Steering and Control of a Passively Articulated Robot

Benjamin Shamah, Michael D. Wagner, Stewart Moorehead, James Teza, David Wettergreen, William “Red” Whittaker
Field Robotics Center, The Robotics Institute, Carnegie Mellon University

ABSTRACT

The need for light weight yet highly mobile robotic platforms is driven by the limitation of available power. With unlimited energy, surface exploration missions could survive for months or years and greatly exceed their current productivity. The Sun-Synchronous Navigation project is developing long-duration solar-powered robot exploration through research in planning techniques and low-mass robot configurations. Hyperion is a rover designed and built for experiments in sun-synchronous exploration. This paper details Hyperion’s steering mechanism and control, which features 4-wheel independent drive and an innovative passively articulated steering joint for locomotion.

1. INTRODUCTION

Robotic explorers on the planets and moons are fundamentally limited by power, specifically power required for locomotion. With unlimited solar energy, surface exploration missions could survive for months or years and greatly exceed their current productivity in terms of area explored and science objectives achieved. The Sun-Synchronous Navigation project at Carnegie Mellon University is seeking to reduce the required power of these robotic explorers while adding the ability to intelligently gather solar power.

Sun-synchronous 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[6]. The robot must estimate how much energy is required to achieve its goals and plan a course of action in position and time that optimize its use of available power. It is a technique that involves tracking the sun while exploring terrain and is accomplished by traveling opposite to planetary rotation in synchrony with the sun to always remain in sunlight. Robotic explorers on the planets and moons can employ sun-synchronous navigation to maintain continual exposure to sunlight in order to acquire the power necessary to sustain operation for extended periods of time. By travelling at speeds appropriate to their latitude and navigating to avoid shadows cast by local terrain, these solar-powered robots will be able to operate for extended periods of time.

We have applied the concept of sun-synchronous robotic exploration in the Canadian Arctic where the sun-angle

Figure 1: The Hyperion rover on Devon Island, Canada.

Table 1: Hyperion specifications

mass	157 kg
size	2 x 2.4 x 3 m
speed	30 cm/s nominal
power	90 W computing and sensing 0 - 150 W locomotion

bshamah@ri.cmu.edu; phone 1 412 268-1202; fax 1 412 268-5895; http://www.frc.ri.cmu.edu; The Field Robotics Center, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh PA 15213, {mwagner, sjm, jpt, dsw, red}@ri.cmu.edu .

is low and a nearly vertical solar array is necessary. We have shown that at polar latitudes rovers can maintain continual exposure to solar insolation sufficient for sustained operation and have demonstrated this during several 24 hour experiments conducted in July 2001 on Devon Island, Canada. Powered by a 3.45 m^2 solar panel in the persistent daylight of the high Canadian Arctic, Hyperion showed the feasibility of long-duration, solar-powered exploration of terrestrial polar regions[5].

The prime innovation of the Hyperion chassis is the implementation of a passively articulated steering system. It uses 4-wheel independent drive and a passively articulated steering joint for locomotion. This steering configuration offers the mechanical simplicity of traditional skid steering vehicles while maintaining the steering performance of an articulated frame vehicle. The absence of a steering actuator further reduces mass. However, the trade-off in creating such a mechanically simple steering mechanism lies in added complexity of the control system. We will evaluate the locomotion system of Hyperion in terms of steering accuracy, maneuverability, reliability and suitability as a robotic platform with experimental results from its expedition to Devon Island and from other test sites.

This paper discusses the locomotion system design, detailing the configuration trade-offs and the final design. The control system for steering the passive articulation is described and its performance is analyzed.

2. LOCOMOTION SYSTEM DESIGN

2.1 Configuration

The Hyperion rover was configured to perform multiple 24 hour sun-synchronous routes. Configuration parameters were defined such that the vehicle would be able to demonstrate a meaningful sun-synchronous traverse. An initial configuration study that examined the available solar power determined that a speed of 1km/hr and a mass of 125 kg was feasible which includes enough margin for variable conditions. Several vehicle concepts were investigated, which led to the design of a combination of skid steering and axle articulation. The idea being to leverage on a simple (and lightweight) mechanical design while maintaining an efficient steering scheme.

One way to reduce vehicle mass is to limit the number of actuated motions. During the configuration stage various steering and drive modes were studied to determine the best combination of mobility and simplicity required to traverse the desired terrain. Independent, Ackerman, axle, and skid steering configurations were studied[2,1]. Each mode has advantages in certain areas as shown by Table 3.

Independent steering explicitly articulates each of the wheels to the desired heading. Apart from the issues of actuation complexity and accuracy of coordination control, this scheme provides advantages to the maneuverability of mobile robots, especially those operating in unprepared terrains. A common variation of independent all-wheel steering, not attainable by the other schemes, is crab steering in which all wheels turn by the same amount in the same direction. As a result, the robot can move in a sideways fashion. Coordination of driving and steering allows efficient maneuvering and reduces the affect of internal losses due to actuator fighting.

The most common type of steering on passenger cars is Ackerman steering which mechanically coordinates the angle of the front two wheels. In order to maintain all wheels in a pure rolling condition during a turn the wheels need to follow curved paths with different radii originating from a common center[8]. Advantages of explicit steering include more aggressive steering with better dead reckoning (due to less slip of the wheels) and lower power consumption. The downside of explicit steering is a higher actuator count, part count, and the necessary swept volume.

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

Skid steering can be compact, light, require few parts, and exhibit agility from point turning to line driving using only the motions, components, and swept volume needed for straight driving. Skid steering is achieved by creating a differential thrust between the left and right sides of the vehicle thus causing a change in heading. The downside is that skidding causes unpredictable power requirements because of terrain irregularities and non-linear tire-soil interaction. Skid steering also fails to achieve the most aggressive steering possible which can be achieved with explicit steering because the maximum forward thrust is not maintained during a turn.

Table 2: Hyperion rover locomotion configuration specifications

driving	speed	0 - 1 km/hr
	distance	up to 24 km in 24 hours
obstacles	max positive / negative	30 cm
slope	max pitch	20 deg
	max roll	20 deg
miscellaneous	ground clearance	40 cm
	terrain - firm sand to ice	rolling resistance 0.2 - 0.3

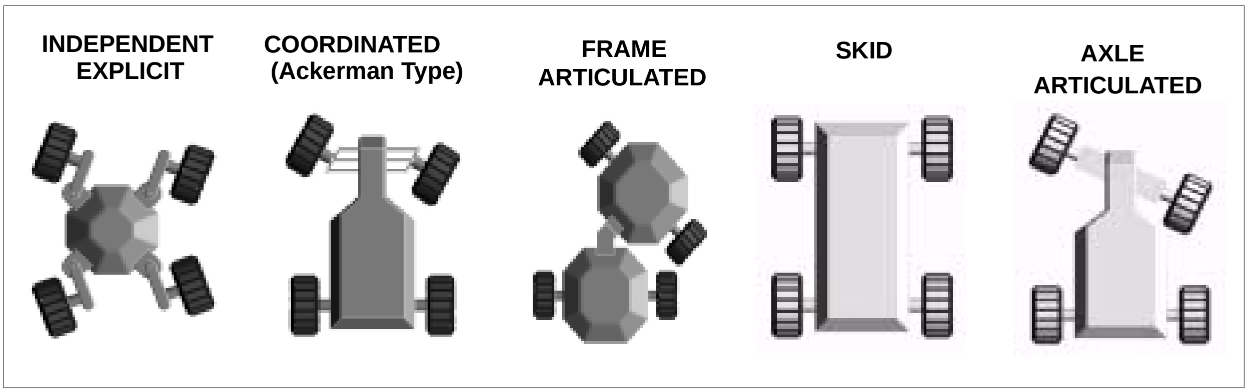


Figure 2: Kinematics of major steering schemes.

Table 3: Steering evaluation for 4 wheel rovers

	INDEPENDENT EXPLICIT	**COORDINATED ACKERMAN**	**FRAME ARTICULATED**	**SKID**	**AXLE ARTICULATED**
maneuverability	med / high	med	med	high	med
mechanical complexity	med	med/high	low	low	low
control complexity	low	med / low	med	low	med / high
drive power during steering maneuvers	med	med / low	med	high	low
number of actuated joints for steering	4	1	1	0	0

Passively articulated axle steering is performed by adding a free pivot to one of the vehicle axles. This type of steering is common in wagons or carts. One disadvantage of single axle steering is that the wheels run in separate tracks when going around curves. Under difficult ground conditions this requires increased drive propulsion as each wheel is driving over fresh terrain[2]. The advantages include mechanical simplicity, relatively low steering power, and moderate maneuverability. The innovation of the design implemented on Hyperion is to electronically control the velocity of the front wheels to maintain a desired angle of the front axle.

2.2 Detail Design

The philosophy behind the Hyperion chassis design was to utilize simple assemblies (identical where possible) that could be easily assembled in the field. The chassis is composed of 7.6 cm diameter aluminum tubing in a rigid "T" structure. All modules are clamped to the tubing which allows the design to be easily modified. An added benefit to the steered front axle is the ability to point the perception sensors with the front axle. The steered front axle also carries a mast with all of the perception sensors including: stereo cameras, a laser range finder, and a panoramic camera. This type of sensor pointing increases the effective horizontal field of view[3] resulting in more robust autonomous navigation.

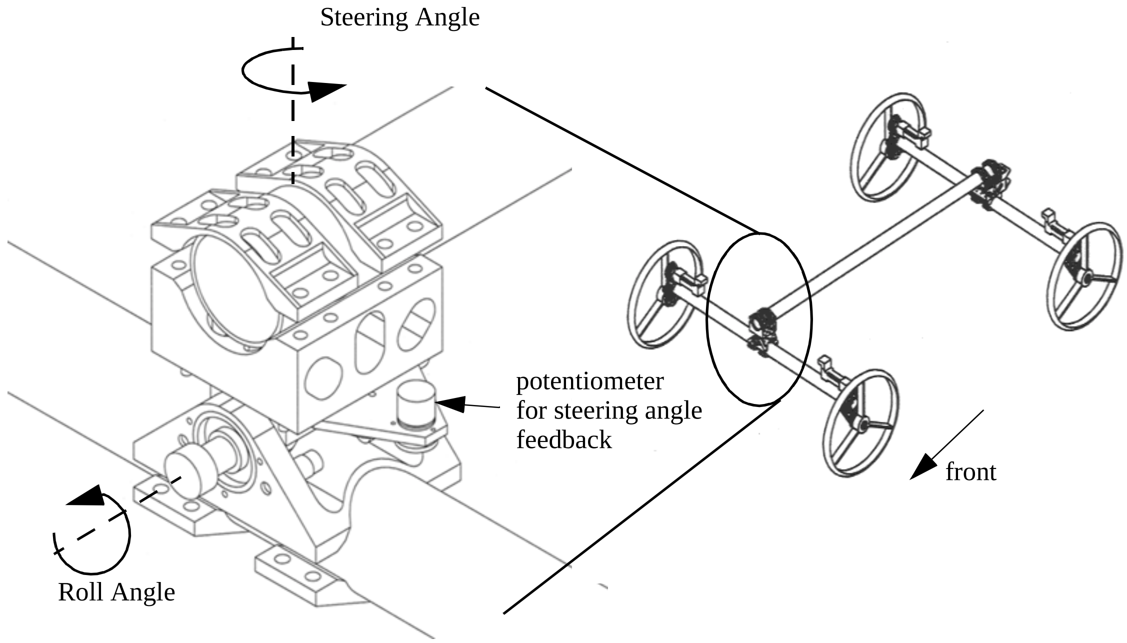


Figure 3: Detail of the front axle steer and roll pivot

The passively articulated steering joint is composed of two free rotations. The first is about the vertical axis which allows the change in heading of the front axle. The second rotation allows a roll motion of the front axle which is necessary to enable all four wheels to contact the ground over rough terrain. A spring suspension is not used because Hyperion travels at relatively low speeds.

Each wheel of the four identical wheel modules is composed of a motor, gearhead, and chain drive. A brushless DC motor (Kollmorgen Goldline part # mt1506b) is used to provide a maximum continuous torque of 1.5 Nm at 6.25 amps. Motor feedback is provided by a 2048 line per revolution incremental encoder. The output of the motor is connected to a harmonic drive (Harmonic Drive Systems part #: CSF-25-80-2A-GR-IV) 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 primarily because of the high quality of design and the extremely light weight.

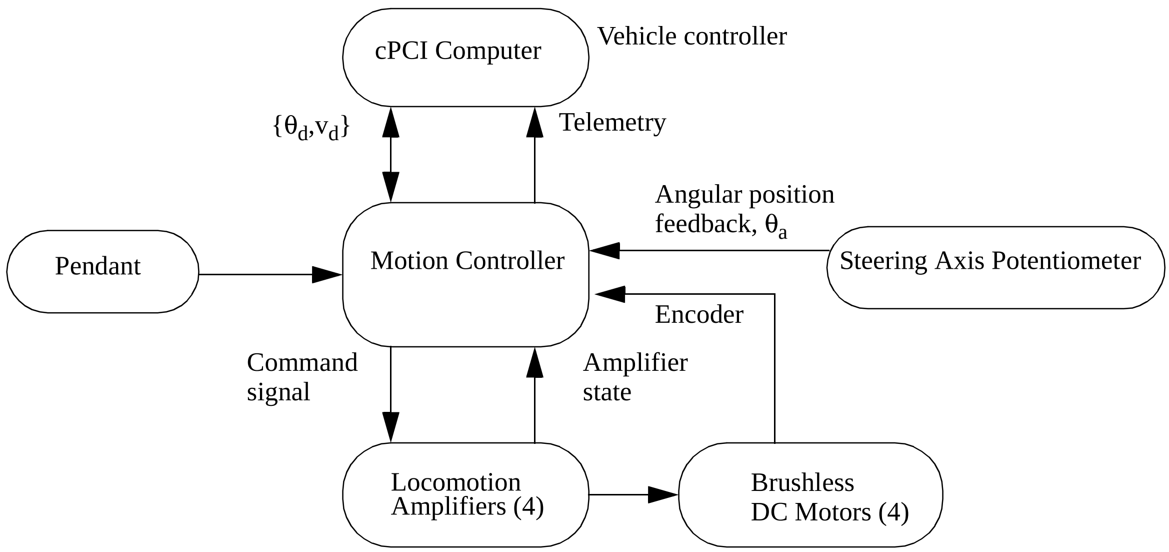


Figure 4: Locomotion hardware architecture

3. CONTROL SYSTEM DESIGN

The mechanical simplicity of Hyperion's underactuated chassis presents challenges to its motion control system. To support teleoperation and autonomous driving, mobility and steering must be:

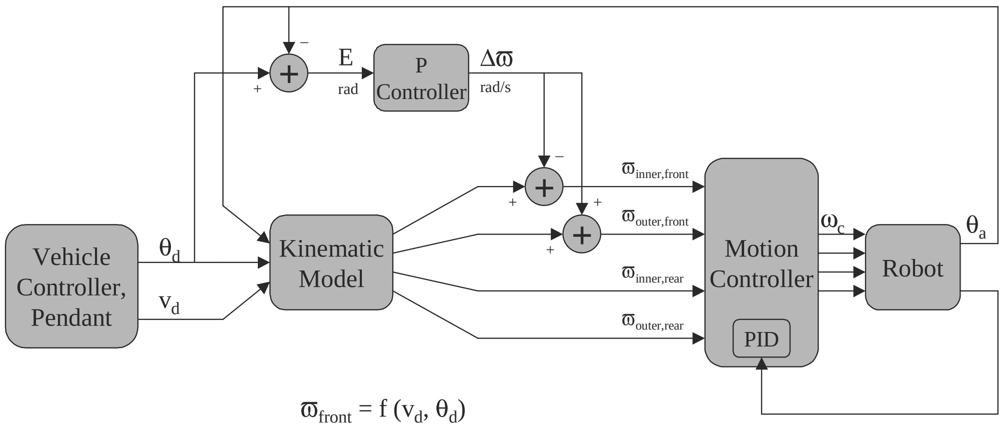
- *Accurate* so mobility commands are carried out as expected.
- *Maneuverable* to enable capable avoidance of obstacles. Furthermore, maneuverability must be well defined, meaning that it can be mathematically described to the higher level autonomy software. This makes path planning more predictable. High maneuverability is achieved by minimizing the response time of the transition of the vehicle into the desired radius.
- *Reliable* so the chassis avoids dangerous configurations. This is especially critical because no mechanical limits exist to stop the passive steer pivot

To satisfy these design goals control software was designed to maintain a desired steering axle angle. This software executed as an application on a Galil DMC-2080 multi-axis controller, utilizing built-in PID motor velocity control. The motion controller was selected for ease of implementation and support for reasonably sophisticated application programs. These programs were stored in nonvolatile memory and began execution immediately upon power up. This provided increased reliability since driving maneuvers did not require Hyperion's on board computer, a CompactPCI Pentium III running Red Hat Linux. See Figure 4 for a block diagram of the control hardware.

As shown in Figure 5, mobility commands consisting of desired steering axle angle (θ_d) and overall rover velocity (v_d) come from either the vehicle controller or the pendant. The Vehicle Controller process runs on Hyperion's on board computer, which communicates with the motion controller via a serial line. It is responsible for responding to commands from the higher level autonomy software and producing telemetry. Autonomy software commands are represented as a triple:

{ *rover_speed, steering_arc_radius, command_timeout* }

The variable *rover_speed* represents the desired overall rover ground speed, given in meters / second. Commanded steering is represented by the variable *steering_arc_radius*. This describes the radius (in meters) of the resulting arc path. The kinematic equations in the motion controller application software allow this commanded radius to be converted into a commanded steering axle angle. Finally, the *command_timeout* variable describes how long the last steering command should be executed before the rover should stop. Timers built into the motion controller application enabled reliable adherence to these time-outs. Driving commands may also come from a pendant that can be attached to the rear panel of Hyperion's electronics box. An analog joystick on the pendant is connected to two channels of the motion controller's analog input.



$\varpi_{front} = f\,(v_d, \theta_d)$
$\varpi_{rear} = f\,(v_d, \theta_a)$
$\Delta\varpi = K_p(\theta_d - \theta_a)$

Figure 5: Hyperion's steering axle control software

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

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

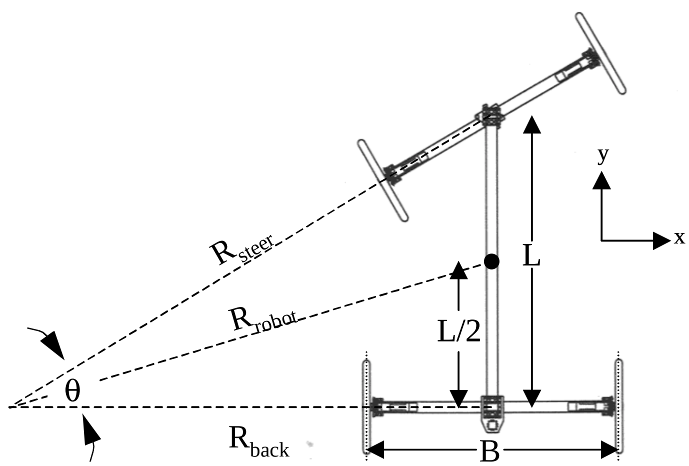
$$R_{back}(\theta) = \sqrt{R_{steer}(\theta)^2 - L^2}$$

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

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

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

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



L = length of chassis [meters]

B = wheel base [meters]

r_{wheel} = wheel radius [meters]

θ_a = actual steering axle angle [radians]

θ_d = desired steering axle angle [radians]

v_d = desired robot velocity [meters / second]

Figure 6: Steering kinematics and equations

The $\{\theta_d, v_d\}$ mobility commands must then be translated into angular velocities of each of Hyperion's four wheels. This calculation is performed using a kinematic model of the rover, which consists of the seven equations shown in Figure 6.

Note that the front wheel angular velocities are based on the desired steering axle angle, θ_d. In contrast, the rear wheel velocities are based on the current actual steering axle angle, θ_a. If the rear wheels, which are attached to a non-pivoting axle, do not spin at a rate based on the actual steering axle angle, the chassis undergoes excessive stresses and disturbs the velocity control of the wheel motors. Because the front axle can freely pivot, the front wheels can spin at a rate based on the desired steering axle angle, which forces the axle to the proper angle.

Simply commanding the front wheel velocities based on desired steering axle angle does pivot the axle. 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. Speeding up these transitions increases the rover's maneuverability. To do this, a proportional controller was added. The output of this controller is a $\Delta\varpi$ that is subtracted from the front inner wheel and added to the front outer wheel. The output is based on the difference between the desired and actual steering axle angle. The proportional term, specified in units of encoder ticks/s/rad, is increased to speed up the steering response of the robot.

Once the wheel angular velocities are calculated, closed-loop velocity control was accomplished in the motion control hardware using PID loops. The Galil DMC programming interface made this a straightforward task, and eliminated the need for custom control software and real-time operating systems.

4. EXPEDITION RESULTS

4.1 Steering Accuracy

The first test characterizing Hyperion's steering accuracy involved simply commanding a straight path and fitting a line to the resulting ground track. Differential GPS (DGPS) was used to track the rover's path, with an expected accuracy of less than 5 cm. Care was taken to eliminate data gathered while the DGPS system was not fully converged. The tests were performed on mostly flat, slightly sloped terrain. The ground was composed of a compacted fine soil with rocks less than 20 cm high distributed throughout the area. In this test, Hyperion was commanded to travel at 30 cm/s. The proportional gain on the steering control was 0.2 ticks/s/rad. See Figure 7 for plots showing 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. The errors are also centered about zero. Note that the expected accuracy of DGPS is just less than 5 cm, so it is difficult to differentiate steering and DGPS effects in these data.

In addition to straight paths, further tests in the same area involved commanding given steering arcs to Hyperion and allowing the rover to follow the arc for approximately one complete circle. A Matlab script was used to fit a radius to the rover's actual path. Table 4 summarizes the results of the circular path trials. Figure 8 depicts one such trial.

For all but one trial, the error in the fit radius to commanded radius was less than 3%; although in the 20 m right turn trial an error of 11.8% was calculated.

4.2 Maneuverability

Maneuverability, meaning the ability of the steering controller to provide responsive direction control allows the robot to negotiate natural terrain. Hyperion's steering axle joint has a 90 degree range of mechanical rotation, enabling steering radii as tight as 2.5 m. In simulation, this proved capable of safely maneuvering the robot through terrain with obstacle densities of approximately one obstacle per 10 m^2. This simulation assumed instantaneous steering response, so real world tests were needed to model latencies in Hyperion's steering control system.

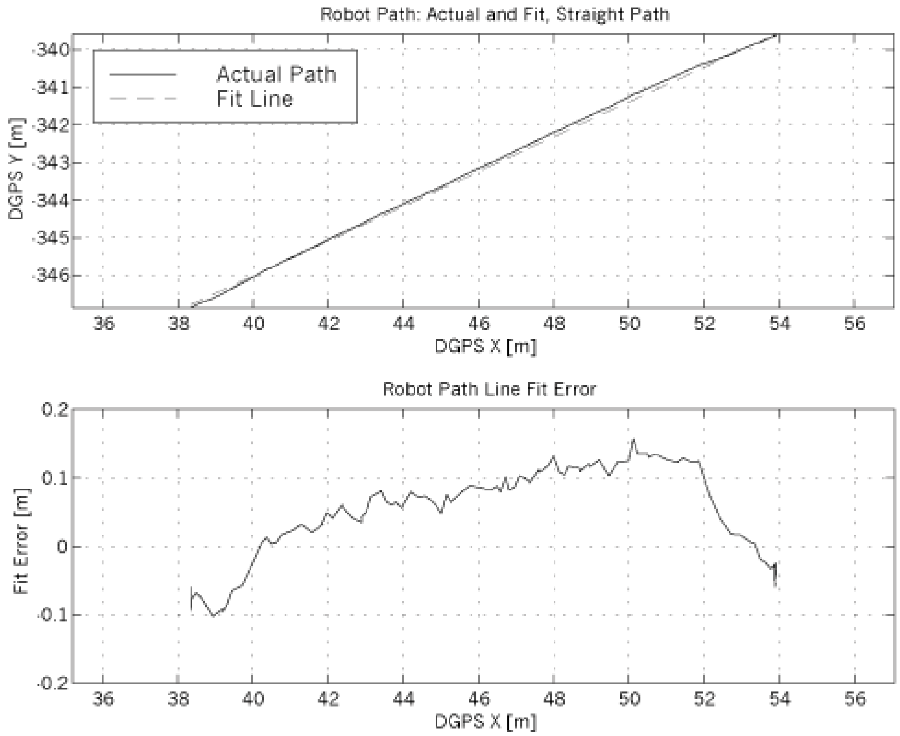


Figure 7: Above: plot of straight rover path along with a line fit to the path data. Below: plot depicting line fit error across path.

.

Table 4: Hyperion steering accuracy results for circular paths

Commanded Path Radius	Radius Fit to Actual Path	Radius Error	Number of Data Points
20 m left turn	19.70 m	-0.30 m / 1.5%	1848
20 m right turn	17.65 m	-2.35 m / 11.8%	1804
10 m left turn	10.12 m	0.12 m / 1.2%	949
10 m right turn	9.79 m	-0.21 m / 2.1%	984
5 m left turn	4.99 m	-0.01 m / 0.2 %	719
5 m right turn	4.95 m	-0.05 m / 1.0%	558
2.5 m left turn	2.57 m	0.07 m / 2.8%	410
2.5 m right turn	2.51 m	0.01 m / 0.4%	406

.

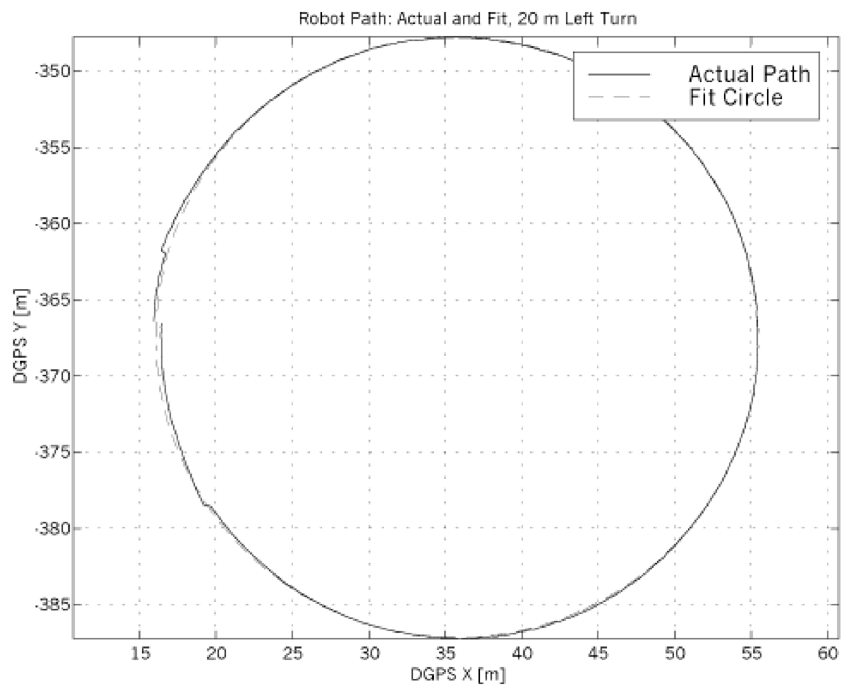


Figure 8: Plot of the 20 m left turn trial. The discontinuities are due to DGPS position error.

Before the expedition, tests were performed to examine the effect of the proportional gain on the steering system. As expected, the response of the system was proportional to the gain. Furthermore, we did not experience overshoot as the gain was increased, supporting the assumption that the steering plant is a first-order system which eliminates the need for a more complicated, PID controller. However, the proportional gain was bounded by the maximum wheel velocity, so the steering response of the rover was limited.

The steering arc transition tests were carried out in the same location where the accuracy tests were performed. In these trials, Hyperion was commanded to travel at 30 cm/s along a given arc. Once in steady state, a new arc was commanded. Steering axle angle data were recorded during the steering transitions, and an exponential curve was fit to these data to determine the time constant of the steering system and steady state angle errors. Table 5 shows the time constants along with the steady state fit angle. The time constants were initially calculated in units of samples rather than seconds, leading to a slight uncertainty in the time constant due to the variability of the number of samples taken per second, which was found to be on the order of 0.04 s.

Table 5: Hyperion Steering Maneuverability Test Results

Initial Arc Command	New Arc Command	Fit Exponential Time Constant	Fit Exponential Steady State Angle
Straight / 0 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 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

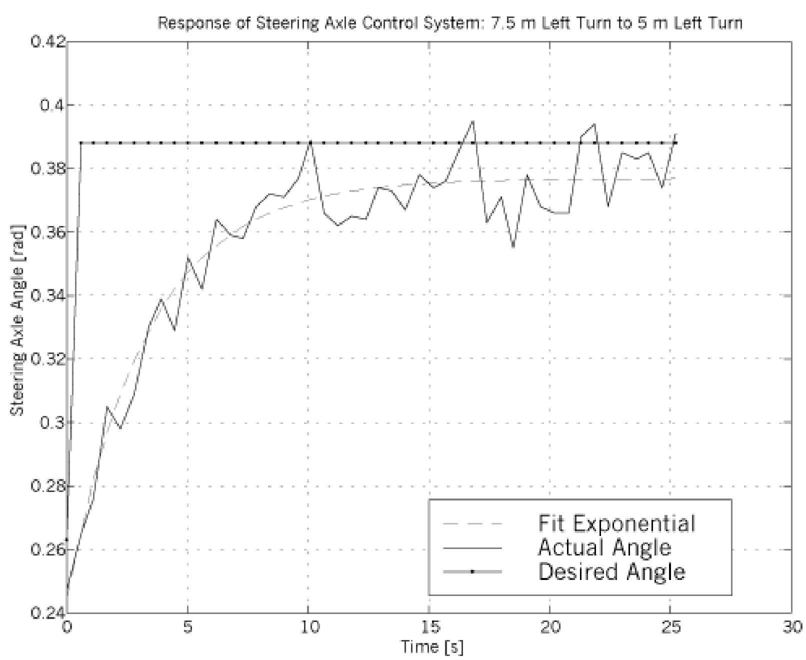


Figure 9: Plot showing transition between 7.5 m left turn arc and to a 5 m left turn arc. The solid line shows how the steering system responds to the step function change in command.

4.3 Reliability

Robots require reliability at various levels. The reliability of Hyperion's mobility system is particularly important because failures may result in damage to the chassis or drive motors. Several failure modes were anticipated and experienced throughout development and testing.

One obvious failure mode would be rover damage due to excessive steering axle angle. The underconstrained nature of Hyperion's chassis created the possibility that invalid commands or obstacles could rotate the steering axle far enough to collide with the rover's electronics box. Furthermore, the potentiometer mounted on the steering axle pivot could be damaged by excessive pivoting, crippling all driving capability. The simple chassis did not adequately support hard, mechanical steering limits. Therefore software limits were implemented in the control software. Added reliability came about by executing this software on the motion controller. The motion controller applications do not support commonly used but error-prone software approaches such as dynamic memory allocation. Furthermore, a separate motion control application need not contend for CPU time with several other processes as on many standard computers.

Gear damage due to excessive drive motor torque was another potential failure. Circuitry was integrated to set a hard limit on the peak and continuous torque generated by the drive motors. This avoids damage to Hyperion's harmonic drives.

While analysis of the steering controller showed stability when driving forward, driving in reverse proved to be unstable with the current control scheme. When driving in reverse, forces rotate the steering axle angle that seem to be proportional to the magnitude of the angle itself. Therefore small perturbations in the steering axle angle lead to ever-increasing axle angles until software limits are reached and the rover's motion is stopped. In Figure 10 the robot is commanded to follow a straight path forward and in reverse. The steering axle angle is then recorded over a period of approximately 40 seconds. No steering control is attempted in these cases, so the divergence seen in reverse driving are due solely to the mechanism itself. This instability becomes a reliability issue for the rover because backup maneuvers are generally executed if the robot fails to find a safe path through obstacles. When backup maneuvers were needed, reverse driving commands were only executed for a few seconds and steering was avoided.

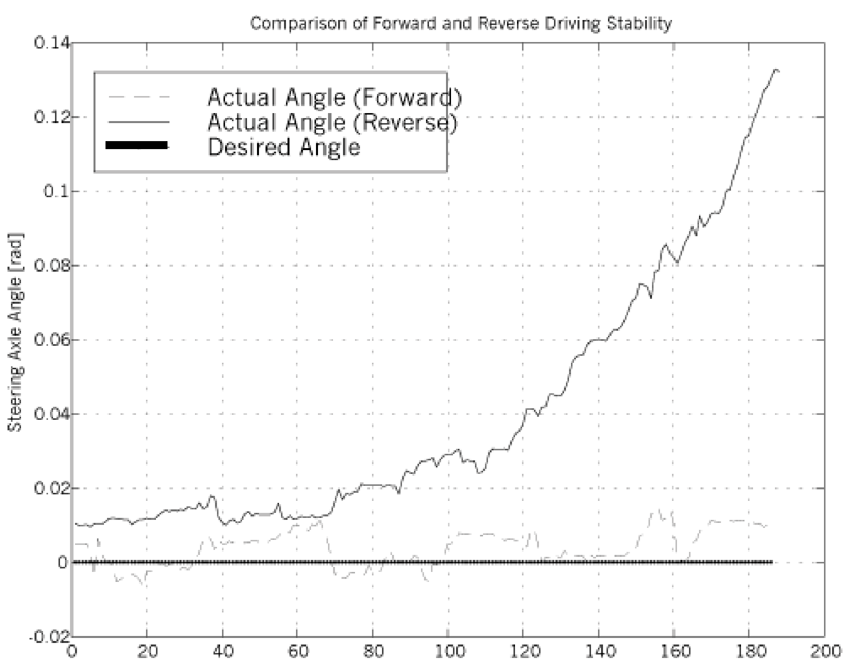


Figure 10: Plot comparing the stability of forward driving with the instability of reverse driving without a steering controller. Within about 40 seconds of reverse driving, the axle angle reaches nearly 0.14 rad (8 deg) while forward driving remains centered about 0 rad.

5. CONCLUSION

The concept of sun-synchronous operation has been brought forward as an enabling technology to help extend the distance traveled by solar powered robots. The terrestrial experiments performed in July of 2001 showcased the development of a new solar powered rover named Hyperion. The configuration of a terrestrial solar powered rover compelled the design of a novel steering system that enabled the Hyperion rover maintain a high terrainability with efficient steering. The underactuated steering added new complexity to the control system. In order for the steering to be used by the autonomous navigation software the robot must be able to follow predictable path commands. The predictability of a path was decomposed into the steady state accuracy and maneuverability of the robot. The accuracy, or the ability of the rover to track predictable arcs was verified to less than 3 percent. The maneuverability was verified in terms of the response time for the robot to transition between arcs.

With over 6 km of autonomous and 25 km of teleoperated driving in outdoor terrain the Hyperion rover proved to be capable of handling the various disturbances of natural terrain. The control system was able to maintain a desired axle angle even when traversing over 20 cm obstacles. Future work includes an analysis of the power draw of this rover in order to quantify the benefits of the passive articulated steering. The development of the control algorithm must be enhanced for backwards driving.

ACKNOWLEDGEMENTS

This paper describes the work of the Sun-synchronous Navigation project and all of its members are important contributors. We acknowledge and thank Dimi Apostolopoulos, Jesse Boley, Bernardine Dias, Reid Simmons, Sanjiv Singh, Surya Singh, Tony Stentz, Chris Urmson, Vandi Verma, and David Wilkinson. Field experimentation was conducted in collaboration with the NASA Haughton-Mars project, Pascal Lee, Principle Investigator. This work is supported by NASA under grant NAG9-1256, Chris Culbert, Program Manager. Dr. Melvin Montemerlo has also added his support and inspiration to this work.

REFERENCES

1 D. Apostolopoulos, “Analytical Configuration of Wheeled Robotic Locomotion,” tech. report CMU-RI-TR-01-08, Robotics Institute, Carnegie Mellon University, April, 2001.

2 P. A. Dudzinski, “ Design Characteristics of Steering Systems for Mobile Wheeled Earthmoving Equipment,” Journal of Terramechanics, Vol. 26, No. 1, pp. 25-82, 1989.

3 A. Kelly, A. Stentz, “Rough Terrain Autonomous Mobility - Part 1: A Theoretical Analysis of Requirements,” Autonomous Robots, No. 5, May 1998, pp. 129-161.

4 K. Shillcutt, “Solar Based Navigation for Robotic Explorers,” Ph. D. thesis, CMU-RI-TR-00-25, Robotics Institute, Carnegie Mellon University, October 2000.

5 Sun-Synchronous Navigation web page, http://www.frc.ri.cmu.edu/sunsync/

6 P. Tompkins, A. Stentz, W. Whittaker, “Automated Surface Mission Planning Considering Terrain, Shadows, Resources and Time,” i-SAIRAS, Montreal, Canada, June 2001

7 D. Wettergreen, B. Shamah, P. Tompkins, W. Whittaker, “Robotic Planetary Exploration by Sun-Synchronous Navigation,” i-SAIRAS, Montreal, Canada, June 2001

8 J. Y. Wong, *Theory of Ground Vehicles*, John Wiley & Sons, Inc., New York, 1993.