David Wettergreen
Paul Tompkins
Chris Urmson
Michael Wagner
William Whittaker

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890, USA
{dsw, pauldt, curmson, mwagner, red}@ri.cmu.edu

Sun-Synchronous Robotic Exploration: Technical Description and Field Experimentation

Abstract

Sun-synchronous robotic exploration is accomplished by reasoning about sunlight: where the Sun is in the sky, where and when shadows will fall, and how much power can be obtained through various courses of action. We conducted experiments in the Canadian high arctic using a solar-powered rover to prove the concept of Sun-synchronous exploration. Using knowledge of orbital mechanics, local terrain, and locomotion power, the rover Hyperion planned Sun-synchronous routes to visit designated sites while obtaining the necessary solar power for continuous operation. Hyperion executed its plan, beginning and ending each 24-h period with batteries fully charged, after traveling two circuits of more than 6 km in barren, Mars-like terrain.

The objective of the Sun-Synchronous Navigation project (http://www.frc.ri.cmu.edu/sunsync) was to create hardware and software technologies needed to realize Sun-synchronous exploration and to validate these technologies in field experimentation. In the process, we learned important technical lessons regarding rover mechanism, motion control, planning algorithms, and system architecture. In this paper we describe the concept of Sun-synchronous exploration. We overview the design of the robot Hyperion and the software system that enables it to operate in synchrony with the Sun. We then discuss results and lessons from analysis of our field experiments. This paper describes rover and planetary exploration research at Carnegie Mellon during 2000–2002.

KEY WORDS—sun-synchronous robotic exploration, autonomous navigation, polar solar power

1. Introduction

Robotic exploration of planetary surfaces is restricted by available power. With sufficient energy, surface exploration missions could last for months or years. New capabilities for reasoning about resources in real time, to balance the demands of locomotion, communication, and science investigation, will help to enable long-duration exploration.

Sun-synchronous navigation involves reasoning about sunlight: where the Sun is in the sky, where and when shadows will fall, and how much power can be obtained through various courses of action. It is accomplished by traveling opposite to planetary rotation, synchronized to the orbital motion of the Sun, to control exposure to sunlight (Shrounk 1995). On the planets and moons, robots may employ the Sun-synchronous navigation technique to acquire the solar energy necessary to sustain exploration for extended periods of time (Wettergreen et al. 2001). There are other potential benefits such as ensuring visibility and adequate lighting, and moderating thermal extremes by planning solar exposure. Sun-synchronous navigation is a potential means to provide for persistent, in some cases perpetual, presence to explore, dwell in, and develop regions of planets and moons.

Sun synchrony can potentially enable global circumnavigation when the robot's speed of traverse is sufficient to match the rate of planetary rotation, as is possible at the equator of Mercury or the poles of the Moon. On bodies with axial inclination, such as the Earth and Mars, Sun-synchronous routes occur at polar latitudes and follow a path of continuous exposure to the Sun which circles above the horizon in summer. This means that by setting speed appropriate to latitude and navigating to avoid shadows cast by local terrain, solar-powered rovers can obtain continuous energy from the Sun.

The goal of this research is to develop the technologies and to prove the concept of Sun-synchronous exploration in field demonstration. Here on Earth that means during the summer in a polar region with continuous daylight. The following objectives were set:

- to devise resource-cognizant navigation algorithms that produce plans to avoid shadows and navigate par-

The International Journal of Robotics Research
Vol. 24, No. 1, January 2005, pp. 3-30,
DOI: 10.1177/0278364904046632

Fig. 1. Hyperion in the Mars-like terrain of Devon Island, Canada casts an elongated shadow from low-angle polar sunlight.

tially unknown, time-varying, natural environments with an effective balance of power production and consumption;

- to optimize terrain perception, obstacle avoidance, and traversability evaluation to achieve autonomous navigation at 1 km per hour;
- to create a solar-powered robot whose scale, capability and technology are suitable for terrestrial polar operation;
- to perform field experiments featuring multikilometer, semi-autonomous traverse in 24-h circumnavigation of a planetary-analog region, to convincingly demonstrate Sun-synchronous exploration.

We conducted our experiments in July 2001 on Devon Island in the Canadian high arctic (75°N) with the solar-powered rover Hyperion (Figure 1). Hyperion's mission planner used models of orbital mechanics, terrain elevation, and vehicle performance to construct Sun-synchronous routes between designated sites while obtaining the necessary solar power for continuous operation. Hyperion executed its plan and returned to its starting location with batteries fully charged after 24 continuous hours navigating the barren, Mars-like terrain of Devon Island. The project thus achieved its goal.

This paper is organized in two parts: first, a complete technical description of the Hyperion robot and its software system, particularly the global mission planning and local navigation components, and secondly a review of the field experiments conducted and their results. We conclude with our insights and future direction.

2. Technical Description

2.1. Polar-Environment Robot

We created the robot Hyperion to exploit the advantages of Sun-synchrony and named it after the Greek titan Hyperion whose name translates as "he who follows the Sun." Hyperion represents a class of polar robot notable for reduced complexity, mass and power, and solar panels oriented vertically to catch the low-angle polar sunlight (Wettergreen et al. 2002).

In the configuration of Hyperion we strived for mechanical simplicity in order to reduce mass and thus reduce necessary locomotion power. To obtain sufficient solar energy for sustained locomotion in a terrestrial polar environment requires a careful balance between the amount of energy needed for locomotion and the amount collected by a given solar array area. There is a design spiral that occurs as rover mass increases, the locomotion power increases and thus the solar array size must increase, thereby increasing the rover mass further. Conditions for sustained locomotion are actually more favorable on the Moon with one-sixth of the gravity and greater solar flux due to the lack of atmosphere than they are on Earth (Table 1).

For Hyperion, a fixed-orientation solar array was chosen rather than an actively pointed panel to minimize rover structural mass and actuated motions. Reducing actuated motions reduces mechanical, electronic, and control complexity. We had a second more philosophical motivation: that a fixed array would represent the most challenging planning problem for Sun-synchronous navigation. Planning for Sun-synchrony is useful with an actively pointed array for predicting locomotion power and to determine energy needs and pointing requirements but we believed it was truly necessary with a fixed array which would not track the Sun exactly because of the vagaries of the terrain. If Sun-synchrony could be devel-

Table 1. Hyperion Has Low Mass and Required Power But Ample Size to Stably Support the Solar Array Needed for Polar Operation

Parameter	Value
Mass	156 kg
Size	1.8 m width, 2.0 m length (between axles), 3.0 m height
Panel	3.45 m^2
Speed	0.25 m s^{-1} nominal, 0.3 m s^{-1} maximum
Power	75 W continuous + 50–200 W locomotion

oped for a fixed array, the hard case, it would work for arrays pointable in one or two degrees of freedom.

Hyperion's chassis is composed of three tubes and clamping brackets, all of identical design, that can be easily assembled in the field. The chassis structure is 7.6 cm diameter, 3 mm thickness, aluminum tubing. Assemblies for joints, drivetrains, enclosures, and masts clamp to the tubing, which allows the vehicle to be easily reproportioned. The rear "T" frame is joined to the front axle by a steering joint that provides yaw and roll motions (Figure 2).

The passively articulated steering joint is composed of two free rotations. The first is about a vertical axis which allows the front axle to yaw relative to the rover body, in order to steer the vehicle. The second rotation is a roll motion of the front axle, which allows all four wheels to maintain contact with the ground on uneven terrain. A sprung suspension is not used because Hyperion travels at relatively low speeds and impact forces are minimal.

Another advantage of the steered front axle is that by fixing perception sensors to the axle their view is always directed along the driven path. The steered front axle carries a sensing mast with stereo cameras, a laser range finder, and a panoramic camera. Having the sensor mast on the steered front axle increases the effective horizontal field of view while turning.

Each of Hyperion's four identical wheel modules is composed of a motor, gearhead, and chain drive. A brushless DC motor (Kollmorgen Goldline) is used to provide a maximum continuous torque of 1.5 Nm at 6.25 amps. Motor feedback is provided by a 2048 count per revolution incremental encoder. The output of the motor is connected to a harmonic drive (HD Systems) for an 80:1 reduction providing a maximum continuous torque of 90 Nm (including the 75% efficiency of the harmonic drive). A bicycle chain connects the output of the harmonic drive to the drive wheel. The chain, sprockets and wheels are mountain bike components that were chosen because of their durability and light weight. Pneumatic tires are of course for terrestrial use only.

Hyperion is designed for natural terrain. It surmounts all obstacles up to 20 cm in height using four-wheel independent drive. It operates at a nominal speed of 0.25 m s^{-1}, slightly under 1 km h^{-1}.

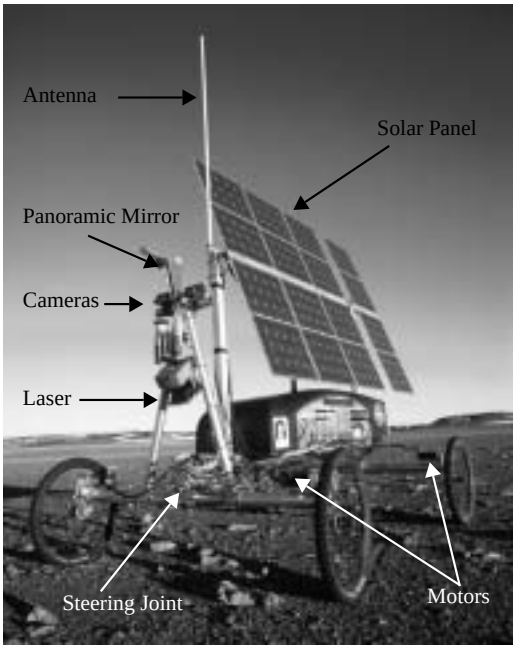


Fig. 2. Hyperion's main components, most notably the vertically oriented solar array mounted atop a low-mass high-stability chassis.

Although it is configured for polar operation, it is intended for operation during the period of continuous daylight near mid-summer when temperatures in the arctic are moderate. Hyperion is sealed and splash-proof but not extensively insulated.

2.2. Passively Articulated Steering

An important innovation is that Hyperion uses differential velocities on all four wheels to steer the front axle relative to the vehicle. This strategy drives the vehicle smoothly and efficiently along a desired path without a steering actuator; four drive motors are all that are needed. This configuration offers the mechanical simplicity of skid steering while maintaining the power efficiencies of articulated-steering vehicles. Velocity control on all four wheels eliminates slipping and skidding thus improving locomotion efficiency. The absence of steering actuators reduces mass but increases control complexity. Rather than performing position control on a steering motor, the front axle angle must be controlled by commanding front drive motor velocities properly, even when traveling over uneven terrain (Shamah et al. 2001).

We studied various steering and drive mechanisms during our rover configuration analysis to determine the best combination of mobility and simplicity for the traverse of the terrain that we expected to encounter in a polar desert environment. Independent and Ackerman steering, frame and

axle articulation, and skid steering configurations were considered (Figure 3). Independent all-wheel steering explicitly actuates each of the wheels to the desired heading. Apart from actuation complexity and control, this provides advantages to maneuverability, especially when operating in unprepared terrains, due to instantaneous omnidirectional motion. Independent steering also enables crab motion in which all wheels turn to the same direction and advance. As a result, the robot can move in a sideways fashion. Advantages of independent explicit steering include more aggressive steering with better dead reckoning (due to less slip of the wheels) and lower drive power consumption. There is an energy cost in steering the wheel assemblies and sometimes unintended soil work.The downside of explicit steering is a higher actuator count, part count, and the necessary swept volume.

The most common type of steering on passenger vehicles is mechanically coordinated with an Ackerman mechanism, which couples the angle of the front wheels. To maintain all wheels in a pure rolling condition during a turn, all the wheels must follow curved paths with different radii originating from a common center along the rear axle axis. Coordination of steering allows efficient maneuvering and reduces the affect of internal losses due to actuator conflict and slipping. The Ackerman mechanism however introduces a non-holonomic constraint in motion planning.

Articulated frame steering is prevalent in large earth-moving equipment. The heading of the vehicle changes by articulating the hinged chassis. For large vehicles, articulated frame steering has the advantage of allowing the vehicle to be more maneuverable than a vehicle with coordinated steering. Articulated frame steering has the advantage over skid steering in that the maximum thrust provided by the traction elements is maintained while turning.

Skid steering can be compact and lightweight, require few parts, and exhibit agility up to point turning using only the components and motions needed to drive straight ahead. Steering is achieved by creating a differential thrust between the opposite sides of the vehicle to cause it to skid and change heading. A disadvantage is that skidding causes unpredictable power requirements due to terrain irregularities and non-linear tire–soil interaction. In addition, while turning, skid steering loses efficiency in soil work so it cannot maintain maximum forward thrust, in contrast to explicit steering which can maintain thrust while turning and minimizes soil deformation.

Articulated axle steering is performed by adding a pivot to one of the vehicle axles. This type of steering is common in wagons although their wheels are typically unpowered. One disadvantage of articulated axle steering is that the wheels run in separate tracks when following a curved path; in soft terrain this requires greater power due to plowing as each wheel is driving over fresh terrain. The articulated axle design suffers from lesser tolerance of drivetrain (motor and gearing) failure than other designs. The other chassis configurations may be able to resort to skid steering if a steering actuator fails and will have reduced efficiency and control if a drivetrain fails free. The articulated axle design can tolerate drivetrain failure in the rear if even both drivetrains fail but continue to rotate freely. If the front wheels partially lose traction, more control effort is needed to steer properly. However, if a front drivetrain fails the vehicle cannot steer predictably. In benign terrain travel is possible with great control effort, but in rough terrain response to terrain features and irregularities will dominate and progress is unlikely. The advantages of an articulated axle steering design include mechanical simplicity, relatively low steering power, and good maneuverability. Our comparative analysis showed the articulated axle design to be promising for the Hyperion rover (Table 2).

To better understand the articulated axle steering approach, we prototyped a full-size chassis model. The model enabled several insights, including the need to place the front axle's steer pivot above the roll pivot (Figure 4). The steering pivot, which is about a vertical axis, allows the rover to change its heading. The roll motion is necessary to continuously maintain four-wheel contact with the ground. If the roll pivot is placed on top, the front wheels do not necessarily remain perpendicular to the ground plane and the front axle can be forced into a singularity that causes the front axle to collapse. Once this property was understood, the full-scale prototype showed that the articulated axle steering approach was feasible.

2.3. Steering and Drive Control

Hyperion's passive steering approach is by definition underactuated, which presents challenges to its motion control system. To support teleoperation and autonomous driving, mobility and steering must be: accurate, so mobility commands are carried out as expected; reliable, so the vehicle avoids limiting configurations; and maneuverable, to enable capable avoidance of obstacles. Furthermore, maneuverability must be predictable so that it can be modeled by the navigation algorithms. High maneuverability is achieved by minimizing the response time of the transition of the vehicle into the desired radius. To satisfy these design goals, control algorithms were designed to maintain a desired steering axle angle using velocity control of all four wheels. Figure 5 provides a schematic diagram of the control hardware.

Figure 6 describes the flow of data through the control system. Mobility commands consisting of a desired steering axle angle, θ_d, and overall rover velocity, v_d, come from a vehicle controller process. It is responsible for responding to commands from the navigation software and for producing telemetry. Commands from either source are represented by rover speed and steering arc radius. These commands are converted into angular velocities for each of Hyperion's four drive wheels using a kinematic model of the rover, which consists of equations given in Figure 7.

Note that the front wheel angular velocities are based on the desired steering axle angle, θ_d. In contrast, the rear wheel

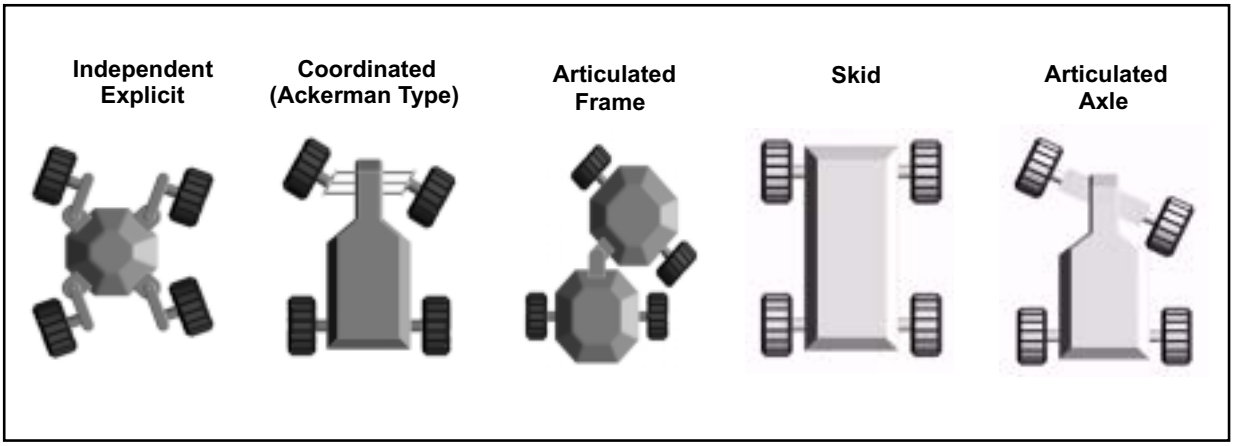


Fig. 3. Configuration of steering types for four-wheeled rovers.

Table 2. Comparison of Steering Types

	Independent Explicit	Coordinated	Articulated Frame	Skid	Articulated Skid
Maneuverability	Medium/High	Medium	Medium	High	Medium
Mechanical complexity	Medium	Medium/High	Low	Low	Low
Control complexity	Low	Medium/Low	Medium	Low	Medium/High
Drive power during steering maneuvers	Medium	Medium/Low	Medium	High	Low
Number of actuated joints for steering	4	1	1	0	0

velocities are based on the current actual steering axle angle, θ_a. If the rear wheels, which are attached to a rigid axle, do not spin at a rate based on the actual steering axle angle, the chassis is subjected to stresses that slip the front wheels and disturb the velocity control loops. Because the front axle freely pivots, its wheels spin at a rate based on overall vehicle speed plus an additive term, which drives the axle to the desired steering angle.

Simply commanding the front wheel velocities based on the desired steering angle does pivot the angle; however, the response time is slow. Early testing showed that such a system is stable. The axle will eventually reach the desired angle and will tend to stay at this angle, slowly righting itself even if the rover surmounts an obstacle and the front axle unpredictably rotates. Speeding up these transitions increases the rover's maneuverability. To do this, a proportional controller was added (shown at the top of Figure 6). The controller's output represents a change in drive wheel velocity, $\Delta\omega$, that is subtracted from the front wheel inside the steering arc and added to the front wheel outside the arc. The controller output is based on the difference between the desired and actual steering axle angles. The proportional term is increased to improve the steering response of the vehicle. Once the wheel angular velocities are calculated, closed-loop velocity control is accomplished in the motion control hardware, which uses PID velocity control loops.

2.4. Polar Solar-Power System

To gather and store power Hyperion employs two arrays of solar cells, two maximum power-point trackers (MPPT), two sets of batteries, and components for conversion and distribution of power to the various subsystems (Figure 8). Hyperion's solar arrays consist of eight paired modules, visible in Figure 2, fabricated from 12.8% efficient silicon cells. The array is tilted perpendicular to the average Sun elevation, in this case 21°. The total area of solar cells is 3.45m^2 which can provide 220 W given isolation of 500 W m^{-2}. MPPTs allow operation of each of the arrays with maximum efficiency for various insolation and thermal conditions and support two separate battery buses at a nominal 24 V. Two sets of lead acid gel-cell batteries can power the robot during conditions of shadowing or peak consumption during slope climbing. The two battery buses also supply power to a main bus for the computing, sensing, control and communication subsystems. Without input energy, the batteries have capacity to power

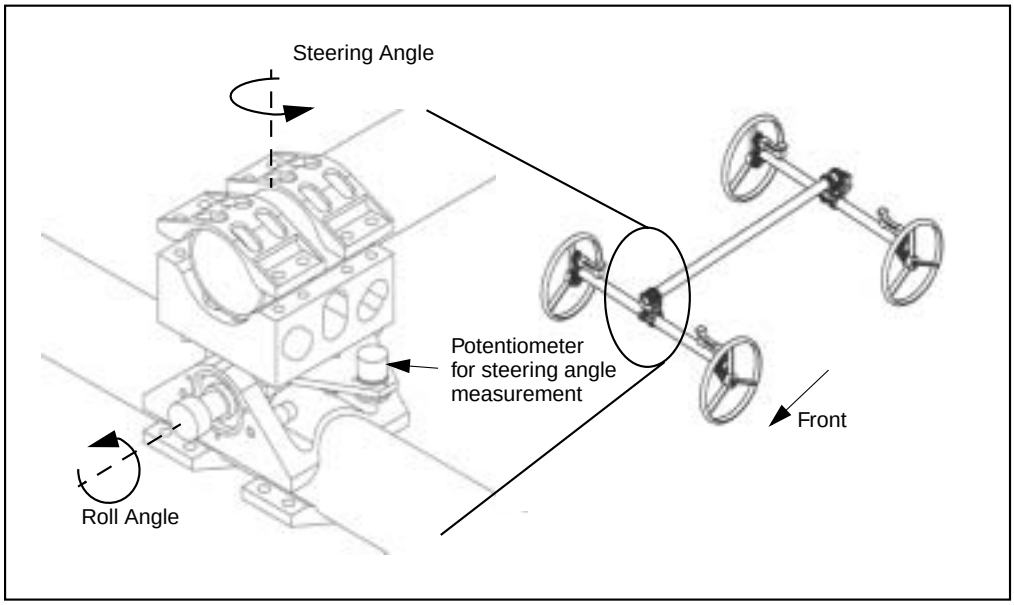


Fig. 4. Hyperion's front axle steering and roll joints are passive pivots.

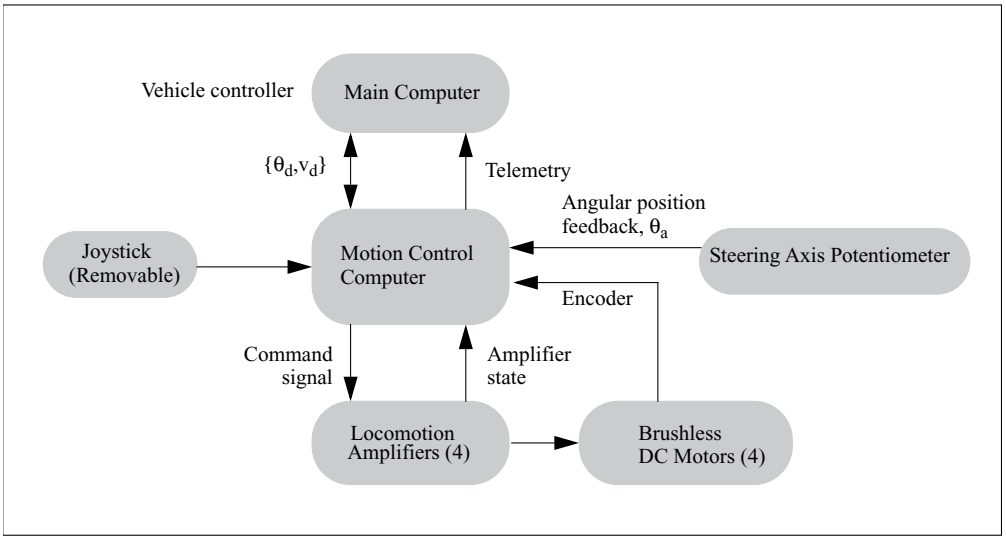


Fig. 5. Hyperion's real-time motion control computing and electronics enable autonomous guidance.

Hyperion for approximately 2 h under typical operating conditions. Power system parameters, such as bus voltage, battery and load currents, are sensed continuously and monitored.

2.5. Sensing and Computing Systems

The computing hardware is designed to support on-board autonomy including high-throughput components such as stereo perception and local navigation as well as a diverse set of processes that require fewer computational resources. The computational complexity of robot autonomy can always make full use of high-performance processors, yet to meet the competing demand for energy efficiency the main computing system uses a single 500 MHz Pentium III (PEP CP302) processor board in a 3U CompactPCI form factor. A CompactPCI solution provides scalability through the numerous bus interface and processor cards available and is designed for mechanical reliability in shock- or vibration-prone envi-

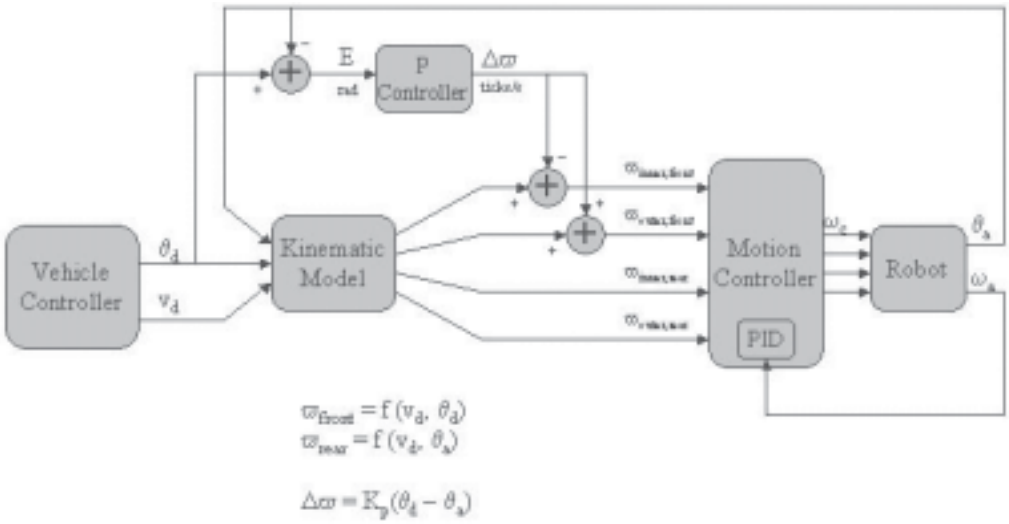


Fig. 6. A steering controller for an articulated axle vehicle computes the speed of each wheel for no-skid-slip driving along the current arc and feeds forward to the front wheels to servo the axle to the desired arc.

$$R_{steer}(\theta) = \frac{L}{\sin\theta}$$

$$R_{robot}(\theta) = \sqrt{R_{back}(\theta)^2 + \left(\frac{L}{2}\right)^2}$$

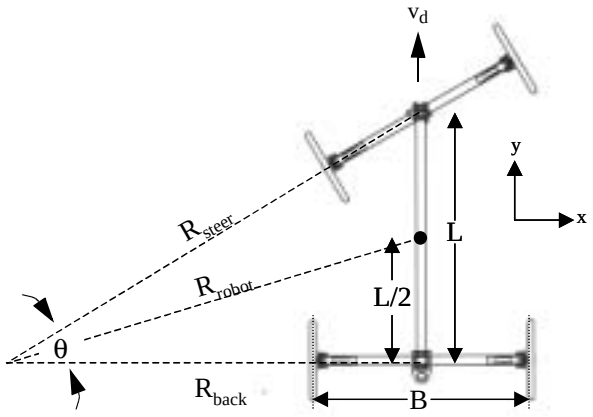
$$R_{back}(\theta) = \sqrt{R_{steer}(\theta)^2 - L^2}$$

$$\omega_{innerfront} = \frac{v_d}{2\pi r_{wheel}} \frac{R_{steer}(\theta_d) - \frac{B}{2}}{R_{robot}(\theta_d)}$$

$$\omega_{outerfront} = \frac{v_d}{2\pi r_{wheel}} \frac{R_{steer}(\theta_d) + \frac{B}{2}}{R_{robot}(\theta_d)}$$

$$\omega_{innerrear} = \frac{v_d}{2\pi r_{wheel}} \frac{R_{back}(\theta_a) - \frac{B}{2}}{R_{robot}(\theta_a)}$$

$$\omega_{outerrear} = \frac{v_d}{2\pi r_{wheel}} \frac{R_{back}(\theta_a) + \frac{B}{2}}{R_{robot}(\theta_a)}$$



L = length of chassis [meters]
B = wheel base [meters]
r_{wheel} = wheel radius [meters]
C = encoder_ticks_per_wheel_rev x gear_ratio [ticks / rev]
θ_a = actual steering axle angle [radians]
θ_d = desired steering axle angle [radians]
v_d = desired robot velocity [meters / second]

Fig. 7. Steering kinematics and equations for an articulated axle vehicle.

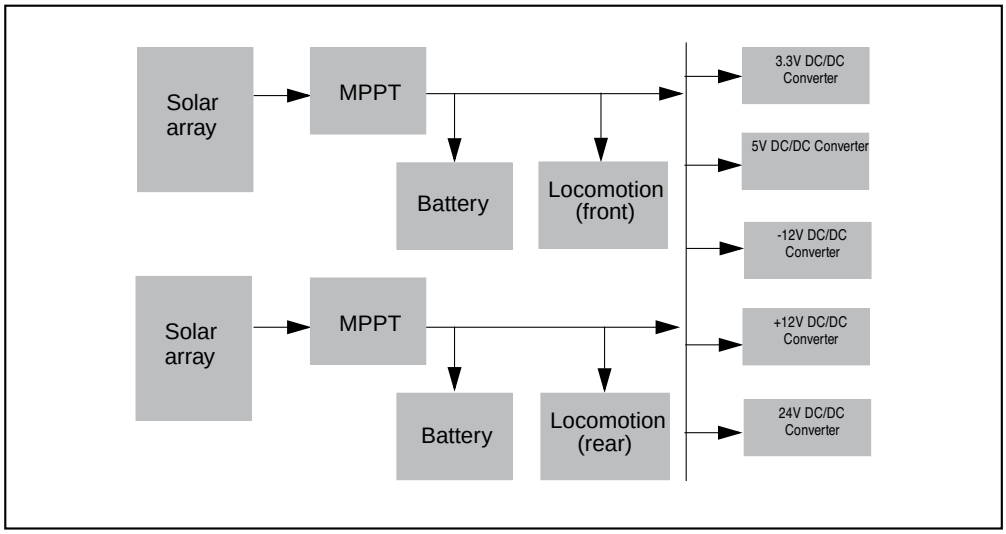


Fig. 8. In Hyperion's power system energy flows from parallel solar arrays and MPPTs onto the common bus and front and rear drives with over/under capacity flowing to/from batteries.

ronments, an important feature for a field robot. The on-board operating system is Red Hat Linux 7.1, kernel 2.4.2. We find the soft real-time requirements of an event-driven system are handled by the standard (meaning not real time) Linux operating system when minimal devices and adequate processor speed are applied. Hard real-time control is handled by dedicated motion control hardware (Galil DMC-2142).

The computing system interfaces over IEEE-1394 buses for devices including two digital cameras (Sony DFW-V500) and a digital video camcorder and RS-232 buses for a motion controller, a laser rangefinder (SICK LM20), a tilt/roll sensor (Crossbow CXTILT02E), and the power management and distribution microcontroller.

Hyperion's real-time drive and steering motion control is accomplished by the Galil multi-axis controller that provides PID motor control using encoder feedback. The controller was selected for ease of implementation and support for reasonably sophisticated programs for steering and arc following. These programs are stored in non-volatile memory and begin execution immediately upon power up. An independent motion controller increases reliability since driving (and stopping) maneuvers do not require Hyperion's main computer. This strategy provided the accuracy needed by the navigation software, the maneuverability to ably avoid obstacles, and the reliability to prevent chassis damage (Shamah et al. 2001).

2.6. Wireless Communication and Telemetry

To conduct long-distance navigation experiments calls for a wireless connection with the robot. An operator interface for telemetry and video display can easily utilize a 1 Mbps data rate and for Hyperion we maximize use of available bandwidth (Manavalan and Wagner 2001).

Wireless Ethernet bridges (Cisco Aironet 340), 2.4 GHz using the 802.11(b) protocol, are used with a directional 19 dB antenna at the control station and an omnidirectional 15 dB antenna on Hyperion. A transportable omnidirectional repeater was introduced as needed to maintain continuous line of sight. The bridges provide 100 mW of transmit power and theoretically provide 11 Mbps data rates beyond 10 km with appropriate mounting.

Communication on board Hyperion is implemented using the IPC protocol (see IPC Technical Document at http://www.cs.cmu.edu/afs/cs/project/TCA/ www/ipc/ipc.html). IPC is a publish/subscribe, message-based protocol for which a message dictionary was defined specific to Hyperion's software design. IPC performs anonymous message subscription, so a Telemetry Manager process was implemented to filter and record message traffic among modules. The resulting telemetry can be replayed or analyzed to understand experimental results.

2.7. Sun-synchronous Navigation Architecture

To operate Sun-synchronously, Hyperion must optimize the orientation of its solar panel with respect to the Sun. This poses significant constraints for the navigation software; it must go beyond avoiding obstacles and reaching goal locations to also follow a schedule and maintain an orientation governed by the position of the sun. To receive 90% or more of the available solar energy, the robot must maintain panel orientation within 15° of the sun (as related by the cosine of

the incident angle). Hyperion can run into difficulty not just from obstructed paths but also from improper orientation or being off schedule. Architecture is emerging in planetary exploration research as a crucial challenge for rovers that have complex skills and true autonomy in isolation (Volpe et al. 2001). This has motivated us toward an architecture that enables rigorous fault detection and flexibility in the command structure to facilitate fault recovery either automatically or by judiciously seeking assistance.

2.7.1. Sliding Autonomy

Hyperion's software architecture exhibits a property of sliding autonomy such that the current conditions dictate the operational mode. This concept and architecture has evolved through many planetary rovers with our approach traced to Dante II (Bares and Wettergreen 1999). The operator can interact with the robot by directly teleoperating its actions, by allowing it to safeguard operator commands, or by enabling it to navigate autonomously. Figure 9 shows the software components active during each of the operational modes.

Monitored teleoperation allows direct low-level commands to the robot and provides minimal safety—safety monitors warn the operator of possible faults but they are superseded by operator commands. An operator is able to move Hyperion by appropriately setting individual wheels speeds, which can be tedious, but it is occasionally necessary to overcome obstacles or escape entrapment. Monitored teleoperation is implemented first on our rovers initially to enable development of the mechanism and drivetrains, verification of sensors, and direct interaction with low-level device drivers. It is rarely used in field experimentation (at which point most systems are mature and reliable) and then only in a situations of recovering from a specific fault.

In the safeguarded teleoperation mode, the operator guides the robot with coordinated motion commands and receives state directly from the sensors, including on-board cameras. The State Estimator integrates sensor information including position, orientation, and speed. The Laser Mapper detects nearby obstacles and signals the Health Monitor of potential collision. When the Health Monitor detects this or other anomalous conditions, motion is stopped and the fault is signalled to the operator. Figure 10 diagrams the modules in this operational mode.

In its autonomous mode, the Stereo Mapper classifies terrain, generating a traversability map from stereo imagery. The Navigator evaluates the map and selects a path that best leads the robot to the next goal. Using position goals, a Mission Planner produces a scheduled plan to guide the rover along a Sun-synchronous route. It then commands goal sequences to the on-board Mission Executive for execution. These goals lead the Navigator along a Sun-synchronous route. If the local goal cannot be achieved or if the time to reach the goal jeopardizes Sun-synchrony, the Mission Executive signals the

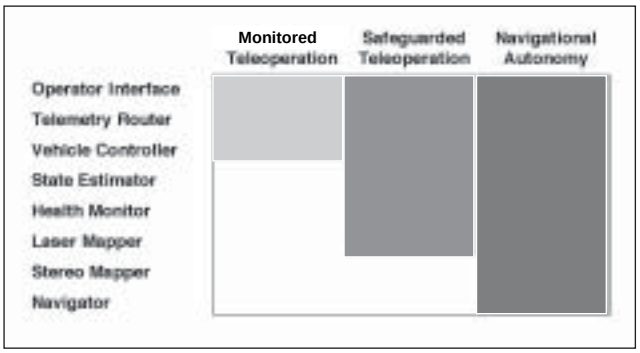


Fig. 9. In each operational mode, different software components are active.

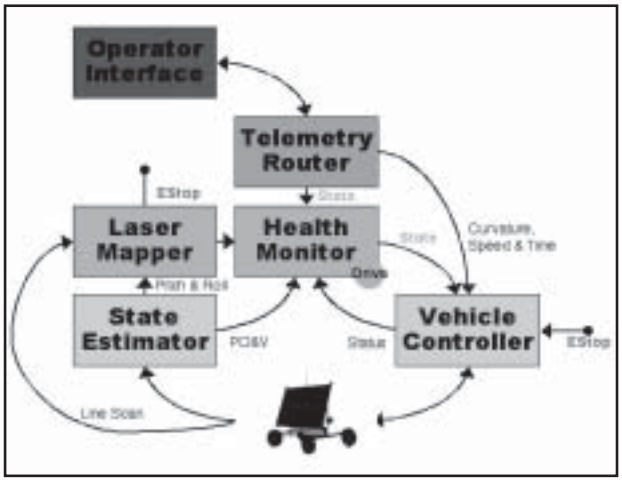


Fig. 10. During safeguarded teleoperation the software architecture provides monitoring of rover state and command safing.

Mission Planner to replan the route. Figure 11 show the additional modules active in this operational mode.

2.8. Health Monitoring

In exploring the unknown, circumstances will likely arise that challenge the capabilities of an autonomous rover. The rover must detect and recover from faults as well as monitor developing conditions so that it may modify its behavior appropriately. This self-awareness becomes crucially important to achieving long-duration operation where faults should not necessarily halt rover operation until the next communication opportunity.

In the Hyperion software system, the Health Monitor samples approximately 100 state variables including temperatures, voltages, currents, positions, orientations, and velocities. It also monitors software state, such as process activity

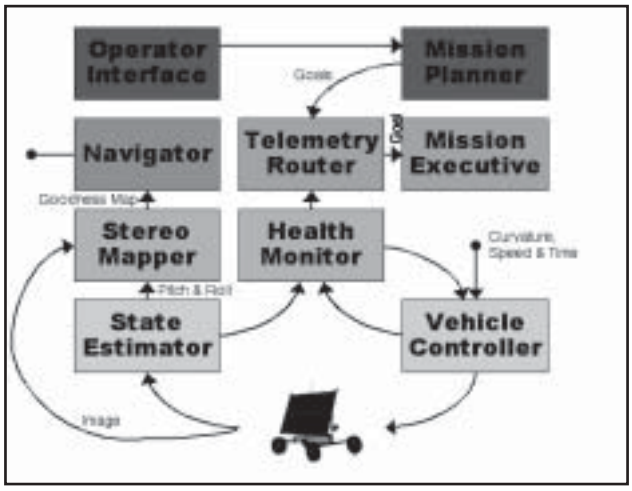


Fig. 11. During navigational autonomy the software architecture activates modules for sensing and modeling of terrain and planing and executing Sun-synchronous routes (Laser Mapper not shown).

and uncertainties in stereo correlation and terrain modeling. This information is used to detect faults in the system which are signaled to the operator (Verma, Langford, and Simmons 2001).

The Health Monitor also determines the system operating mode, meaning whether the rover acts autonomously or not. When individual faults or a combination of faults occur the Health Monitor changes the operational mode accordingly. For example, if the uncertainty in stereo matching becomes large, indicating a fault and that the robot cannot perceive the terrain ahead, the Health Monitor signals a fault. On this and other critical faults, the Health Monitor commands an emergency stop which halts motion and puts the vehicle in a safe mode until operator guidance is received (or in this case if imagery becomes available and the fault clears). When the operator engages the autonomous operational mode, the Health Monitor first checks for faults before allowing the robot to begin driving.

Hyperion's Health Monitor has been designed and developed incrementally. As simple faults become better understood and modeled, they can be recovered from automatically, for example re-enabling amplifiers after current limiting. There are capabilities that may be necessary, including monitoring for complex faults such as wheel entrapment or vehicle high centering, that require the inference from a number of observables. Eventually some form of probabilistic reasoning becomes appropriate.

2.9. Perception and Local Navigation

Hyperion uses a combination of stereo camera imaging and laser ranging for terrain perception. The nature of the terrain encourages the use of an "optimistic/pessimistic" local navigation strategy. Terrain that is unseen by stereo vision, for example due to occlusion or distance, is optimistically assumed to be traversable. The laser operates as a "virtual bumper" that stops the robot prior to collision when detecting obstacles missed by the optimistic evaluation, thus it acts as a pessimistic safeguard. In practice, most of the terrain is eventually sensed by stereo vision and all is swept by the laser before it is traversed. This combination of optimistic and pessimistic strategies allows for efficient navigation with neither undue risk nor the need for perfect terrain knowledge.

We confronted directly the problem of navigation in natural terrain when much robot navigation research has dealt with planar environments and discrete obstacles. There is increasing interest in algorithms capable of driving robots over rough terrain. Often this work maintains the assumption that space can be easily divided into what is and what is not traversable (for example, Stentz, 1995; Bussey, Robinson, and Spudis 1999; Laubach and Burdick 1999). Natural terrain rarely provides this simple distinction as there is a gradation between terrain that is easily traversed and that which is impassable. Without modeling this variability, a robot may be unnecessarily cautious or risky. An abrupt boundary between traversable and impassable can also lead to instability in the navigation algorithm as the terrain is viewed from a changing perspective. Some approaches explore descriptions of terrain that encode its variability, e.g. continuous measures of terrain have been implemented as fuzzy sets (Martin-Alvarez et al. 1999; Seraji 1999). These representations are generally combined with fuzzy logic controllers which suffer local minima problems similar to potential field approaches. There have also been several efforts that describe traversability as a continuous value based on the weighted combination of terrain metrics (Biesiadecki et al. 2001). Generally, obstacle detection is incorporated as a reactive behavior with little or no consideration for global planning (Seraji 2001).

Our work serializes the obstacle avoidance and local navigation algorithms and utilizes complementary sensors to consider terrain (Urmson and Dias 2002). Previously, the output of local obstacle detection software had been combined in parallel with global planning information through an arbiter (Singh et al. 2000). Now, steering commands are selected by the Navigator using the obstacle avoidance and global information in serial, avoiding potential conflicts between global and local information.

By combining an optimistic visual terrain evaluation with a pessimistic virtual bumper, the process by which data are combined into a global navigation map can be streamlined. Furthermore, the robot does not have to be overly cautious about entering terrain it has not fully sensed using stereo vision alone.

2.9.1. Obstacle Avoidance

The specific characteristics of the intended terrain are critical to the design of perception and navigation algorithms.

The high arctic, where Hyperion is designed to operate, is typical of a terrestrial polar environment and shares many properties with regions of Mars and the Moon. The terrain is gently sloping with impassable gullies interspersed at low density. The ground generally consists of loose shattered rocks or hard-packed soil which forms a rough substrate. Boulders are sparsely distributed. This type of environment allows for many optimizations to the navigation algorithm that would be inappropriate in more dense terrain.

The physical characteristics of a robot are also important to determining the configuration of its terrain sensors. Key parameters are the stopping distance, speed and position at which sensors can view the terrain (Kelly and Stentz 1998). Optimizing the characteristics of a stereo vision system involves trading between range accuracy, minimum viewing distance, stereo rate, and field of view. To increase the effective field of view of the stereo system, the cameras can be actively pointed. In the case of Hyperion with stereo camera mast attached to the steered front axle, this allows the use of relatively narrow (60°) field-of-view cameras that are effectively made wider by the swept motion. Hyperion utilizes a 20 cm stereo baseline that yields a range resolution of approximately 15 cm at 7 m. Limiting the number of disparities in the stereo correlation dramatically reduces the time taken to process each frame into a range image, but also limits the minimum distance at which range data can be acquired. Due to this effect, Hyperion is unable to reliably recover accurate range information closer than 1 m from the cameras, but with cameras mounted 1.5 m above the ground this arrangement provides a workable trade-off between accuracy at range and minimum sensing distance.

Traversability is determined from range information by extracting three metrics: the slope, roughness, and discontinuity of the near-field terrain in rover sized patches of the local region. Slope is the steepest of the pitch or roll the vehicle would encounter and is calculated from a plane fit to the range information. Hyperion is designed to be able to climb slopes and navigate side slopes of up to 15°, thus any patch with a slope greater than this is considered impassable. Roughness is calculated as the chi-squared residual of the plane fit. The discontinuity metric is an estimate of the variance of the height of points in the cell. It detects step features in the terrain. Each metric is normalized and traversability of the rover sized patches is considered to be the worst of the three measures. Figure 12 shows the evaluation of a region of terrain with three-dimensional geometry of terrain patches shaded based on traversability. Regions of the stereo field of view that contain more points usually generate a better estimate of the actual terrain. To capture this, a certainty value is computed for each patch. Certainty is calculated as a function of the number of points in a patch and the evenness of the distribution of the points over the patch.

The basic cycle of the Navigator when evaluating possible paths to the next goal is shown in Table 3. The Navigator autonomously guides the rover from its current location to the next goal location, or waypoint, by selecting the lowest cost combination of possible steering arcs and remaining path to the goal as determined by the D* algorithm (Stentz 1995). The robot travels a fraction of the chosen path and then the navigation cycle is repeated—at 1.5 Hz so smooth transition among steering arcs is achieved.

2.9.2. Localization

Hyperion utilizes odometry, inertial sensing, and carrier phase differential GPS to track its location relative to a satellite-derived digital elevation model. Its dGPS (Novatel Beeline) provides accurate heading, pitch and location information using two receiver antennae but is subject to drop-out. By filtering the dGPS and incorporating odometry and measurement of roll and pitch, stable and accurate pose information is obtained. Accurate localization, within 10 cm and 1°, is important for collecting ground truth data on robot performance for analysis, but is not critical to navigation which operates as described in the local reference frame. Foregoing GPS and applying a less accurate positioning system, such as localization from inertial sensing and visual features, would be sufficient for an actual planetary mission, and would be necessary beyond Earth.

2.10. Mission Planning for Sun-synchrony

The Mission Planner solves for plans that enable a robot with a fixed solar array to navigate multikilometer circuits over long periods of time. It must reason about the complex interactions between motion of the Sun in the sky, terrain, shadows, solar array pointing, energy consumption and finite battery capacity (Tompkins, Stentz, and Whittaker 2001). Since Hyperion's solar array is body-fixed, the robot must synchronize its route timing with the position of the Sun to achieve

Table 3. Steps in the Cyclic Algorithm That Drives Autonomous Navigation

	Navigation Cycle
1	Update the robot position via the State Estimator
2	Obtain range information from the Stereo Mapper
3	Compute evaluation metrics into the discretized terrain map (Figure 12)
4	Update position to account for movement during computation
5	Evaluate the cost of motion along a discrete set of arcs
6	Estimate the cost of reaching the goal with each arc (using the D* algorithm)
7	Choose the arc that has the lowest value of total cost
8	Send radius, speed, and time for arc to the Controller
9	Repeat

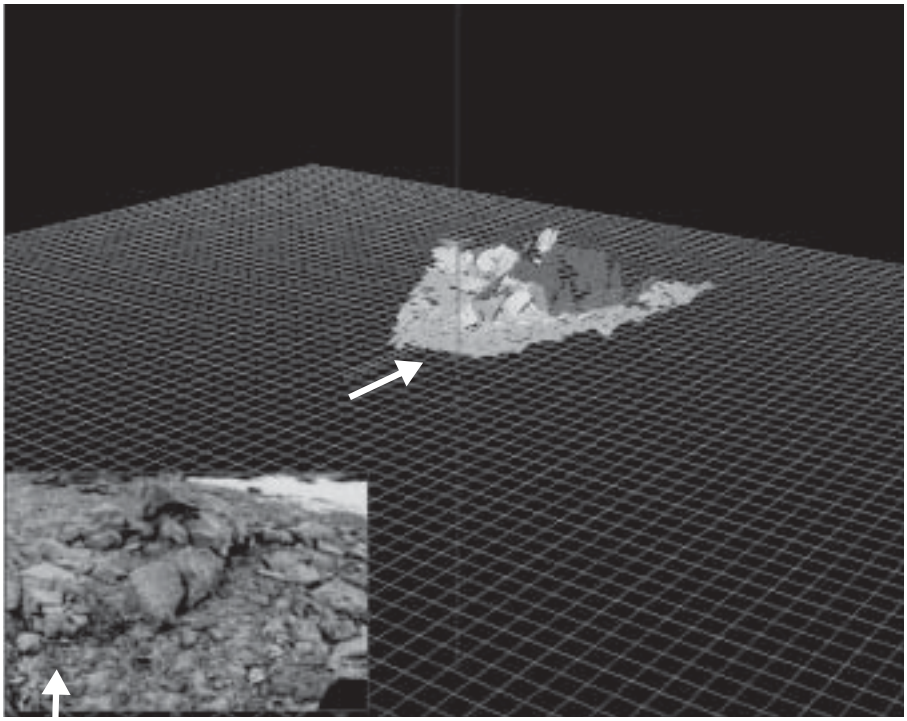


Fig. 12. Terrain evaluation map where medium gray is easily passable, light gray has moderate difficulty, and dark gray is impassible. The inset view of terrain from left stereo camera shows the impassible rock outcrop modeled for traversability and white arrows correspond to the view directions.

array pointing that yields adequate energy for driving and other loads. To point the solar array at the Sun, the orientation of a path must continuously change to match the angular rate of the Earth's rotation. On an idealized planar surface without obstacles, optimal paths are circular. In practice, terrain, shadows and intermediate goals (specifically science targets) typically prevent circular path solutions. Because non-circular routes prevent the solar array from always pointing directly at the Sun, the specific choice of route and timing must be carefully planned to maintain Sun-synchrony and appropriate power levels.

Automated activity planning and scheduling software has been successfully deployed on spacecraft and prototype planetary rovers, including Remote Agent (Bernard et al. 1999) and ASPEN (Chien et al. 2000). In particular, the ASPEN system and its derivative CASPER were separately integrated onto the JPL Rocky 7 rover and used to produce coordinated activity schedules based on science and engineering team requests. These activity scheduling experiments considered resources and environment effects but demonstrated only loose coupling to path planning, focusing primarily on conflict resolution through event rescheduling or reordering. The Sun-synchronous navigation problem prompts a solution that tightly couples route selection and resource management.

Our Mission Planner is called TEMPEST, a planner that searches through a space of two-dimensional position, time and battery energy to find Sun-synchronous paths (Tompkins, Stentz, and Whittaker 2002). It defines the Sun-synchronous navigation domain in terms of a number of models that aid in computing costs for gross robot actions. At its core, TEMPEST uses the Incremental Search Engine (ISE) algorithm (Stentz 2002) to determine the sequence of actions, and hence trajectory in the four-dimensional search space, that is optimal in terms of navigation and energy criteria. ISE is a graph-theory-based search algorithm optimized for planning and replanning in high-dimensional spaces under global constraints. Resulting plans comprise a queue of time-sequenced position–time–energy waypoints that are transferred to the Navigator for execution.

We first describe how TEMPEST defines the planning domain and then discuss the planning procedure.

2.10.1. Domain Models

TEMPEST defines the mission domain in terms of the environment (terrain, ephemeris, lighting, and solar flux), rover performance, rover state and actions, mission objectives, and operational constraints (Figure 13).

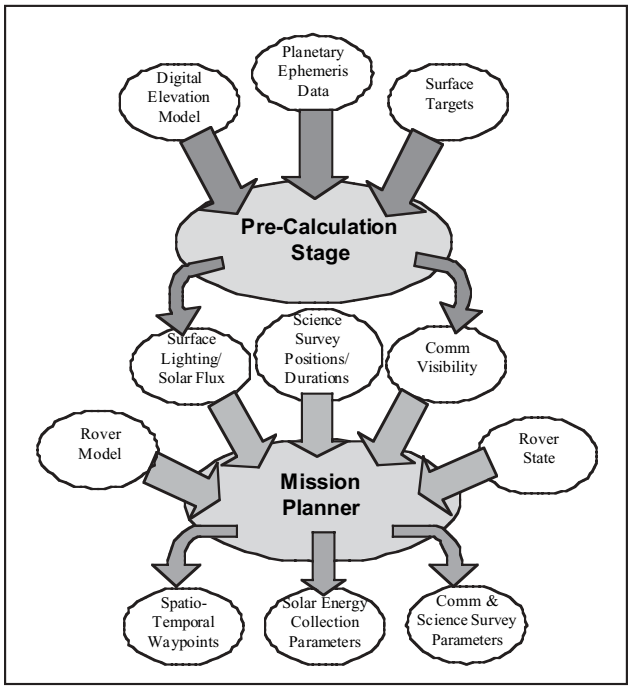


Fig. 13. TEMPEST pre-calculates lighting maps, which are stored for later use. Planning combines objectives, constraints, lighting maps and the rover model to produce mission plans.

Environment: globally-referenced digital elevation models (DEMs) encode elevation values to grid cells. TEMPEST designates position waypoints at the spatial resolution of the DEM, typically 25–100 m. Using the elevation data, it estimates local slopes that aid in determining rover locomotion energy costs and rover orientation as a function of driving direction. TEMPEST uses the JPL CSPICE software (Acton 1996) to determine the relative position and orientation of Solar system bodies, in this instance the Sun azimuth and elevation from the local Earth horizontal. Using terrain and planetary ephemeris data and ray-tracing calculations, TEMPEST computes maps of instantaneous lighting exposure. Sequences of these maps represent the dynamic lighting on the surface over time including where and when shadowing will occur. TEMPEST also incorporates models of predicted solar flux (power per unit area) at Earth locations as a function of time to determine available energy.

Rover performance: a kinematic model describes the orientation of the fixed solar array relative to rover driving direction. A battery model specifies the maximum energy storage capacity of the rover. Other models estimate: rover locomotion power as a function of speed, slope and direction; steady-state electronics power; solar array power as a function of solar flux, array area, efficiency and orientation; and battery energy accumulation as a balance between rates of energy intake and expenditure.

Rover state and action: rover state is represented by position, time and battery energy (a four-dimensional state space). TEMPEST considers two gross rover actions: Drive and Charge. The Drive action moves the rover from the current cells to adjacent cells on the eight-connected DEMs, and allows solar energy collection while in motion. The Charge action points the rover in the optimal direction for solar energy collection and charges the batteries for a fixed duration. Each action results in time and battery energy costs, although energy costs can be negative if collected energy exceeds expended energy.

Mission objectives and constraints: TEMPEST requires operators to specify mission objectives in terms of start position, an ordered list of intermediate goal positions, minimum battery level at the final goal, and the window of allowable starting times. Intermediate goals serve several functions. First, they provide the skeleton route that is a satisfactorily large circuit for exploration. They can be selected to avoid known terrain hazards not apparent in the DEM. In a planetary exploration scenario, they might also represent intermediate science targets. In conjunction with intermediate goals, the Mission Planner can enforce a cyclic path by co-locating the final goal position and the start position. Operators can optionally specify a number of local constraints on actions to impose a minimum allowable Sun angle or maximum allowable slope, prevent entry into any shadowed region, or disallow paths that enter user-defined exclusion zones. Paths are further globally constrained by the finite energy storage capacity of the battery; the planner disallows any path that requires a greater battery capacity than the maximum.

2.10.2. Path Generation

Early developmental tests showed it was not computationally practical to globally optimize entire paths through the four-dimensional search space in a single call to ISE (typically > $5x10^7$ states). To combat this problem, we introduced the intermediate goal mechanism to reduce the spatial dimensions of the search, and time windows to limit the temporal dimensions of the search. TEMPEST plans in reverse order, from goal to start, intersecting each designated intermediate goal. Clearly, locally optimizing each "path segment" between adjacent goals might result in highly suboptimal, or even incomplete, planning behavior. Therefore, TEMPEST plans several path options for each segment, each sharing common start and goal positions and all arriving within an allowable goal window, but all starting at different times, and with potentially different battery energy levels. By chaining the solutions of later segments to the planning of previous segments, TEMPEST extends all the plan options until they reach the global start position or are deemed infeasible. The surviving plans are evaluated according to a set of heuristics to yield the overall

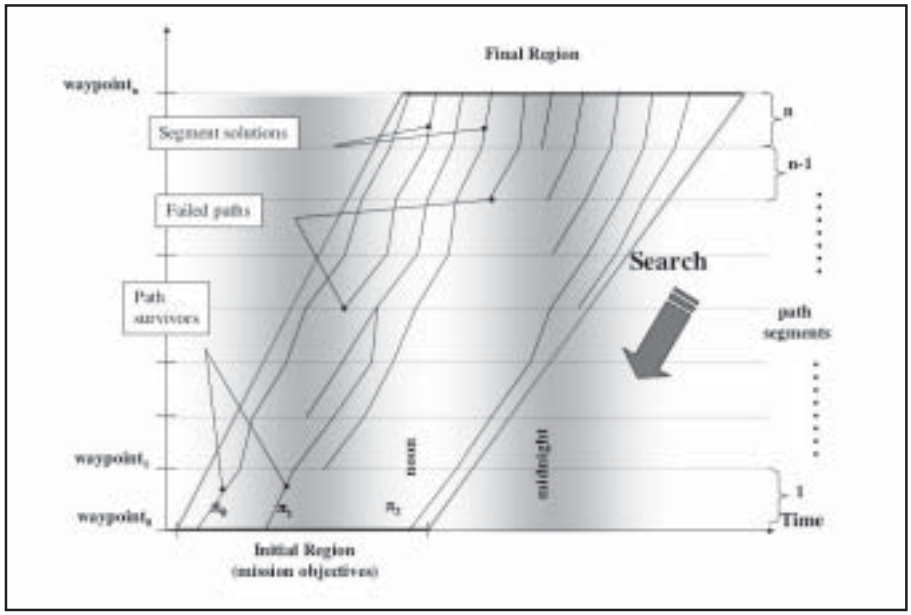


Fig. 14. Depiction of backward-chained plan search. TEMPEST defines start and goal windows, and searches for the optimal path segments spread over time. By passing the start states of later segments as goals for earlier segments, TEMPEST builds complete paths for evaluation.

best plan. The algorithm proceeds according to the following steps, illustrated in Figure 14.

- An operator designates the start location and a window of allowable start times.
- TEMPEST estimates reasonable time windows for goal arrival for each goal, based on projections on the minimum and maximum average rover speed and path distance, and the allowable start time window.
- The software creates a list of final goal states that share the same position, but are distributed evenly over the goal arrival time window.
- The planner defines equally sized, adjacent start time intervals over the window of the previous goal.
- For each of these start time intervals, TEMPEST performs an incremental search for an optimal path segment starting at the segment start position and within the time interval to any of the goal states. (In this heuristic search, non-monotonic objective functions are disallowed because they force the search to linger at states that drive costs down rather than achieving the goal.) It optimizes energy, a non-monotonic function, and solves for segments that have the minimum energy cost below a maximum designated path duration, thereby limiting the amount of time a plan lingers at energy-rich states.
- TEMPEST saves all path segment solutions, and abandons the start states from which feasible solutions cannot be found.
- If the current path segment is the first path segment, the planner records all chains of path segments as complete path options. Otherwise, it designates the start states for each incomplete path option as goal states for the next earliest segment, and continues planning backwards.

In this way, TEMPEST builds complete paths in reverse order, from finish to start. The complete plan options are ordered lists of time-sequenced positions and minimum battery energies that satisfy all specified local and global constraints. Each is the optimal plan to reach all of the goals from its respective start time interval.

2.10.3. Plan Option Selection

To select the single best plan from the list of surviving plan options, the Mission Planner applies one or more of the following evaluation functions on path state values to prune possible plans and select the best path for execution. The first two heuristics return a list of all tied paths and the final heuristic returns a unique path.

Minimum initial energy: select the plans requiring the least initial battery energy to reach the goal.

Minimum peak required energy: select the plans whose peak energy requirement over the plan is minimum. (The peak

energy in a plan represents the greatest demand on the battery over the traverse so finding the minimum peak energy minimizes the peak demand on battery reserves.)

Earliest start time: select the earliest available plan option. Given a list of available start times, often it is operationally desirable to select the earliest option, thereby leaving the greatest number of fallback options in the event of operational delays.

TEMPEST applies these selection heuristics in series to select the best path. Note that the Mission Planner does not mandate optimal solar array pointing, but rather seeks a balance between navigational requirements to reach all the intermediate goals, and energy requirements to satisfy operational costs. This means the rover can, and sometimes does, travel with its panel suboptimally pointed or through shadowed terrain but it has always planned for sufficient energy to complete its traverse.

3. Field Experimentation

The Arctic Circle, at 66.55°N, marks the southernmost latitude of the northern polar regions at which the Sun does not set on the summer solstice. Haughton Crater is located at 75.36°N, 89.68°W on Devon Island about 600 miles north of the Arctic Circle (Figure 15) where the Sun stays above the horizon for over three months in summer. Haughton is particularly notable for the lunar-like breccia inside the crater and Mars-like "planitia" to the north-west of the crater rim (Figure 16). It is the site of ongoing investigation by the NASA Haughton Mars Project (ArcticMars, see http://www.arcticmars.org) and, through the support of the HMP, the location for our field experimentation.

Fig. 15. Map of location of Haughton Crater on Devon Island, Nunavut, Canada (75.36°N, 89,68°W) (Map source: National Geographic Society).

The overall goal of the field experiment was to prove the concept of Sun-synchronous exploration by navigating a Sun-synchronous circuit in 24 h, finishing with batteries fully charged so that, in principle, the operation could be repeated indefinitely while sunlight persists. An important aspect was to determine the practical effectiveness of synchronizing with the Sun, requiring the measurement of ground-truth power input and output, and to learn the achievable vehicle speed and endurance (on Earth). Our three-week field experiment in July 2001 involved numerous hardware and software component tests, and culminated in two 24-h Sun-synchronous circuits.

Typically, an exploration robot's route might be defined by navigation goals that may be selected by planetary scientists who identify sites of scientific interest in orbital images and data. In these experiments, the field team selected a sequence of intermediate waypoints that defined a loose framework for the Sun-synchronous path. Rather than representing scientifically interesting sites, the waypoints defined the topology and scale of the circuit that the Mission Planner transformed into a list of Sun-synchronous waypoints. Although we strived for the highest degree of autonomy that Hyperion could attain, we allowed continuous moderate bandwidth communication to the rover, and occasional teleoperation. Such a scenario would be feasible for lunar polar exploration.

Given a start position, the list of intermediate goals, and a 24-h range of allowable start times, the Mission Planner, which ran on a laptop computer, produced the optimal Sun-synchronous plan including the recommended start time. Beginning at this time, waypoints from the plan were sent to Hyperion individually for execution. The robot would autonomously navigate the terrain, avoiding obstacles while seeking the waypoint. Operators monitored robot progress throughout the field experiment. Vehicle telemetry and low-resolution imagery from a panoramic camera were continually available at 1 Hz to operators through the user interface. When the robot reached a waypoint, it would wait until the designated time to execute the next waypoint, thus maintaining the schedule imposed by the plan. In effect, this also simulated the effects of conducting a science exploration in conjunction with navigation. Importantly, Mission Planning was too computationally demanding to allow replanning during a traverse. Therefore, the operations team had to enforce the navigation schedule imposed by the plan, or take action to resynchronize Hyperion with the plan in the event of operational delays.

Fig. 16. Terrain of Von Braun Planitia north-west of Haughton Crater on Devon Island.

The first 24-h, Sun-synchronous experiment was completed on July 19, 2001. Hyperion traveled 6.1 km with 90% of the mission completed autonomously (Figure 17). Teleoperation was used upon indication from Hyperion's health monitor that a fault had occurred. In this experiment, the only faults to occur were failures to find a forward path and laser obstacle detection. No intervention was required by field observers for safety or any other reason, despite winds up to 27 kph. Teleoperation typically involved very brief instances of human driving to place the robot in more benign conditions from which it could continue autonomously. In several instances during the experiment, Hyperion fell behind its plan due to the difficulty of the terrain it encountered, but each time it was able to catch up to the plan when it reached more benign terrain. The first experiment successfully demonstrated the concept of Sun-synchronous navigation; solar array and battery power were sufficient for continuous operation and the rover battery was fully charged at the completion of the circuit.

A second integrated experiment was conducted in which more challenging terrain and greater distance were attempted. In this experiment Hyperion traversed 9.1 km with 50% autonomy and 50% safeguarded teleoperation, shown in Figure 18. During this experiment a single instance of manual intervention occurred when, in driving close to a large rock, the steering axle tripped a limit switch when a rear wheel was obstructed and the rover was forced to halt. A software patch to override the fault would have resolved the problem, but in the interest of maximizing the experimental data return, this time-consuming option was not exercised. Instead, the axle was repositioned manually and the traverse resumed. Between this incident and numerous communication outages due to the distance and line-of-sight visibility to the base camp, Hyperion fell behind its plan by 3 h. This delay resulted in poor solar array Sun angles, and also in stereo camera blinding from the Sun directly ahead. To recover, teleoperation was needed on 50% of the path (still following the Sun-synchronous navigation plan, but without delays to reacquire the schedule). Safeguarded teleoperation was initiated to command the rover to stop and orient for stationary recharging to build battery reserves. Hyperion regained its schedule after 6 h and, after several more hours of nominal performance, completed the circuit with batteries fully charged. This second experiment served to illustrate how deviating from Sun-synchrony caused many operational difficulties.

3.1. Rover and Control Performance

3.1.1. Accuracy

A first test characterizing Hyperion's steering accuracy involved simply commanding a straight path and measuring error in the resulting ground track. Differential GPS was used to track the rover's path, with an expected accuracy of less than 5 cm. The tests were performed on slightly sloped ground composed of a compacted fine soil with rocks less than 20 cm high distributed throughout. In this test, Hyperion was commanded to travel at 0.3 m s^{-1}. Figure 19 shows plots of the rover's actual path, a line fit to the path and the error between actual and fit paths. The mean error magnitude of the actual path versus fit line is 6.0 cm, with a standard deviation of 3.2 cm. Further tests in the same area involved commanding steering arcs from 2.5–20 m radius for Hyperion to follow for approximately one complete circle.

Figure 20 depicts one such experiment for 20 m radius arcs, while Figure 21 depicts an experiment for a 2.5 m arc. For all but one experiment, the error in the fit radius to commanded radius was less then 3%; although in the 20 m right turn trial

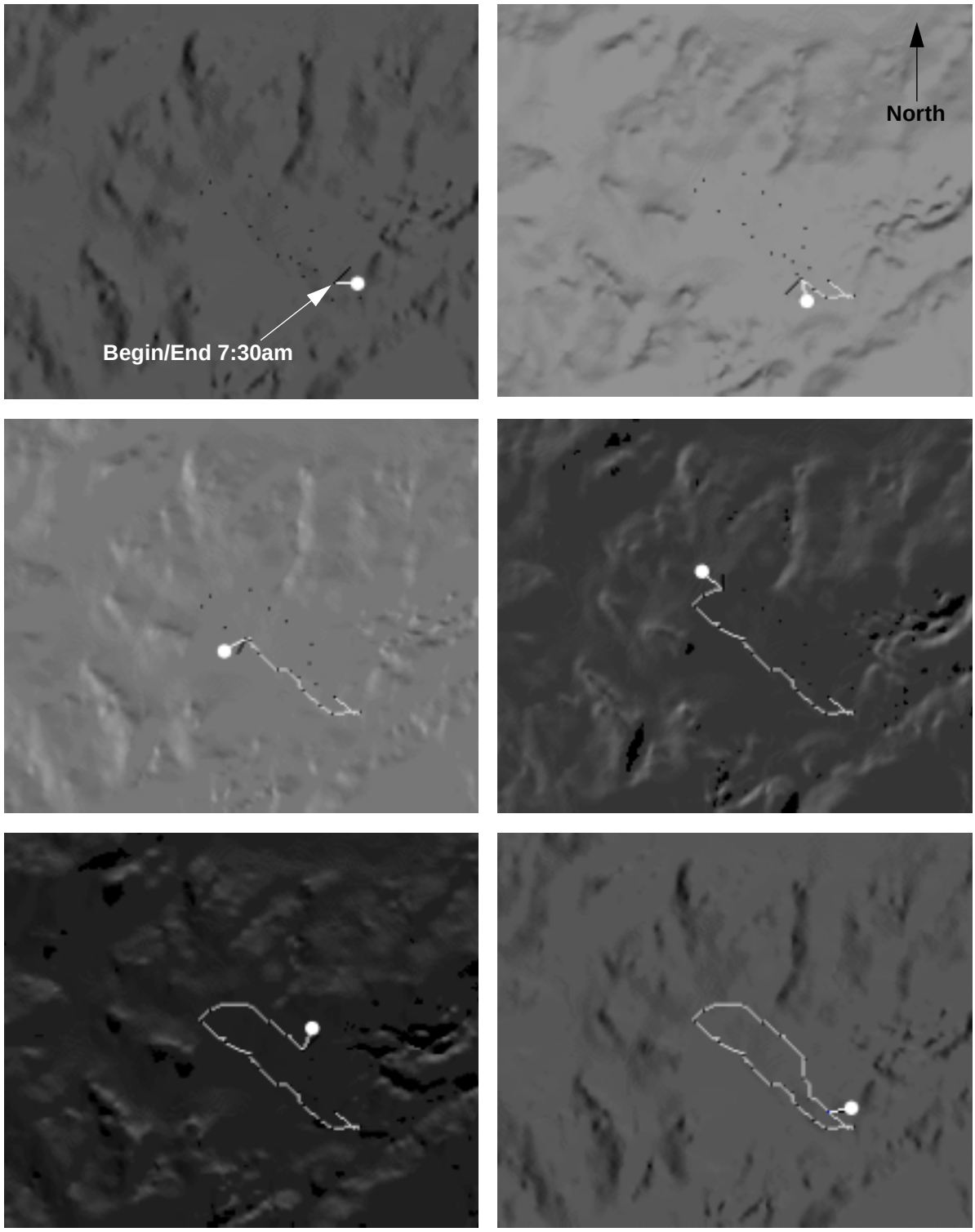


Fig. 17. Sun-synchronous plan for the first experiment, a 6.1 km, 24-h traverse, overlaid on maps of dynamic lighting. The grayscale backgrounds indicate Sun incidence angle, where lighter tone is higher-angle incidence, and black is full shadow. The images are "snapshots" of the plan, showing remaining intermediate goals (black dots), the executed path (white trace), and the rover position. The vectors emanating from the rover position are the Sun direction (circle-end vector) and the solar panel normal vector (black vector). Each pixel represents 25 m^2.

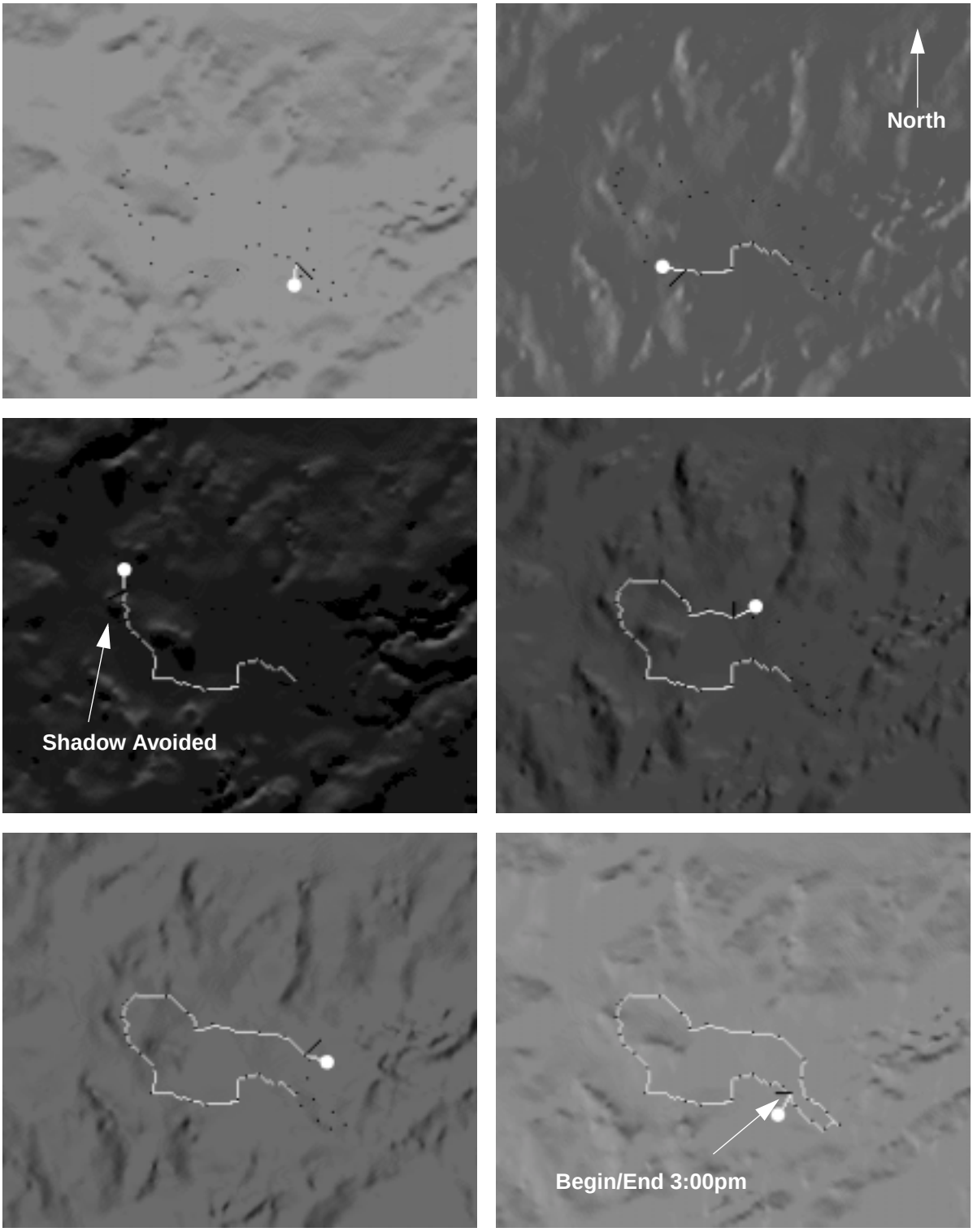


Fig. 18. Sun-synchronous plan for the second experiment, a 9.1 km, 24-h mission plan overlaid on maps of dynamic lighting. Note deviation in the traverse route to avoid area of full shadow. Notation as in Figure 17.

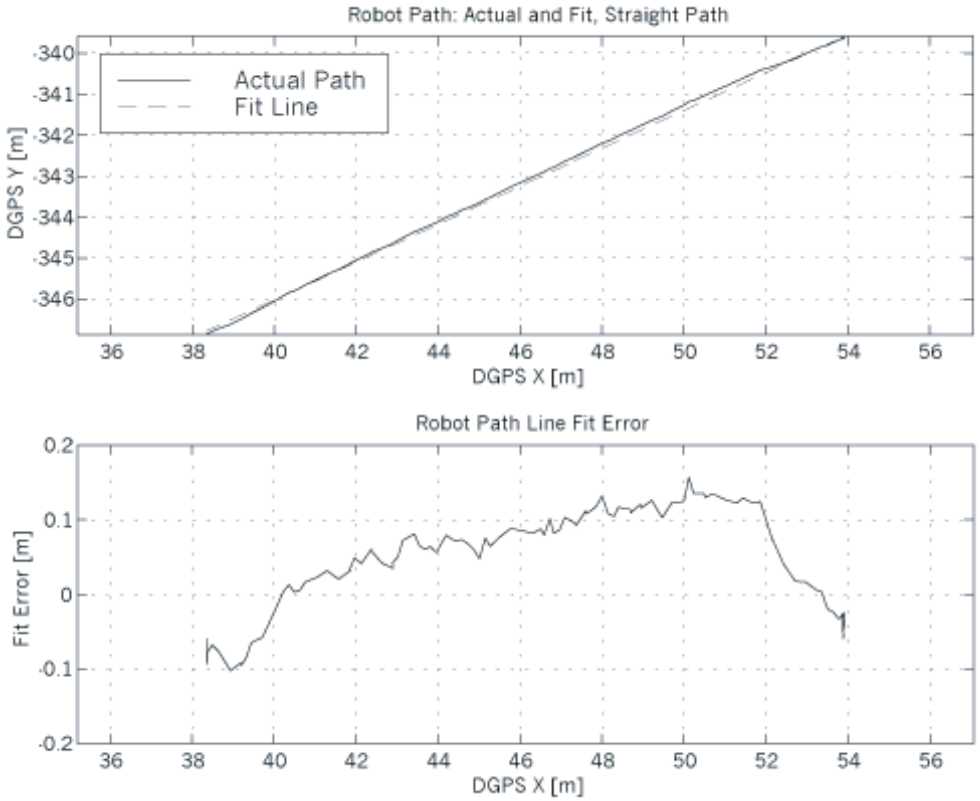


Fig. 19. Response of passive steering mechanism to a straight path in benign natural terrain (soil and rocks smaller than 0.1 m).

(Figure 20, right) an error of 11.8% occurred. Note that the position noise in Figure 21 comes from errors inherent to dGPS and does not represent actual rover motion.

3.1.2. Maneuverability

Maneuverability, meaning the ability of the Controller to provide responsive direction control, enables the robot to negotiate natural terrain. Hyperion's steering axle joint has a 90° range of mechanical rotation, enabling steering radii as tight as 2.5 m. In simulation, this is sufficient to maneuvering through terrain with obstacle densities of one obstacle per 10 m^2. This simulation assumed instantaneous steering response, so physical tests were needed to model latencies in Hyperion's steering control system.

In the steering latency experiments, Hyperion was commanded to travel at 0.3 m s^{-1} along an arc, again in representative terrains. Once in steady state, a new arc was commanded and steering axle angle data were recorded during the transition. An exponential curve was fit to these data to determine the time constant of the steering system and steady-state angle errors. Table 4 shows the time constants along with the steady-state fit angle and Figure 22 shows a typical response profile.

3.2. Solar Power Performance

Analysis of data obtained during the course of the experiment shows the total mean solar input to the power trackers to have been 144 W. This represents the time-averaged power generated by the solar panels. The total mean load power of the robot was 113 W. The relatively low mean load reflects the nature of the planned path, which included extended periods of charging during which the locomotion load was negligible. The difference of 31 W corresponds to power tracker conversion and battery inefficiency. The batteries were in the same fully charged state at the start and conclusion of each experiment.

3.3. Autonomy

Hyperion spent 90.0% of its time in the first experiment operating in autonomous mode, 9.9% of time in safeguarded teleoperation mode and 0.1% was in monitored teleoperation

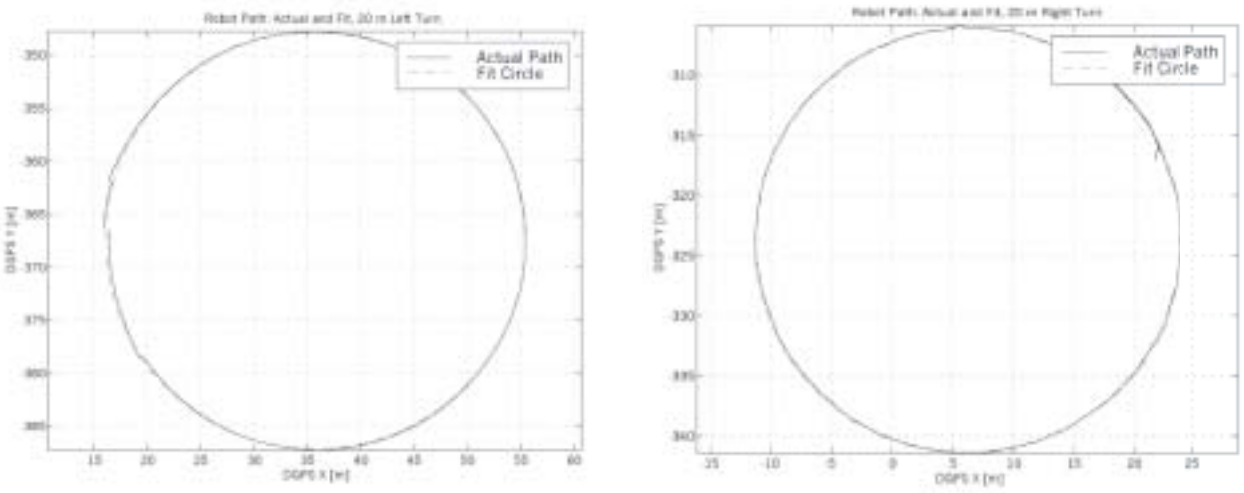


Fig. 20. Response of passive steering in 20 m radius turn, left and right, in natural terrain.

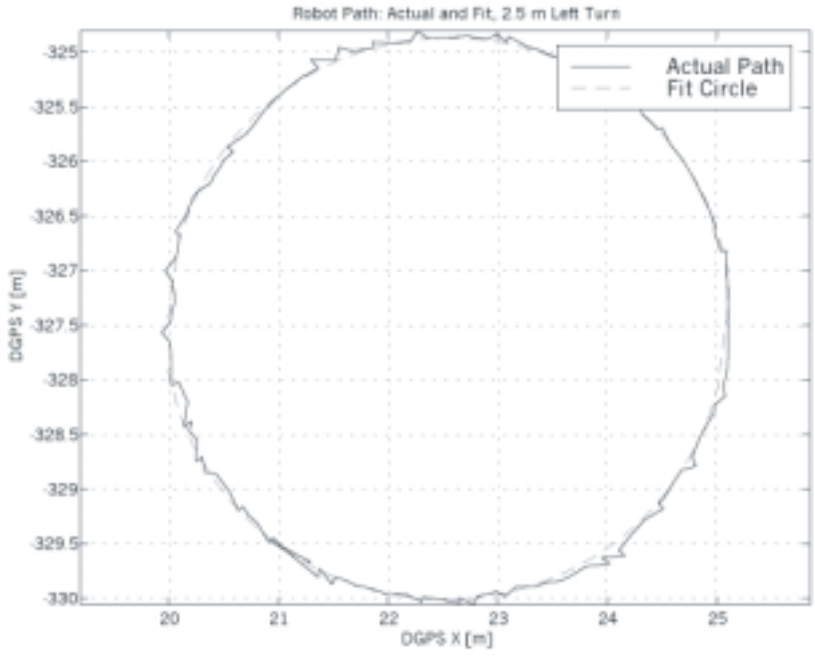


Fig. 21. Response of passive steering to 2.5 m radius left turn, in natural terrain.

Table 4. Time Constant and Steady-State Angle Fit in a Maneuverability Test of Hyperion's Steering Transition Latency

Initial Arc Command	New Arc Command	Fit Exponential Time Constant	Fix Exponential Steady-State Angle
Straight / 0.000 rad	10 m left turn / 0.198 rad	3.25 s	0.192 rad
10 m left turn / 0.198 rad	7.5 m left turn / 0.263 rad	3.11 s	0.253 rad
7.5 m left turn / 0.263 rad	5 m left turn / 0.388 rad	3.36 s	0.377 rad
5 m left turn / 0.388 rad	10 m left turn / 0.198 rad	3.42 s	0.186 rad
Straight / 0.000 rad	10 m right turn / –0.198 rad	3.04 s	–0.197 rad
10 m right turn / –0.198 rad	5 m right turn / –0.388 rad	3.75 s	–0.384 rad
5 m right turn / –0.388 rad	2.5 m right turn / –0.718 rad	3.53 s	–0.701 rad
2.5 m right turn / –0.718 rad	10 m right turn / –0.198 rad	3.05 s	–0.195 rad
Average:		3.31 s	0.008 rad / 0.46 degree absolute error

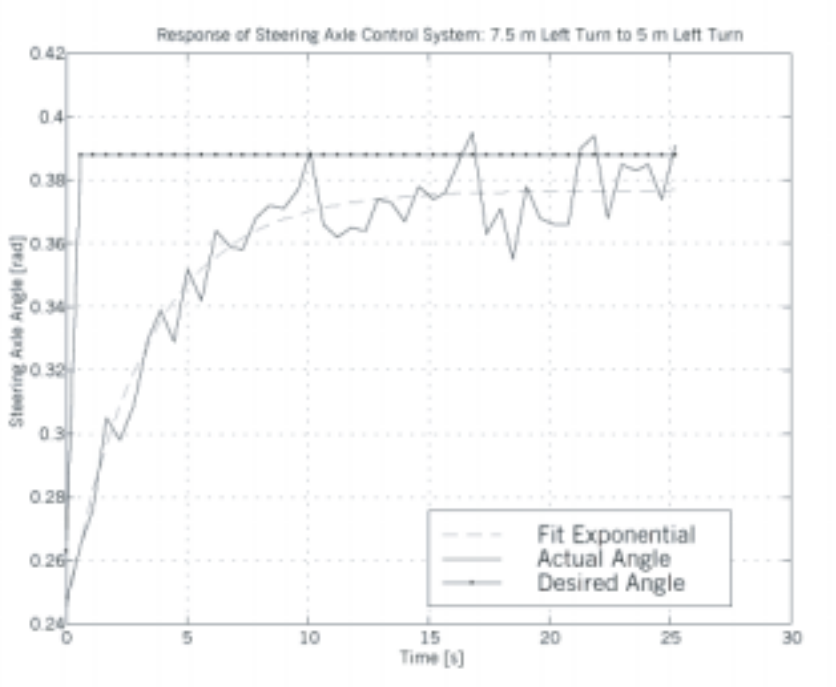


Fig. 22. Response of passive steering control in transition between 7.5 m and 5.0 m radius turn.

Fig. 23. Hyperion crossing flowing water, not expected on Mars.

mode. These measurements only take into account Hyperion's traverse activity, not time spent waiting at a waypoint for timing or recharging purposes.

As the results suggest, nominal waypoint navigation was performed autonomously. In autonomous mode, driving decisions were based on perception from stereo sensors and the location of Sun-synchronous mission goals. Operators were limited to two actions in autonomous mode: submitting goals and setting the nominal driving speed. An operator would generally decrease driving speed to 0.15 or 0.20 m s^{-1} if rocky terrain was perceived ahead; in retrospect this conservative measure was probably unnecessary.

Teleoperation was initiated in two situations: when traversing areas observed to be dangerous due to flowing water and when the Health Monitor triggered a fault. Remote operators detected dangerous areas primarily with images from the panoramic camera and streams and snow patches were typical concerns (Figure 23). During the course of the field experiments, operators never needed to intervene to stop Hyperion; its Health Monitor always detected the condition and changed operational mode, and stopped, as necessary. Laser-detected obstacles were the most common fault triggered by the Health Monitor. Less common faults included loss of heartbeats from processes, typically from temporary communication loss.

3.4. Obstacle Avoidance

The navigation software proved reliable and capable of avoiding obstacles in environments with obstacle density as high as 30%, meaning that 30% of the area the robot drove through was impassable. Figure 24 shows the display of detected obstacles on the approach to the waypoint in one path segment. On average, the robot maintained a heading error of less than 5° while traveling through terrain with an average obstacle density of 6.9%. Important to note in Figure 25, which shows the occurrence of the full range of non-zero heading errors, are several instances of large heading error (+/– 160°) where Hyperion was navigating to avoid obstacles and came fully about to get out of impassible regions.

3.5. Sun-synchronous Performance

The Mission Planner was successful in generating Sun-synchronous routes for Hyperion despite the degree to which terrain prevented an ideal circular route. Figures 17 and 18 depict the plans generated for the two 24-h Sun-synchronous navigation experiments. Each figure is a sequence of six "snapshots" of the plan, running from upper left to lower right across three rows. The snapshots show the terrain lighting (grayscale background; normal incidence is white, and shadows are black), the remaining intermediate waypoints, and the path executed so far. Each pixel in the image is a 25 m cell, corresponding to the DEM spatial resolution. The bottom-right snapshot in each sequence shows the completed route, with the rover in the start and finish position. Note that the snapshots also indicate, from the current rover position, the vectors to the Sun and of the solar panel normal. The solar panel normal appears 90° to the left of the driving direction, as with the Hyperion rover.

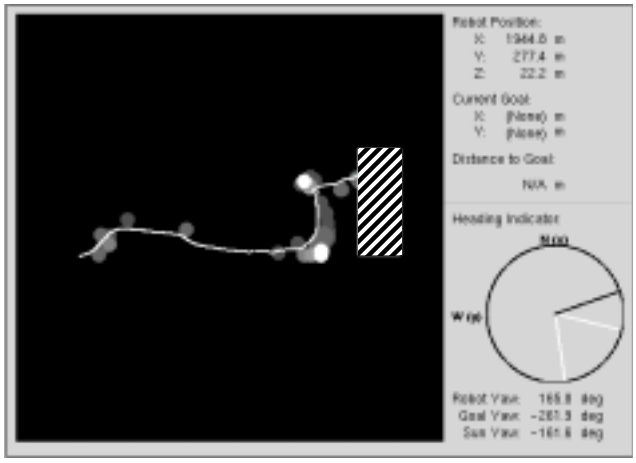


Fig. 24. Obstacle avoidance information display showing actual path taken to avoid detected hazards (shaded circles) on approach to a goal region (slashed). Note deviation over 90° to avoid impassible (white) obstacles.

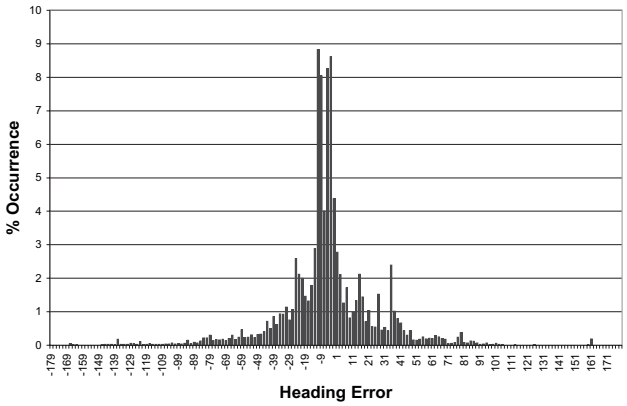


Fig. 25. Relative occurrence of heading error during the first 24-h continuous experiment.

The non-circular shape of the routes stems from natural terrain that confined driving. Streambeds ran along the outside of both diagonal legs, preventing excursions outside those bounds. The first experiment plan stopped short of a rocky promontory to the north-west end of the route, while the second experiment, largely over the same terrain, went all the way around the promontory. The field team selected the intermediate goal points to avoid terrain hazards below the scale of the elevation map for the region, at intervals of approximately 400 m. Hence, the Mission Planner had to plan the precise route between these positions and, more importantly, reason about how to best time the route to minimize the inevitable solar array mispointing forced by the non-circular circuit.

Figure 17 (top right) shows early progress of the plan for the first experiment. Note the Sun direction is aft of the solar array normal. Meanwhile, in Figure 17 (middle left), the Sun direction is forward of the solar array normal. These aft and forward biases reflect TEMPEST's ability to schedule the path to achieve the best average Sun angle over the path. The length of the north-west leg in these frames, coupled with a limitation in rover top speed, prevents an ideal Sun angle over the course. Therefore, TEMPEST biases the Sun aft at the beginning of the leg (Figure 17, top right) in anticipation that the Sun will overtake the rover near the center of the leg (middle left), and be biased ahead of the rover by an equivalent angle at the north-west end. A similar behavior occurs on the return, south-east leg of the traverse (Figure 17, middle right, bottom left, and bottom right).

The second experiment plan (Figure 18) was far longer than the first, and circled around the rocky promontory at the north-west end of the route. The behavior here was similar to that in the first experiment, except complicated by the highly concave route shape. The second experiment plan forced the rover to turn its solar array away from the Sun (for example, Figure 18, middle left) more frequently than for the first experiment. We suspect that these deviations, along with the greater overall plan distance, caused a greater sensitivity to the delays experienced during this plan's execution.

The histograms in Figure 26 illustrate the ability of the Mission Planner to maintain solar array Sun exposure in the first experiment plan (Figure 26(a)) and Hyperion's ability to do the same during execution (Figure 26(b)). Each depicts the angle between the projections of the Sun vector and solar array normal vector into the horizontal plane. A value of zero degrees indicates optimal pointing, while negative and positive values indicate Sun-aft and Sun-forward conditions, respectively. The plan histogram shows that the Mission Planner achieved optimal solar array pointing a large percentage of the time, but also that the irregular path forced other suboptimal configurations. The similarity between planned and executed pointing indicates the accuracy to which the mission plan was executed. Differences in these profiles are attributed to mispointing as a result of obstacle avoidance maneuvers taken by the Navigator. In the plan, the bias in Sun angles toward the aft of the robot is attributed to the Mission Planner's path selection criteria; all other things being equal, ties between path options are broken by selecting the earliest path opportunity. In the northern hemisphere, for which Sun-synchronous paths are clockwise, earlier timing will bias Sun angles aft.

4. Conclusions

Research in planetary rover technologies for autonomous traverse seeks to enable capabilities today that will be needed by robotic explorers in the coming decade (Bresina et al. 2001; Schenker et al. 2001; Miller et al. 2003). These technologies are moving forward into flight systems (Biesiadecki et al.

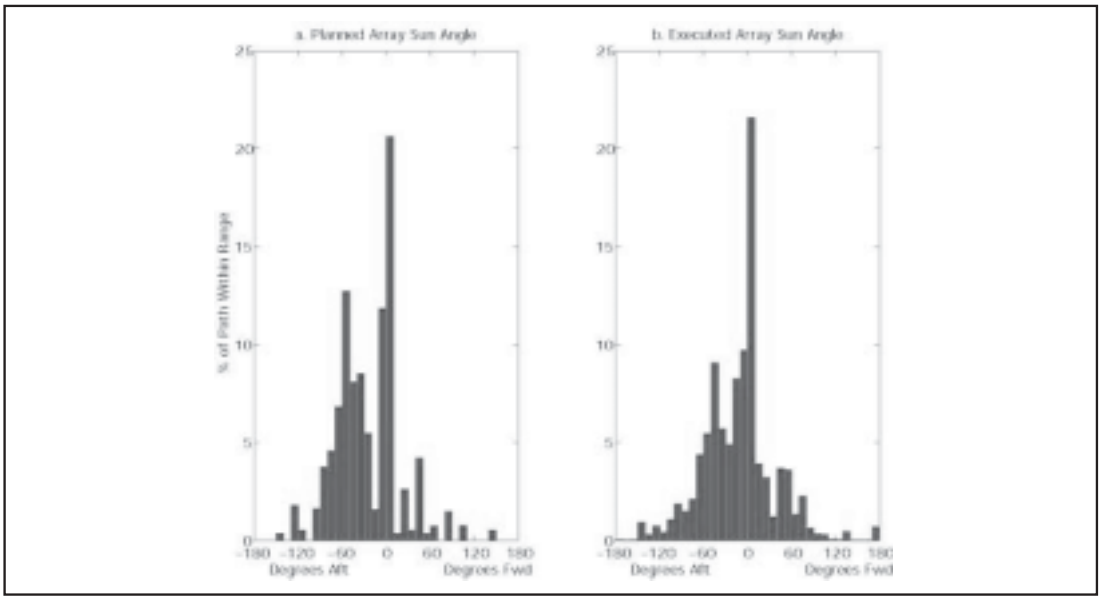


Fig. 26. Comparison of planned (a) and actual (b) solar array pointing performance shows close agreement. Plots show Sun-to-array pointing angle versus percentage of the traverse at that angle. Dominant angle is 0° with the path putting the array directly in line with the Sun and the bias is with an aft pointing angle, indicating the rover is typically ahead of the Sun.

2001) as we continue to create new capabilities like those for Sun-synchronous navigation.

4.1. Future for Sun-synchrony

Capitalizing on solar energy is an opportunity for robotic operation on all the inner planets. On the Moon and Mercury in particular, mitigation of extreme temperature changes is important and another objective in reasoning about solar flux. These factors and the abundance of solar energy argue that Sun-synchrony could enable missions by rovers lasting months or years.

At mid-latitudes Sun synchrony means traveling opposite to planetary rotation. On Earth, equatorial Sun synchrony is not feasible due to high speeds, and therefore power, required. On Mercury the solar insolation is nine times higher than on Earth, gravity is one-third that of Earth and planetary rotation takes 176 Earth days. In theory, a rover circumnavigating Mercury's equator requires a small solar array and a nominal speed of only 4 km h^{-1} on average, although conditions on the surface would likely dictate a higher but still moderate peak speed.

Table 5 considers several hypothetical planetary missions. For each case, the map distance, average speed, average power, and required solar array size are calculated. The required average speed is calculated from the traverse's map distance and the diurnal period of the Sun through the sky. Actual traverse distance is usually 1.1 to 3.0 times longer than the map distance. The power required is from an idealized calculation given by the equation

$$P = m_{rover} g v r_{soil} + 50W.$$

The constant of 50 W is an assumed value that includes the constant power for all systems such as computing and communication except locomotion. For this comparison, a rover mass of 100 kg and a soil resistance of 0.1 are assumed. The solar panel area is estimated by

$$A = \frac{P}{k_{efficiency} E_{irradiance}}$$

with an assumed efficiency of 20%. These simplified equations reveal the effects of gravity and speed on required power, and highlight the scenarios under which Sun-synchronous solar power may be feasible.

At high latitudes, the traverse speed decreases with the effective planetary circumference. At 80° latitude on the Moon, a rover must travel at 3 km h^{-1} to track the Sun. The effective circumference is large (1895 km) but the insolation is high (1368 W m^{-2}), the gravity is weak (1.6 m s^{-2}) and the rotational period is long (27.3 d), combining for a viable solar-powered mission (Figure 27).

A region of continual sunlight exists seasonally on planets with axial tilt. At high latitudes on Earth and Mars, continuous direct sunlight occurs seasonally with a duration that is dependent upon latitude. In the polar regions above arctic circles, a robot's solar panel must daily sweep 360° either

Table 5. Navigating in Synchrony with the Sun in the Dawn Terminator in the Lunar South Polar Region Has Advantages to Power and Thermal Regulation

	Mercury	Earth	Moon	Mars
Equatorial				
Distance (km)	15327	40074	10914	21344
Speed (km/h)	3.6	1670	15	864
Power (W)	87	45412	119	8930
Solar array area (m^2)	0.04	227	0.4	75.8
Polar, 80° latitude				
Distance (km)	2662	6959	1895	3706
Speed (km/h)	0.6	290	3	150
Power (W)	56	7927	62	1598
Solar array area (m^2)	0.03	40	0.2	13.6
Circuit, 5 km radius				
Distance (km)	31	31	31	31
Speed (km/h)	0.01	1.3	0.04	1.3
Power (W)	50.1	85	50.2	63
Solar array area (m^2)	0.03	0.4	0.18	0.5

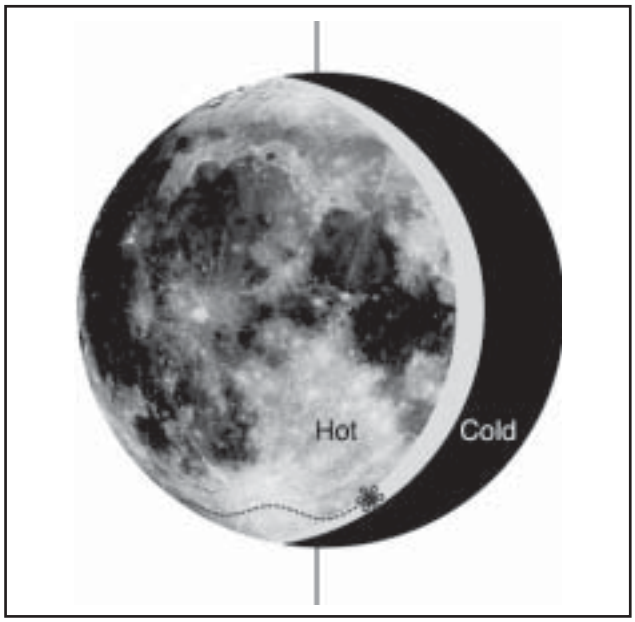


Fig. 27. Navigating in synchrony with the Sun in the dawn terminator in the lunar south polar region has advantages to power and thermal regulation.

through rotation or by following a spiraling path to maintain Sun-synchrony. During the arctic summer of the Earth or Mars, traverses by rovers with fixed, vertically-deployed solar arrays could circumnavigate a feature with a 5 km radius at an average speed of 1.3 km h^{-1}. The possibility of a spiraling path that explores a wide swath of terrain is intriguing.

With a concurrence of features such as moderate temperatures, extended periods of sunlight and the possibility of *in situ* volatiles, polar regions of moons and planets offer excellent opportunities for long-term missions. Sun-synchronous exploration enables coverage of vast regions far from a landing site.

4.2. Lessons for Robotic Exploration

The performance of the Hyperion system was sufficient for the situations encountered in field experimentation but certainly there is opportunity for improvement and refinement for more demanding environments, like those of space. We learned important lessons (sometimes repeatedly) through this experience.

Sun-synchrony is sufficient and necessary. The two 24-h navigation experiments successfully demonstrated the benefits of Sun-synchrony. The first experiment proceeded nearly flawlessly, suggesting that the Sun-synchronous method was sufficient for operations in the Arctic environment. The second experiment, through its operational difficulties, indicated that a Sun-synchronous strategy is necessary, as well as sufficient, given the scenario and Hyperion's constrained ability to point its solar array. The robot's performance degraded as it fell behind schedule, but became sustainable again once it re-synchronized with the plan. As noted, Hyperion's fixed-orientation solar array was beneficial to rover mass and

complexity but a worst-case condition for vehicle path planning. Sun-synchronous navigation enabled Hyperion to operate over many kilometers, and returned to its starting point with full batteries in each case. For rovers with actuated solar arrays, the power-cognizant planning techniques are still relevant in terms of considering power production and consumption. Secondary benefits come from the ability to predict and avoid shadowed regions and to mitigate any failures in the actuation of an actively pointed solar array.

Specify goals. For this work we found it beneficial to have a specific project goal that we could measure progress against and use to make decisions about which research activities would be necessary and which were tangential. We reduced the motivations and technical interests down a concept scenario: "a solar/battery powered rover will traverse a complete Sun-synchronous circuit in 24 h." Our requirements for essential research and design for field experiments followed from this goal.

Simplify design. We strived for a simple mechanical design, with a few innovations such as the steering method, and were rewarded with reduced development time and reliability in the field. Similarly, the fixed-orientation solar array on Hyperion reduced actuated motions and supporting mechanical structure. Electronically, we used the opportunity of a large volume for the enclosure to spread out, organize components and make clean connections. In software, we focused on the essential not the ultimate features. The message here is mixed, however, as simplifications can have some surprising results such as the steering instability that must be controlled in order to drive backwards and the (intentional) complexity introduced in planning Sun-synchronous paths for a fixed solar-array rover.

Prototype concepts. Building quick prototypes contributed to success. In particular, we built scale models of chassis design concepts and a full-size rolling chassis with the passive steering mechanism and a layout of all electronic components. These were fabricated quickly using stock parts, bicycle components, and wood, but the verification provided allowed us to proceed with confidence. Similarly, we prototyped the software architecture by stubbing out modules still in development and we tested the navigation and planning algorithms with synthetic data to confirm feasibility under simulated conditions.

Buy, don't build. We were well served by using commercial off-the-shelf hardware and software, including computing elements, wireless communication, motor controllers, and operating systems. We used publicly distributed device drivers, messaging, filtering, and machine vision software libraries rather than implementing anew. Again, this sped development and improved reliability. It is hard to match the engineering and manufacturing skill that has been applied to lightweight, durable mountain-bike wheels, so why expend effort trying to reinvent when there is a wheel that works. Our approach is to build only where necessary and focus on research issues.

Find helpful tools. Glare from direct sunlight not withstanding, Hyperion's stereo vision implementation proved very capable of detecting obstacles and keeping the rover safe without excessive steering activity. Tuning stereo vision for optimal performance can be very time-consuming and sensitive to changing environmental conditions. This task was simplified significantly with the development of a stereo configuration interface that not only displayed range data in three dimensions but simultaneously superimposed the results of traversability analysis onto the three-dimensional terrain (Figure 12). This significant innovation provides the vision system developer with superior understanding of how well the entire perception system is functioning. Problems could be diagnosed relatively quickly since the interface provides insight into all components of the stereo perception system.

Avoid the rocks first. It is always important to get the basics solid. In this case it was the obstacle detection and avoidance software that made it possible to navigate long distances without crashing into anything. In the two integrated experiments, over 15 km of traverse, Hyperion only once encountered an obstacle that exceeded its capability (and this was by a rear wheel during a steering maneuver). In this regard, the optimistic/pessimistic strategy for evaluating terrain worked well at maintaining continuous exploration.

Some plans will fail. The Mission Planner showed that a solar-powered rover could meet the simultaneous constraints of local navigation and global resource optimization. The ability to replan as the robot falls behind or ahead of schedule is needed to add robustness. In the second integrated experiment, we had to rely on operator guidance to reschedule but not redirect the original plan, where eventually a more sophisticated Mission Executive might handle the situation. Fundamentally the route could be completed Sun-synchronously but real-time replanning is likely as important as initial planning. A more capable Mission Executive and run-time replanning with TEMPEST are current developments.

Design fail-safe. Both plans and components will fail and in any case the system should tend to safe itself and await rescue. We designed the Health Monitor to detect faults and when possible to recover but at least to stop the rover and wait. While the ability of the robot to autonomously operate was effective, there were times when teleoperation was required and robot initiated human teleoperation (supervisory control) functioned smoothly. This was particularly important for a small team on 24-h experiments, which with set-up and shut-down typically ran 40 h, since everyone had a hand in operating the system and monitoring progress.

Design for the field. Calculations of wind pressure on a vertical solar array are much more important in the field than in the lab. In our first outing near Pittsburgh we were hit by a 13+ m s^{-1} (30 mph) strong breeze that lifted Hyperion and collapsed its rear wheel sidewall. Calculations following showed that the vehicle was stable against 11 m s^{-1} (25 mph) lateral winds. On Devon Island the peak wind speed was just 7 m s^{-1}

(15 mph) which is a moderate breeze but not a cause for concern given our design. Similarly, sealing the robot against moisture and regulating internal temperature require attention. We successfully sealed Hyperion but we lost quick and ready access to components which were enclosed, sealed, and secured (with waterproof tape). These types of issues are not cutting-edge research but attention to these details are crucial to enable research validation in the field.

Verify in the field. An important technical result of field rather than laboratory experimentation is that system models are verified in actual operation. In retrospective analysis, it appears that Hyperion's solar power system was overdesigned. The solar flux experienced was greater than global flux models predicted and Hyperion used less power than our worst-case scenario; it was well within its design margin. In the first integrated experiment, Hyperion effectively ran from its solar power alone and rarely dipped into its batteries for peak power. In the second experiment, despite a 3 h delay and with batteries nearly depleted during the "night", the remaining 12 h were sufficient to build reserves and regain the schedule. As we move forward we are refining the models of solar power generation and vehicle locomotion performance and actually reducing the solar-array size while increasing rover speed. This is one of many insights only possible with field testing.

Get comfortable. Field experimentation, for all the benefits to verification and validation in the relevant environment, introduces certain difficulties. Logistics become a factor in experiment design and success. This can even impact rover design, for example mass and size limitations for transport by light airplane and component packaging for assembly in the field. As well, keeping everything organized, from labeling all the cables to who should get a nap, sometimes dominates the technical agenda when it comes to overall experiment success. Most important is ensuring everyone stays calm, well rested, and warm (or cool).

The 24 h Sun-synchronous exploration experiments showed the feasibility of continuous operation of a solar-powered rover. The ability to operate Sun-synchronous and maintain the necessary power levels was clearly demonstrated. The capability for resource-cognizant exploration, like that used for Sun-synchronous navigation, is important to robotic exploration. It is the basis of creating robotic agents to conduct geologic and biologic investigations. (Atacama, see http://www.frc.ri.cmu.edu/atacama). These rovers will increasingly become tools for scientists at Earth's extremes and beyond.

Acknowledgments

We acknowledge and sincerely thank Dimi Apostolopoulos, Jesse Boley, M. Bernardine Dias, Stewart Moorehead, Ben Shamah, Reid Simmons, Sanjiv Singh, Tony Stentz, James Teza, Vandi Verma, and David Wilkinson for their contribution to this work. Field experimentation was conducted in collaboration with the NASA Haughton Mars Project, Pascal Lee, Principle Investigator. This work was supported by NASA under grant NAG9-1256, Chris Culbert, Program Manager. This paper describes the work of the Sun-synchronous Navigation project at Carnegie Mellon University. Opinions expressed are those of the authors and not necessarily those of NASA or the NASA program manager.

References

Acton, C.H. Jr. 1996. Ancillary data services of NASA's navigation and ancillary information facility. *Planetary and Space Science* 44 (1):65–70.

Bares, J., and Wettergreen, D. 1999. Dante II: technical description, results, and lessons learned. *International Journal of Robotics Research* 18(7):621–649.

Bernard, D. et al. 1999. Spacecraft autonomy flight experience: the DS1 remote agent experiment. *Proceedings of the AIAA*, Albuquerque, NM.

Biesiadecki, J. et al. 2001. The Athena SDM Rover: a testbed for Mars rover mobility. *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space (i-SAIRAS 2001)*, Montreal, Canada, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18–22.

Bresina, J.L., Bualat, M.G., Edwards, L.J., Washington, R.M., and Wright, A.R. 2001. K9 operations in May 2000 Dual-Rover Field Experiment. *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space (i-SAIRAS 2001)*, Montreal, Canada, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18–22.

Bussey, D.B.J., Robinson, M.S., and Spudis, P.D. 1999. Illumination conditions at the lunar poles. *Lunar and Planetary Science* 30, abstract 1731, Lunar and Planetary Institute, Houston, TX, March.

Chien, S. et al. 2000. ASPEN – automating space mission operations using automated planning and scheduling. *SpaceOps 2000*, Toulouse, France, June.

Kelly, A., and Stentz, A. 1998. Rough terrain autonomous mobility. Part 1: a theoretical analysis of requirements. *Autonomous Robots* 5:129–161.

Laubach, S., and Burdick, J. 1999. An autonomous sensor-based path-planner for planetary microrovers. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Detroit, MI, May.

Manavalan, P., and Wagner, M.D. 2001. A data-rate aware telemetry scheduler. CMU-RI-TR-01-12, Robotics Institute, Carnegie Mellon University, May.

Martin-Alvarez, A. et al. 1999. Fuzzy reactive piloting for continuous driving of long range autonomous planetary micro-rovers. *Proceedings of the IEEE Aerospace Conference*, Snowmass, CO.

Miller, D.P., Hunt, T., Roman, M., Swindell, S., Tan, L., and Winterholler, A. 2003. Experiments with a long-range planetary rover. *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2003*), Nara, Japan, May 19–23.

Schenker, P.S. et al. 2001. FIDO: a field integrated design & operations rover for Mars surface exploration. *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space (i-SAIRAS 2001)*, Montreal, Canada, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18–22.

Seraji, H. 1999. Traversability index: a new concept for planetary rovers. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Detroit, MI, May.

Seraji, H. et al. 2001. Safe navigation on hazardous terrain. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, May.

Shamah, B., Wagner, M.D., Moorehead, S., Teza, J., Wettergreen, D., and Whittaker, W. 2001. Steering and control of a passively articulated robot. *Proceedings of SPIE, Sensor Fusion and Decentralized Control in Robotic Systems IV.* Vol. 4571. October, Boston, MA.

Shrounk, D. 1995. Sun-synchronous operation. Workshop discussion, ISE Lunar Conference.

Singh, S. et al. 2000. Recent progress in local and global traversability for planetary rovers. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, April 24–28.

Stentz, A. 1995. The focused D* algorithm for real-time planning. *Proceedings of the International Joint Conference on Artificial Intelligence*, August, Montreal, Canada.

Stentz, A. 2004. Optimal incremental search for high-dimensional, constrained path finding problems. Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-04-39.

Tompkins, P., Stentz, A., and Whittaker, W. 2001. Automated surface mission planning considering terrain, shadows, resources and time. *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space (i-SAIRAS 2001)*, Montreal, Canada, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18–22.

Tompkins, P., Stentz, A., and Whittaker, W. 2002. Mission planning for the Sun-synchronous navigation field experiment. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15.

Urmson, C., and Dias, M. 2002. Stereo vision based navigation for Sun-synchronous exploration. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15.

Verma, V., Langford, J., and Simmons, R. 2001. Non-parametric fault identification for space rovers. *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space (i-SAIRAS 2001)*, Montreal, Canada, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18–22.

Volpe, R., Nesnas, I., Estlin, T., Mutz, D., Petras, R., and Das, H. 2001. The CLARAty architecture for robotic autonomy. *Proceedings of the 2001 IEEE Aerospace Conference*, Big Sky, MT, March.

Wettergreen, D., Shamah, B., Tompkins, P., and Whittaker, W. 2001. Robotic planetary exploration by Sun-synchronous navigation. *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space (i-SAIRAS 2001)*, Montreal, Canada, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18–22.

Wettergreen, D., Dias, B., Shamah, B., Teza, J., Tompkins, P., Urmson, C., Wagner, M., and Whittaker, W. 2002. Experiments in Sun-synchronous navigation. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15.