both collision and obstacle avoidance to perform a desired experiment. It then advances the target counter until the last target is reached. A target is considered failed if the rover does not advance on it or escape a given radius after a given amount of travel, as prescribed by [11]. Failure results in mission stoppage to conserve energy until operator intervention.

Telemetry

The telemetry function can operate in either the background or foreground. In background mode, it periodically sends position, obstacle, and free distance updates to the operator station, while servicing non-destructive sensor sampling requests and modifications to targets and parameters. Alternatively, the user may override semi-autonomous mode to set trajectories directly in supervisory mode, in which case telemetry is the highest priority foreground task.

3.7 Implementation

The control code was developed in standard "C." It is easily ported between the robot and simulation (described in Section 5) by replacing stump I/O functions.

Control parameters were initially determined from Monte-Carlo simulations and fine tuned on the actual robot. Obstacle avoidance throughput was traded off with background task cycle rates and obstacle mapping resolution and memory. As a result, most background tasks cycle at 10 Hz, while obstacle avoidance repeats at 1 Hz.

4. Operator Station

4.1 Operation Objectives

The capabilities of the operation station should include graphical monitoring and intervention for supervisory control, on-the-fly troubleshooting and reprogramming for semi-autonomous control, and data logging and replay for post-mission analysis of the rover. Communication bandwidth, time delay, interference, and range are restrictions on operation effectiveness.

4.2 Real-Time Display

A typical operator screen is shown in Figure 6, composed of a graphical window and text interaction areas. The text buttons emulate the optional keypad interface to the robot.

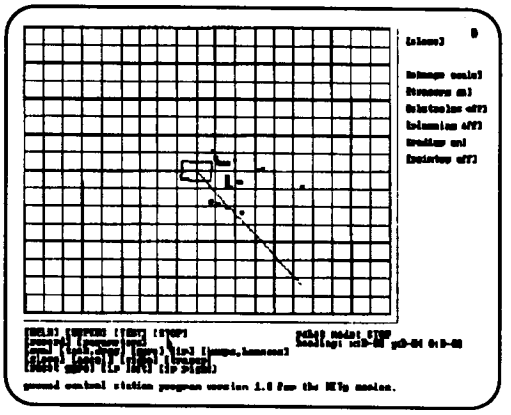


Figure 6: Ground Station GUI

A graphical window displays robot telemetry in real-time and allows user interaction with goal points. Robot position, obstacles, and free distances are updated independently at about 1 Hz onto a reference grid. These can be toggled for display as well as various driving aids for supervisory control. The scale and offset of the window relative to the robot can be adjusted on-the-fly. The screen follows robot motion by means of a travel boundary, which recenters the screen on the robot when crossed.

The button interface provides all means of interaction with the robot, from parameter and goal changes to sensor sampling requests. It also backs up graphical interactions, such as goal positioning and window resizing, with precise entry ability. A message window displays robot responses to user interactions and shows the precise navigation state at all times.

4.3 User Intervention

The operator interacts with the robot in one of three operation modes: semi-autonomous, supervisory, ready. Human factor issues in monitoring and shared control are regarded, as introduced by Sheridan [12]. The robot is placed into ready mode on power-up.

Ready Mode

In this mode the robot can service troubleshooting and reprogramming requests by the operator; all control parameters, goals, and sensors are accessible for viewing and modification. The drive system is inactive in this mode for safety reasons. Parameter sets may be saved and loaded from the operator station for testing or optimization purposes. The robot's initial position and path are set in this mode based on static video imagery; the mouse can set target destinations graphically. A panic button returns the robot to ready mode from other active modes.

Supervisory Mode

From supervisory mode the operator can command speed and steering to the robot. Arrow keys drive the robot forward or backward from 0-30 cm/s and can steer down to 63 cm arcs. Graphical aids for supervisory control include superimposed turning arcs, lines between targets, and unerased telemetry data. This mode is used for fine maneuvering or trap extraction, often relying on real-time video imagery in addition to position and obstacle telemetry. Although obstacle avoidance is inactive on the robot, it will still stop for collisions in case of operator error. To push an object or climb out of craters, collision parameters can first be modified in ready mode.

Semi-Autonomous Mode

In its baseline mode, the micro-rover autonomously travels between target destinations, which are initialized in ready mode. The operator may guide the robot in real-time by moving the current target with arrow keys, which move it radially and axially about the robot. The rover performs homing and obstacle avoidance at its nominal speed, freeing the operator to interact only when required. Real-time video can provide long-range information to the operator for maneuvering the rover around hazardous regions, rather than individual hazards.

4.4 Analysis Tools

Post-mission analysis involves logging and replaying telemetry data at a desired rate and display perspective. Telemetry signals may be recorded to a file from any mode during the mission. At operator station

startup, either real-time or replay operation may be chosen. The latter allows the user to load a telemetry file and analyze it with the same graphical aids available to the real-time mode, even if those aids were not active during recording.

4.5 Implementation

The operator station code is currently written in BASIC. A transmission delay of 0.2 sec was found by comparing the speed of implementing a simple command from the remote operator station to that from the optional keypad on the robot. This delay consists of both protocol overhead and radiomodem throughput; it is minimized by customized asynchronous and buffered message passing.

5. Performance Results

The MITy system has been implemented and tested in both the field and simulation. The simulation is used for development and Monte-Carlo performance evaluations.

5.1 Simulation

The rover simulation embodies the control code and two dimensional models of the platform and environment. It runs on an IBM/320 workstation at about 100 times actual rover speed.

Modeling

The vehicle, sensors, and environment models are low-fidelity, two dimensional representations intended for control system development. The vehicle is modeled kinematically as a bicycle with first-order steering dynamics. Perfect traction and rigid body assumptions are made at nominal speed, which has been shown reasonable for the micro-rover in low gravity packed powder environments [2]. The laser scanner and inclinometer models also have first-order dynamics; the drive motors, platform suspension, and other sensors are quasi-static at the time scale of interest. Rover dimensions and locations of sensors are represented faithfully. The laser is modeled as a ray and the bumpers as line segments in an environment of circular obstacles. All time constants and kinematic parameters were determined from experimental data, and sensor returns are considered ideal.

Monte-Carlo Statistics

Performance statistics were compiled on batch runs of the rover through random obstacle fields. In the nominal run, the rover must travel to a target 46 m (50 yd) away through 25 cm obstacles with 6% aerial density, the median distribution of rocks on Mars [13]. Runs were arbitrarily considered failures if the rover traveled over 92 m or crossed outside the 31 x 61 m field boundaries.

Measures of rover behavior and performance were selected by inspection and correlation studies, which describe time and power usage, path features, overall safety and navigational error, and failure modes of each run [4]. For a batches of 100 runs, statistics on these measures were compiled over variations in obstacle fields and control parameters; a few of these are listed in Table 1 for various obstacle sizes and densities. The mean passing distance is the average distance to the closest obstacle, which reflects rover safety. Normal deviation is of the distance away from the straight line path to the target, to estimate the area necessary to penetrate a given obstacle field. The total path distance shows power usage, while the distance in reverse indicates collision frequency. These measures were only compiled for successful runs, which is the foremost indicator of performance.

Table 1: Statistical Measures

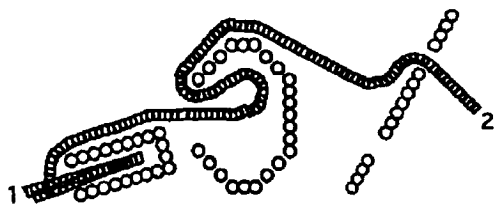
performance	obstacle radius (cm)/ aerial density (%)				
measures	25/ 6	25/ 10	25/ 3	30/ 6	20/ 6
mean pass (m)	0.8	0.7	1.2	0.9	0.7
norm dev (m)	1.4	2.5	4.3	1.2	2.1
total dist (m)	59	94	50	54	77
rev dist (m)	2.5	14.2	0.4	1.3	8.4
success (#)	100	44	99	99	94

Case Studies

Individual case studies highlight special abilities of the MITy control system. A rover that can recover from dead-ends without operator intervention, as shown in Figure 7(a), is desirable for cluttered obstacle fields with limited sensor ranges. In 7(b), the rover escapes the type of large concave obstacle that often troubles potential field methods of obstacle avoidance. Lastly, methods that ignore turn radius constraints would suffer in the hole-in-the-wall test, which demonstrates the unification of target homing with obstacle avoidance in 7(c).

5.2 Field Tests

The rover and portable operator station were transported to various locales to test real system performance. The results presented here are qualitative and more telling of hardware performance than the simulation.



(a) dead end (b) concave obstacle (c) hole-in-wall

Figure 7: Simulated Case Studies

Effects of Environment

Navigation and hazard detection were affected by different environments as listed in Table 2. Only problems are noted; otherwise, the rover performed as expected. Specular reflection and misorientation of the laser increased the frequency of collisions. Navigation was generally over 95% accurate outdoors, even over sand and under shade, due to dynamic sensor arbitration.

Table 2: Environment Effects

Environment	Nav. problems	Hazard problems
Hallways	gyro drift	specular reflection at low incidences
Pavement w/ traffic cones	sun reflectance off building windows	specular reflection on edges of cones
Sandy beach w/ sand piles	wheel slippage on sand	specular reflection off polished sand
Rocky beach	wheels not in contact w/ ground	pitching/rolling of laser scanner
Grass field w/ people	no problems	no problems

Supervision Effectiveness

Overall system performance and robustness were increased by using the various rover modes in conjunction with each other, which eases user fatigue during semi-autonomous segments while allowing detailed supervisory operations. In fact, much of the control system was debugged using the operator station for the insight and flexibility it provides. Real-time video images were mainly useful for target homing rather than obstacle avoidance, because the spatial relationship between the rover and observed obstacles was not intuitive to the user.

6. Continuing Work

Future plans in sensing are to incorporate a quartz gyro and phase-locking to a modulated laser for more accuracy and less power. The operator station is currently being integrated with customized 3D simulation and animation packages for operator training and further system verification. Further team effort will hopefully culminate in space qualification of the final MITy prototype for consideration in the NASA MESUR mission.

7. Acknowledgments

The authors acknowledge the support of the MITy team and its technical advisor, David Kang, PhD. Special thanks to Matthew Fredette for implementing operator station and driver designs, and Anthony Lorusso for implementing laser and sun sensor electronics. Lastly they appreciate Dr. Joseph Shea and Dr. Thomas Sheridan at MIT for their academic advisement. This project is supported by the C. S. Draper Laboratory and MIT Space Grant.

8. References

[1] Schondorf, S., *Systems Engineering for a Mars Micro-Rover*, MSAE Thesis, MIT, June 1992.

[2] Gilbert, J., *Design of a Micro-Rover for a Moon/Mars Mission*, MSME Thesis, MIT, Dec. 1992.

[3] Ma, C., *Dynamics, Control, and System Simulation of a Planetary Micro-Rover*, MSME Thesis, MIT, June 1993.

[4] Malafeew, E., *An Autonomous Control System for a Planetary Micro-Rover*, MSME Thesis, MIT, June 1993.

[5] Kaliardos, W., *Sensors for Autonomous Navigation and Hazard Avoidance on a Planetary Micro-Rover*, MSAE Thesis, MIT, June 1993.

[6] Layman, W. and Matijevic, J., "Micro-Rover Technical Baseline: Highlights, and Design Philosophy," *JPL Interoffice Memorandum*, July 1993.

[7] Krafft, B. and Pien, H., "Terrain Reconstruction from Rover Images," presented at *AIAA SP&T Conference*, Huntsville, Sept. 1993.

[8] Payton, D., "An Architecture for Reflexive Autonomous Vehicle Control," *IEEE Trans Rob Auton*, 1986, pp. 1838-1845.

[9] Borenstein, J. and Koren, Y., "The Vector Field Histogram--Fast Obstacle Avoidance for Mobile Robots," *IEEE Trans Rob Auton*, 1991, pp. 278-288.

[10] Amidi, O. and Thorpe, C., "Integrated Mobile Robot Control," *SPIE Mobile Robots V*, 1990, pp. 504-523.

[11] Lim, W., "Self-Awareness in Mobile Robots," *SPIE Mobile Robots VI*, 1991, pp. 58-67.

[12] Sheridan, T., *Telerobotics, Automation, and Human Supervisory Control*, MIT Press, Cambridge, 1992.

[13] Christensen, P., "The Spatial Distribution of Rocks on Mars," *Icarus*, v.68, 1986, pp. 217-238.