UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

EFFECTS OF PERCEPTION RANGE ON MOBILE ROBOT PATH EFFICIENCY

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

MATTHEW J. ROMAN
Norman, Oklahoma
2011

EFFECTS OF PERCEPTION RANGE ON MOBILE ROBOT PATH EFFICIENCY

A DISSERTATION APPROVED FOR THE
SCHOOL OF AEROSPACE AND MECHANICAL ENGINEERING

BY

Dr. David Miller, Chair

Dr. Peter Attar

Dr. Kuang-Hua Chang

Dr. Andrew Fagg

Dr. Dean Hougen

Acknowledgements

This project has been funded on part by NASA's Mars Technology Program and MSSS Inc.

Contents

List of Tables

List of Figures

Abstract

Path efficiency is the length of the path actually traversed versus the length of the optimal path. If a robot has complete knowledge of the terrain between it and its goal and unlimited time, then it is possible to plan an optimal path to that goal. Complete knowledge is not available because robot sensors have limited range. Knowledge is also limited by features in the terrain that are shadowed by other portions of terrain. This research uses data from both simulations and actual field trials to determine the effects of sensor range on rover path efficiency. Results from both simulation and field trials with the SR2 rover indicate that in terrain typical for planetary rovers (i.e., terrain without foliage and with broken features) sensor range has a surprisingly low impact on path efficiency.

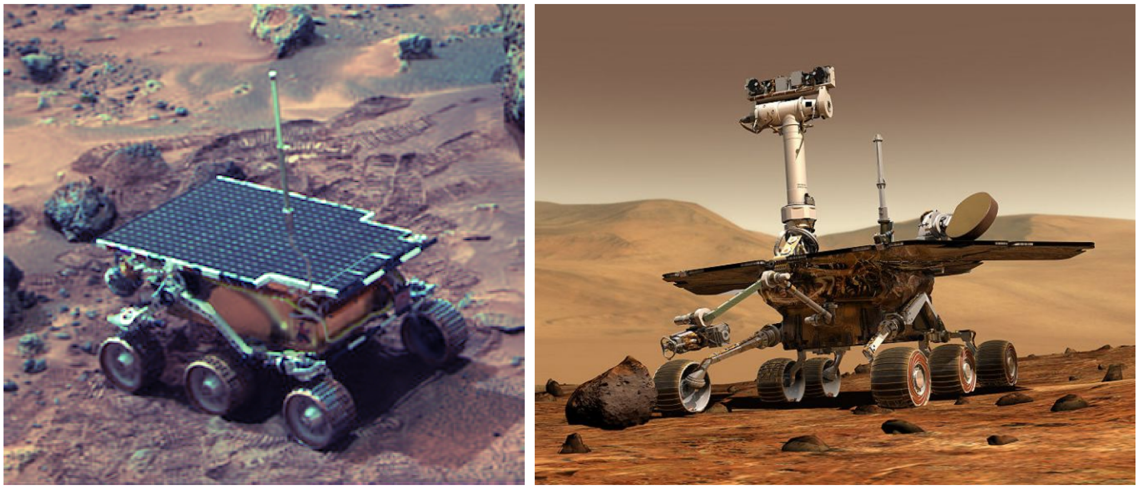
Chapter 1

Introduction

The idea of landing a mobile robot on the surface of another planet is to allow earthbound scientists access to specific areas of interest to do in-situ experimentation. It is likely that science objectives will be separated by many kilometers to better define the planet. Ideally, these areas of interest would be selected prior to mission launch, meaning that a suitable landing site might be located tens of kilometers away [9]. Or, multiple science outcrops may be defined near the projected landing location. Though current technology for placing a craft on Mars has a landing site estimation that covers hundreds of square kilometers [42, 64]. A mobile robot for planetary exploration must have the capability to autonomously move long distances in a matter of weeks to achieve these science goals.

A manned mission to Mars or any other terrestrial surface outside Earth will be vastly more expensive to launch and maintain than a robotic mission. Some feel that robotic craft are unreliable due to the fact that many of the probes sent to Mars have failed to make it to the surface. This argument hardly justifies sending humans to explore the unknown. Turner [62] stated that robotic explorations are limited for two main reasons: limited control of scientific procedures and limited range of mobility. However, there have also been many experiments indicating that long distance traverses are possible with robotic vehicles. It is believed that the development of robotic spacecraft for scientific exploration should and will continue until a more reliable and economical way to put humans into space, maintain a solid foothold, and land them safely on the surface of any orbiting body is achieved. Even then it is likely mobile robots will be used to support astronauts during their mission.

The various spacecraft that have landed on Mars provide sufficient evidence that its surface is hard enough to support a small mobile robot, herein referred to as a rover. The images taken from the surface indicate that it contains geological formations similar to places on Earth. Some have stated that the areas resemble places in the deserts of Arizona and California where there is little vegetation [45, 52]. The images from the Viking landers in the 1970's and the Pathfinder mission in 1997 show rolling hills littered with rocks of various sizes. In 2004 the twin Mars Exploration Rovers named Spirit and Opportunity, landed on opposite sides of the planet from each other. Spirit's landing site has similar features as the Viking and Pathfinder sites. Opportunity landed on a smooth dust covered area with an occasional impact crater.



(a) Sojourner [49]

(b) MER

Figure 1.1: NASA Mars Rover technology. Images taken from NASA.

The Mars Pathfinder mission included a small 11kg rover named Sojourner. It traveled a little more than a meter a day on average for a total traverse of about 100m. The Mars Exploration Rovers (MER) landed in January 2004, each rover weighing approximately 176.5kg [32]. These larger rovers increased the traverse rate by an order of magnitude. As of August 2011 Spirit and Opportunity traveled 41.22km between them, 33.49km by Opportunity alone [26]. During the mission the rovers progress up to 48m per martian day on average. This rate of progress excludes days in which they were engaged in science operations. This validates that long-range traversals of tens of kilometers are possible on the surface of another planet.

Present day Mars rovers are micromanaged to the point that most of their driving is accomplished by tele-operation commands. Due to the communication time delay between Earth and Mars, the command cycle time is on the order of a day. This makes long traverses difficult because the operators can only plan as far as the surface images show a clear path. The time delay also places a lot of stress on the rover operators as they have to literally live on Mars time. Each Mars sol is about 40 minutes longer than an Earth day. Future missions will require rovers to make longer traverses than what has previously been achieved. Mission operations will have to rely more on the autonomy of the rover if larger distances are going to be driven. It is possible that surface operations cost will decrease as rover autonomy increases, because a detailed plan for each days traverse will no longer be required. Therefore, fewer command cycles will be needed reducing the stress placed on the operation personnel to complete the mission successfully.

1.1 Anatomy of a rover

The "rover scenario" or "rover problem," as stated by Doyle [16] and Matthies et al [27] is defined as: A mobile robot in unknown, rough terrain must autonomously navigate from a start position to its goal. The rover must do this using onboard sensors and processors to determine a safe passage through its environment. Planetary rovers must solve this problem efficiently due to the limited resources available to complete such a complex task.

A long-range rover on the surface of Mars may encounter drastically different regions during its journey between science objectives. A closer look at the makeup of a rover will reveal four major parts that affect its ability to navigate natural terrain.

First is the mobility system, which includes the drive motors, suspension, and wheels. This system defines how the rover moves and what terrain or obstacles it is physically capable of crossing. Typically higher mobility means larger obstacles can be surmounted. A well designed suspension allows the rover to negotiate densely populated rock outcroppings as well as efficiently make its way across vast dust-covered plains.

Second, the perception system allows the rover to detect the state of its local environment. This system must accurately represent the environment in a format that can be used for the path planning software to navigate around obstacles. The system uses a sensor suite to take measurements from the environment. The amount of computation to process the sensor measurements and extract obstacle information can affect the refresh rate of the perception system. The time between perception updates should be small enough so the rover's progress is not slowed.

The third major part is the planning and navigation software. This software gives the rover the ability to make decisions on its own. It is designed to make decisions that affect the rover's pose based on its state of health and information about the environment provided by the perception system.

Finally a power system must keep the rover alive. It will need to generate enough energy for all the systems to function properly. If this system is based on a local resource that is not continuously available, like solar energy. The rover must be able to store enough energy to survive during periods when the resource is unavailable.

The physical design of a spacecraft is limited by the size of the launch vehicle and the spacecraft's destination. Rover designers must work with a finite amount of resources, i.e. mass, volume, available power, and life span, to solve the "rover problem". Each of the rover systems will require a portion of each resource and in return will affect the performance of the rover. For example, a rover that has a high degree of mobility will be able to traverse large obstacles more safely than a rover with limited mobility. The sensor suite on the high mobility rover will not have to detect smaller obstacles, potentially simplifying the perception system and planning software. However, it is likely that the resources required for the increased mobility will exceed the mass and size limitations, meaning the rover will be too heavy or large to launch into space. The example rover may require excessive amounts of power to support the mobility system or drive so slow that it will not be able to reach its science objectives within the mission timeline. Each rover subsystem affects one another. Therefore, design tradeoff decisions must be made when selecting system components and their mode of operation.

1.2 Problem Statement

Currently, sensor selection is based on the assumption that high fidelity sensors are needed to navigate efficiently in natural terrain. A rover with sensors that provide information about obstacles at a greater range can plan a more efficient path around them. However, as the range of the sensor increases more resources are needed to support it. Without enough information about the local environment the rover will not be able to successfully maneuver around obstacles to reach its goal. So, how does the sensor range of a rover affect its ability to autonomously navigate natural terrain?

The rover's path efficiency is the key measurement in quantifying the effects of varying sensor range on rover performance. Path efficiency is a useful parameter to measure the effectiveness or performance of a system or path planning algorithm [58]. Research in multi-robot systems use path efficiency as a measure of how well a swarm of robots converge on a particular location or leader robot [41, 51, 4]. A path is defined by the route a rover takes between a start and goal position. The path length is typically many times greater than the length of the rover. The path is determined using the rover's onboard sensors to avoid obstacles in the environment. Assuming the goal position is reachable there will likely be multiple pathways around obstacles in natural terrain. If the rover has complete knowledge of the environment between the start and goal positions an optimal path can be generated given enough processing time. In the case of this research the optimal path is equivalent to the shortest path. All paths are assumed to be safe from collisions with obstacles. Therefore, the path efficiency can be calculated as the ratio between the lengths of the actual path the rover takes and the optimal path

$$\eta = \frac{\text{Optimal path length}}{\text{Actual path length}}. \tag{1.1}$$

Hazards must be detected sufficiently ahead to allow time for evasive maneuvering or stopping before collisions are encountered. The maximum distance between the sensor and a detectable obstacle is known as the look-ahead distance [22]. If the time between sensor updates is too long the speed of the rover can be affected. The rover must not drive farther than the look-ahead distance between sensor updates. Therefore, the maximum speed of the rover can be affected by the refresh rate of the sensors. The refresh rate may be limited by the electronic circuitry that drives the sensor transducer or by the algorithms needed to transform sensory data into obstacle data. The sensor's range can be affected by its mounting location on the rover. A sensor may be attached to a support structure such as a mast or boom and lifted to extend its range. The structure will require more mass to be allocated to the sensor system, limiting the design of another subsystem. Sensor selection should be an integral part of the rover's design as it affects power, mobility and processing requirements.

1.3 Sensors and Rover Operations

The current rover sensor suite of choice relies heavily on vision as the primary source for obstacle detection. Cameras are low power and robust sensors and the images they return can be easily decoded by a human. A person can look at an image of the surrounding terrain and almost immediately identify obstacles and characterize new types of terrain hazards. A large amount of computation is needed for a machine to complete the same task. A rover with limited processing power will need a significant amount of time between sensor updates to detect hazards from vision.

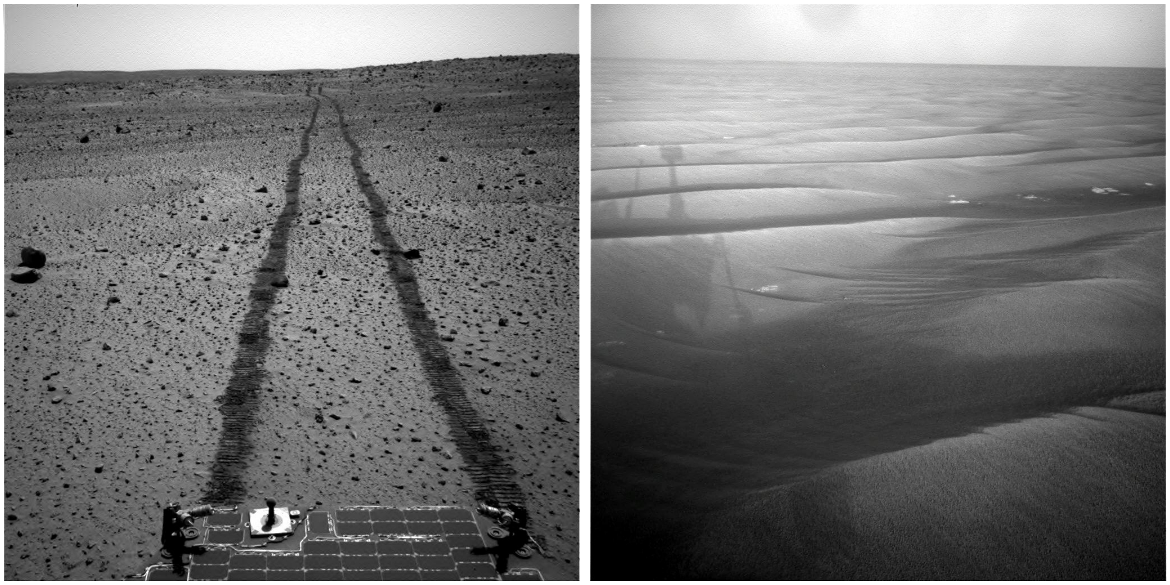
The Sojourner rover on Pathfinder used a structured light vision sensor for detecting obstacles. But due to the risky nature of the surrounding terrain the rover was never really given the chance to perform any fully autonomous traverses. Sojourner also never strayed more than a few meters from the landing site during its operational period [34]. Driving was accomplished by sending a list of commands to the rover each day. The commands were step-by-step instructions essentially telling Sojourner which motors to drive and for how many rotations. The command list was created by analyzing stereo images of the terrain taken from a pair of cameras on a mast that had been raised by the lander.

During one instance the estimated distance between the rover and a large rock was incorrect. The images from the next day showed one wheel of the rover on the rock, after the rover executed the command list. The navigation software on Sojourner kept it from driving too far which could have tipped it over ending the mission [46].

Science targets for MER could not be defined until the rovers landed and their locations identified. Possible targets were located farther than expected from the landing site. This forced operators to drive the rovers faster than planned to reach the targets within the primary mission timeline [39]. The rovers are mechanically limited to a top speed of 4.6cm/second [32] meaning they physically cannot travel very far each day. Unfortunately, to maintain the highest traversal rate the rovers were mostly driven in a fashion similar to what was done on Pathfinder. For MER this is known as Directed or "Blind" driving. The autonomous navigation software does not run while in directed mode, hence "Blind" driving. The Earth based operators create a path using primitive drive functions consisting of arcs, straight lines, and turn-in-place points. The paths are generated from the previous days stereo PanCam images, where the rover operators can measure obstacle sizes and locations. All image analysis is done on Earth based processors. The range of detectable obstacles is limited by terrain features but is typically about 70m [29]. During 2010 Opportunity covered an average of 48m per sol under directed control [26]. On some sols a maximum rate of 120m/hr was achieved [6, 7, 39]. The difference is due to the density of potential hazards in the local environment of each rover.

Spirit and Opportunity drove approximately 28% and 21% of their mileage under autonomous control respectively, during the first 550 days on Mars [39]. The rovers use multiple sets of stereo camera pairs to calculate safe paths and execute them automatically. This mode of driving is called AutoNav. Opportunity progressed 213m on Sols 384 and 385, all of which was done in AutoNav mode. Opportunity also achieved the longest single sol traverse of 220m on Sol 410 (March 20, 2005), about 110m in AutoNav mode [8, 26]. Typically the autonomous traverses were completed during long distance driving periods on level terrain that appeared to be relatively free of obstacles as seen in figure 1.2.

Directed driving is used because the rovers can only achieve an average speed of 0.6cm/second or 30m/hr under AutoNav mode [7, 36]. Terrain height and depth information is calculated from the HazCam images, at a range from 0.5m to 5m, or the NavCam images, at a range from 2m to 20m [6]. The image size is reduced, by lowering pixel resolution, then filtered before stereo matching is performed. About 70 seconds are needed to compute a path from a pair of stereo images on MER's 20MHz



(a) Gusev Crater [49]

(b) Meridiani Planum [49]

Figure 1.2: Images taken from Mars Exploration Rovers Spirit (a) and Opportunity (b).

processor. In some cases the rovers switch into AutoNav mode after the commands from a directed path are completed [36, 26]. This allows the rovers to progress a few extra meters at the end of a sol where the rover operators could not see.

The MER rovers software is capable of correcting its own position estimation. Features are tracked in consecutive stereo images to determine a change in position when driving. The technique is known as Visual Odometry or VisOdom [40, 11, 35]. VisOdom is used for driving in and out of craters where the rover can lose traction and slip due to the increased slopes and sandy surfaces. This aids in safely maneuvering on the edge of crater rims which are particularly interesting to scientists.

MER's traverse rate drops to 10m/hr under VisOdom mode [7]. The rovers can only drive a few meters per sol in this mode. During the primary mission phase VisOdom was used for less than 10% of the total mileage [61]. The rovers can only move 60cm or rotate 15 degrees between sensor updates because the image pairs need 60% overlap to reliably track features. Again, due to MER's relatively slow processor VisOdom calculations take up to three minutes to provide an accurate position solution [30].

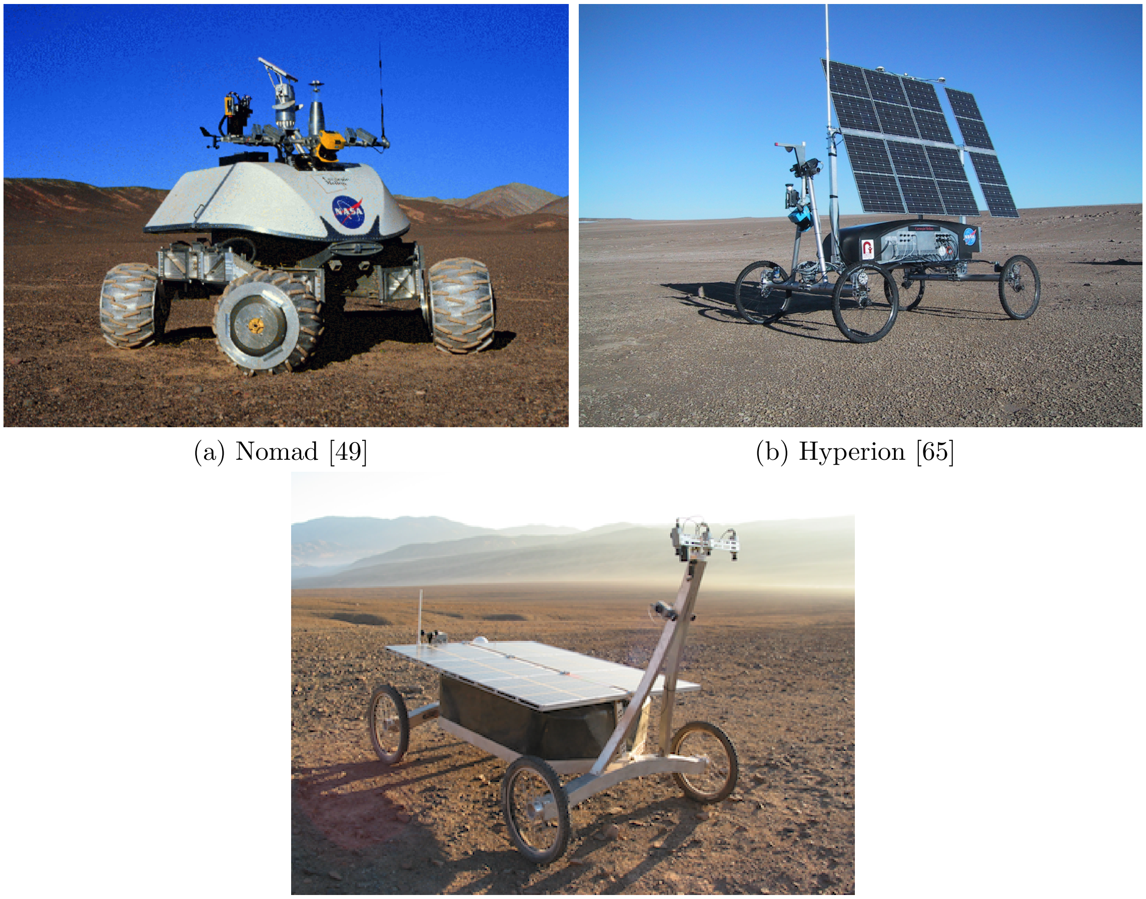
MER is limited in speed because the rover has to stop every few seconds to compute its next motion. A lower definition sensor requiring less computation could be used to achieve obstacle detection at a higher rate. With this lower definition comes limited sensing range that could make it difficult to find optimal pathways around obstacles. How could this affect the rover's path efficiency over long distances from a global standpoint?

1.3.1 Other Rover Prototypes

A few other prototype rovers have accomplished long-range traverses through natural terrain. During the late 1990's Nomad, a four wheeled 725kg rover from Carnegie Mellon University, traveled over 30km autonomously across the Atacama Desert in northern Chile and Antarctica [47, 66, 2]. The Hyperion rover, also from CMU, navigated on a 6.1km course during a 24hr sun-synchronous experiment in the Canadian Arctic. In 2003, Hyperion was taken to the Atacama as a preliminary test and traversed 18km during 90 experiments [63, 69, 55, 68]. Hyperion evolved into the 200kg Zoe robot in the fall of 2004. Zoe autonomously traveled 55km through the Atacama at one point traversing 3km in a single day. Returning in 2005, Zoe successfully navigated 202km with an improved navigation system [67, 70, 65].

In one instance, Nomad's camera system became disabled requiring the rover to use only its single laser range scanner. Nomad traveled more than 10km autonomously in this mode [47]. Although the field trials were located in areas of very low obstacle densities.

The rovers from NASA and CMU are highly capable and have achieved important scientific contributions. However, they rely on complex sensor suites and a high degree of human input to safely navigate long distances. The high mileage of MER was only



(a) Nomad [49]

(b) Hyperion [65]

(c) Zoe [65]

Figure 1.3: Nomad and Zoe in the Atacama desert.

possible after multiple mission extensions over a six year timeline. Nomad and Zoe require large amounts of power to process sensor data quickly enough to achieve higher speeds. Nomad has a gas generator onboard to supply electricity for the computers. The vehicle frames must also be large enough to support this, making it difficult to show there is a flight equivalent solution.

1.4 Approach

A sum of 345km of natural terrain have been traversed autonomously by the experiments of MER, Zoe, Hyperion, and Nomad. Zoe itself completed 257km or about 3/4 of that total distance. Much of the long distance traverses were completed in areas where the obstacle density is very low. Densely populated obstacle fields that are encountered have been traversed by using low level commands or numerous closely spaced waypoints. Human operators essentially plot a path for the rover to follow requiring little or no autonomy. Almost all of the experiments were done using vision as the primary obstacle detection scheme. Only Hyperion and Nomad used a laser range finder as a virtual bumper to safeguard against inaccurate stereo vision results.

The experiments that have been performed indicate that there is limited data on the autonomous capabilities of rovers traversing terrain densely populated by obstacles. A more efficient design for long range driving through low density obstacle fields may be achieved by using a different type of sensing. The experiments do not test any correlation between different sensor ranges and the path to the goal. For long distance traverses through natural terrain, it is likely there will be multiple paths to the goal. While only one path may be optimal from a global frame of reference the difference between it and the remaining paths may be insignificant.

1.4.1 SR2

The remainder of this section briefly describes Solar Rover 2 (SR2), a 22kg solar powered rover seen in figure 1.4, and a comparable virtual simulation (RoverSim) to test the effects of a rover's sensor range on path efficiency. SR2 was built to determine if Mars missions could achieve lower cost through a more simplified design and surface operation procedure. The design goals of SR2 are to simplify the perception system, navigation software, and mobility system. Therefore, requiring less power to achieve efficient long-distance traverses of 1km per day. A day of driving is typically less than a four hour period. It was expected that in simplifying the design, accuracy in positioning the rover would be reduced. It was unclear how large the position error would be due to the reduced information set from the simplified perception system.

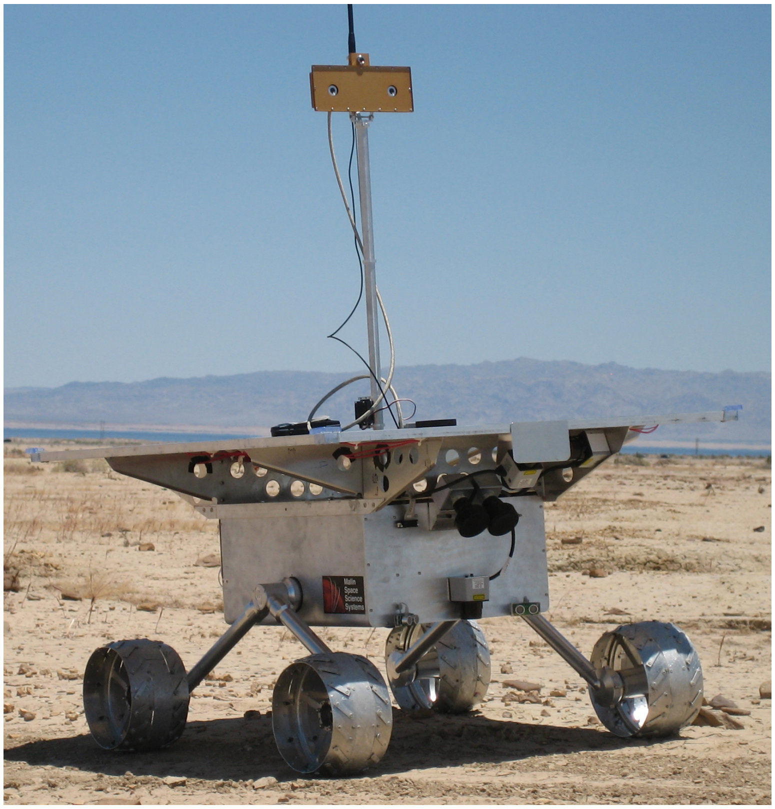


Figure 1.4: SR2 during a field test in Anza Borrego.

A series of field experiments to characterize this error were carried out in the Anza Borrego desert of Southern California. SR2 was capable of autonomous speeds 25 times faster than MER with an average of 15cm/s through Mars-like terrain [44, 54]. Results from the field tests demonstrate numerous days of traverse averaging about 1km per day through more rugged terrain than what has been encountered by MER.

Unlike the previously mentioned rover systems, SR2 uses a set of two scanning laser range finders as its primary obstacle detection sensor. The advantage of using this type of sensor is that it provides depth information directly without large amounts of computation. The information produced by each sensor is low definition and limited to a single plane. With this particular sensor suite SR2 has autonomously traveled over 14km in a matter of days during multiple field tests. Only a few cases were seen in which a higher definition sensor suite may have exhibited significant improvement.

The hypothesis to be tested in this research is: The path efficiency of a mobile robot in Mars-like terrain is not affected by limiting its sensing range. A simplified perception system may lead to a more power efficient and faster rover due to its lower overall complexity. The experiments completed by existing planetary rovers indicate the performance gained from using complex obstacle detection systems to increase sensor range does not justify the resources required to support them.

The number of field experiments with SR2 were limited due to the amount of time and support available. To decrease the time needed for software development and testing, a virtual simulation of SR2 was created. This allowed the autonomous navigation system to be debugged and tested while the rover's hardware was inaccessible. Following the field tests the simulator was upgraded to allow for multiple runs of the rover under various sensor configurations. Thousands of kilometers of path data can now be generated from different terrain and obstacle layouts to better determine how sensor range affects path efficiency.

1.5 Outline of Dissertation

The remaining sections in this dissertation include a description of the work done for studying the affects of sensor range on a rover. Chapter two contains design details of the Solar Rover 2 and each of its subsystems. Chapter three describes the RoverSim program and how well it substitutes for doing more field work. The experiments conducted by SR2 during the field tests in Anza Borrego are described in chapter four. The analysis of the simulation are in chapter five. Finally, the effects of sensor range on rover efficiency are concluded in chapter six. This is followed by a discussion about how equivalent RoverSim is to the real world field tests and what areas require further attention.

Chapter 2

A Mars Solar Rover

Rovers are important tools for planetary scientists to retrieve data from interesting sites by making the instruments they carry mobile. However, our understanding of what is needed for an efficient rover design is limited due to the low mileage currently driven autonomously under conditions similar to Mars. The terrain that has been covered by the aforementioned prototype rovers is in areas with low obstacle densities. A large part of the terrain covered by flight rovers has been achieved using a tele-operation style of navigation. Most of the obstacle navigation sensors used are based on complex and computationally intensive vision systems that generate large amounts of data which must be parsed to identify a safe path. It is unclear whether this type of sensing limits the speed of the rover with a constricted power supply.

For rovers to remain statically stable they can not travel fast enough for the wheels to leave the surface and become dynamic [72]. The velocity limit for MER is approximately $56cm/s$ with a wheel diameter of 20cm on Mars. A similar size vehicle traveling faster than these speeds will require a damped suspension system, which is likely to increase complexity and mass. Also, when traveling at speeds where the wheels become dynamic the rover has a much greater chance of tipping over rendering these types of vehicles unlikely to be selected for near future space exploration missions.

The current mobility system used for Mars exploration is the rocker bogie suspension developed at JPL by Donald Bickler [5]. A rover with this suspension can traverse obstacles 1.5 times taller than the diameter of its wheel. While this is a highly capable system, it is complex and unclear that it is necessary for the terrain explored by current Mars rovers. At least ten motors are needed to drive the rocker bogie, each accompanied by a gear train and exposed to the environment. During the Pathfinder and MER missions few large obstacles have been encountered that the rover could not simply drive around.

The remainder of this chapter describes how SR2 is a more simplified rover and what assumptions its design is based on. Details about the RoverSim application and how it compares to SR2 follow in chapter 3.

2.1 SR2 design

2.1.1 Mechanical

The primary design goal of SR2 is to simplify the mobility, sensory, and intelligent control subsystems. One of the mission goals of SR2 is to travel 1km per Martian sol. This drove many decisions that influenced the mobility system. To save weight the need for a physical steering mechanism seemed unnecessary due to an initial assumption that much of the rover's traverse would be nearly straight. This led to a differential drive train with all wheels parallel. Even without the ability to steer each wheel SR2 can turn on a single point.

SR2's unique drive system uses only two motors each located inside of the body which houses the electronics. On a flight version of the rover the body would be temperature controlled to reduce the adverse effects from the environment. Each side of the rover is driven by a split beveled gear drive shaft where the left and right pairs of wheels are linked to their own motors respectively. The left and right sides of the suspension are connected to the body through a passively geared differential. This ensures that all four wheels maintain ground contact with equal force, similar to the rocker bogie suspension. The differential combined with the large ground clearance are what allow the rover to traverse hazards as large as a wheel diameter. The wheels are 20cm in diameter and 11cm wide; they have been designed to reduce drag on relatively flat hard surfaces, while maintaining good tractive properties over rough terrain [44]. A more detailed description of SR2's mechanical design can be found in *Design and Analysis of a Four Wheeled Planetary Rover* by Roman [53].

2.1.2 Electrical

Much of SR2's electronic control system is built from off-the-shelf components to reduce cost and development time. The onboard processors include a PowerPC Macintosh Mini computer running at 2.0GHz and a real time Xport Botball Controller (XBC) [31]. All external devices relay data through a multi-port RS-232 hub connected to the Mac Mini through a USB interface. The Mac Mini runs the Command-Center server application that executes the rover's autonomy software but can also receive high level commands from a client through wireless communication [71]. The XBC performs many of the low-level tasks such as motor control and sensing the internal state of the rover. In particular the Back-EMF voltage is measured and used in a PID loop to control the speed and position of the motors. The XBC is limited to only a few amps of current through its stock motor driver chips so, a separate H-bridge circuit was designed to handle the higher currents pulled by the drive motors. Two 12V Li-ion batteries, connected in parallel, are used to store energy from the solar panel. To prevent damage from overheating caused when the batteries are charged thermistors measure their temperature. Electrical current meters measure the amount of power flowing between the batteries, solar panel, motors and remaining electronics.

Collecting sun light with solar arrays has proven to be an effective form of power generation. At first it was believed that there would not be enough sunlight on the surface of Mars to sustain operations. Sojourner was capable of driving across the terrain and producing data from extended periods of science operations. During the MER mission it was believed that Martian dust settling on the surface of the solar panel would cut power to a minimum and eventually suffocate the rover. Luckily, dust devils created in the thin atmosphere wash the panels allowing light to pass through to the solar cells. Both Pathfinder and MER have lasted many times longer than their initially stated lifespan.

SR2 has a $0.98m^2$ solar panel composed of 3 strings with 40 silicon cells per string potentially supplying a maximum of 160 watts. Together the batteries and solar panel produce enough power for a four hour daily traverse. The power subsystem is based on solar energy because it is a safe and readily available technology and has been flight proven. Therefore, the design efforts to simplify it were marginal. The transition to a flight ready system will see a decrease in solar flux, or reduction in available power, due to the greater distance between Mars and the Sun. However, the rover will require less power for mobility due to a decrease in gravity.

A Honeywell HMR3000 magnetic compass is a cost effective solution to an expensive inertial measurement unit (IMU) that would be found on a flight system. Even though Mars does not have a magnetic field to support such a sensor it is assumed that heading can be computed from the yaw rate gyro of an inertial measurement unit. Where drift in the gyro can be corrected by locating the position of the sun in images taken by the mast cameras. The HMR300 provides pitch and roll angles up to $\pm 45^o$ as well as heading information.

The navigation system of SR2 is entirely based on dead reckoning position estimations from the number of wheel rotations and heading information. Currently the rover does not have a mechanism for measuring when a wheel is slipping therefore it can not account for position errors caused by external forces. A few examples include slippage when crossing loose sandy regions, traversing obstacles, or unexpected collapses in the soil. As part of the experiments conducted with SR2 it was important to characterize these types of errors. A Garmin SportTrack Global Positioning System (GPS) provides ground truth location data of the rover's actual position. The GPS data was not used for navigation.

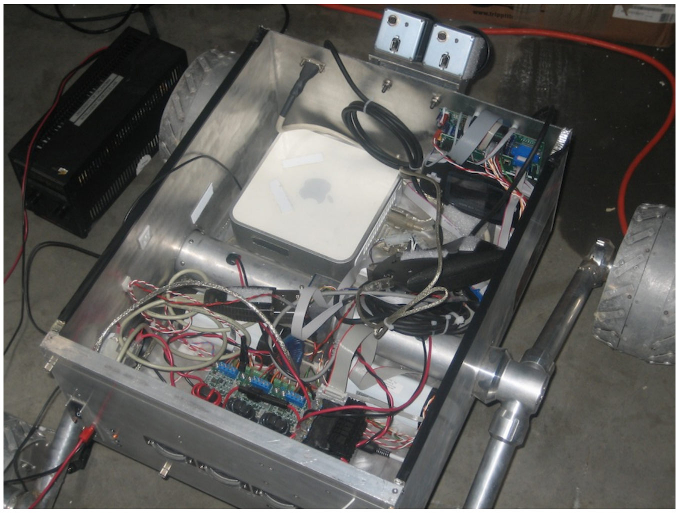


Figure 2.1: Inside the SR2 Chassis: Mac-mini, XBC, Power management electronics and stereo navigation cameras (near top) are not used for this work.

2.1.3 Sensing

A rover in natural terrain will encounter two types of obstacles; positive and negative. Rocks that are above the ground plane are considered positive obstacles. Holes, craters, and gullies are examples of negative obstacles. Most rovers will stay clear of negative obstacles for fear that they may get stuck or damaged from a fall more easily than bumping into a positive obstacle. Whatever the type of obstacle the rover must carry a perception system in order to avoid it. The perceptive system is built on the sensor suite and the processing power needed to analyze the sensory data to determine obstacle size and position.

Sensors for mobile robots can be broken into two types: external and internal. External sensors read the state of the environment surrounding the robot. Examples of these states may measure the distance to an obstacle, color of an object, or ambient temperature. Internal or propriocepive sensors measure the state of the robot itself. Examples of such measurements include: wheel position, battery voltage, and manipulator joint angles. Each sensor can further be classified as active or passive. An active sensor must emit some form of energy into the environment. The change in the amount of returned energy is related to the property being measured. A passive sensor relies on the energy already provided by the environment.

Important properties for selecting sensors to be used for obstacle navigation include: Complexity, Cost, Error Rate, Mass, Power, Sensing Range, Size, and Speed of Operation [17, 48]. Tradeoff decisions between the items listed have to be made when selecting sensors for a perception system to detect obstacles. Sensors that have a larger range like a stereo vision system generate a large amount of data. Processing the data so obstacle locations can be extracted is computationally intensive and can require significant time between updates lowering its speed of operation. A scanning laser range finder can also have a large range and achieve a higher refresh rate when compared to stereo vision. Data values from the scanner represent depth directly, therefore less processing is needed to extract obstacle information between updates. Bump switches mounted on the front of the rover could detect contact with obstacles. These simple sensors require very minimal mass, power, and processing and are very cheap. However, laser scanners and contact switches suffer from much lower resolution than a stereo vision based perception system. Contact switches severely reduce the sensing range and can not detect negative obstacles. It is important to note the major differences between the various types of perception as their selection will affect the design and performance of the rover.

A brief description of the sensor types commonly used for sensor suites on mobile robots is given below to identify their characteristics.

Contact or tactile sensors are used as bump sensors or on end-effectors for grasping. More complex tactile sensors can measure the amount of force on an object.

Bump sensors are tactile sensors for collision detection in mobile robots. In some cases contact sensors are a last resort and if the robot's speed is sufficiently high the damage caused by colliding with an obstacle cannot be avoided by the use of these types of sensors.

IR Infrared sensors emit IR light then correlate the recovered intensity to distance. Typical ranges vary from a few centimeters to a meter or two depending on the sensitivity of the receiver. IR is adversely affected by intense ambient light such as sunlight. Dark materials can absorb the emitted beam. In some configurations these sensors are used as non-contact bump sensors since they are relatively simple, cheap and low power while having a sensing range large enough for the rover to maintain a safe distance from any hazards.

Ultrasound or ultrasonic sensors operate by emitting a cone-shaped shock wave from a transducer and receiving its echo. Ultrasonic sensors use time-of-flight of the pressure wave to estimate distance. However, it is difficult to record any angular precision without multiple receivers. False readings can be recorded by neighboring ultrasonics due to cross talk and specular reflections. Ultrasonics are blind to close obstacles due to the settling time of the transducer after the outgoing pulse is sent. Readings are unreliable during this interval. While an ultrasonic sensor could be designed for use in the Mars environment the power required for the emitter to produce a sonic pulse large enough to detect its reflection in the thin atmosphere would be impractical[14].

LIDAR or LADAR light detection and range sensors operate on the time of flight, phase shift, or triangulation principals of an emitted laser beam [19]. TOF and phase-based sensors are more efficient since the laser can be pulsed due to the fact that the beam remains collimated or parallel over long distances. A single detector can be used along with a rotating mirror to obtain 2D range data. The range and bearing of detected obstacles relative to the position of the sensor head is recorded. Depending on the power of the laser system these range finders can obtain readings from a few meters to a kilometer or more. To obtain 3D range information the sensor is typically mounted at an angle so the sensing plane sweeps out a volume as the robot moves. Spurious data is caused when the rover is pitched up or down on uneven terrain. This can be corrected by fusing the LIDAR output with internal roll and pitch sensors.

Mechanical parts limit the life and stability of the sensor but micro mirror arrays could reduce these effects as well as improve functionality and resolution by splitting the primary beam into multiple smaller beams (MemsLadar). Scanning laser range sensors are limited to low definition sensory information due to the use of a single receiving array. However, technology based on forward looking infrared detectors (FLIR) has been used to create Flash LADAR [59, 60]. A 2D chip like a CCD or CMOS image sensor that incorporates additional timing circuitry to each pixel is the foundation for a Focal Plane Array (FPA). This type of sensor can generate a low-resolution image where each pixel value is related to range information providing a depth map of the scene without any computation[10]. Currently the cost of producing the FPA is very high when compared to other sensing technologies.

Laser sensors typically do not have sufficient resolution or data rates for object recognition. They suffer from similar effects seen in IR sensors where not enough light is returned from light absorbent surfaces. The additional timing circuitry in flash LADAR increases pixel size on the FPA, which limits resolution.

Vision provides a rich source of sensory information. Various dimensions of an object can be detected like size, distance, and color. If the same object is detected in multiple frames the objects relative motion may be determined. The ability of a vision system to perform these functions can be affected by many parameters such as camera focal length or FOV, mounting position, look-ahead, resolution, and robot motion [17, 48, 25, 38, 15].

Stereo vision systems use two or more cameras and require a computationally intensive algorithm to extract depth or range to obstacles. The difference in the location of pixels corresponding to the same object on each image is known as disparity and is inversely proportional to the distance to the object. The error rate in calculating a depth map can be very high if there is not enough texture variation on the obstacle surface. This has been the case on Opportunity and Nomad during operations on terrain with near uniform surface properties such as fine sand and snow respectively. Stereo is a passive sensor that does not require any projected signal and uses no moving parts making it more robust than active sensors. Also the imagery data can easily be decoded by a human in the event the error rate becomes unacceptable, which is a dominant reason it is selected for flight missions.

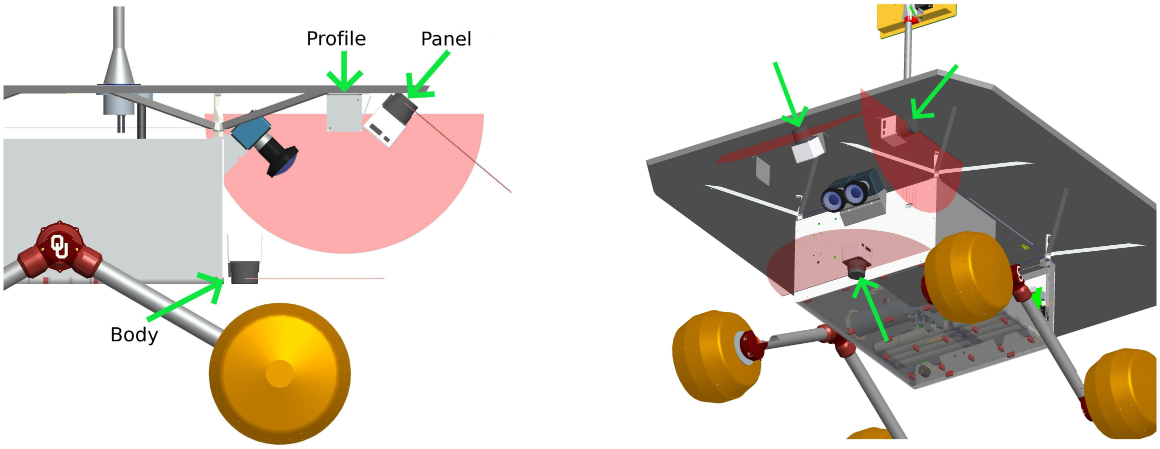
Depth from focus or defocus is another type of vision-based sensor using variations in the focal length of the camera to determine distance. The amount of computation to obtain distance measurements by depth from focus can be reduced when compared to stereo vision. Computation is reduced because the algorithm for detecting focal points of an image can be done in a single pass over the pixel data [50]. The distance of an object or part of an object that is in focus is proportional to the focal length of the camera lens. The focal length of the camera lens can be changed and the process is repeated to create a depth map over varying distances. However, this system requires the use of multiple cameras or the rover must remain stationary while capturing multiple images to create an accurate map.

The primary obstacle avoidance sensors for SR2 are LIDAR based Hokuyo URG-04LX scanning lasers. They were selected because of their ability to measure distance to obstacles with very minimal computation at a scan rate of about 10Hertz while drawing only 2.5W and weighing 160grams. The range data is sent through a standard serial port making the interface to the rest of the navigation system straight forward. Each scanner has an effective range of 4m over a 240^o swath in a single plane. The range has been programmatically limited to 2m on SR2 to reduce the error rate caused by the intense ambient sun light in the field.

Three laser scanners are mounted to the body and underside of the solar panel on the forward end of the rover, seen in figure 2.2. The body laser is mounted upside down just above the bottom surface of the body. The body is approximately 20cm or 1 wheel diameter above the ground. This is the maximum height of an obstacle the rover can physically surmount. The laser sweeps a semicircular swath 2m ahead to detect if the rover should maneuver around an obstacle to the left or right. This laser can only detect positive obstacles because the scan pattern remains relatively parallel to the terrain. It can not detect holes or negative obstacles without the scanner pointed downward.

The profile laser scanner is mounted with the scan plane vertical and parallel to the length axis of the rover. It is used to detect the profile of the terrain in front of the rover. If the entire body laser is blocked the rover proceeds ahead at a reduced speed. At this point the profile laser is used to detect whether the rover is driving up a slope or into a wall. Unfortunately, the data from the profile laser proved to cause more confusion for the avoidance software during testing as the rover pitched up and down, so it was removed.

The panel laser is mounted underneath the solar panel's front edge at a 45^o downward angle. It detects the distance to the ground approximately 60cm in front of the wheels on level ground. Both positive and negative obstacles can be detected by this sensor. If a positive or negative obstacle is detected in front of a wheel or a positive obstacle is detected in front of the body, the rover will turn to avoid it, stopping and performing a point turn if needed. The definition of an obstacle by the panel laser is determined in part by the roll and pitch of the rover at that time. For example if the rover is going down hill then a positive obstacle is considered less of a hazard and may simply be the bottom of the slope.



(a) Profile view

(b) Iso view

Figure 2.2: Hokuyo URG-04LX laser scanner placement on SR2 indicated by green arrows. Note the scanning pattern of each sensor shown as red semicircular segments though not at full radius.

2.1.4 Rover Control

SR2 navigates by a list of waypoints selected by a person from satellite imagery of the test area. The waypoints provide a more global reference path for the rover to follow. A plan consists of the list of waypoints selected as well as an optional function the rover must satisfy as it reaches each of the points.

The CommandCenter application is a graphical interface for creating a plan and viewing the rover's state of health. The application runs on a separate computer and connects to the rover control server through a local wireless network connection.

Various science objectives such as panoramic imagery or spectral analysis may be selected for SR2 to perform. Plans can be sent from CommandCenter for the rover to execute. In the event of an error the user can tele-operate the rover with a joystick interface. During normal operation the user can view the rover state in real time. This information includes the current rover location and GPS position for comparison, power readings, compass heading, sensor readouts, and any error messages.

Waypoint Navigation

The plan from CommandCenter contains waypoints in rectilinear coordinates where the origin is the rover's starting position at the beginning of the traverse. The onboard system checks to see if there are any outstanding waypoints. If so, it takes the next point, calculates the vector between the current position and the waypoint and starts toward the new point. If the vector to the goal is more than 8^o from the current heading a course correction is performed. A point turn is executed for any large angle turns greater than 30^o. Waypoints are successfully reached if the rover is within one meter of them. Upon reaching the point the rover will pause and perform any science activities or functions that were previously specified in the plan.

The power consumed by the motors and electronics is continuously monitored and compared with the battery capacity. If a low voltage state is detected the rover will stop, issue a warning to CommandCenter and enter a timed sleep mode. All sensors are powered down and the Mac Mini is put to sleep for a period of 20 minutes to allow the solar panel time to recharge the batteries. If sleep mode is entered later than 3pm local time it is assumed there is insufficient sunlight for a quick charge and the rover will remain asleep until the following morning.

Obstacle Avoidance

Since the resolution of the satellite images is limited to approximately a meter the rover must autonomously avoid any obstacles not visible between waypoints. The obstacle avoidance part of the navigation software is based on simple reactive behaviors dependent on SR2's current pose and range information.

A hazard may be encountered that is perceived insurmountable by the perception system while the rover is moving toward a waypoint. The body and panel lasers cover a 124^o wide circular sector divided into 16 segments. At a two meter radius each segment for the body laser is 27cm wide. On level terrain four segments of the body laser and eight segments of the panel laser cover the width of the rover. The range of each segment is defined by the shortest distance measurement of 22 laser readings across the segment. Obstacles are detected when three or more consecutive segments indicate an object 15cm above or a void 15cm below the ground plane.

When avoiding an obstacle the rover must pass it by two vehicle lengths before it will try and resume its path toward the waypoint. If another obstacle is detected prior to reaching this safe path distance, the rover will avoid and proceed two vehicle lengths past the most recently detected obstacle before resuming its course. If the rover turns left at the first obstacle it detects, it will turn left at all future obstacles until it is able to resume its path to the waypoint. A similar strategy is pursued if the rover had turned right. If an obstacle is detected within two meters of the current waypoint then that waypoint is skipped and SR2 proceeds to the next point on the list. This prevents the rover from continuously circling a goal in the event the waypoint is encompassed by obstacles. A no progress limit of 15 minutes keeps the rover from straying too far off course while avoiding obstacles. This means an error is indicated and the rover stops to ask for help from CommandCenter if it has not made progress toward the goal for 15 consecutive minutes.

SR2 has two speeds when driving. On relatively level ground clear of obstacles it is programmed to drive up to 20cm/s. In areas of potential hazards the rate is slowed to about 12cm/s. The slower rate allows more update cycles of the avoidance software per meter traveled to better define the location of obstacles.

A few extreme conditions have been defined to keep the rover safe and ensure all systems are working properly. If any of the conditions are met the rover will stop immediately and request help from the user through CommandCenter. The following is a list of the possible errors.

- The difference between updates in rover position is greater than 2m. Indicates failure of the serial communication to the XBC.
- If excessive amounts of current are flowing through the motors they are assumed to be stalled.
- The rover has been avoiding obstacles longer than the 15 minute time limit.
- The rover pitch angle is above 15^o
- The rover roll angle is above 20^o

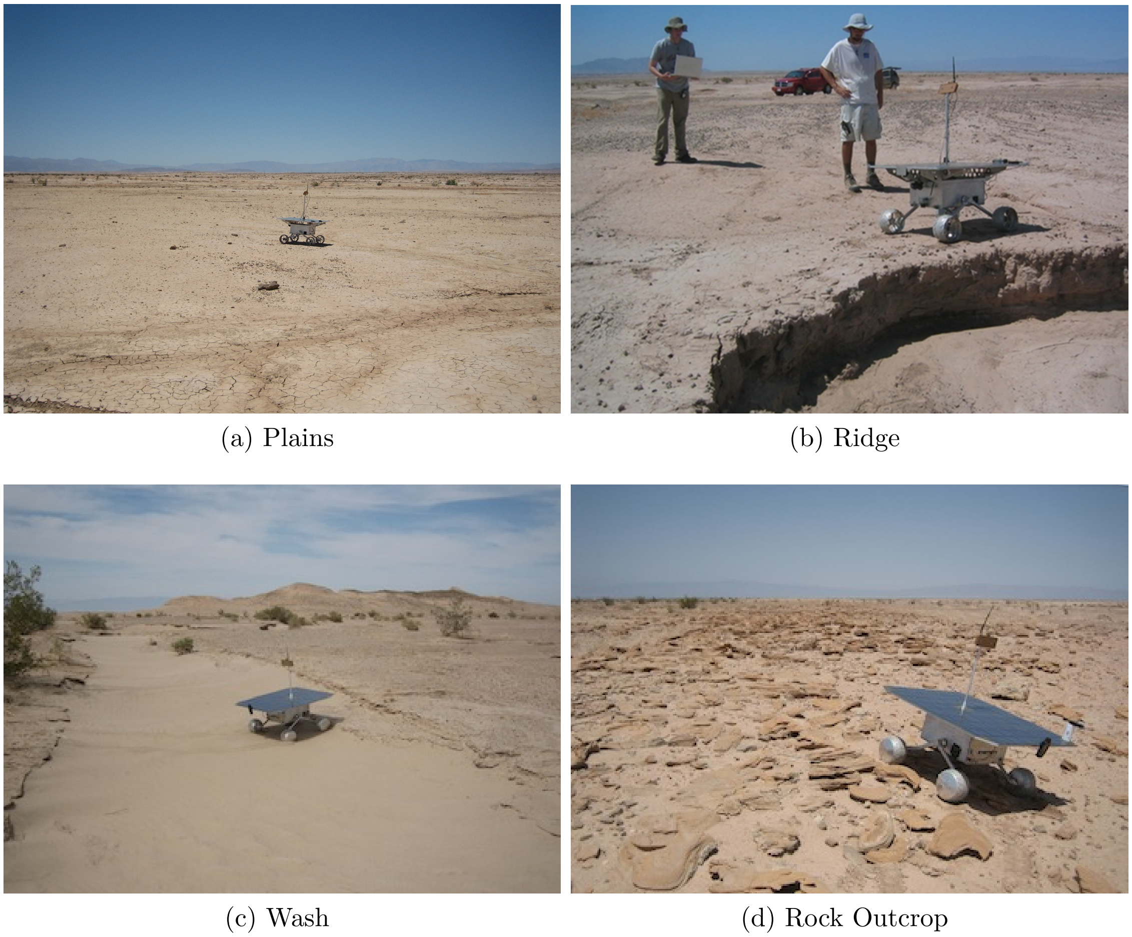
At any time the rover can be interrupted by CommandCenter, except when it is in sleep mode. The user can send a newly edited plan or use the joystick for manual control. The latter is usually only done when the rover gets into a situation where it asks for help.

The rover side server and the CommandCenter applications have been written in various programming languages throughout the life span of the field work. While the underlying flow of the program has not changed, the graphical design of CommandCenter has been improved for better visibility of sensor readings and rover location.

2.2 SR2 Field Experiments

Three field tests were completed in the Anza Borrego desert over a total of 19 days during the summers of 2006, 2007 and 2008. The field test site was selected because it reflects similar geological features that are analogous to the most easily accessible layered or sedimentary materials on Mars [37]. The topography is relatively low, with elements of non-traversable terrain such as steep slopes, escarpments, deep channels and large rocks relative to the scale of the rover. This configuration of terrain elements is seen on Mars where layered rock is created by wind and lake-water sediments. The test site is located near the western edge of the Salton Sea Lake in Southern California where similar processes are taking place [45, 52]. The images in figure 2.3 show some of the elements the rover crossed during the tests.

Many of the types of terrain covered by SR2 were also seen by the MER rovers. Figure 2.4 is a set of panoramic images taken from each MER rover. Figure 2.4a is from Opportunity at the rim of Victoria crater and shows sharp drops surrounded by vast plains similar to the ridges seen by SR2 in 2.3b. The terrain encountered by Spirit during its climb to the summit of Husband hill is more hilly and sparsely covered with rocks, figure 2.4b. The rock distributions in the outcrops near Spirit do not appear as densely populated as the ones found by SR2 in figure 2.3d.



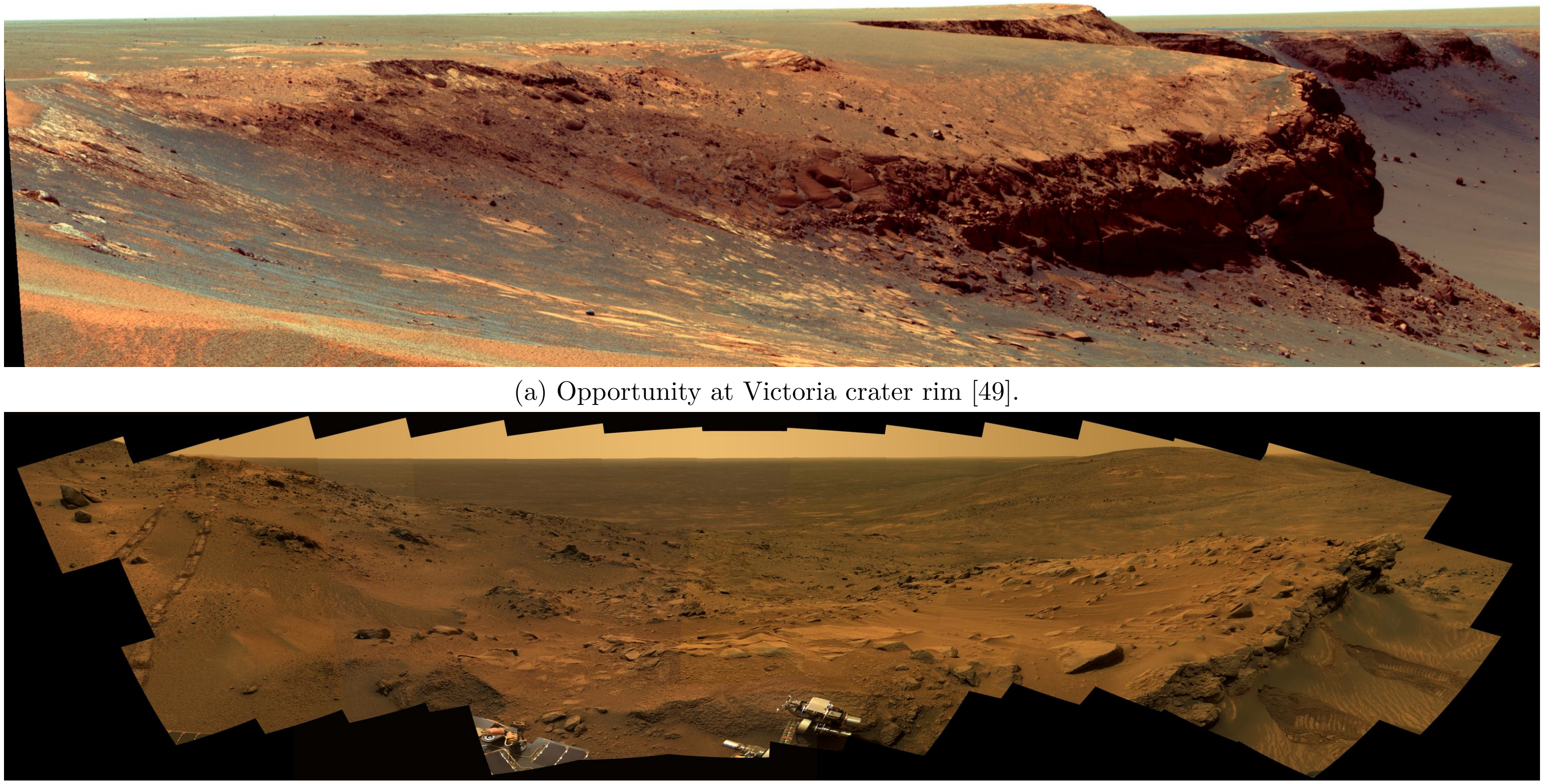
(a) Plains

(b) Ridge

(c) Wash

(d) Rock Outcrop

Figure 2.3: Terrain elements found in the Anza Borrego field test site.



(a) Opportunity at Victoria crater rim [49].

(b) Spirit near the summit of Husband Hill [49].

Figure 2.4: Mars Exploration Rover terrain elements

SR2 was able to traverse along a path of waypoints determined by satellite images. Google Earth images were used to select the waypoints prior to the start of the test. A total of 218 waypoints were imported by CommandCenter which converted the Lat/Long coordinates, from the KML file produced by Google Earth, into relative positions in meters from the start. The operator selected various target waypoints to conduct science, then sent the final plan to the rover. Before executing the plan the rover's position was calibrated to the current GPS position. Each of the three tests were conducted in the same manner over the same set of waypoints with slight variations in hardware.

A day of driving was completed by letting SR2 autonomously navigate from waypoint to waypoint. Much of the time the field personal just followed behind and monitored the rover from CommandCenter. Occasionally some form of vegetation was encountered which is not readily detected as an obstacle by the perception system. A person would stand directly in front of the plant to provide a wider surface for the sensors so the rover could determine it was an obstacle. A large tarp was eventually used to wrap the vegetation if it was in the rover's path. This is shown in figure 2.5. As far as the rover was concerned, vegetation became large rocks. This problem would not exist on a true Mars mission due to the lack of vegetation.

Operator intervention events occurred only a few times throughout the field trials. These situations were resolved by tele-operating the rover a few meters away from the incident spot. During the 2006 field test SR2 bore a set of wheels that were open or did not have "hubcaps" as shown in figure 2.6a. Half of the events requiring operator assistance during the 2006 trial were caused by rocks catching the inner edge of the wheel as it made a skid turn. Similar situations during the second and third field tests were avoided with a new fully enclosed wheel design seen in figure 2.6b. In the cases where the rover was stuck it would initially detect a stalled drive motor and begin a series of maneuvers to free itself. When this failed the rover stopped and a message

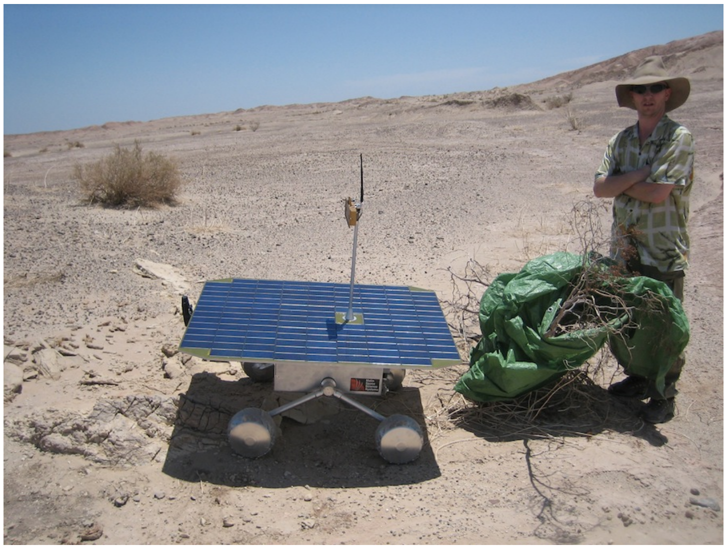
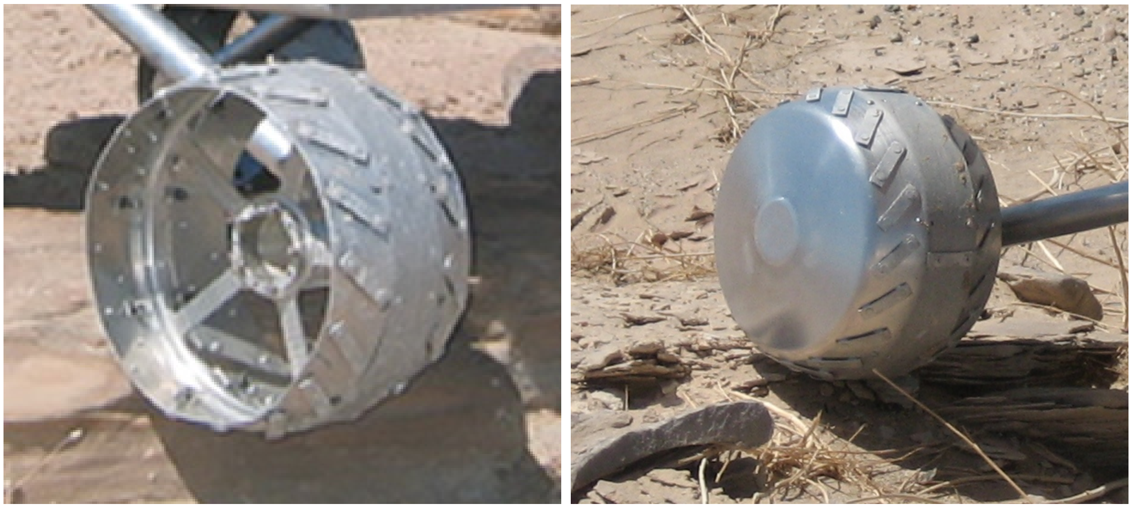


Figure 2.5: The green tarp is used to create more surface area over vegetation for the range finders to obtain proper readings.

was sent to CommandCenter indicating an error. The user could then evaluate the situation and determine the appropriate action for the rover to proceed.

During one event the soil gave-way underneath SR2 causing it to slide down the slope it was trying to climb. By the time the rover stopped its roll angle was considered too severe. It called for help and was tele-operated about a meter whereupon it could continue on its own. On an actual mission an egress of a few meters to extract the rover may take a few days to plan which would be determined from the panoramic images of the surrounding area by the mast camera. Each time SR2 was tele-operated it was never moved farther than a few meters, well within the range of the mast cameras.

There were instances in which all sensors indicated the rover was performing properly and in no danger but was not making any progress toward the goal due to massive wheel slippage. This happened when the goal waypoint required the rover to cross a dried-up wash or gully with the basin covered in fine loose sand. SR2 could easily drive into these areas and follow the wash for a few meters to find an acceptable place to climb out without corrupting its dead reckoning position estimate.



(a) Open wheel design

(b) Closed wheel design

Figure 2.6: The open wheel design used during the 2006 field test caused hang-ups on rocks. This situation was mitigated in 2007 and 2008 with the closed wheel design.

The problem occurs when climbing out. As the front wheels climb the rover begins to pitch up putting more weight on the rear wheels. This creates higher surface pressure causing the soil to give-way. The rear wheels sink and make very minimal progress forward only because it is trying to dig itself out. The rover remains below the pitch limit and the laser sensors do not detect any obstacles so it believes it is moving toward the goal since it has no external position reference. After a few minutes to study this specific scenario the rover was interrupted and tele-operated out of the wash.

The rover continuously recorded health and pose information to a log file on the Mac Mini. The data recorded includes the current time, rover state, waypoint number, position estimate, compass heading, roll and pitch angles, various power measurements, motor speed settings, and GPS latitude and longitude. The average time between samples is 10 seconds except for additional data points for special cases such as reaching a waypoint, stalling a motor or calling for help.

Each day the test was concluded when the voltage on SR2's batteries reached about 10volts. Usually, SR2 would reach this state around 3pm local time. Below this voltage the current required to maintain sufficient torque on the drive motors would blow the main fuse and shut down the entire rover. In a few cases the rover was stopped and put to sleep prior to a low voltage state. It was allowed to charge for a few minutes before continuing to drive a few extra meters or proceed in doing science.

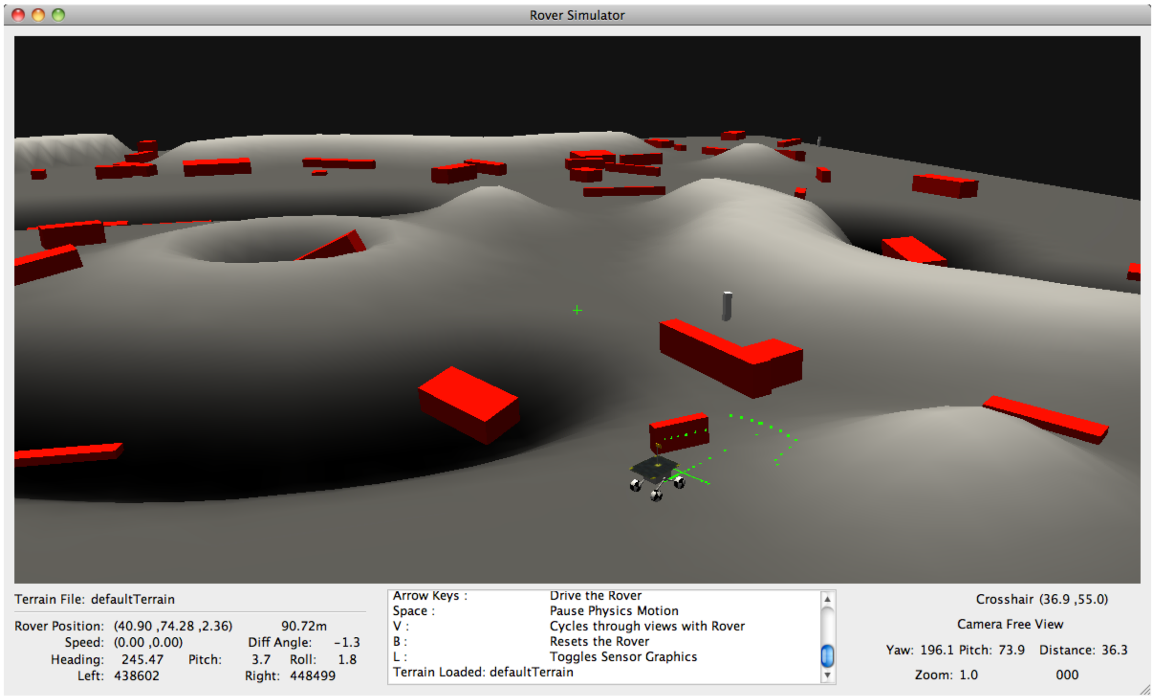
At the end of the day the rover was completely shut down and the solar panel and mast cameras disconnected so it could fit into the support vehicle. During the night the batteries were recharged from an external power supply. It was assumed the remaining daylight periods before and after each days traverse would be enough to recharge the batteries to an equivalent level. This was done to limit the down time of the rover in the field due to the extreme temperatures experienced by the support personal.

Chapter 3

Simulation of a Mars Solar Rover

3.1 RoverSim Design

RoverSim is a 3D virtual environment designed to study the effectiveness of obstacle avoidance software and generate more traversal data without the expense of doing extra field work. It is written in C++ and its graphical interface is built on the Qt framework by Nokia [18]. All graphical representations are displayed on the screen with OpenGL commands. Using Qt and OpenGL not only allows the program to be cross compatible but the source code can also be built on all major operating systems. Figure 3.1 shows an active RoverSim instance with SR2 among a field of obstacles.



```
Terrain File:  defaultTerrain

Rover Position:  (40.90 ,74.28 ,2.36)        90.72m
         Speed:  (0.00 ,0.00)          Diff Angle:     -1.3
       Heading:    245.47      Pitch:       3.7     Roll:     1.8
          Left:  438602                 Right:  448499
```

```
Arrow Keys :               Drive the Rover
Space :                    Pause Physics Motion
V :                        Cycles through views with Rover
B :                        Resets the Rover
L :                        Toggles Sensor Graphics
Terrain Loaded: defaultTerrain
```

```
          Crosshair (36.9 ,55.0)
            Camera Free View
Yaw:  196.1 Pitch:  73.9   Distance:  36.3
     Zoom:  1.0              000
```

Figure 3.1: The RoverSim virtual environment

3.1.1 Why use RoverSim?

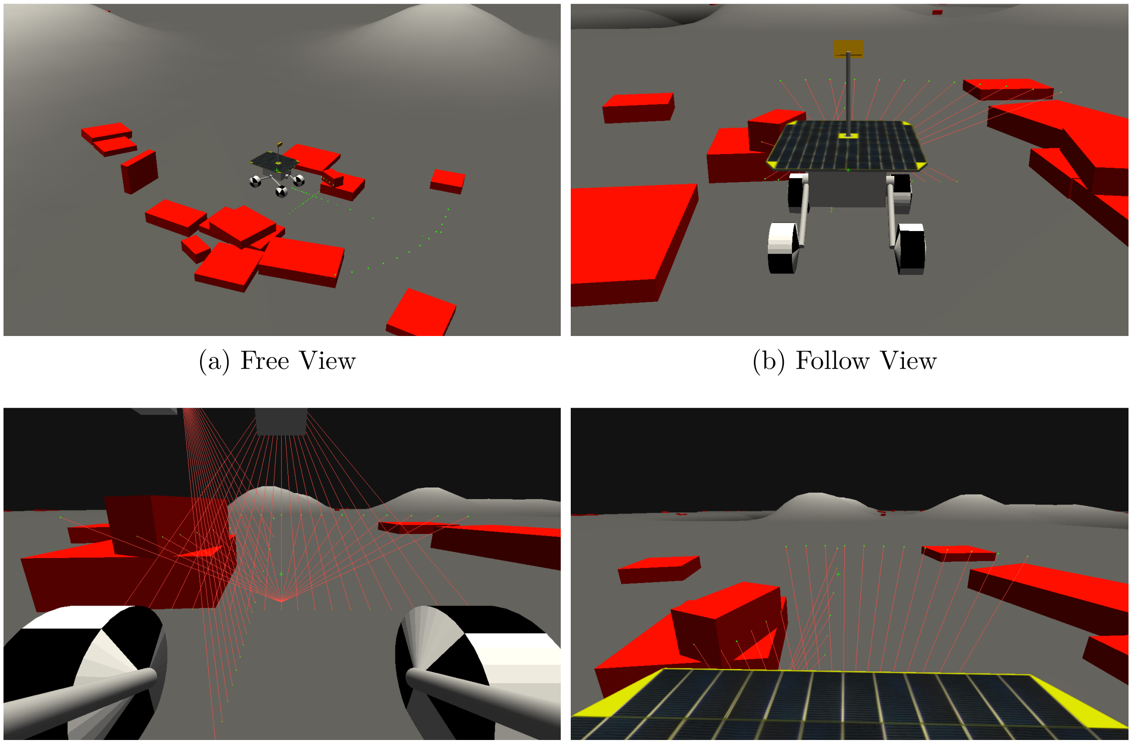
There are multiple 3D vehicle or robot simulators available on the internet. Some like Breve [24] are used to simulate the interaction of artificial life but most are to study algorithms for artificial intelligence. Many of the high fidelity simulators like Cogmation [1] and Webots [43] are based on proprietary software and can not be modified. Simulations that are open source are unsupported at the current date or particular elements of their design are restricted in some manner. SIMBAD [23] is Java based making it cross platform but its simulated robot appears limited to a single dynamic body without suspension or friction properties. The Player/Gazebo Project [20] is likely the best fit since it is open source, includes rigid-body dynamics and has multiple sensors predefined. Though documentation and support for the Player Project is lacking.

The RoverSim application is based on previous experiences and work done by the author for related projects. A large amount of the time to create RoverSim was spent on the proper calculation of configuration space around obstacles and the rover path planner. This work would have to be completed even if a currently available application were used to simulate SR2.

3.2 RoverSim Objects

The simulation is composed of multiple objects including: terrain, obstacles, a rover, path planner, waypoints, data recording and an automator. Graphical tools are available to the user for editing each of the objects as well as some global simulation parameters. Virtual simulation environments are created by loading and editing terrain and obstacles. Once the environment setup is complete a rover and waypoints can be placed on the terrain. The rover may be driven across the terrain manually or the path planner can be initiated to simulate paths the rover would traverse under autonomous control.

Multiple predefined camera views are available to put the user in the perspective of the rover or its various sensors, an example of each view is shown in figure 3.2. These views are very helpful for debugging the avoidance behaviors and defining what scenario caused the rover to take a specific action.


(a) Free View

(b) Follow View

(c) Sensor View

(d) PanCam View

Figure 3.2: User selectable views of simulated SR2 rover. The red lines represent the beam paths of the laser scanner sensors.

3.2.1 Terrain

A height map can be loaded from a gray scale image file to simulate rough terrain. Dark and light areas in the image correspond to low and high areas of terrain respectively. Terrain can be resized and rescaled once loaded. Keyboard hot-keys are also included to further modify the terrain under the center-of-screen crosshairs. All terrain editing features are located under the terrain menu in figure 3.3. The terrain

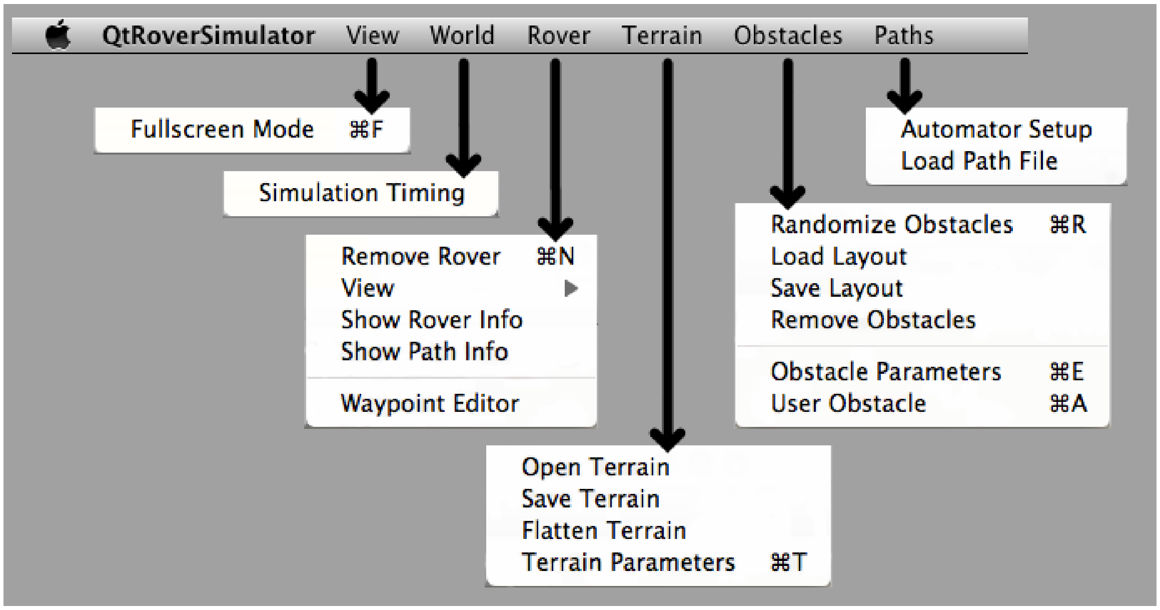
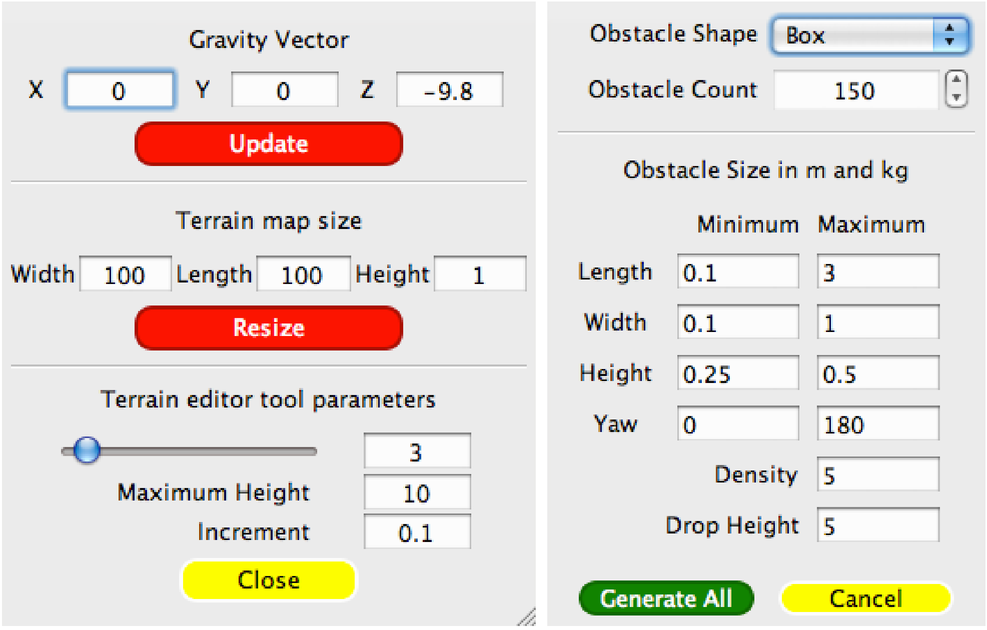


Figure 3.3: RoverSim menu bar layout

tool for setting various parameters is seen in figure 3.4a.

3.2.2 Obstacles

Obstacle layouts can be user-designed by placing objects at specific locations and orientations or created at random during run time. Various parameters are available which affect obstacle quantity, size, shape, orientation, and mass. A drop-down editor window is located under the obstacle menu in figure 3.3. When obstacles are defined randomly their XY positions are randomly selected for the entire region of terrain. All obstacles are assumed positive, meaning they are above the ground plane. The obstacle size and yaw angle are randomly selected between the upper and lower limits indicated under the obstacle parameters window in figure 3.4b. Using the mouse, an obstacle layout can be modified by selecting individual obstacles and changing their position and orientation. All terrain and obstacle layouts can be saved for later use or further modification.


Gravity Vector

X 0 Y 0 Z -9.8

Terrain map size

Width 100 Length 100 Height 1

Terrain editor tool parameters

3

Maximum Height 10

Increment 0.1

Obstacle Shape Box

Obstacle Count 150

Obstacle Size in m and kg

	Minimum	Maximum
Length	0.1	3
Width	0.1	1
Height	0.25	0.5
Yaw	0	180
Density		5
Drop Height		5

(a) Terrain editor

(b) Obstacle editor

Figure 3.4: GUI windows for editing the virtual environment

3.2.3 Simulated Rover

RoverSim incorporates the open source Bullet physics library [13] to simulate the rover's motion and the effects of driving over and bumping into static and dynamic obstacles. The physics engine allows for accurate modeling of SR2's suspension system. The left and right side of the suspension are free to rotate. However, the center differential limits the rover's body to half the angular difference between the two sides. The constraints mechanism in Bullet provides a simplified solution for having these multiple dynamic bodies linked together.

The wheels of SR2 are specifically designed to have a maximum amount of friction orthogonal to the axis of the wheel for climbing. Due to the skid steering design, the wheels have minimal friction in the axial direction allowing them to slide laterally as the rover turns. Bullet allows for anisotropic friction properties to be defined on the surface of an object. This means that the interaction between the rover wheels and the environment can be properly simulated. The friction properties of the wheels are clearly evident when the rover drives up a slope at an angle. The rover experiences lateral motion across the face of the slope in real world field trials. The same effect is accomplished in the virtual environment with the anisotropic friction setting in Bullet.

SR2's three laser scanners are placed at various angles from one another on the body and solar panel. Bullet simplifies generating simulated sensor data based on the pose of the rover through its transformation and ray casting classes. Figure 3.5 shows SR2's suspension and intersections of the laser beams from the sensors with the terrain represented by green dots.

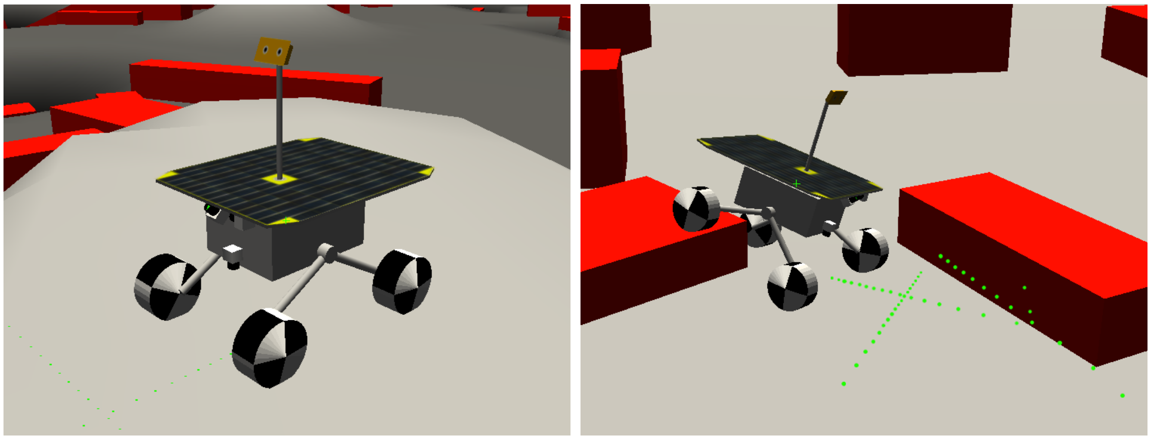


Figure 3.5: Virtual rendering of SR2, the green dots represent laser range finder intersections with the terrain or obstacles.

3.3 Planning a path

The current implementation of RoverSim contains a path planner that is limited to a 2D representation of the simulated environment even though the obstacles and rover are rendered in 3D. Before paths can be generated a few items must be defined that determine how they are created.

3.3.1 Rover Step Size

The distance the rover travels between sensing updates is defined as the step size. The step size is an important metric as parts of RoverSim's path planner refer to it. The length of the step size can be set to any positive distance by the user.

3.3.2 Sensor Range

The sensing range of the virtual rover used during path planning is more simplified than the actual SR2 rover. The range is simply defined by a circular region with the rover at the center. The radius of the circle is equivalent to the sensor range and can be set to any value between the step size and infinity. Only the portions of obstacles that are located within the circular sensing region are used for path planning. Infinite range is equivalent to the rover having complete knowledge of all obstacles. This means obstacles are not shaded from view by one another.

3.3.3 Sensor Visibility

The sensor visibility parameter is selected by the user and can be set to full or limited. Sensor visibility is used to study the effects of sensor placement on the rover. When the parameter is set to limited visibility, obstacles within the sensor range that are shadowed by other obstacles can not be sensed. When the parameter is set to full visibility all sides of the obstacles within range are detected no matter their orientation.

A sensor that is placed high above and looking down is more likely to view obstacles behind one another. Figure 3.6 demonstrates this with views from two different sensor mounting positions. A path planner using information from a sensor mounted low, e.g. the top image, would generate a path to the right of the obstacle directly in front of the rover. The path planner would choose this direction because it is the shortest distance to the goal based on the information from the sensors. However, the bottom image shows the view from a sensor mounted high on the rover. It clearly shows that a path to the right is blocked by obstacles behind one another and a path to the left would be a better choice. Low and high mounted sensors are represented in RoverSim by limited and full visibility settings respectively.

3.3.4 Configuration Space

The path planner can not successfully compute safe paths around obstacles without taking into account the geometry of the rover. A safe path is a continuous path from a start point to a goal where the rover's geometry may come into contact but does not overlap any obstacles. The region of possible configurations of the rover that are not overlapping obstacles is known as configuration space [3, 33]. Building configuration space is a very important step in solving a planning problem. Configuration space creates a level of abstraction between the geometric and kinematic constraints of the rover and the path planning algorithm [28]. In RoverSim configuration space is generated from the obstacles within range of the rover's sensors. To simplify the geometric calculation of configuration space the rover's shape is defined by a circle. It is assumed that the difference between the rover's true form and a circular representation will be negligible due to the relative small size of the rover in comparison to the overall path distance.

All configuration space objects are geometrically based on the underlying obstacles. Each obstacle shape is grown latterly in all directions by the configuration space growth margin parameter. The default value of the growth margin is equal to the radius of the rover. Therefore, configuration space simplifies the path planning problem because the rover can now be considered a point object. Configuration space objects that overlap are linked together in a structured list to simplify the merging of objects that potentially form concave shapes. Figure 3.7 shows a blue circular region repre-

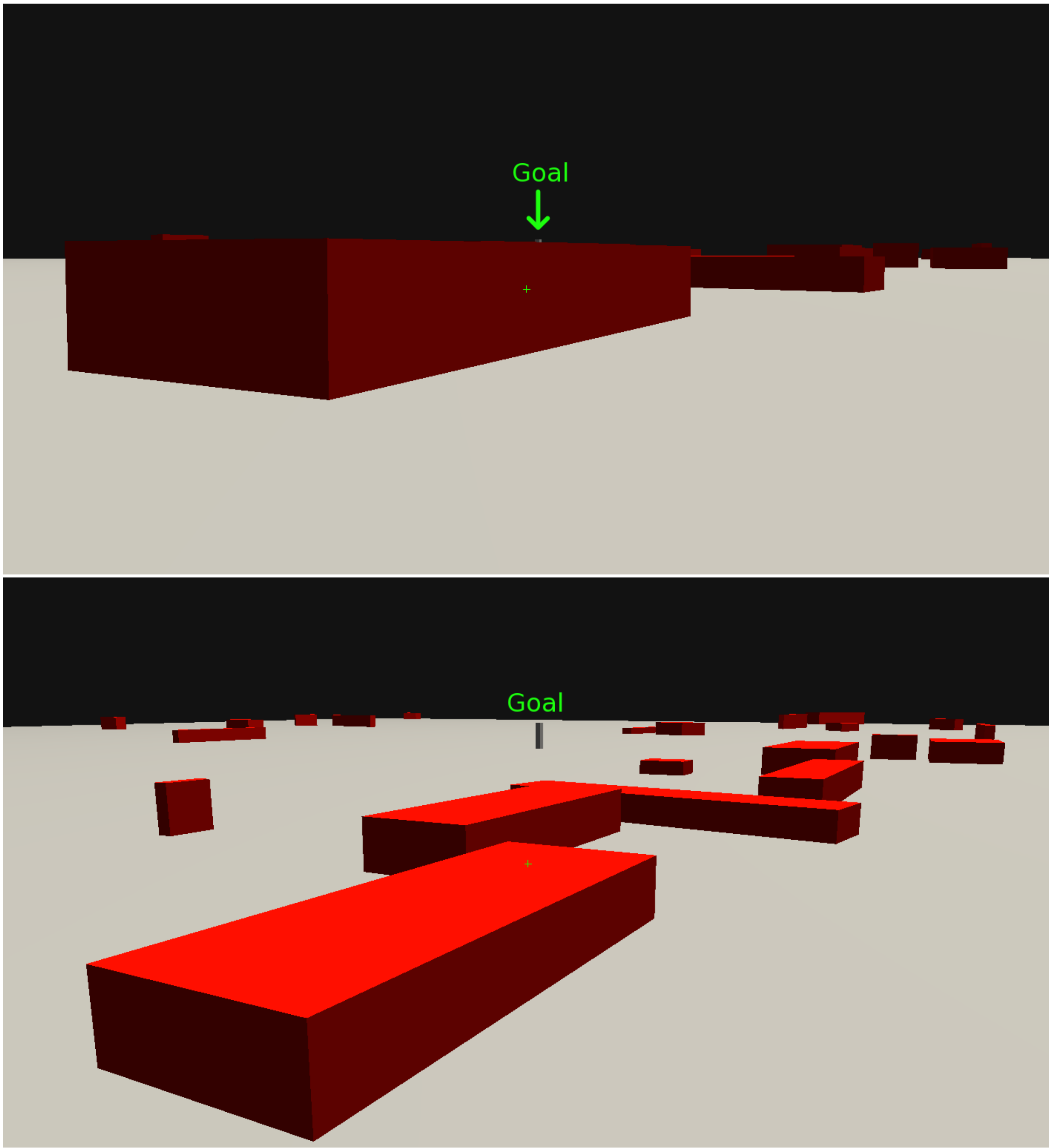


Figure 3.6: Views from a sensor mounted at two different positions on the rover.

senting the sensor range of the rover. All obstacles are colored red. The wireframe structures are configuration space objects generated by the parts of obstacles within range of the sensors. The non-rectangular corners on some of the configuration space objects are created when the edge of an obstacle extends past the range of the sensing region. In figure 3.7 the geometric portion of obstacles used to create configuration space is shaded by the sensor range in a darker red color. The configuration space objects are allowed to extend past the sensor range for the path planner to generate paths that maintain a safe distance from any known obstacles.

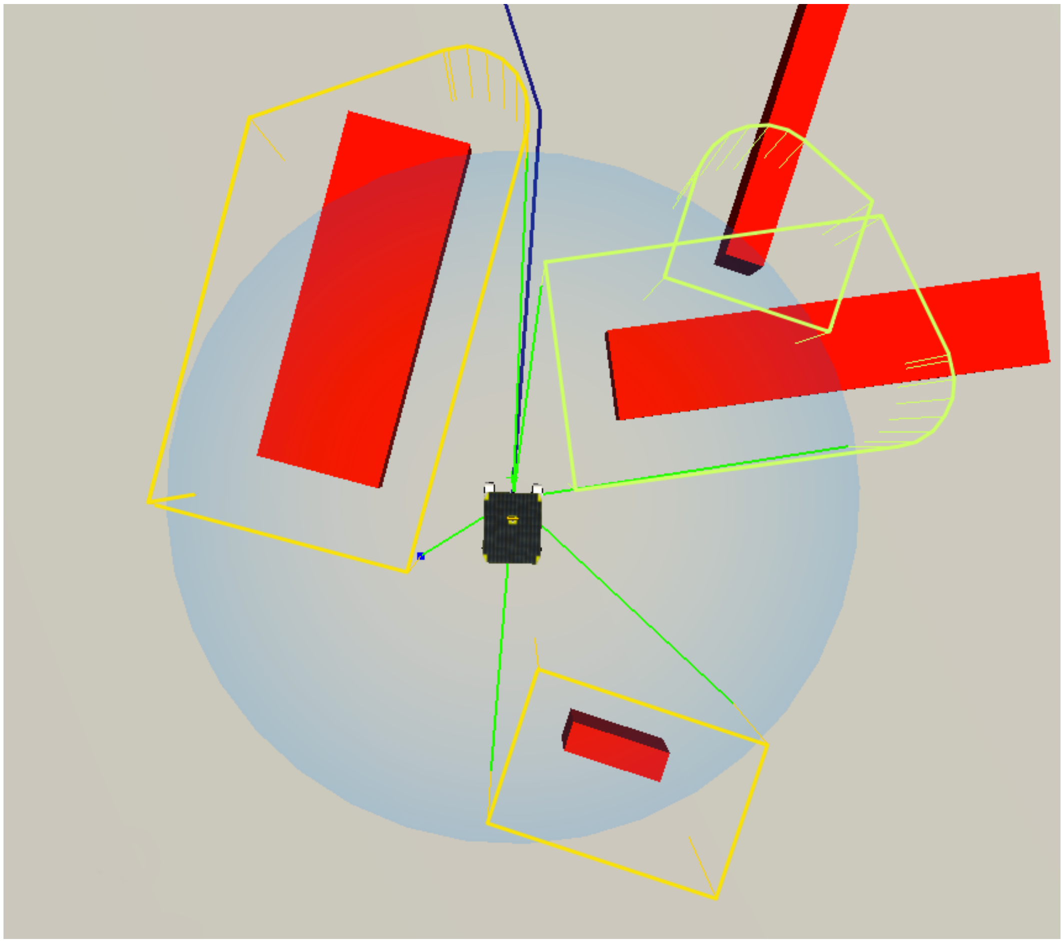


Figure 3.7: Configuration Space objects are defined by the yellow wireframes. They are based on the portion of the red obstacles within the sensor range. Where the sensor range is represented by the blue circle surrounding the rover.

3.3.5 Local Minima

A path planner with incomplete knowledge of the world can not guarantee that it will reach the goal successfully. Part of the planned path that extends past the sensor range of the rover could pass through unknown obstacles. As the rover makes progress toward the goal, new territory is revealed. Decisions made earlier in the path could have lead the rover into an area where forward progress to the goal can no longer be made. This situation is known as a local minima, an example is shown in figure 3.8.

Insert explanation of obstacle spin from Slack [57].

A local minima is detected if the angle between two step segments is greater than 90^o. This will be referred to as a switchback condition and is seen at position 1 in figure 3.8. The initial spin direction of an object is based on the direction of the switchback. In the example the rover will drive toward position 2 indicating a clockwise spin around the obstacle. The rover will follow the edge of any blocking configuration space object until the spin progress limit is reached. A spin progress limit of six meters was used in the example. Therefore, the path length from position 1 to position 2 is approximately six meters. During a spin maneuver the rover may break away from the edge of the object if there is a path toward the goal. The spin progress limit is initially specified under the path scenario parameters. However, to minimize looping paths the spin direction is reversed and the value of the limit is doubled each time the rover encounters the same local minima. A local minima is considered a duplicate if a switchback condition is detected within two step lengths of a previous local minima. The example shows that the rover returned to position 1 after reaching the first spin progress limit. Another local minima is detected at position 1 or within two steps of the previous detection, so the spin direction is reversed and the rover progresses toward position 3.

The spin progress limit defines how far the rover is allowed to travel to extricate itself from a local minima. This value is doubled every time the rover comes back to the same local minima. The spin progress limit is reset to the initial value if a new local minima is encountered.

Path planners exist that will eventually find the goal, assuming it can be reached, even with limited information about the world. However, the paths they are likely to generate will have very low efficiencies. As far as this research is concerned there is not a significant difference between a path with very low efficiency and a path that becomes stuck in a local minima. Both paths represent special situations that are not likely to be seen on a flight mission. The features that form local minima are relatively large when compared to the rover and will readily be detected from orbital imagery of the surface and avoided prior to waypoint selection.

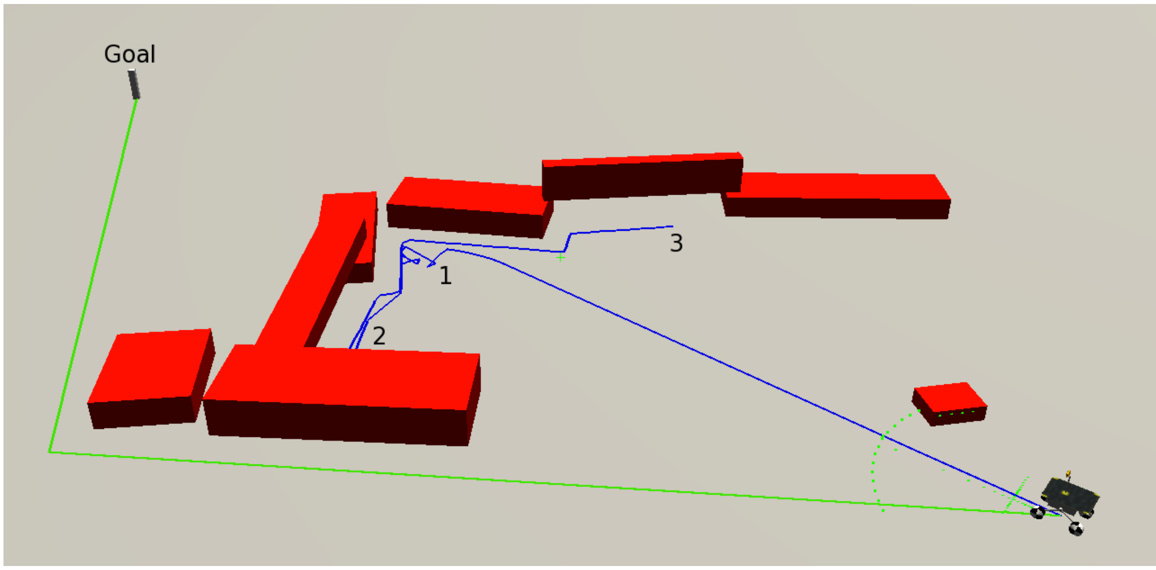
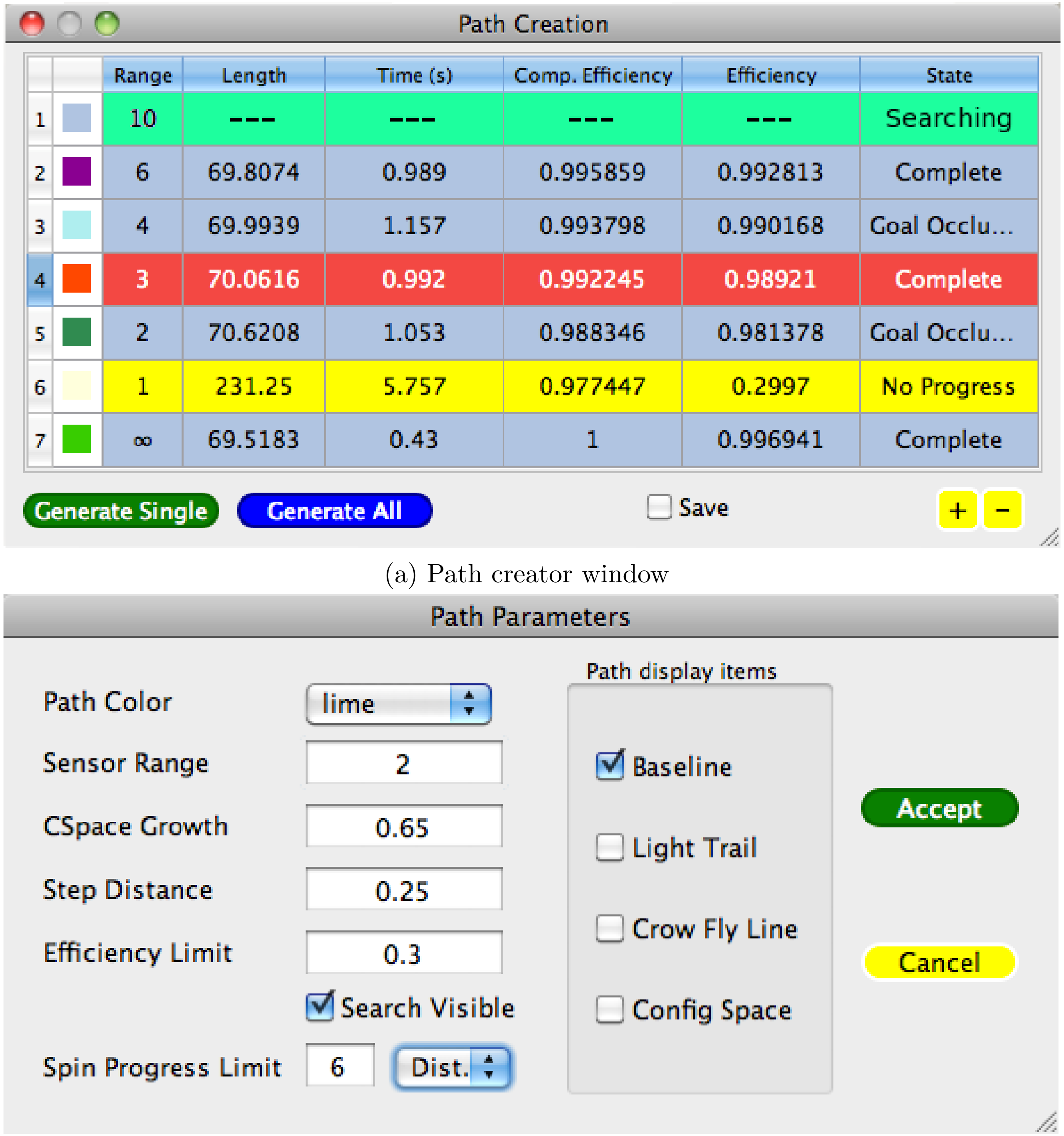


Figure 3.8: The blue path is stuck in a local minima, the green path represents infinite sensor range.

3.3.6 Path Planner

The path planner tool, seen in figure 3.9a, can only be viewed after a rover and goal waypoint have been placed in the environment. This tool allows for the creation of multiple path scenarios based on the search parameters: sensor range, sensor visibility, configuration space margin, step size, and local minima spin progress. When a path scenario is added the path parameter window in figure 3.9b will become active. The parameters can be edited from this window along with graphical features used for debugging.



Path Creation

		Range	Length	Time (s)	Comp. Efficiency	Efficiency	State
1	🟦	10	---	---	---	---	Searching
2	🟪	6	69.8074	0.989	0.995859	0.992813	Complete
3	🟦	4	69.9939	1.157	0.993798	0.990168	Goal Occlu...
4	🟧	3	70.0616	0.992	0.992245	0.98921	Complete
5	🟩	2	70.6208	1.053	0.988346	0.981378	Goal Occlu...
6	🟨	1	231.25	5.757	0.977447	0.2997	No Progress
7	🟩	∞	69.5183	0.43	1	0.996941	Complete

(a) Path creator window

Path Parameters

Path Color: lime
Sensor Range: 2
CSpace Growth: 0.65
Step Distance: 0.25
Efficiency Limit: 0.3
☑ Search Visible
Spin Progress Limit: 6 Dist.

Path display items
☑ Baseline
□ Light Trail
□ Crow Fly Line
□ Config Space

(b) Path Parameters Window

Figure 3.9: All path scenarios are listed in the Path Creator window.

A* Search

The A* algorithm from Hart et al.[21] is used to find the near optimal path to the goal around configuration space objects. The evaluation metric for A* is the straight line distance to the goal from its current search position. A path generated using a sensor range other than infinity is referred to as a limited range path. Limited range paths assume that all space outside of the sensor range is open or free space. For limited range paths the rover computes a path to the goal and moves the distance set by the step size on that path. The process will be repeated until the goal is within the sensor range. A single path is computed from the rover's start point for infinite range path scenarios since complete knowledge of the obstacles are known. This means the rover does not step toward the goal as it does with limited range path computations.

Search Conditions

A path search will stop if one of the following conditions is met:

- The path is successful in finding the goal.
- The path length is significantly larger than the Euclidean distance between the start/goal points.
- A dead end is reached and a path is not possible.
- The path search has taken longer than five minutes of computation.

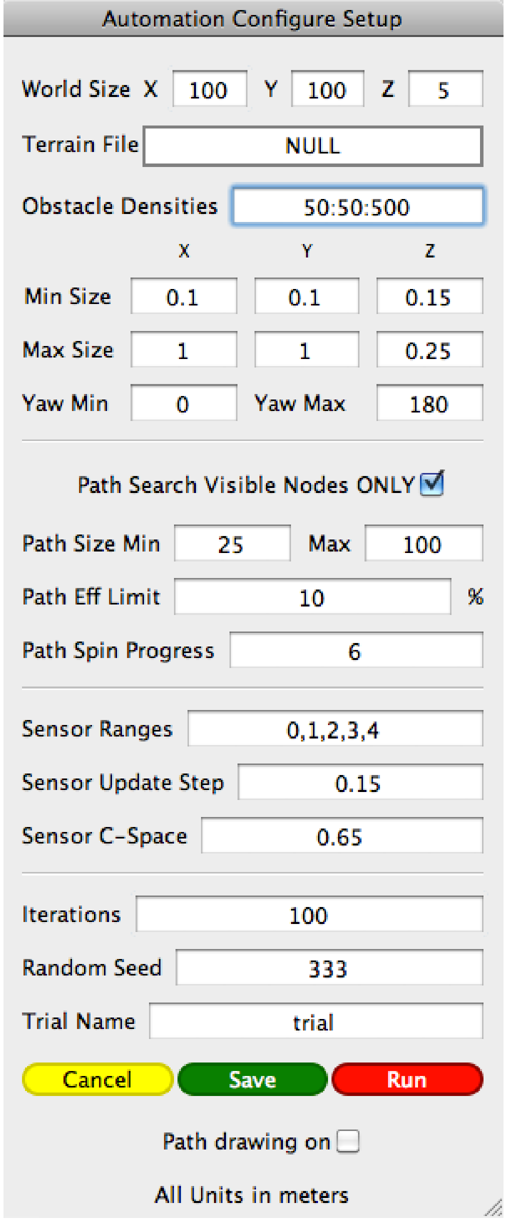
Upon completion of a path search the condition of the search is represented by the state of the computed path. An example of the search conditions can be seen in figure 3.8. The rover's path ends at position 3 because the planner has reached the efficiency limit. The path length has become significantly larger than the distance between the rover's start and goal points. In this case the rover's planner apparently does not have enough information to successfully reach the goal due to the local minima. This is clear when compared to the green path, representing infinite sensor range or complete knowledge of the world. The infinite sensor range path completely avoids the local minima and successfully reaches the goal.

3.3.7 Automator

A configuration file may be created to automatically setup path testing parameters for RoverSim. This feature is accessed through the automator setup under the paths item menu. The default automator window is seen in figure 3.10. It allows the user to define the simulation environment by specifying terrain and obstacle parameters. The following description of the settings and their formats refer to figure 3.10.

One iteration of RoverSim is defined by the series of paths generated from all the sensor range and visibility combinations between a single start/goal waypoint pair in a field of obstacles at a specific density. The number of iterations pertains to the number of randomly placed start/goal positions for each trial. A trial refers to the set of iterations for a specific obstacle density or quantity of obstacles placed on the terrain. If obstacles have been added between trials the process waits until they have reached a steady state. At the beginning of each iteration the rover's start and goal positions are randomly selected. The Euclidean distance between the start and goal must be within the limits defined by the path size fields. Or, a new goal position is randomly selected until the condition is met. Both the start and goal positions are placed outside of any configuration space objects.

The obstacle density field may contain comma separated values where each value represents a trial. Or, it may contain a minimum count, a progression count, and a maximum count separated by colons such as: *min count : progression : max count.* This will automatically add the number of obstacles defined by the progression count to the terrain after each trial is completed. Obstacles will continue to be added until the maximum count is reached after which the simulation is terminated.



Automation Configure Setup

World Size X 100 Y 100 Z 5

Terrain File NULL

Obstacle Densities 50:50:500

	X	Y	Z
Min Size	0.1	0.1	0.15
Max Size	1	1	0.25

Yaw Min 0 Yaw Max 180

Path Search Visible Nodes ONLY ☑

Path Size Min 25 Max 100

Path Eff Limit 10 %

Path Spin Progress 6

Sensor Ranges 0,1,2,3,4

Sensor Update Step 0.15

Sensor C-Space 0.65

Iterations 100

Random Seed 333

Trial Name trial

Path drawing on □

All Units in meters

Figure 3.10: Configuration file editor window

Multiple path scenarios may be added by entering sensor range values separated by commas. Each sensor range corresponds to a single path scenario but all path scenarios have identical step size, configuration space margin and progress limit parameters. If the sensor visibility parameter is set to true then each path scenario of limited range will have a duplicate scenario, meaning one path will be generated with full visibility and the other with limited visibility.

The path efficiency limit condition is a percentage based on the Euclidean distance to the goal. The limit specifies how far the rover is allowed to travel before the planner gives up its search. The travel distance is equal to the Euclidean distance divided by the path efficiency limit. For example if the path efficiency limit is set to 10% the rover is allowed to drive 100 meters total before giving up if the Euclidean distance between the start and goal position is 10 meters.

Before a simulated test begins a random seed number may be specified to ensure there is variation in obstacle placement between tests. The seed number changes after each random variable that is created but all seeds are based on the initial value. The seed number is recorded in a statistics file at the beginning of each trial. This allows for the recreation of the environment and start/goal positions if the setup needs further analysis.

3.3.8 Data Recording

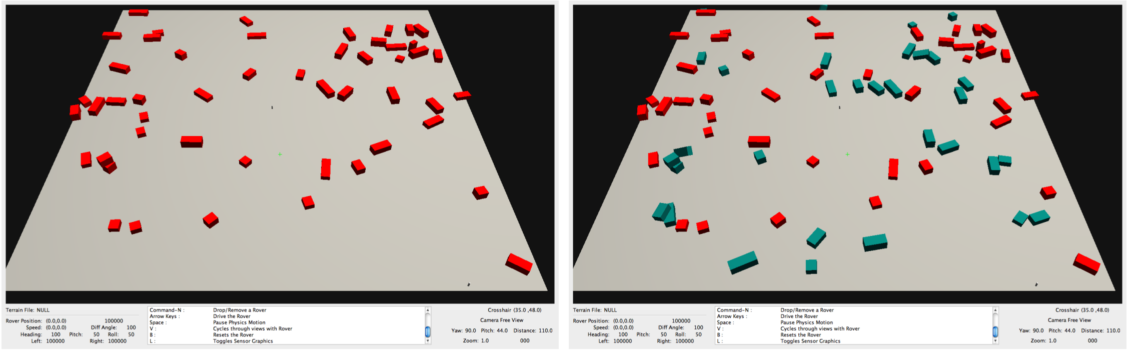
Path data and obstacle layouts are saved in Extensible Markup Language or XML format. Qt provides classes for streaming XML data to and from files easily. An obstacle layout file includes the location of the terrain height map and its scaling factors as well as detailed information about each individual obstacle. A path file includes the location of the obstacle layout file that defines the environment in which the paths were generated. The location of the start and goal positions are listed next followed by detailed information about each path scenario. All the parameters pertaining to a path scenario are saved including the final computed path length, the efficiency compared to the Euclidean distance to the goal, the computation time in milliseconds and the final state of the path. A list of vectors that make up the actual path are also recorded. During an automated run, the data is written to a file after each path scenario is computed to prevent data loss in the event there is a fault in the program. Under the paths menu in figure 3.3, XML files may be loaded to review previously calculated paths that have been saved to disk.

A statistics file is compiled for each trial in a comma separated format to make data analysis easier. The pertinent data for each path scenario includes the sensor range, path length, Euclidean length, efficiency, computation time, path state, and visibility state. The name of the file corresponding to the detailed path information is also listed.

3.4 RoverSim Experiment Configuration

RoverSim was run over multiple weeks of processing on a 64bit 12-core Mac Pro with Hyper-Threading at 2.93GHz. Twelve simulations were run simultaneously, each simulation is a single thread and is confined to one of the processor cores. An instance or thread of RoverSim is referred to as a mission. Every mission begins with a different random number seed. This means that each mission has a unique obstacle configuration even though the remaining simulation settings are identical. Multiple missions were executed so unforeseen effects by noise variables would be minimized in the data. To reduce processing time the openGL commands for drawing the virtual environment were not called upon as part of the simulation.

In its current state RoverSim can only generate paths in a 2D representation of the configuration space. Therefore, all paths were computed on flat terrain with only positive rectangular shaped blocks as obstacles. Obstacle densities started at 50 blocks and increased to a maximum of 800 in increments of 25. At the beginning of each trial 25 new blocks were randomly placed in the field with a uniform distribution, this can be seen in figure 3.11. New obstacles are highlighted in blue. The obstacles were dropped from a height above the terrain and the simulation paused until a steady state was reached in the physics engine before computing paths. The size of the blocks were also randomly selected and ranged from $1500cm^3$ to $250,000cm^3$ or 0.28% to 47% of the rover's volume.



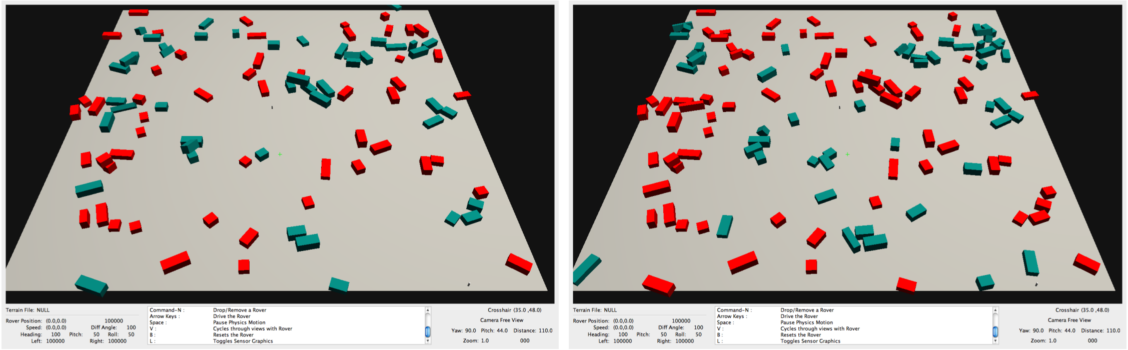
```
Terrain File:  NULL
Rover Position:  (0.0,0.0)              100000
         Speed:  (0.0,0.0)         Diff Angle:    100
       Heading:     100    Pitch:      50    Roll:     50
          Left:  100000            Right:  100000

Command-N :      Drop/Remove a Rover
Arrow Keys :     Drive the Rover
Space :          Pause Physics Motion
V :              Cycles through views with Rover
B :              Resets the Rover
L :              Toggles Sensor Graphics

Crosshair (35.0 ,48.0)
Camera Free View
Yaw: 90.0   Pitch: 44.0   Distance: 110.0
Zoom: 1.0          000
```

(a) 50 obstacles

(b) 75 obstacles



(c) 100 obstacles

(d) 125 obstacles

Figure 3.11: New obstacles, defined in blue, are randomly added to the previous layout at the beginning of each trial.

The terrain covers $10,000m^2$ of virtual space for the rover to operate. The start and goal points are randomly selected to be within this envelope but not on top of an obstacle. The Euclidean distance between the points must also fall in the range of 25m to 100m or the goal point is randomized again until the condition is met.

Once the start and goal points have been established the paths between them can be simulated. For each point pair a series of 15 paths are generated based on the sensor range and visibility of the rover. Seven limited sensor ranges were tested with distances of 1,2,3,4,6,8,10 meters. Each limited sensor range is computed twice, once with full visibility and once with limited visibility. Obstacles in range can be shaded from view by other obstacles based on the relative position of the rover making them undetectable in limited visibility cases. All sides of an obstacle are detectable for full visibility cases no matter their orientation so long as they are within the sensor range. The final or 15th path is computed with complete knowledge of all obstacles and noted as the infinite sensor range path. Figure 3.12 shows an example of a single iteration of simulated paths. The goal waypoint is located in the upper left of the image. Each solid colored line represents a path taken by the rover with a limited sensor range. The infinite sensor range path is shown with a yellow dotted line. It is difficult to see all 15 paths since many of the shorter sensor range paths are very similar and overlap. Some of the paths show the rover had to navigate out of a local minima created by the particular arrangement of obstacles. The paths generated with larger sensor range are visible because the path directs the rover away from obstacles much sooner. The green path represents this situation clearly due to its large sensor range of 10m.

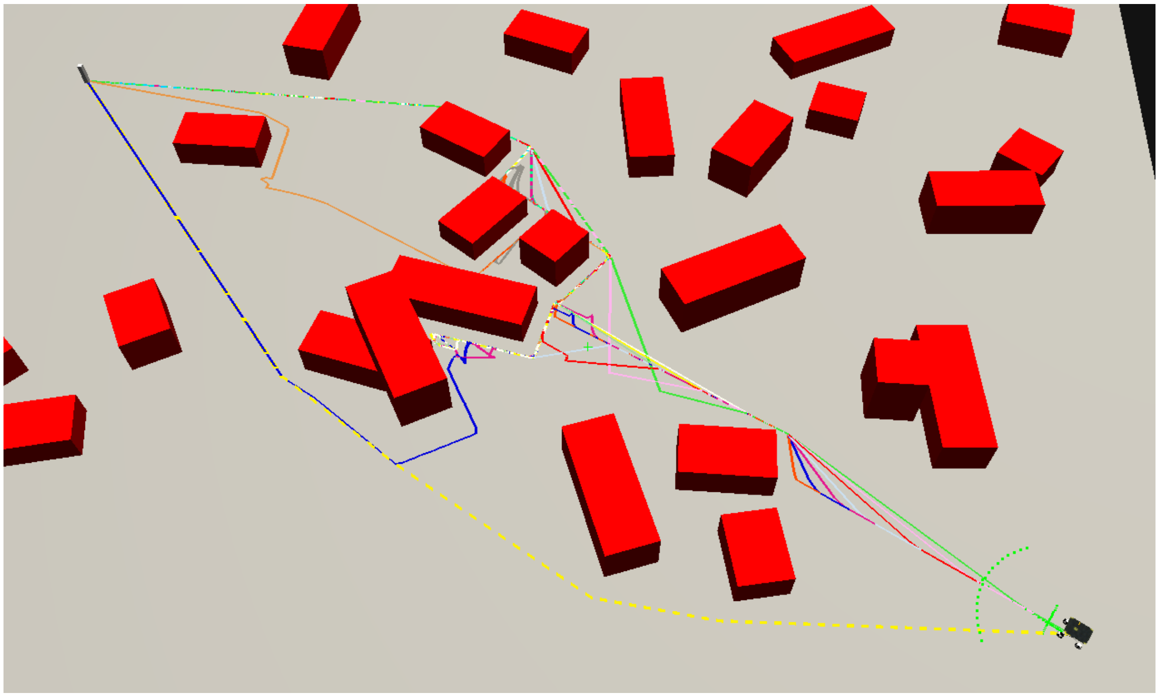


Figure 3.12: Simulated paths generated during a single iteration of RoverSim.

Simulation variables that were held constant for all paths include the following:

World size	100 x 100m
Max Obstacle size	1 x 1 x 0.25m
Min Obstacle size	0.1 x 0.1 x 0.15m
Obstacle initial yaw range	0^o - 180^o
Start/Goal separation range	25 - 100m
No progress efficiency	1%
Spin progress	6m
Step size	15cm
Configuration space size	0.65m

Table 3.1: Simulation variable settings used for experimentation.

The values in table 3.1 are from the configuration file that RoverSim used while in automated mode. The configuration space margin is based on the largest dimension from the center of SR2 to the corner of the solar panel. The no progress efficiency limit forces the path search to quit if the efficiency becomes lower than 1%. In other words the rover has driven 100 times farther than the Euclidean distance between the start and goal.

A total of 732,150 paths were scheduled to be computed producing two sets of data. The first simulation run required 300 start/goal waypoint iterations per obstacle density. Not all paths were completed before a memory leak in RoverSim forced the program to quit prematurely. The second run required 30 iterations so a full set of data could be created in a reduced timeframe.

Chapter 4

Analysis of SR2 Experiments

4.1 Field Trial Results

The outcome of the three field trials in the Anza Borrego desert met the design expectations. The SR2 rover proved capable of safely traversing distances on the order of a kilometer per day over terrain analogous to places seen on Mars. The rover accomplished this using a low fidelity sensor suite and simplified suspension allowing it to reach an average speed of 15cm/s. The mobility system was also capable of surmounting obstacles larger than a wheel diameter while maintaining stability. Even though the rover's mechanism for steering is not an optimal way to maneuver, the power required for mobility remained lower throughout all field trials than what was needed for the CPU, laser scanners and control electronics. The current hazard avoidance system was acceptable in finding a safe path between waypoints. This is significant because the hazard avoidance system is much simpler than that used by MER. SR2 also has theoretically lower overall mobility than MER. These two factors might lead one to expect that the rover would be more prone to get lost or get stuck or have other problems related to ground hazards. Yet this was not the case. SR2 would usually find a way to navigate around obstacles that could not be driven over due to suspension limitations. Only a few situations were encountered where the rover had to request help from the user. Many of the events involved sandy slopes or unexpected collapse of the soil supporting the rover. These types of hazards are not likely to be detected by increased sensor range. The remaining few events may have been avoided with a larger sensor range.

Figure 4.1 shows the distance SR2 traversed each day in the field test. Each of the trial years is marked by a different color. The average distance traveled per day for SR2 is 938m, indicated by the dotted line in figure 4.1. On average fewer than four hours (and never more than four and one half) of driving time were spent each day. This limitation was due to the elevated air temperatures (in excess of $46^o C$) during prime solar power time (noon). The intense heat reduced the solar panel's efficiency limiting the recharging of the rover's batteries. More severely, the extreme temperatures limited the endurance of the field test personnel.

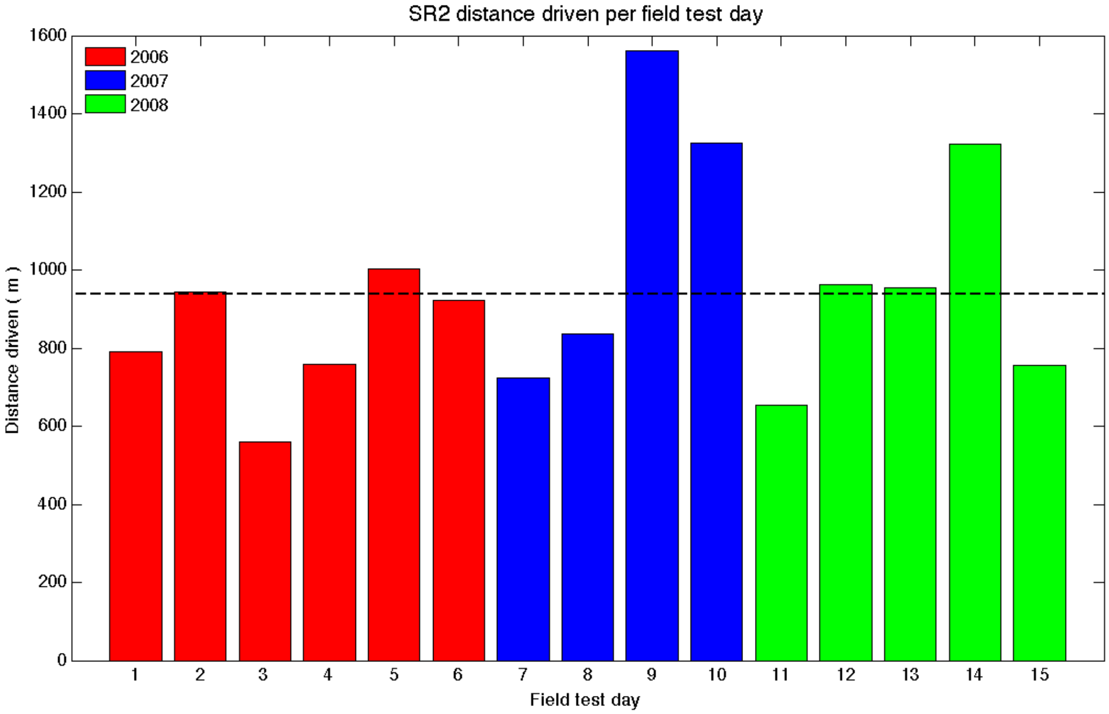


Figure 4.1: The distance traversed by SR2 during each day of the Anza Borrego field trial. Each test year is indicated by a different color. The dashed horizontal line represents the average of traversed data.

The waypoint guide path was approximately 6.64km long and ended in a spiral pattern seen near the bottom of figure 4.2. A total of 14.8km were driven under autonomous control. The light blue path was completed during 2006. In 2007 the trial began where the rover left off the previous year, indicated in green. Once the rover reached the end of the waypoint list it continued in the reverse direction. This can be seen as the green path folds back on itself near the bottom of the figure. The 2008 field test was started at the end of the guide path and made its way toward the start. Three days of traverse deviate from the guide path due to an incorrect magnetic declination offset during compass calibration. One of the days can be seen in figure 4.2 where the dark blue path deviates from the green path near the bottom of the figure. The following day SR2 was placed where it had stopped with the incorrect declination offset. The rover's position was reset and the declination was properly calibrated before the test continued.

The position error is calculated using the distance between the rovers physical position and the intended position in the field. The percentage for each day in figure 4.3 is the error as a percent of the distance traveled for that day. The red bars represent days where SR2 was given incorrect navigation settings that affect navigation such as magnetic declination offset. The average position estimate was 1.29% of the distance traveled and is shown on figure 4.3 as the dashed line.

One area that severely affected the rover's ability to maintain a close approximation of its position is crossing sandy slopes. Typically SR2 maintained an internal position estimate accurate to within 3% of the distance traveled. However, if the rover got stuck while climbing out of a sand laden wash it was unable to detect an error state therefore unable to call for help. During these situations excessive wheel slip caused rapid deterioration of the position estimate and the operators would have to intervene. While crossing these regions did not happen often, detecting sandy slopes is an area of great concern as similar situations have been encountered by Spirit and Opportunity on Mars. These scenarios can halt the rover for weeks at a time before a plan can be formulated to free it.

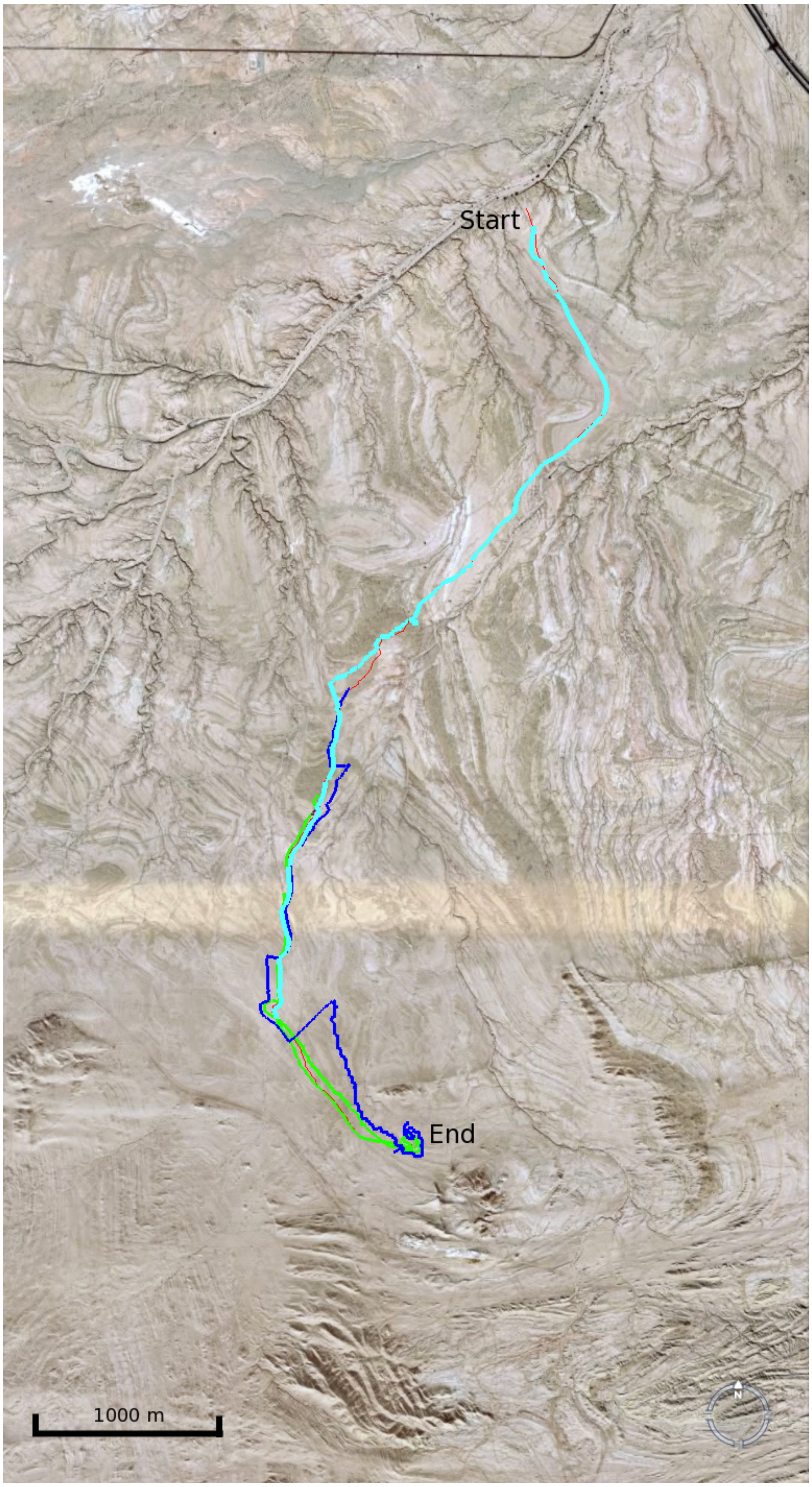


Figure 4.2: Paths driven during the Anza Borrego field test are superimposed on the satellite image taken from Google Earth.

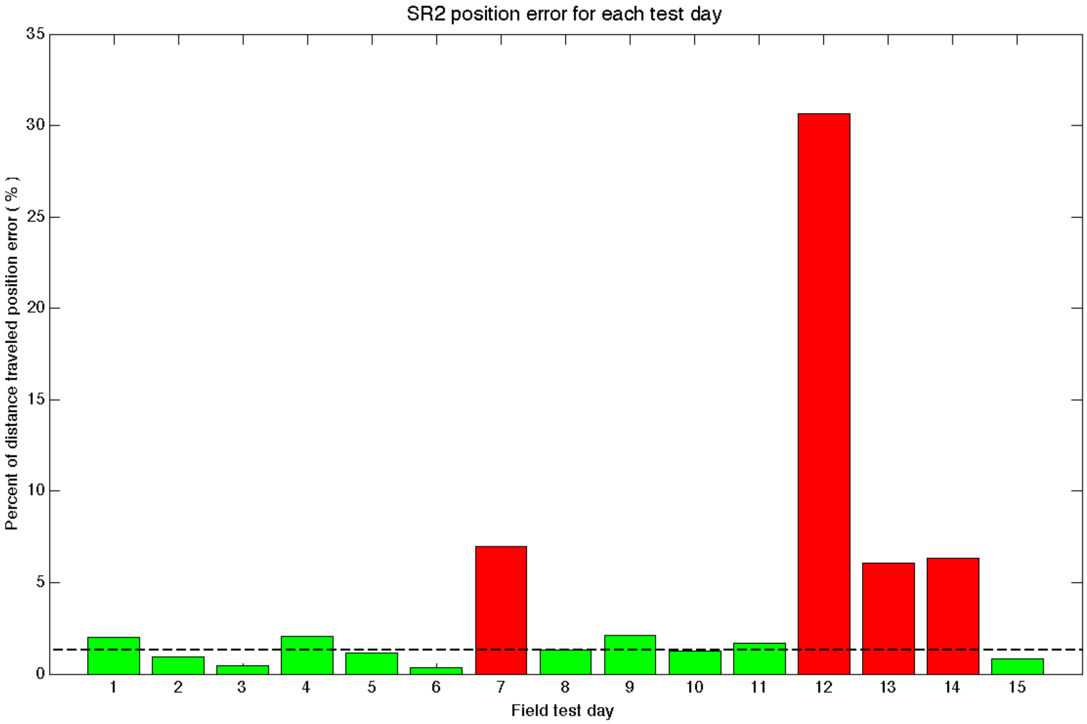


Figure 4.3: SR2 position error measured as a percentage of the distance traveled each day. Days driven with incorrect rover settings are highlighted in red. The mean percent error is indicated by the dashed line and does not include the days with improper settings.

The average speed of SR2 was calculated from the estimated distance traveled based on dead reckoning measurements over the elapsed time between two measurements. Only measurements where the rover was moving were included in generating the speed results shown in table 4.1. The speeds of current Mars flight rovers are also listed for comparison. The value for Sojourner is based on the maximum mechanical speed that it could drive since almost all of the distance it covered was tele-operated. The MER rovers speed is the average speed they were able to drive under AutoNav or autonomous mode. The Mars Science Lab rover (MSL), set to launch in the fall of 2011, has a mechanical speed limit of $\approx 4cm/s$ and its projected average speed under autonomous control over martian terrain will be very similar to its predecessors. Figure 4.4 shows the distance covered over time for each field trial of SR2 in comparison with the other rovers over the same time span. The linearity of the SR2 trials indicate that even when the rover encountered difficult terrain its progress over a kilometer was not greatly affected. Also improvements of the hardware and tuning of the software between trials helped increase the rover's speed.

Rover	Avg. Speed
SR2 2006	10.07cm/s
SR2 2007	11.13cm/s
SR2 2008	15.07cm/s
Sojourner[1]	0.6cm/s
MER	0.6cm/s
MSL[2]	0.8cm/s

Table 4.1: Average speeds of the rovers under autonomous control.

[1] Sojourner's speed is based on the tele-operated movement of the rover

[2] Projected average speed for MSL rover, launch date Fall 2011

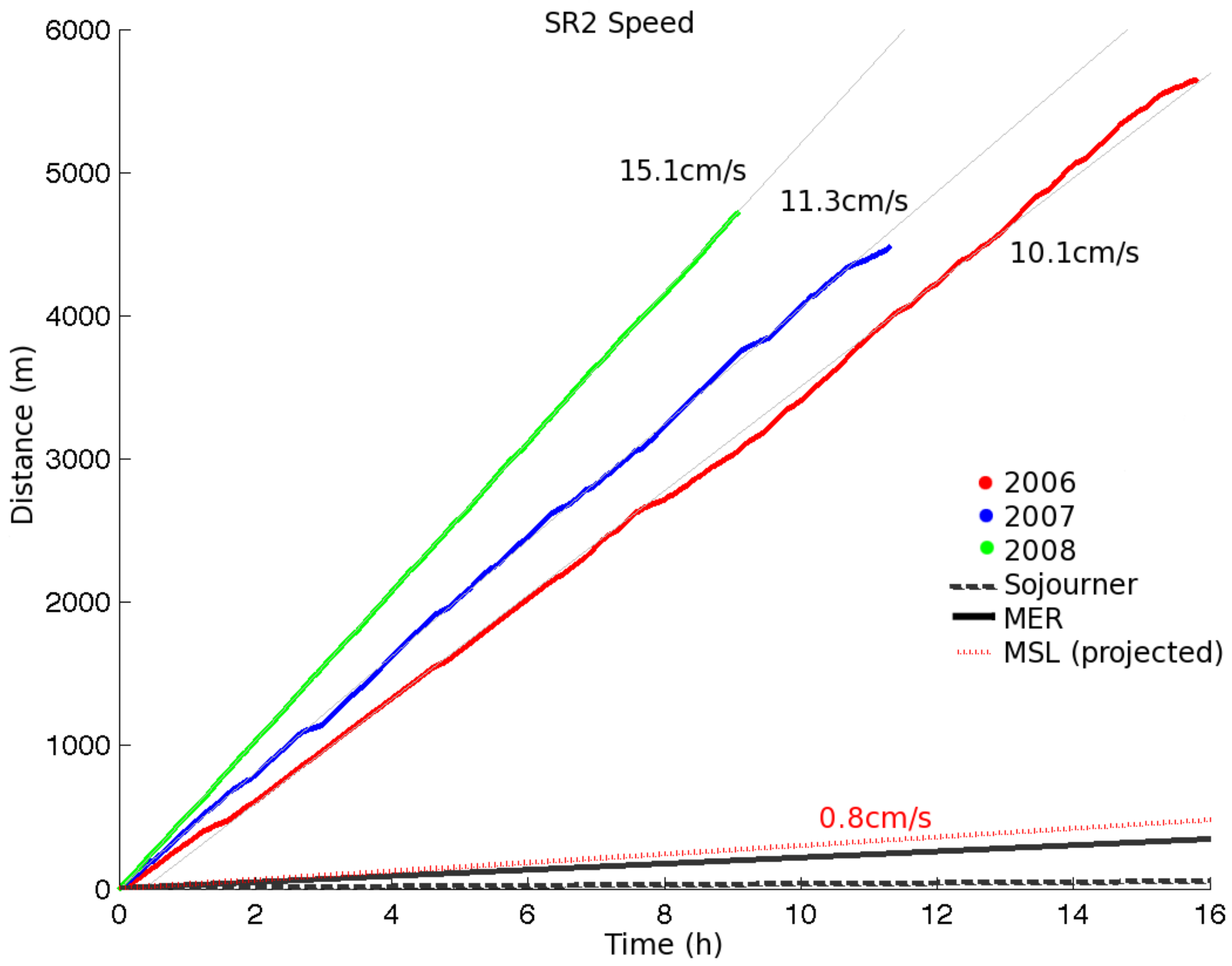


Figure 4.4: SR2 average autonomous speed comparison with Mars rovers during their mission.

In 2008 a new Intel based version of the Apple Mac Mini computer was installed as well as a more powerful pair of drive motors. The rover side and CommandCenter software were also rewritten on a new framework. The underlying functionality and decision making remained mostly the same. These differences are clearly visible in figures 4.5 and 4.6. Trial years 2006 and 2007 follow a similar trend especially when comparing the mobility power in figure 4.6, due to the rover's almost identical setup between trials. The higher power consumed by the electronics in 2008 is likely due to the increased frequency in logging data to the hard disk. Also, images were captured and saved to disk periodically by one of the stereo cameras to be combined in series later to form a video of the rover's traverse. Each time the rover would write data to its hard disk the current drawn by the electronics would spike because of disk seeking. The dips in the 2008 data at 9cm/s and 15cm/s in figure 4.5 are interesting because they are located at the avoidance and cruise speeds respectively. The exact mechanism that caused the fluctuation in the the 2008 power curve is not defined but is related to the navigation software transitioning between the obstacle avoidance and cruise states.

The overall trend in consumed mobility power signifies that more power is used at lower speeds. This is because greater torque from the motors is needed while the rover is executing maneuvers around or over obstacles. The rover's rate of progress is slower due to the greater number of turns made during avoidance maneuvers. This is evidence that significantly more power is used in navigating areas densely populated by obstacles. However, the distribution of SR2's speed shows a minimal amount of distance traveled at low speed. The short distance at low speed indicates the rover was able to navigate around obstacles without deviating vary far from the straight line path between each waypoint.

The power consumed by the electronics and motors during each field trial is displayed in figure 4.8 and the overall averages are given in table 4.2. The power used

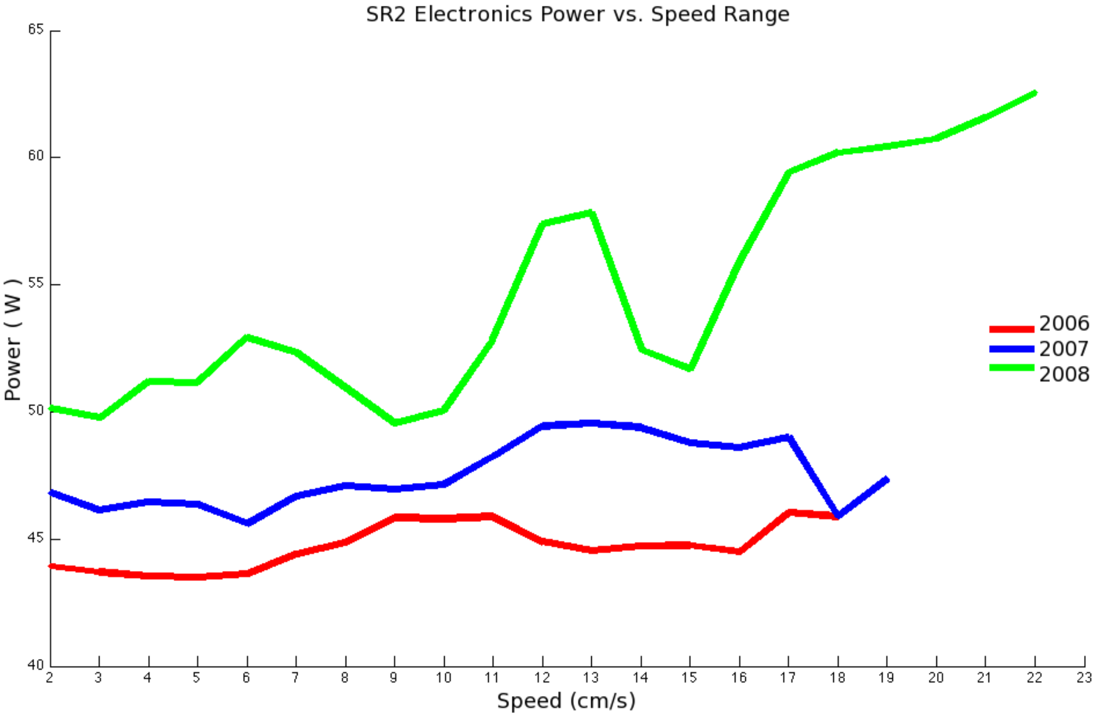


Figure 4.5: SR2 electronics mean power consumed at various speeds.

by the electronics is higher than what is used for mobility throughout all trials except for instances where the wheels stalled and drew excessive amounts of power. Figure 4.7 shows that even at lower speeds, where motor current is at its highest, the average amount of energy used per meter of travel for the electronics is greater and remains greater over the entire speed range. Table 4.3 lists the average amount of energy used per meter for the electronics, motors and combined total. The values for 2008 are lower because the rover was capable of driving at a higher speed due to the more powerful motors. Also, the obstacle navigation system was improved by further tuning the system on RoverSim.

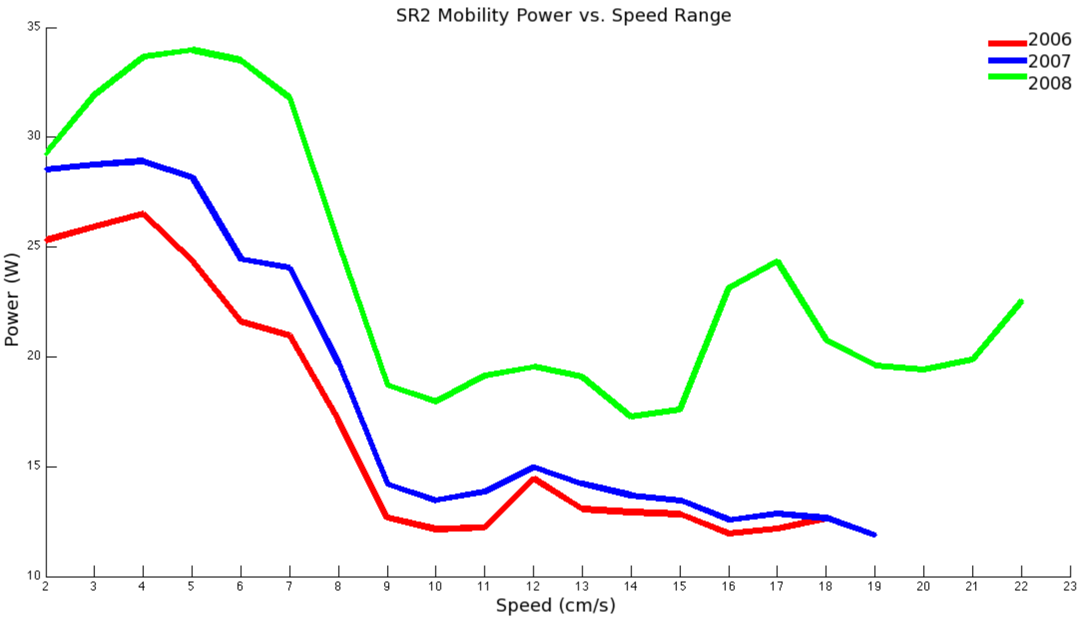


Figure 4.6: SR2 mobility mean power consumed at various speeds.

Test Year	Electronics	Motors	Total
2006	45.09W	15.20W	60.29W
2007	51.76W	19.07W	70.84W
2008	60.59W	23.71W	84.30W

Table 4.2: Average power usage of SR2.

Test Year	Electronics	Motors	Total
2006	150.2mW-h/m	59.3mW-h/m	209.5mW-h/m
2007	159.5mW-h/m	71.3mW-h/m	230.8mW-h/m
2008	128.8mW-h/m	55.8mW-h/m	184.6mW-h/m

Table 4.3: Average energy used per meter by SR2

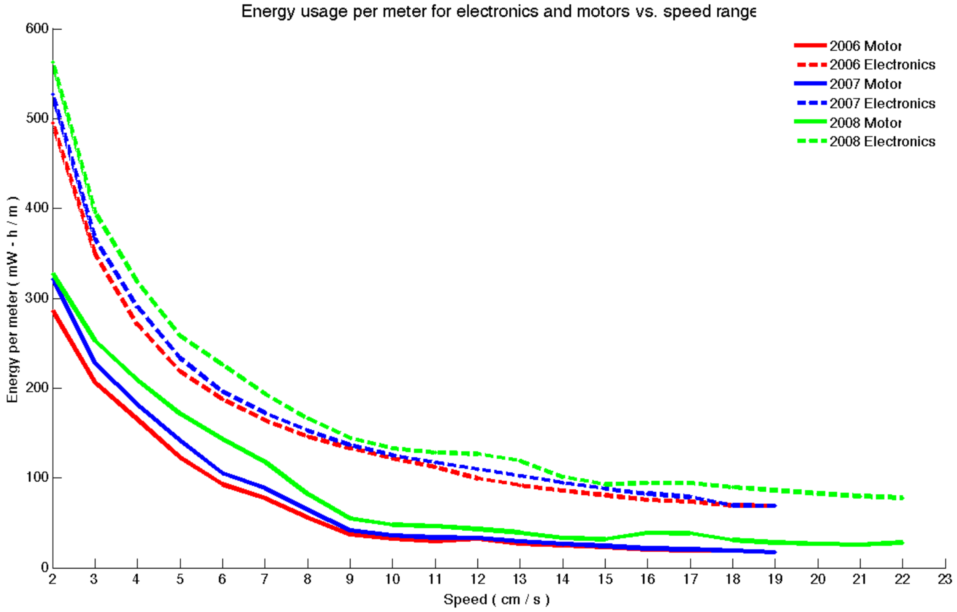


Figure 4.7: Mean energy consumed per meter traveled at various speeds.

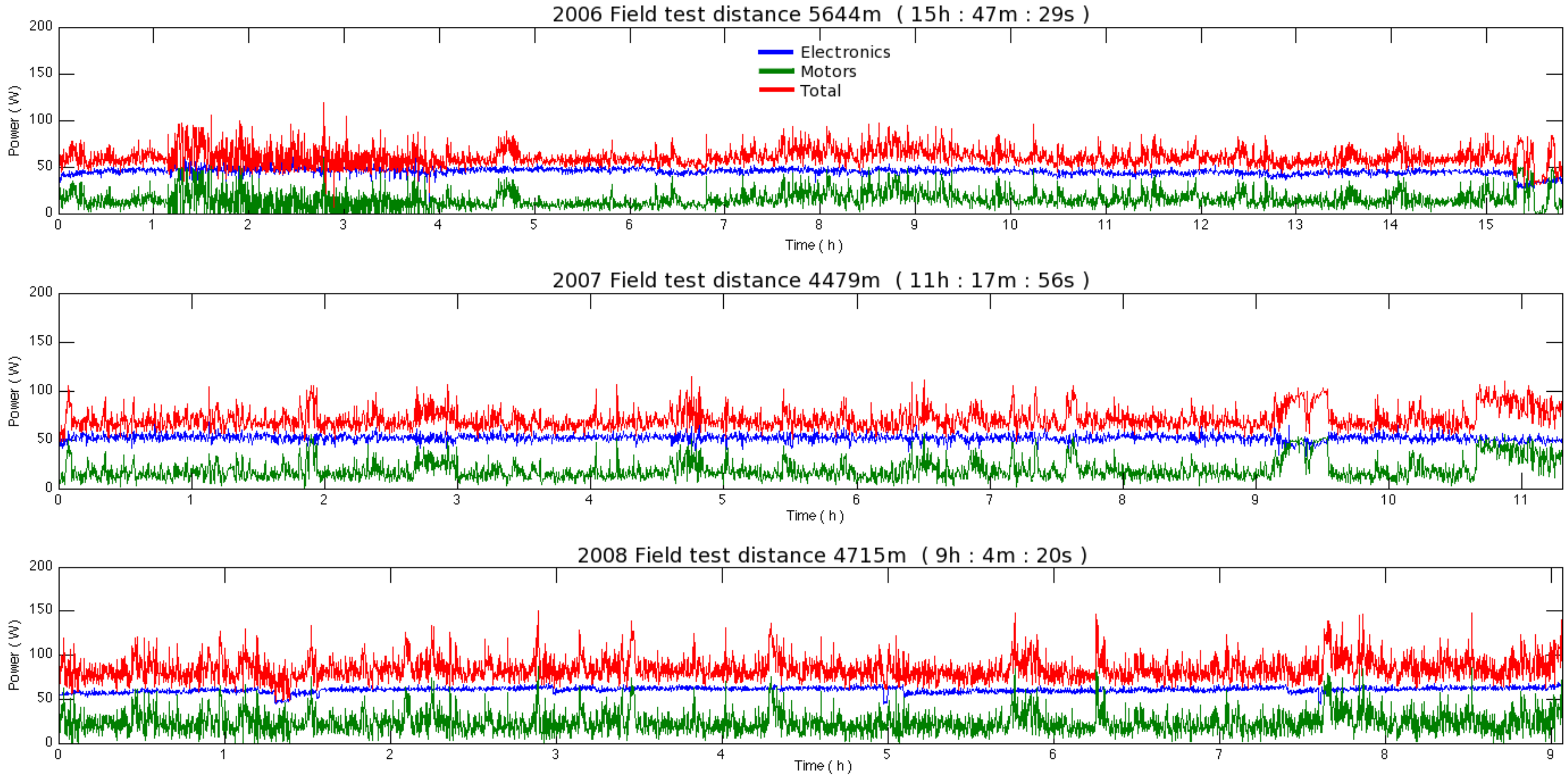


Figure 4.8: SR2 field trial power measurements.

A summary of the major items to take away from the SR2 field trials are:

- A simplified sensor suite is acceptable at finding safe paths on Mars-like terrain.
- Autonomous drives on the order of a kilometer per day were achieved. A day of driving is 3 to 4 hours of operation.
- Mean internal position error is 1.29% of the distance traveled.
- Sandy slopes are major hazards because they are currently undetectable and cause wheel slippage.
- A greater amount of energy is spent on computation than on mobility.

These items helped formulate some of the experiments that were simulated in the RoverSim environment.

Chapter 5

RoverSim Experimentation and Results

Mars rover designs are currently based on the idea that a large sensor range is necessary to traverse natural terrain. Larger sensor ranges can provide more accurate measurements of the environment to the perception system, which detects the location of obstacles relative to the rover. The perception system on SR2 is simpler and has sensors that are shorter range than the perception system of MER. The shorter range allows SR2 to move faster because less time is needed to compute obstacle locations for the navigation system to generate a safe path. SR2's simplified perception system may lead to suboptimal or longer paths to the intended goal.

The Anza Borrego field test experiments show that SR2 is capable of traversing Mars-like terrain at a rate many times greater than the MER rovers. The terrain covered by SR2 contains a higher density of obstacles than the areas traversed on Mars by MER. SR2 accomplished this using a simplified approach to obstacle navigation and a shorter sensor range than MER. The MER rovers have a large sensor range because their sensors take a large number of measurements from different points on the rover to build an accurate model of the environment. The measurement data must be filtered and processed to obtain information about obstacles so a safe path can be planned. The same approach is used on SR2 but the number of measurements taken of the environment is fewer. The major questions generated from the field test are centered on the difference between the sensor suites of SR2 and MER. Even though SR2's sensor range is smaller than MER's, SR2 achieved a higher rate of progress. So, how does a rover's sensor range affect its ability to navigate natural terrain? And, what is the role of sensor placement in determining efficient paths?

The data generated from the SR2 field test is not sufficient to answer the posed questions. Even though SR2 traversed over 14km of Mars-like terrain more data are needed to formulate a conclusion. The field test data are also limited to the perspective of a singe sensor suite and configuration. Unfortunately it would require more resources than what are available to perform further experiments in the field with different rover configurations. Producing a solid conclusion by comparing the available data from SR2 and MER is also difficult because the rover designs and modes of operation are very different. Experiments are conducted in the RoverSim environment to obtain the data needed to determine the performance affects of sensor range on a rover.

RoverSim is a virtual model for experimenting with rover mobility and path planning. It contains simulated terrain and obstacles and a virtual representation of the SR2 rover. Settings can be manipulated that affect the rover's sensor suite and path planner. The model includes tools for configuring the simulation environment and an automator for performing multiple experiments. RoverSim is used to generate tens of thousands of kilometers of data from experiments with varying degrees of sensor range in multiple different obstacle densities.

5.1 RoverSim experiments

The RoverSim experiments are conducted to measure the performance of the rover under different sensor configurations. The simulated rover is driven from a start point to a goal point on terrain containing random distributions of obstacles at various degrees of density. The length of the path the rover takes between the points will be used in comparison with the ideal path length to calculate the efficiency of the rover. This comparison in path length is referred to as the comparative efficiency.

The shortest possible path to the goal without the rover coming into contact with an obstacle is referred to as the *ideal path*. The effects from varying the sensor placement and range of the simulated rover are measured by the comparative efficiency.

The sensor range of the rover is defined in RoverSim by the distance at which the rover can detect obstacles. In the simulated environment the range is equivalent to the radius of a circular region with the rover at the center.

Sensors in RoverSim can have two types of visibility; full and limited. The visibility variable will help determine the effects of a sensor's placement on the rover. The visibility of the rover defines whether or not obstacles in the detection range are shadowed by other obstacles. A shadowed obstacle refers to an obstacle that can not be detected if another obstacle is blocking it's view from the rover's perspective. The two types of visibility relate to two extremes of a sensors placement on the rover. The full visibility type refers to a sensor placed high above the rover looking down on the obstacles. In this situation all sides of the obstacles within range of the sensors are visible to the path planner. The limited visibility type refers to a sensor mounted low on the rover near ground level. A sensor in this situation will not be able to detect obstacles if they are blocked from view even if they are within the sensor range of the rover.

The experimental data from the RoverSim environment are generated in 12 unique instances of the simulation. A mission is defined as a single instance of RoverSim executing on a single processor core of the computer used to conduct the experiments. The terrain for each mission is covered with a unique distribution of obstacles where obstacle size and location are randomly selected. The simulated rover is tested at 31 different levels of obstacle density within a $10,000m^2$ segment of flat terrain. Once the obstacles are placed a start and goal point pair are randomly generated for the rover to plan paths between. A path scenario is a path generated by the rover between the start/goal point pair for a single rover configuration. At each start/goal point pair multiple path scenarios are generated. The scenarios consist of each sensor range and visibility configuration that have been defined for the experiment. A total of 15 different scenarios are used; they include 7 sensor ranges at 2 visibility types each and 1 infinite sensor range path scenario. The infinite sensor range scenario generates the *ideal path* used to compute the comparative efficiency.

5.2 RoverSim results

The experiments in the RoverSim environment required many weeks of computation. The experimental trials produced 722,820 paths total. Of the computed paths 7.7% found no solution in reaching the goal and are removed from the data set. Details of these incomplete paths follow. The total accumulated complete path distance of the rover reached 40,148.9km. This is the path distance computed for all rover configurations over all obstacle densities, not including the *ideal path*s. For reference the path data is equivalent to 1.88 times the circumference of Mars.

The overall comparative efficiency between the *ideal path*s and the rover's limited range paths is 98.3%. This indicates there is generally not a large difference between the paths taken by the rover and the *ideal path* to the goal. The total accumulated Euclidean distance reached 35,239.2km. The Euclidean distance is the straight line distance between the start/goal point pairs but does not account for any obstacles between the points. The total accumulated *ideal path* distance reached 39,456.1km. The accumulated *ideal path* distance is the shortest possible path that the rover could have taken while avoiding obstacles. The difference between the Euclidean distance and *ideal path* distance indicate that obstacles were avoided for the rover to reach the goal.

5.2.1 Path types

The possible types of paths generated in RoverSim are:

- Complete
- Round off
- No solution
- No progress
- Efficiency limit
- 5 minute search timeout

A path that successfully reaches the goal point is a complete path, all other path types are incomplete.

In some cases paths were generated that had lengths shorter than the *ideal path* to the goal. These incomplete paths are found by checking that the comparative efficiency for the path is less than 100%. Meaning that the simulated rover path can not be shorter than the *ideal path* or Euclidean distance to the goal. They are caused by round off errors when comparing the sum of a piecewise path and the *ideal path* to the goal. They are shown in figure 5.1 as *Round off* type and account for 0.55% of the total path count.

All paths in RoverSim are generated between a start and goal waypoint pair using the A* search algorithm created by Hart et al.[21]. The evaluation metric for the search is the Euclidean distance to the goal from the position of the rover. Assuming the goal can be reached by the rover the A* search will find a solution based on the detected obstacles. The search space is defined by all the free space where the rover will not come into contact with any obstacle. All possible configurations of the rover in free space where the rover will not come into contact with an obstacle is known as configuration space. RoverSim simplifies the creation of configuration space by growing all obstacles laterally by the radius of a circle that completely encompasses the rover. The A* search can now assume the rover is a point and will not come into contact with obstacles as long as the point remains outside of an obstacle configuration space.

RoverSim selects a goal point that is not placed on top of an obstacle or within the configuration space of an obstacle before paths are generated. This is done to help ensure there is a possible solution for the A* search algorithm. However, it is possible the goal could be clear of any obstacle but encircled by a group of obstacles rendering the goal unreachable. In this situation the A* search will finish and indicate the goal point is not reachable from the current search location. This scenario is referred to as *No solution* type and is seen rarely, accounting for 0.07% of the total path count.

The *No progress* type is generated when the rover can not make progress toward the goal. An incomplete path is typically referenced by a *No progress* type if it has entered a local minima and the search algorithm can not successfully find a way out. *No progress* types account for 0.28% of the total path count.

An *Efficiency limit* type of incomplete path is generated when the path length becomes excessively large compared to the Euclidean distance to the goal from the start location. The limit at which the path length becomes too long is equal to the Euclidean distance between the start and goal points divided by Υ. Υ is the path efficiency limit set in RoverSim. All missions are simulated with Υ set to 1%, meaning the rover is allowed to drive 100 times farther than the distance between the start/goal points before giving up its search. *Efficiency limit* types account for 2.42% of the total path count.

Finally, if the search for a path ever takes longer than five minutes of computation RoverSim gives up and moves on to the next path simulation. The five minute time limit was selected so the experiment would complete in a timely manner. The *Timeout* type is the largest source of incomplete paths accounting for 4.07% of the total path count. Most of these incomplete paths appear over large obstacle densities where the A* algorithm requires more time at larger sensor ranges to compute a path due to the increased number of vertices to search. This is a potential source of sampling bias because the true type of incomplete path is masked due to the cut-off time which can affect the distribution of the incomplete path data. The distribution of the incomplete path types is displayed in figure 5.1.

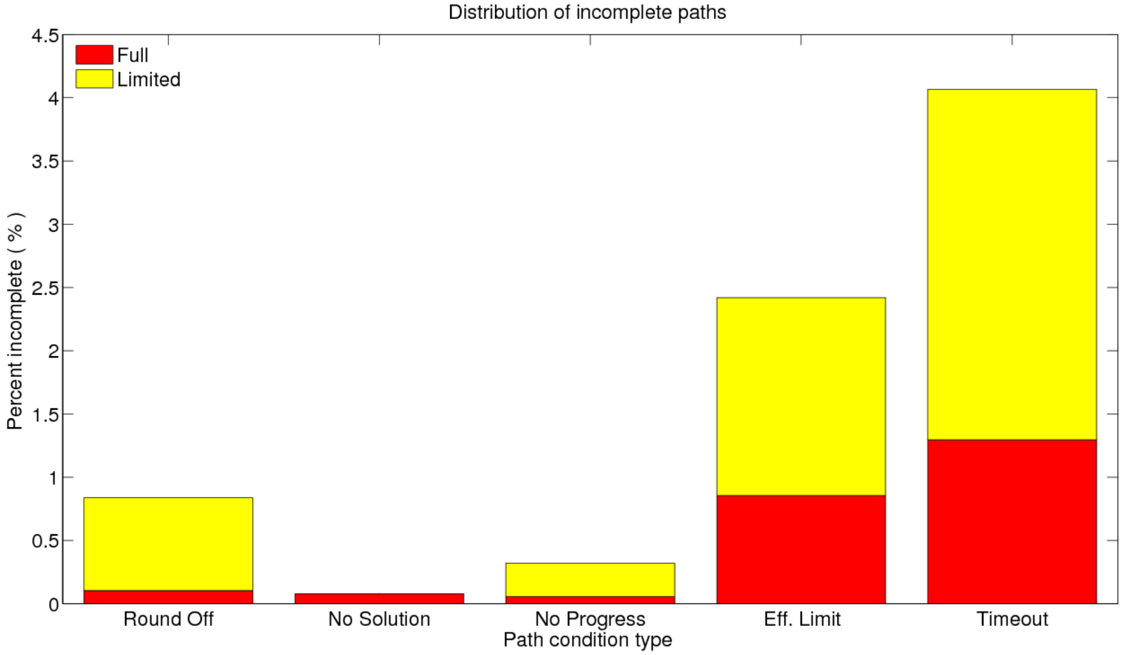


Figure 5.1: The number of incomplete path types separated by path visibility type.

Some of the incomplete paths have been analyzed further as to why they generated such a type. An example of one such path is shown in figure 5.2. Many of the paths in question lead between obstacles and become stuck by continually looping back on itself inside of a local minima. These situations should have generated a *No progress* type of incomplete path because the path does not make progress toward the goal due to the local minima. This indicates that the local minima detection sequence of the simulation is most likely incomplete.

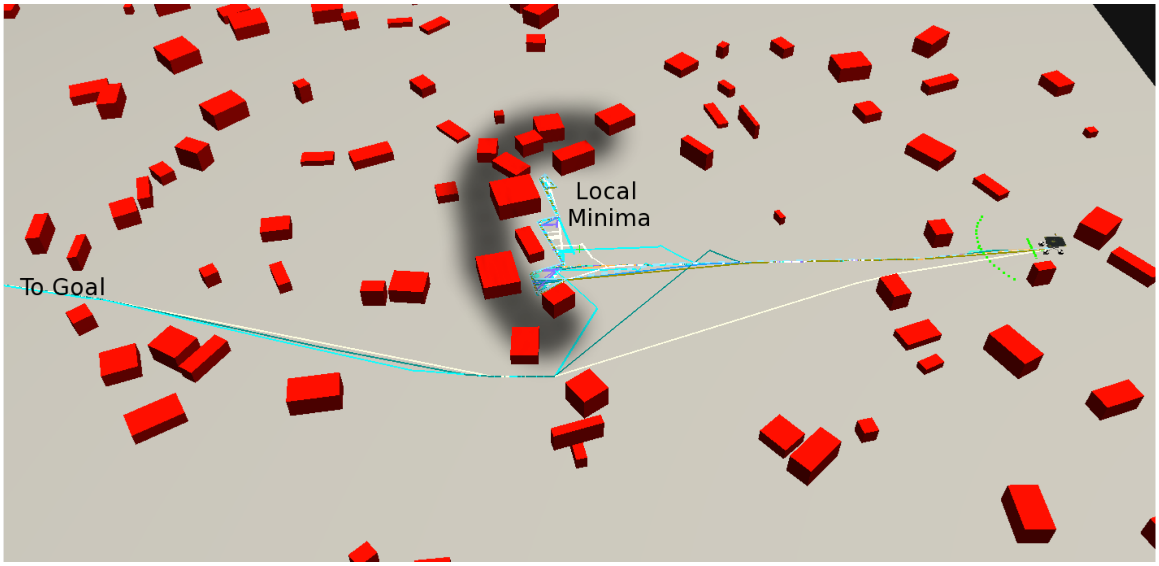


Figure 5.2: Twelve of the fifteen paths simulated during a single trial from mission 10 generated the incorrect type of paths. These paths are referred to as *Efficiency limit* type of incomplete paths which should have been *No progress* type due to the local minima.

The percentage of incomplete paths for each mission are viewed against sensor range in figure 5.3. Dashed and solid lines in the figure represent incomplete paths generated for limited and full visibilities respectively. Sensor range with full visibility does have an effect on incomplete paths. As sensor range increases the number of incomplete paths decrease by 0.2% per meter of sensor range signifying that larger sensor ranges are more reliable results. The rate of incomplete paths decreases slightly as sensor range increases for limited visibility sensors. The rate of decrease is 0.035% per meter of sensor range from shortest to longest sensor range. This indicates that sensors mounted low on a robot, which have limited look-a-head distance, do not produce significantly more reliable results even if their range is increased.

5.2.2 Filtered Data

Incomplete paths have been removed from the data set for the remainder of this analysis. The efficiencies of these paths can not be compared to paths that successfully

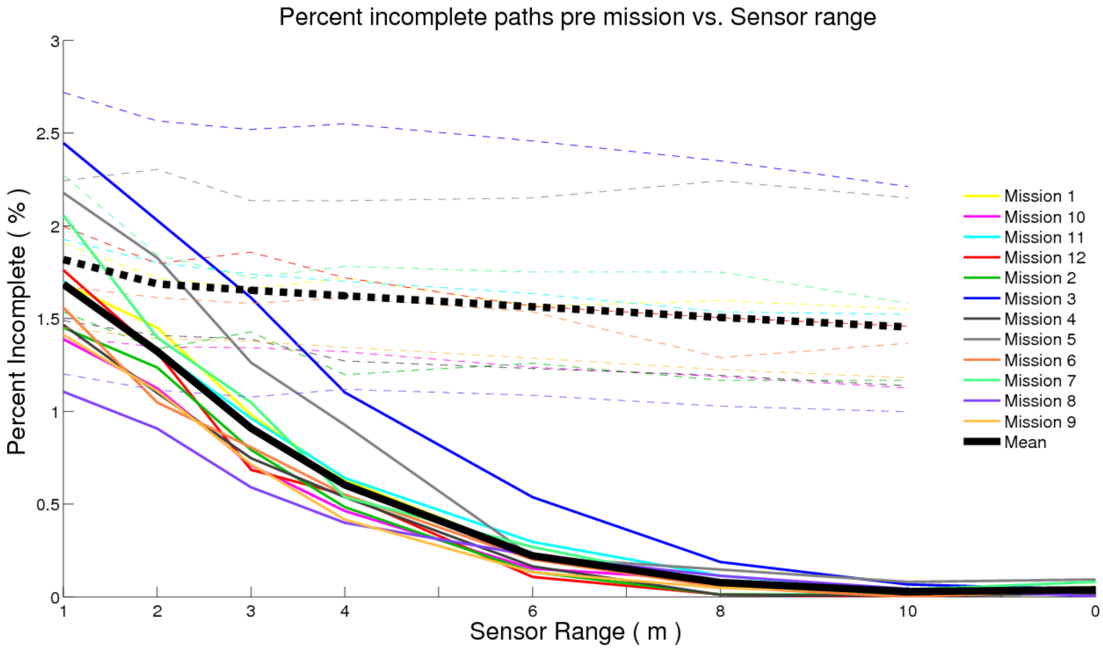


Figure 5.3: Percentage of incomplete paths with respect to sensor range for full and limited visibility type. Dashed lines refer to limited visibility data.

reached the goal because they may cloud the overall picture. A sampling bias may be introduced by removing these data. However, the effects will be negligible due to the small percentage of incomplete paths. The incomplete paths account for 4.6% and 2.3% of the total data set for limited and full visibility sensor types respectively. Two of the missions have been removed as well because they lack a substantial portion of data at high obstacle densities caused by a memory management problem in Rover-Sim. Many of the remaining missions contained incomplete data sets for obstacle density 800 due to the excessive amount of computation time to generate paths at the high obstacle densities. The experiment was stopped before these missions completed. Therefore, obstacle density 800 is removed from all missions. The remaining complete path count is 463,597 paths.

The filtered data are arranged in a matrix. Obstacle densities have been binned into four groups low, medium, high and very high. Each group is defined by eight different density values, i.e. the low density group contains data from densities 50 to 225. The very high density group only contains the last six densities from 650 to 775. The obstacle densities have been binned to increase the readability of figures.

5.2.3 Data Analysis

In order to determine the effects of sensor range and placement on rover path efficiency the data from each mission must be combined. The combined data will provide a more accurate representation of how efficiency is affected. However, the random variability imposed on the creation of each mission to capture a more realistic view of the rover's performance must not differ significantly. Even though each mission is unique all the missions must follow a similar trend. A mission that contains data significantly different from the others may mask the true results when the data are combined. The distribution of the data must be determined in order to select the proper statistical tests to establish that all missions follow the same trend.

Distribution of Data

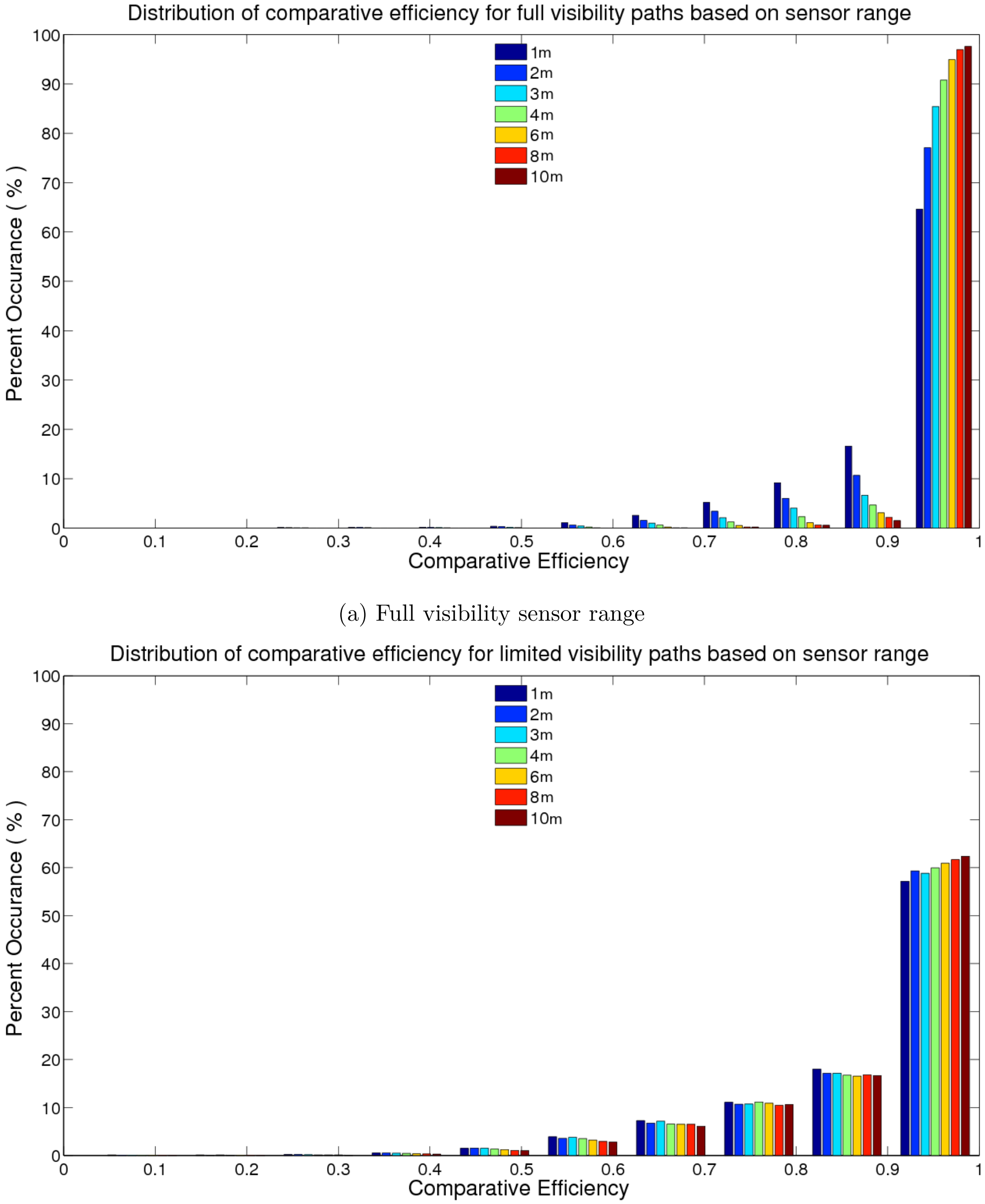
Many statistical methods only produce reliable results if the data are normally distributed. The distribution of the data must be determined before choosing the methods for further statistical analysis. Therefore, histograms of the comparative efficiency are created for both full and limited visibilities with respect to the different sensor ranges and obstacle densities. The histograms can be seen in figures 5.4a to 5.5b. The total number of paths used to calculate the percentage is equal to the number of paths corresponding to the specified visibility type and variable for all missions. The variable is defined as either the sensor range across all obstacle densities or obstacle density group across all sensor ranges. The histograms clearly show that the data have a negative skew. This indicates that neither variable, sensor range or obstacle density, is normally distributed.

To further verify the data are not normally distributed the Shapiro-Wilk test was computed for each sensor range and obstacle density [56]. The null hypothesis for Shapiro-Wilk states that the sample came from a normal probability distribution. The test rejected the null hypothesis for every combination of sensor range and obstacle density at a confidence level of $\alpha = 0.05$. This test verifies that none of the data are normally distributed.

The full visibility sensor range distribution in figure 5.4a shows a difference between longer and shorter ranges. The variance of the distribution increases as the sensor range decreases. This trend follows intuition in that shorter sensor ranges are more likely to produce less efficient paths. 64% of the paths achieved performance levels higher than 90% comparative efficiency.

The distributions for the sensor ranges with limited visibility appear very similar to each other in figure 5.4b. There is still a difference between long and short range sensor paths indicating better performance with longer sensor ranges. However, only a 5% increase in efficiency is gained by increasing sensor range from 1m to 10m. This indicates there is little benefit for increasing sensor range when visibility is limited. Over 57% of the paths achieved performance levels higher than 90% comparative efficiency.

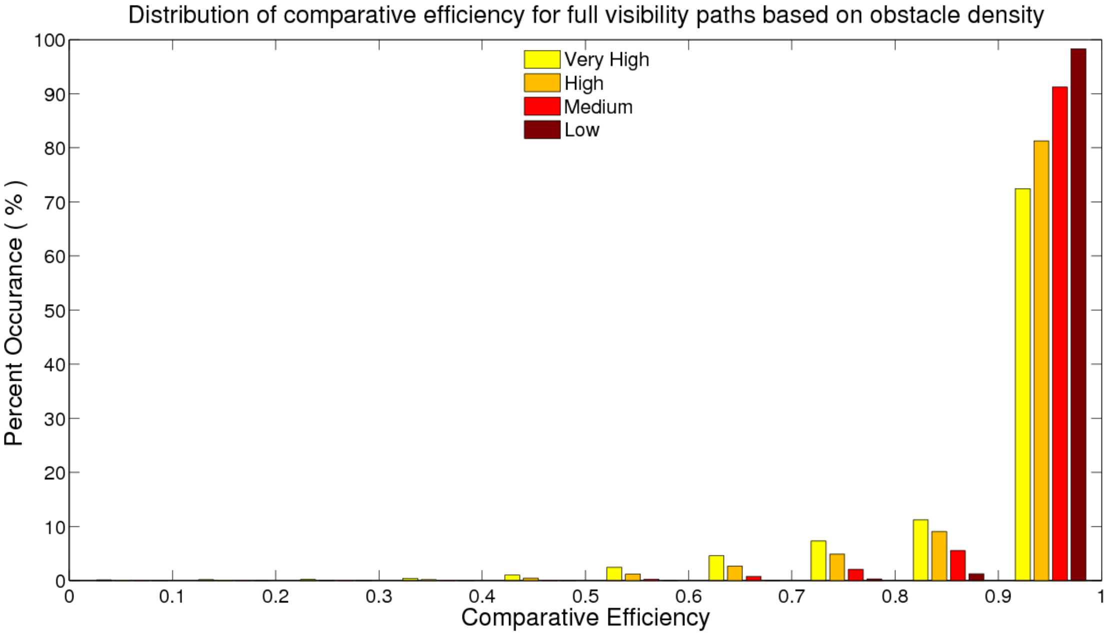
The obstacle density histograms in figures 5.5a and 5.5b both follow the same pattern. As the obstacle density increases the variance in the distribution becomes larger because the rover has a more difficult time navigating through dense obstacle fields. In figure 5.5b for limited visibility paths that completed above 90% comparative efficiency there is a 73% drop in the percent of paths from low to very high densities. This indicates that a change in obstacle density has a large affect on limited visibility path efficiency.



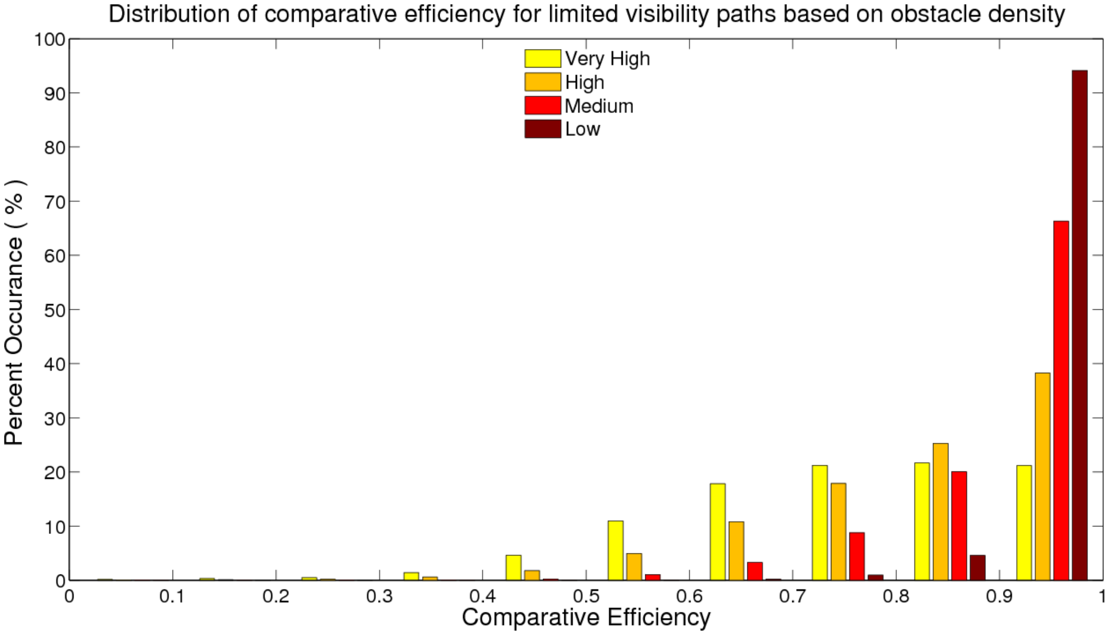
(a) Full visibility sensor range

(b) Limited visibility sensor range

Figure 5.4: The histogram shows the distribution of comparative efficiency for each sensor range. The comparative efficiency is measured by the ratio of the *ideal path* length to the rovers path length.



(a) Full visibility obstacle density



(b) Limited visibility obstacle density

Figure 5.5: Distribution of comparative efficiency based on each of the binned obstacle densities. The comparative efficiency is measured by the ratio of the *ideal path* length to the rovers path length.

Mission Congruence

The RoverSim experiments included multiple missions to obtain more realistic variation in the environment and to reduce unforeseen effects from noise variables. The only planned variation between the missions is due to the random placement and size of obstacles. Even though the variability in the random factor is based on a uniform model, it is possible that a mission will contain patterns within its obstacle layout that are significantly more difficult or easy to navigate than another mission. It is important to verify that all missions are congruent or agree upon the same trend for further analysis to show the true effects on path efficiency.

Friedman's analysis of variance statistical test does not rely on the assumption that the data is distributed normally. The null hypothesis for this test states that all the missions are based on a similar probability distribution. The alternate hypothesis states that at least one or more of the missions differ from the rest. The null hypothesis and alternate hypothesis are defined as:

$$\begin{aligned} &H_0 : \text{All missions are equal} \\ &H_a : \text{One or more missions differ} \end{aligned} \tag{5.1}$$

Table 5.1 contains the statistics generated from the Friedman test. Every sensor range and visibility combination is tested at $\alpha = 0.05$. Some of the test statistics (F_r) are very close to the critical value of 16.919 taken from the χ^2 distribution with 9 degrees of freedom. However, there is not enough statistical information to indicate that any of the missions are significantly different from one another. It is assumed that all missions have similar distributions and their data can be combined without negatively affecting further analysis.

Visibility	Sensor Range	F_r	P-value
Full	1	F= 7.695	p=0.565180
Full	2	F= 13.839	p=0.128182
Full	3	F= 9.374	p=0.403510
Full	4	F= 11.222	p=0.260828
Full	6	F= 16.348	p=0.059967
Full	8	F= 15.919	p=0.068598
Full	10	F= 16.627	p=0.054895
Limited	1	F= 9.658	p=0.378863
Limited	2	F= 16.790	p=0.052115
Limited	3	F= 9.646	p=0.379893
Limited	4	F= 7.337	p=0.602038
Limited	6	F= 5.182	p=0.818159
Limited	8	F= 10.692	p=0.297390
Limited	10	F= 10.495	p=0.311895

Table 5.1: Statistic values from Friedman test of all sensor ranges. Null hypothesis is rejected at $F > 16.919$.

Merging Data

The data are merged into a single matrix. Each cell of the matrix represents the comparative efficiency from the combined data of all missions for the corresponding obstacle density and sensor range with specific visibility type. Due to the large amount of skew in the distribution of the data the efficiency value for each cell was calculated using the median of the data. The mean and median are measures of central tendency of a data set, but the median is less susceptible to drift caused by outliers. The median is also considered to be a more accurate representation of the center of data in cases where the data are heavily skewed.

Variable Interactions

The merged data are graphically represented with respect to obstacle density in figure 5.6. Limited visibility data are indicated by dashed lines in the figure. As obstacle density increases it has a negative effect on comparative efficiency for both types of visibility. The full visibility path data, referred to by solid lines, indicate that reduced sensor ranges also cause efficiency to decrease. There appears to be an interaction between sensor range and obstacle density on comparative efficiency for full visibility paths. The interaction is visible from the growing separation between the solid lines for each sensor range moving toward higher obstacle densities.

Rover path efficiency from limited visibility sensor type does not appear to be affected by sensor range in figure 5.6. It is unclear if there is a significant difference in comparative efficiency between long and short range limited visibility sensors due to the small separation between the dashed lines. The near parallelism of the dashed lines suggest there is no interaction between sensor range and obstacle density. Rover efficiency appears to only be negatively affected by increases in obstacle density for limited visibility sensors.

The effects of sensor range and visibility type at low obstacle densities do not significantly affect rover efficiency. A ceiling effect can be seen at low obstacle densities (< 250) for all sensor ranges because most paths achieve efficiencies very close to the best possible performance. The ceiling effect can make it difficult to determine how the efficiency of the rover is affected. However, the path efficiency in low obstacle densities remains above 98% for all sensor ranges and both visibility types. The data from low obstacle densities is also represented in fig 5.7 with respect to sensor range. At a 98% level of comparative efficiency the rover would traverse 2m more than the *ideal path* to a goal 100m away.

The effects of sensor range and obstacle density on path efficiency for full visibility sensor type are further shown in figure 5.7. Again, as sensor range decreases so does the comparative efficiency. This is visible as each solid line drops off near the left edge of the figure. The vertical separation between each obstacle density group in the figure shows the negative influence on efficiency as density increases for both visibility types.

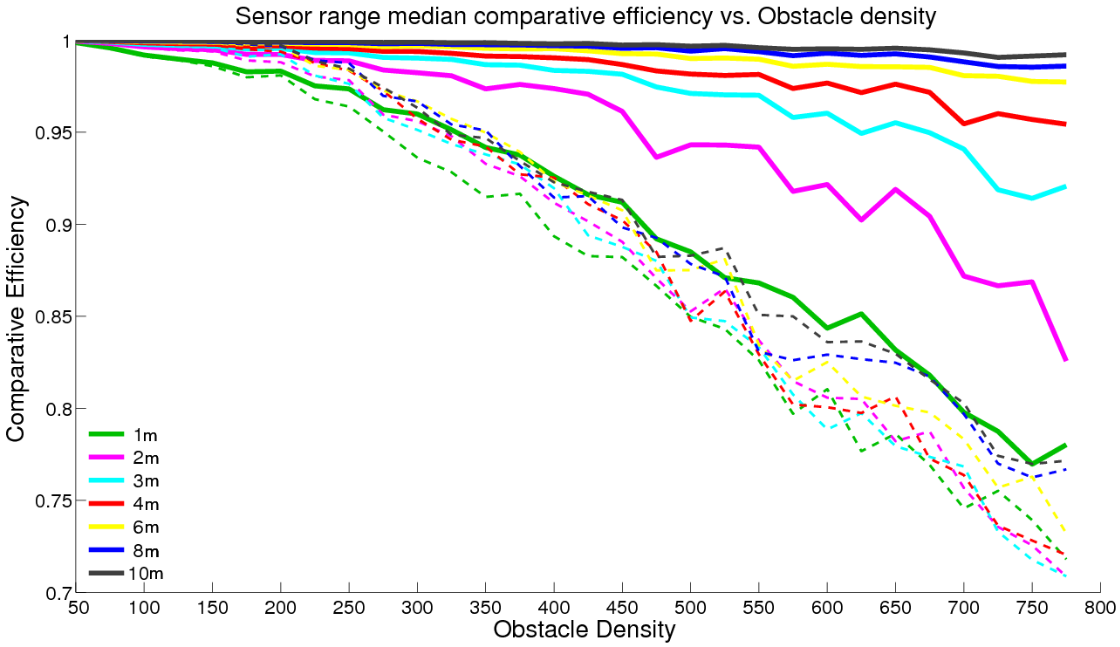


Figure 5.6: Comparative efficiency is affected by obstacle density for each sensor visibility type. Sensor range does not greatly affect efficiency for limited visibility types. The path data for limited visibility type are shown with dashed lines.

The limited visibility data represented in the figures indicate there is very little, if any, increase in comparative efficiency to increasing range of limited visibility sensors. The near horizontal trend in each limited visibility dashed line in figure 5.7 signifies that sensor range has a very small affect on efficiency. A similar effect is visible in figure 5.6 in regards to the closely spaced dashed lines. The resampling method of approximate randomization tests if two samples of range data come from the same distribution [12]. The data from each combination of of limited visibility sensor range were tested. All combinations proved to be statistically different at a confidence of $\alpha = 0.05$ except for the comparison between sensor ranges 2m and 3m which generated a p-value of 0.358. There is not enough data to support there is a difference in the comparative efficiency of paths generated with limited visibility sensors at range 2m or 3m.

Even though the approximate randomization tests verify that longer sensor ranges

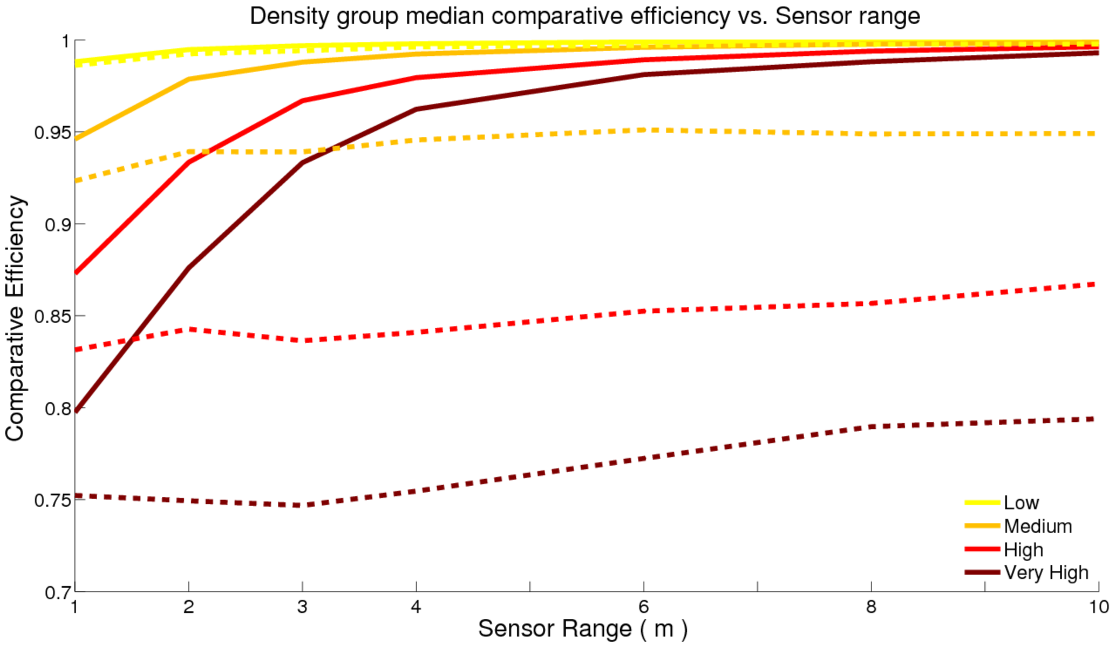


Figure 5.7: Comparative efficiency is not significantly affected by sensor range or visibility type in low obstacle densities. The sensor range of limited visibility type sensors does not greatly affect efficiency in any obstacle density. The path data for limited visibility type are shown with dashed lines.

statistically produce more efficient paths the increase is very small. There is only a difference of 2.5% in the mean efficiency between the shortest (1m) and longest (10m) sensor ranges with limited visibility.

5.2.4 Summary

The comparison of the results from the SR2 field test to the Mars Exploration Rovers generated questions concerning the effects of various types of sensors used for navigation. Specifically, how does the range and placement of a sensor affect the performance of the rover? The SR2 field test was limited in many ways making it difficult to derive a solid conclusion as to how performance is affected. To overcome this, experiments were conducted in the RoverSim virtual environment.

Over 40 million meters of path data are generated under multiple sensor configurations of the simulated rover. Sensor range and placement are the two main variables concerning the sensors configuration. The sensor range pertains to the distance the rover can detect obstacles in RoverSim. Placement is defined by the sensor visibility type in RoverSim and has two states; full and limited. The visibility determines whether or not obstacles are visible if they are blocked from view by another obstacle. The effect of varying sensor range and visibility type are measured by the comparative efficiency of the rover's path. The comparative efficiency is calculated by comparing the shortest possible path between a start and goal point to the path taken by the simulated rover.

The simulation results revealed there is a 75% probability that a limited visibility sensor configuration can achieve a comparative efficiency greater than 80% through even the most densely populated obstacle fields. In low obstacle densities the effects of visibility type and sensor range are negligible. The minimum comparative efficiency for all sensor configurations in low obstacle densities remain greater than 98%.

The comparative efficiency of limited visibility type sensors is not significantly affected by varying sensor range. Due to the non-normal distribution of the simulation data, a statistical resampling method was applied to determine if increasing sensor range with limited visibility affects efficiency. The results show a positive trend. Path efficiency does statistically increase with sensor range. The comparative efficiency of the simulated rover increases by approximately 0.25% for every additional meter of sensor range from 1m to 10m. However, for a practical application the benefit of using a longer sensor range with limited visibility is unlikely worth the additional resources needed to support such a sensor.

The results based on full visibility type of sensors clearly indicate that efficiency can be improved by increasing the sensor range. However, there is an interaction between obstacle density and sensor range. This means that the comparative efficiency is influenced by the sensor range as well as the obstacle density the rover is traversing through. The interaction makes it difficult to determine the exact influence that increased sensor range has on efficiency. On average a 1% increase in comparative efficiency is gained for every meter of sensor range from 1m to 10m.

The analysis of the data suggest that greater sensor visibility can lead to increased efficiency in higher obstacle densities. The sensor visibility is related to the physical placement of the sensors on the rover. The full visibility sensor type simulated in RoverSim refers to a sensor with a completely unobstructed view and capable of detecting all sides of the obstacles within range. A practical discussion follows to determine the physical configuration of a rover that can achieve a high degree of visibility similar to the full visibility in the simulation.

5.3 Practical Analysis

The results from the RoverSim experiment must be related to the practical application of sensors on a rover. A variety of sensor ranges have been simulated; therefore it is important to determine the physical requirements to achieve them. This section outlines some ideas for making a connection between the simulation work and a physical rover. An equation is derived for calculating the effective sensor range of a rover based on various physical parameters. SR2, MER, and the next Martian rover Mars Science Lab (MSL) are used in comparison examples. Once the effective sensor range is computed and the obstacle density is defined the rover's performance can be estimated. Images of Martian terrain are used to determine the obstacle densities encountered on Mars. The data from equivalent obstacle densities created in RoverSim are used for comparing the path efficiency of multiple configurations of rovers.

5.3.1 Effective Sensor Range

The analysis of the simulation experiments indicate the visibility of a sensor can affect the path efficiency of the rover. As the obstacle density increases the sensor visibility or the ability to see obstacles behind one another increases the effectiveness of the path planner at generating efficient paths.

Sensor Visibility

The sensor visibility of a physical rover is related to the mounting location of the sensor on the rover. The effective sensor range diagram in figure 5.8 represents a typical obstacle avoidance scene for a rover. More importantly it shows the relationship between the height at which the sensor is mounted, typically on a mast, and the sensor range. The mounting location of a sensor can affect its visibility for a given obstacle height. There are two extremes of sensor visibility both of which were experimented in the RoverSim trials.

The first visibility type is represented as a sensor mounted on a mast infinitely tall and looking down on the obstacles within range. This gives the rover a complete view of each obstacle and the ability to see obstacles behind one another. During a real mission this view is somewhat equivalent to satellite imagery thought at lower resolution. Sensors of this type are referred to as having high visibility.

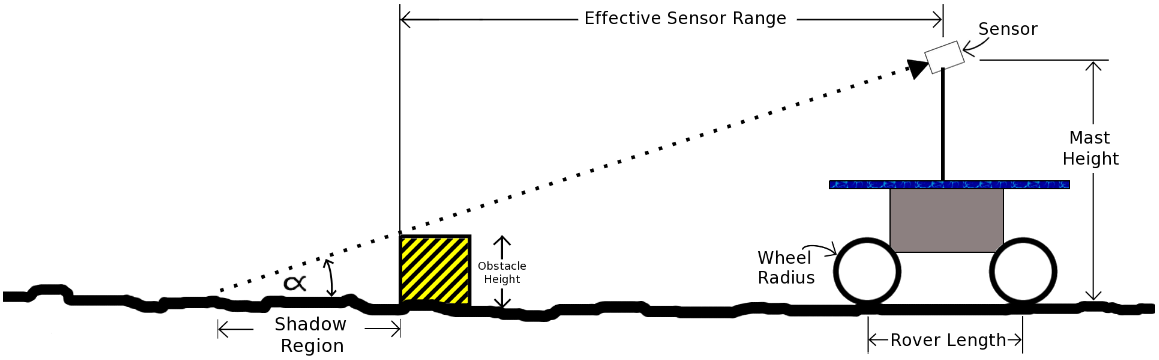


Figure 5.8: A diagram of a typical obstacle avoidance problem. The dimensions of the rover and its surroundings define the effective sensor range which is explained in section 5.3.1.

The second sensor visibility type has a mast height of almost zero. The sensor is mounted just above the ground and looks outward horizontally. Only the front side of obstacles can be seen in this case, other obstacles behind them are blocked from view making them impossible to plan for. Sensors of this type are referred to as having very low visibility.

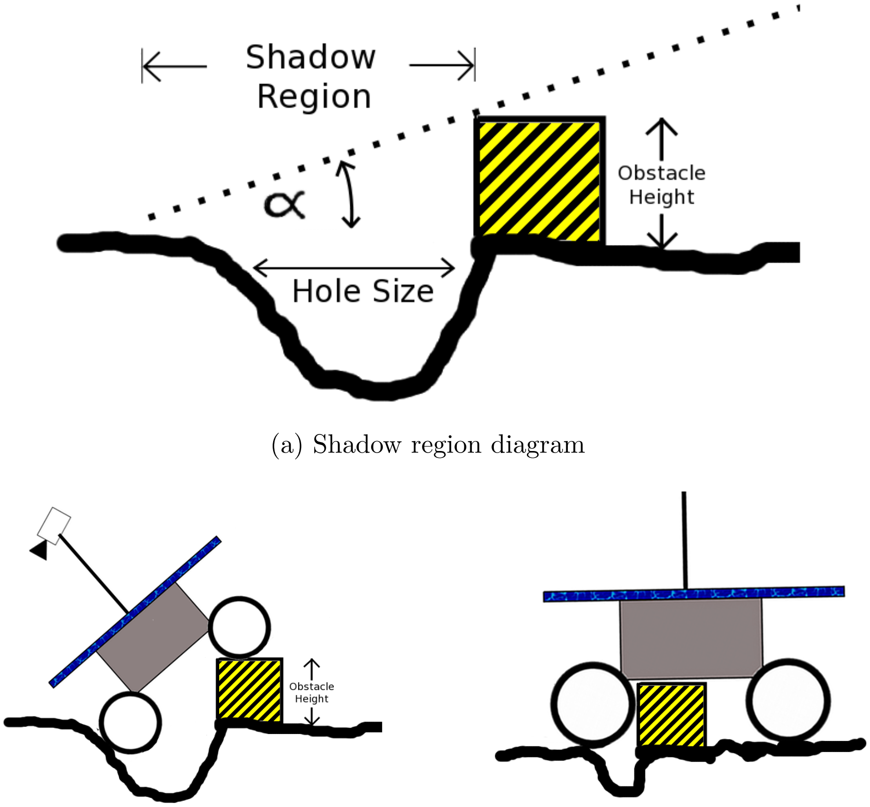
The full and limited visibility types simulated in RoverSim refer to high and low degrees of visibility respectively.

The effective sensor range is defined for this research as the maximum distance that obstacles can be detected while still achieving a high degree of visibility that is adequate for planning. The scenario depicted in figure 5.8 relates the physical dimensions of the rover, the placement of the sensor and the height of the obstacle to determine the effective range of the sensor.

The effective sensor range can be calculated from a few known dimensions of the rover and the height of an obstacle. The obstacle height for this definition of effective sensor range is equal to the minimum obstacle height. The smallest height of an object that is considered an obstacle for the rover is the minimum obstacle height. It is assumed the rover can safely traverse objects shorter than the minimum obstacle height.

The Shadow Region

The shadow region depicted in figure 5.8 to the left of the obstacle is the distance behind an obstacle the rover must sense for visibility to become fully effective. Figure 5.9 represents a worst case obstacle scenario that defines the limit of the shadow region. In the scenario a positive obstacle is immediately followed by a negative obstacle. Positive obstacles extend upward from the ground plane while negative obstacle descend below the ground plane. If the dimension of the shadow region is too large the hole will not be detected. The rover may become stuck if the distance between the top of the positive obstacle to the bottom of the negative obstacle is larger than the rover can surmount as suggested in figure 5.9b. The rover must be able to sense terrain behind an obstacle to a distance of one wheel radius for it to safely surmount obstacles below the minimum obstacle height.



(a) Shadow region diagram

(b) Rover crashes into hole

(c) Rover clears obstacle

Figure 5.9: The shadow region must be small enough for the rover to reliably detect hazards. The maximum size of the shadow region must be one wheel radius to ensure no collisions with obstacles.

Effective Range Equation

The dimensions needed to calculate the effective sensor range are labeled in figure 5.8 and their corresponding symbolic definitions are:

$$\begin{aligned} \alpha &\Rightarrow \text{View Angle} \\ h_m &\Rightarrow \text{Mast Height} \\ l_r &\Rightarrow \text{Rover Length} \\ l_w &\Rightarrow \text{Wheel Radius} \\ h_o &\Rightarrow \text{Obstacle Height} \\ S_d &\Rightarrow \text{Shadow Region} \\ R_s &\Rightarrow \text{Effective Sensor Range} \end{aligned} \tag{5.2}$$

The view angle of the sensor is defined by the obstacle height and the shadow distance behind the obstacle. Simple trigonometry leads to α calculated from

$$tan(\alpha) = \frac{h_o}{S_d} \tag{5.3}$$

The angle α relates the height above the rover a sensor must be placed to achieve a given sensor range from

$$tan(\alpha) = \frac{h_m}{R_s + S_d} \tag{5.4}$$

Substituting α in equation 5.3 gives

$$\frac{h_o}{S_d} = \frac{h_m}{R_s + S_d} \tag{5.5}$$

The ratio of mast height to rover length is labeled as γ and serves to define the physical configuration of the rover as

$$\gamma = \frac{h_m}{l_r} \tag{5.6}$$

Equation 5.6 is rearranged and the mast height is substituted to obtain

$$\frac{h_o}{S_d} = \frac{\gamma \cdot l_r}{R_s + S_d} \tag{5.7}$$

Rearranging equation 5.7 to give effective sensor range produces

$$R_s = S_d \left(\frac{\gamma \cdot l_r}{h_o} - 1 \right) \tag{5.8}$$

Equation 5.8 relates the physical shape of the rover to the effective sensing range. Comparisons can now be made between various rover designs through the combination of the RoverSim data on path efficiency, the effective sensor range equation in 5.8, and measurements taken from physical rovers.

5.3.2 Computed Rover Sensor Range

Equation 5.8 will determine the effective sensor range of current rover designs for comparison. The equation relates the geometry of the rover (l_r, l_w), the location of the sensor (γ) and the potential obstacle height (h_o) to the sensor range (R_s). For the following comparisons the shadow region (S_d) is equal to the rover's wheel radius (l_w). The effective sensor range diagram indicates that the measurement range of the sensor, referenced by the dotted line in figure 5.8, must be larger than the effective sensor range for obstacles to be detected with a high degree of visibility. Equation 5.8 assumes the sensor transducer attains enough resolution over the effective range to detect obstacles. The effective sensor range can now be calculated by knowing a few measurements of the rover's physical shape.

The parameters of SR2, MER and MSL that affect sensor range are listed in table 5.2. The minimum obstacle height in the table refers to the limit at which an object is considered an obstacle that the rover must avoid. For comparison purposes the mast camera height of SR2 is included even though it is used only to simulate the scientific operations during the field experiments. The images from the mast camera could be used for navigation in much the same way they were used during the MER mission. The images are downloaded and paths are planed by human operators who select waypoints for the rover to traverse through the terrain.

The effective sensor range displayed in figure 5.10 is computed from equation 5.8 using the parameters for each rover in table 5.2. The difference between the two sensor positions on each rover is shown by the dashed and solid lines in figure 5.10. The solid lines represent the effective sensor range at various obstacle heights if the mast camera of each rover is used for obstacle detection. From the figure it is clear

Rover	Sensor Height	Wheel Base	γ	Minimum Obst. Height
SR2 panel laser	0.54	0.64	0.84	0.20
SR2 mast camera	1.0	0.64	1.56	0.20
MER nav. camera	0.53	1.4	0.38	0.20
MER mast camera	1.52	1.4	1.07	0.20
MSL nav. camera	1.0	1.9	0.53	0.66
MSL mast camera	2.13	1.9	1.12	0.66

Table 5.2: Parameters of the rovers used for comparison in figure 5.10. All dimensions are in meters.

that the height of the sensor has the largest influence on effective sensor range. Even though MER and SR2 use different technologies for obstacle detection the effective sensor ranges are approximately the same for sensors mounted low on the rover at the same minimum obstacle height. This is indicated by the blue and green dashed lines in figure 5.10. The MSL rover appears it will have a similar sensing range to its predecessor relative to the minimum obstacle size for each rover. The effective sensor range of all three rovers is less than 0.3m for the low mounted navigation sensors at their respective minimum obstacle heights.

A rover must be able to see over nearby obstacles to reliably take advantage of increased visibility. The sensor must be placed well above the obstacles it encounters typically requiring the sensor to be mounted on a mast. However, the difficulty in making the rover fit into an acceptable stowed configuration for flight will place an upper bound on the mast length potentially limiting visibility. Very tall mast heights can decrease rover stability and induce unwanted motion or vibration on the sensor.

The significance of figure 5.10 is that a rover can be designed to carry a sensor suite capable of a high degree of visibility. However, the effective sensor ranges of these configurations are very limited. The effective sensor range for each rover at the minimum obstacle height and highest mounted sensors is less than 1m of range. Based on the data from the RoverSim experiments the efficiency of each rover is not increased by having this high degree of visibility due to its short sensor range.

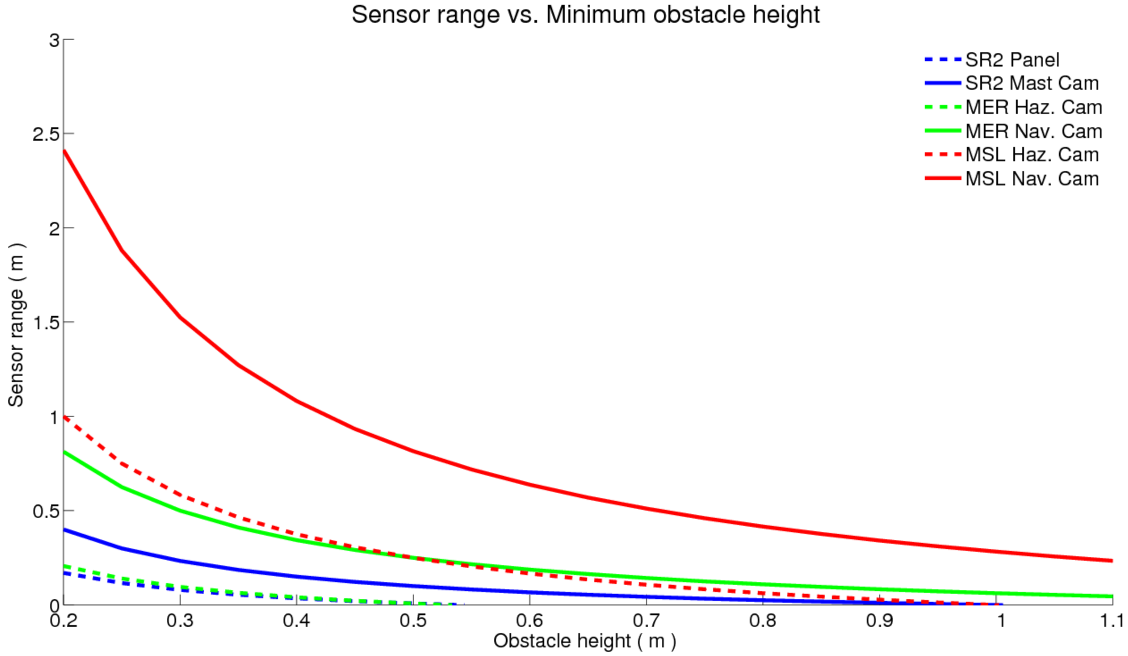


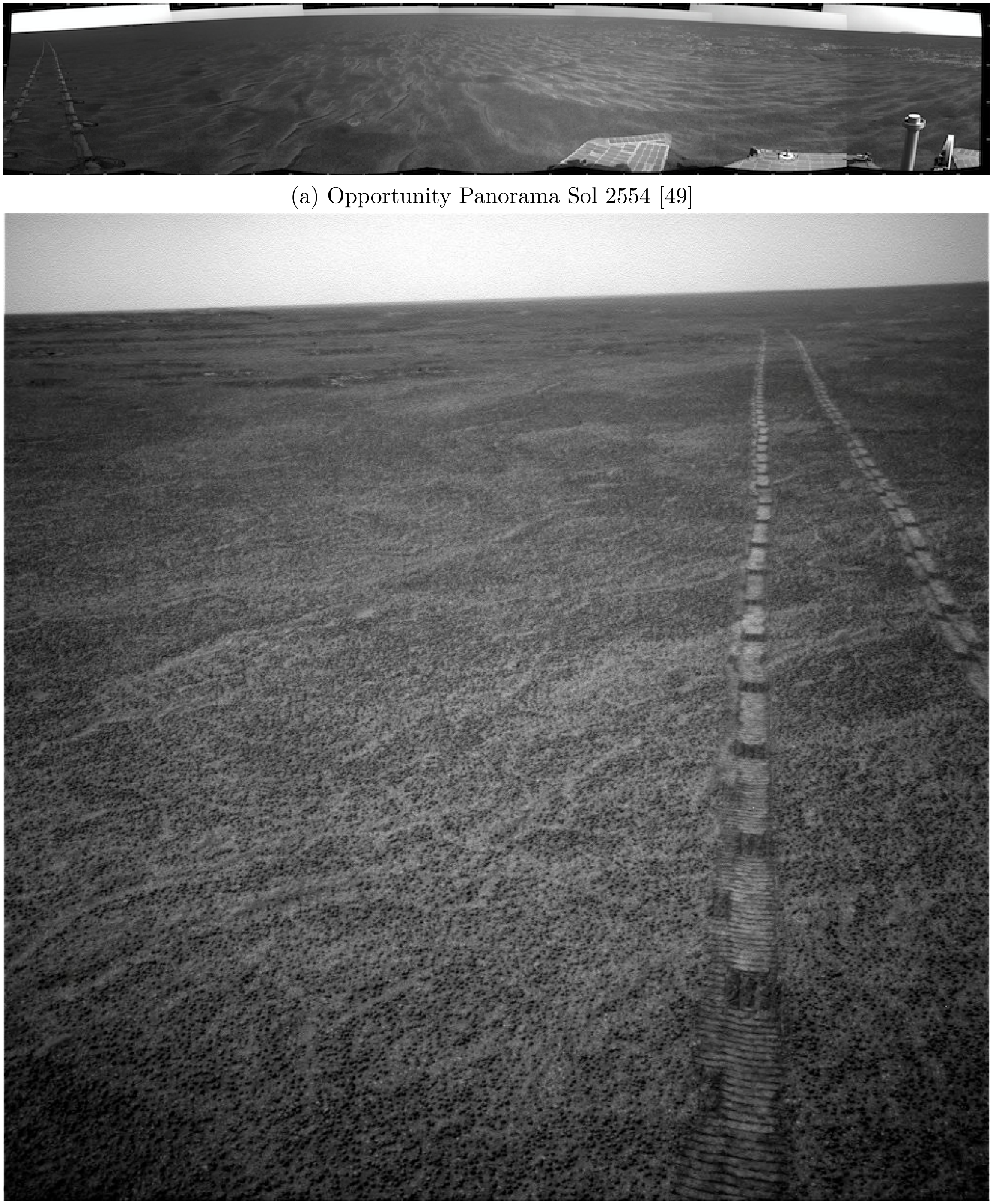
Figure 5.10: Effects of obstacle height on sensor range for various rovers using Eq. 5.8. Vertical dotted lines indicate the minimum obstacle height for the corresponding rover.

5.3.3 Real Terrain Comparison

Very low densities of obstacles were encountered during the MER mission. Both MER rovers explored areas almost free of obstacles that could be detected by the hazard avoidance sensors and on board software. Large scale obstacles, larger than the rover, are visible from satellite imagery and could be avoided by the mission controllers from Earth. Obstacles such as the sand dune formations in which both MER rovers became stuck for a period of time are assumed to be a special case obstacles. These type of hazards are not included because they are not remotely detectable by the sensors onboard the rover, presenting a completely different problem for obstacle navigation. The images 5.11a,5.11b, and 5.3.3 are taken from each MER rover are included for comparison. The width between the rover tracks in the images is approximately 1m for scale.

Examples of the obstacle densities created during the RoverSim trials are displayed in figure 5.13. It is clear that even at a density of 50, the lowest density, the likelihood of encountering at least one obstacle on the way to the goal is greater than actual Martian terrain seen in the images taken by MER. Even though the current implementation of RoverSim does not account for large variations in the terrain, macro scale features are created by the grouping of multiple obstacles when simulating large densities. Figure 5.13b is an image of the RoverSim environment at obstacle density 700.

The RoverSim experiments conducted at low obstacle densities indicate that sensor range has a very limited effect on rover performance for any degree of visibility. Based on the results of the RoverSim analysis, SR2 with its shorter sensor range and insignificant measure of visibility is capable of performing nearly as well in the locations being explored by the MER rovers because of the low obstacle density.



(a) Opportunity Panorama Sol 2554 [49]

(b) Opportunity NavCam Sol 2674 [49]

Figure 5.11: Obstacles are almost non-existent in Meridiani Planum where the Opportunity MER rover is located.

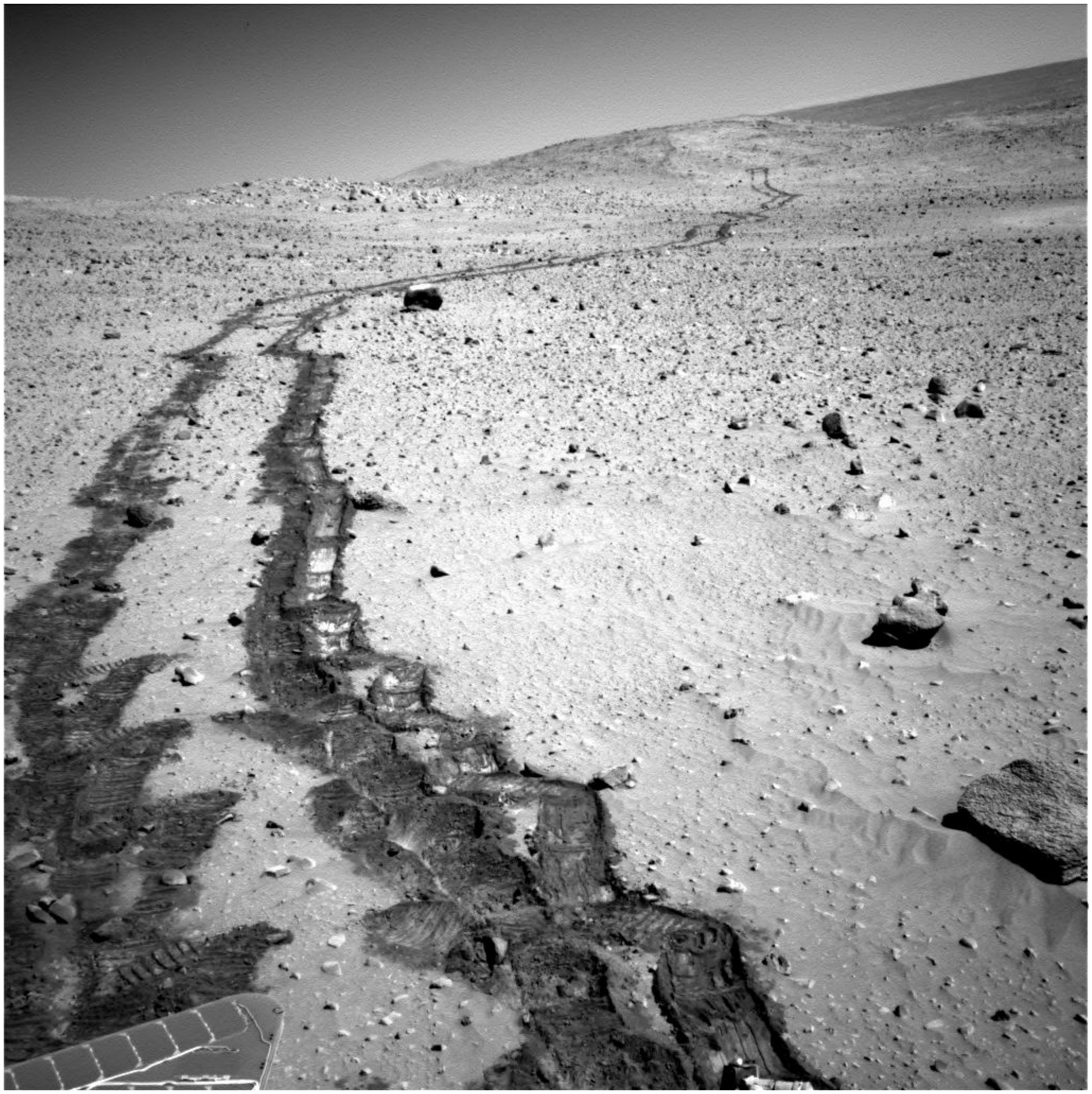
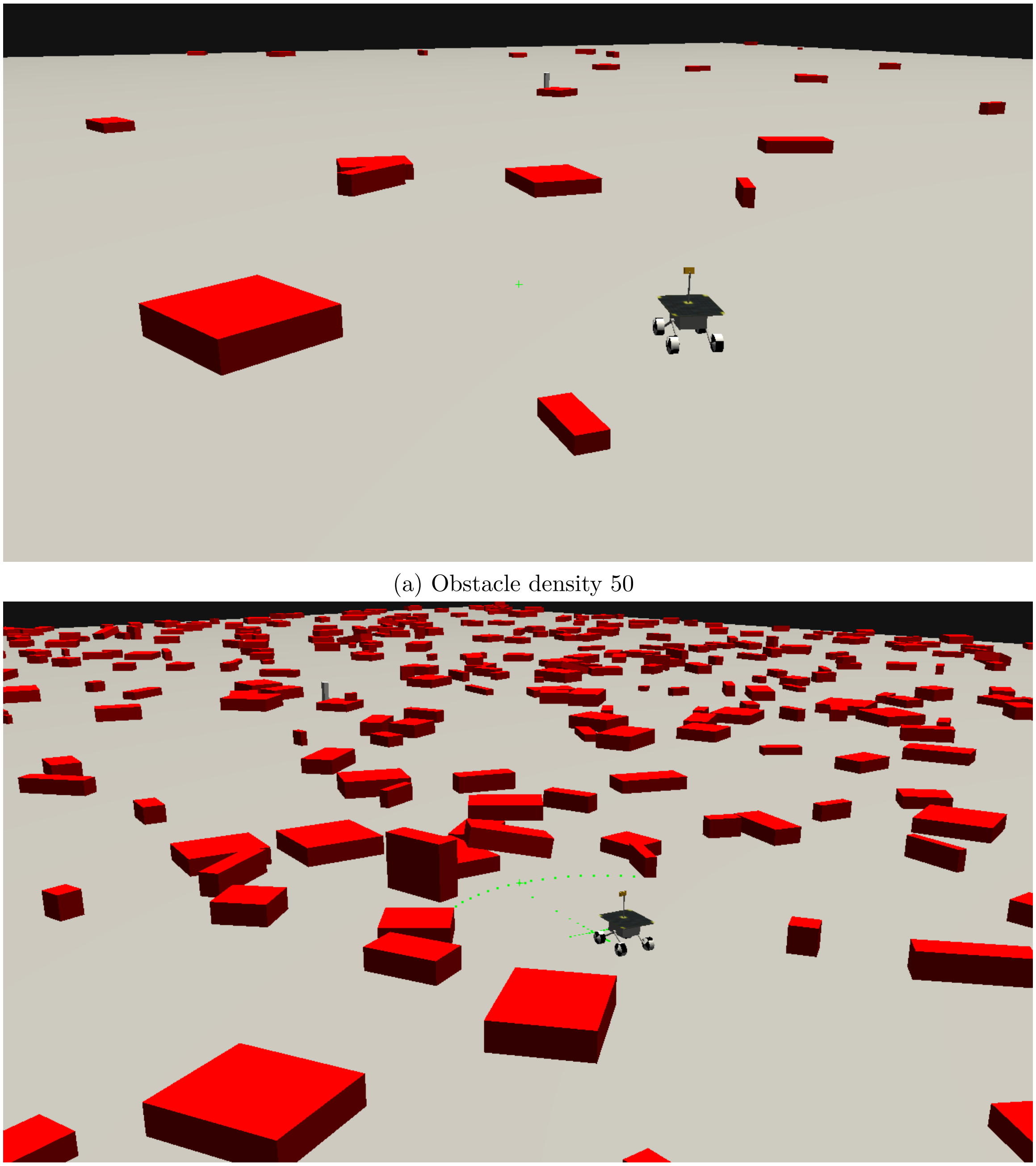


Figure 5.12: Very low obstacle densities on Mars seen from the Spirit MER rover on Sol 327. The distance between track marks is about 1m [49].



(a) Obstacle density 50

(b) Obstacle density 700

Figure 5.13: RoverSim obstacle densities with the simulated SR2 rover for scale comparison.

5.4 Rover System Perspective

The following is a discussion of multiple example situations involving the SR2 and MER rovers. The situations demonstrate the rover's capabilities given a particular environment and rover configuration. The environments are based on the simulated obstacle densities created in RoverSim. The rover configurations are defined by the actual dimensions of SR2 and MER. The effective sensor range for each rover is calculated based on its configuration using equation 5.8. All examples assume the rovers are traversing the same terrain specified for the particular example.

Each example situation assumes the rover is given a 1 hour period to traverse toward a goal point in an environment filled with a specific density of obstacles. The obstacles within the example environment are all equal to the minimum obstacle height for the rover. Both SR2 and MER have a minimum obstacle height of 0.2m. The distance the rover is physically capable of driving is computed based on its average speed while detecting obstacles and the 1 hour time period. It is further assumed that the rover is continuously making an effort to achieve the goal point under autonomous control. The rover's dimensions and sensor configuration are used to calculate the effective sensor range in equation 5.8. With the sensing range known the efficiency of the rover can be estimated from the data generated by the RoverSim experiments. Figure 5.6 is used to estimate the efficiency for each example. The corrected distance is the distance the rover traveled toward the goal. It is calculated using the estimated efficiency and the distance the rover drove during the 1 hour period. A comparison is made between the corrected distances calculated for each example situation.

The example outlined in table 5.3 compares the distances SR2 would achieve in a low density obstacle environment. Three sensor configurations are compared based on the height of the mast the sensor is mounted on. Two of the configurations are the low mounted panel laser sensor and the higher mounted mast camera. The third configuration is the calculated mast height required to achieve a 10m sensor range with a high degree of visibility. The efficiencies are estimated based on effective sensor range and listed near the bottom of the table. The corrected distance is then calculated by multiplying the efficiency by the drive distance. As expected from the RoverSim analysis in low obstacle densities the difference between the corrected distances is only 4m. However, to achieve the 1% gain in efficiency the mast height would have to become 20.2m high. A mast of this height is completely unacceptable and would never be implemented on a flight capable rover to achieve such a small improvement in efficiency.

Low Obstacle Density Example

	SR2-laser	**SR2-mast cam**	**SR2-10m range**
Speed	10cm/s	10cm/s	10cm/s
Mast Height	0.54m	1.0m	20.2m
Eff. Sensor Range	0.17m	0.4m	10m
Wheel Radius	10cm	10cm	10cm
Drive Dist.	360m	360m	360m
Efficiency	98%	98%	99%
Corrected Dist.	**352m**	**352m**	**356m**

Table 5.3: Comparison example of SR2 in low obstacle density field.

In a similar example, the same SR2 rover setup is used but the obstacle density is very high. The difference in efficiency for each effective sensor range is visible near the bottom of table 5.4. A gain of only 3% is seen in efficiency by using the mast cameras for navigation in comparison to the laser, which is mounted half as high. This gain equates to about 11m of additional progress toward the goal during the hour period. While a 3% gain in efficiency is significant it is not a large a increase. A path efficiency of 94% could be achieved with an effective sensor range of 10m. Once again, to achieve the larger effective sensor range the mast height must be 20.2m high which is not capable of being carried by a rover of similar scale to SR2.

Very High Obstacle Density Example

	SR2-laser	**SR2-mast cam**	**SR2-10m range**
Speed	10cm/s	10cm/s	10cm/s
Mast Height	0.54m	1.0m	20.2m
Eff. Sensor Range	0.17m	0.4m	10m
Wheel Radius	10cm	10cm	10cm
Drive Dist.	360m	360m	360m
Efficiency	65%	68%	94%
Corrected Dist.	**234m**	**245m**	**338m**

Table 5.4: Comparison example of SR2 in very high obstacle density field.

The following two example scenarios compare the SR2 and MER rovers in low and very high obstacle densities.

The sensor range for MER is calculated with the effective range equation 5.8 with regards to the mast cameras. In the one hour period for the example MER covers a 70m distance. During Opportunity's mission on Mars it is typically commanded to drive in 70m increments indicating these examples reflect the true capabilities of the rover.

The first comparison between SR2 and MER takes place in a low obstacle density environment. The efficiencies are estimated from figure 5.6 based on the effective sensor ranges listed in table 5.5. SR2's mast height is based on the laser scanner mounted underneath its solar panel. In low obstacle densities the rover efficiencies are all very high. The corrected distances indicate that even though SR2 has limited sensing range compared to MER, the speed SR2 is capable of navigating while avoiding obstacles is a much larger advantage. MER would require a mast height of 16.2m to obtain an effective sensor range of 10m. A mast height of this size is much to large and is not capable of effectively increasing efficiency.

The second comparison is located in a very high density environment. The effective sensor range for SR2 is based on the laser scanner and MER is based on the mast cameras. The distances driven in the one hour period by each rover are the same as

Low Obstacle Density Example

	SR2-laser	**MER-mast cam**	**MER-10m range**
Speed	10cm/s	2cm/s	2cm/s
Mast Height	0.54m	1.4m	16.2m
Eff. Sensor Range	0.17m	0.81m	10m
Wheel Radius	10cm	12.5cm	12.5cm
Drive Dist.	360m	70m	70m
Efficiency	98%	98%	99%
Corrected Dist.	**352m**	**68m**	**69m**

Table 5.5: Comparison example of SR2 and MER in low obstacle density field.

the previous example. However, the efficiencies of each rover configuration are lower because of the very high obstacle density. The corrected distance of SR2 and MER show that the speed advantage of SR2 allows it to make significantly more progress toward the goal even though SR2 has the lowest path efficiency. The capability of MER is still substantially lower than SR2 even if it were possible to increase its efficiency to 95% by increasing its effective sensor range to 10m. Again, this is not possible because of the excessively large mast height required for an effective sensor range of 10m.

Very High Obstacle Density Example

	SR2-laser	**MER-mast cam**	**MER-10m range**
Speed	10cm/s	2cm/s	2cm/s
Mast Height	0.54m	1.4m	16.2m
Eff. Sensor Range	0.17m	0.81m	10m
Wheel Radius	10cm	12.5cm	12.5cm
Drive Dist.	360m	70m	70m
Efficiency	65%	70%	95%
Corrected Dist.	**234m**	**49m**	**66m**

Table 5.6: Comparison example of SR2 and MER in very high obstacle density field.

The examples demonstrate that an excessively large mast height is required to achieve a large sensor range with a high degree of visibility. And, if it were possible to build a system to achieve the large sensor ranges they only improve path efficiency in terrain densely populated by obstacles. However, based on the terrain encountered on Mars by the MER rovers the obstacle densities are very low. This signifies that large sensor ranges are not a requirement for navigating on Mars.

The comparison of SR2 and MER indicate that the speed a rover is capable of navigating at has more of an effect on performance than its sensor range. SR2 is capable of achieving a higher speed while avoiding obstacles than MER because it uses a more simplified perception system. By reducing the time between updates in the obstacle detection system the rover can make more progress in less time. Although, to achieve the higher sensor update rate the range of SR2's sensors is reduced. As stated above the obstacle densities seen on Mars are low therefore the efficiency of the rover will not be affected by a limited sensor range.

Chapter 6

Conclusions

Rovers are an important tool for planetary exploration. For the near future they will be carrying out most planetary science objectives due to the massive safety and budgetary limitations involved in sending humans to perform similar tasks. Once humans plant a strong foothold in space, autonomous rovers will likely be used as support platforms to assist astronauts during surface operations. This research has been guided to better understand the effects of sensor range on rover efficiency. The terrain that has been traversed by rovers related to this research has been done so using vision based sensing. This led to the creation of the Solar Rover 2 and RoverSim. The experiments conducted from both SR2 and RoverSim indicate that sensor selection can affect the overall design of a rover as well as its performance. The results indicate that from a practical stance sensor range does not affect the path efficiency of a rover in natural terrain. Sensor visibility has a large affect on rover performance as sensing range increases. However, it also affects the physical design of the rover by requiring sensors to be placed above obstacles to maintain a high degree of visibility. While the performance is lower for a rover with limited visibility there appears to be almost no benefit to increasing sensor range of limited visibility and no practical way of achieving full visibility.

SR2 has illustrated that a rover with a low resolution short range sensor array mounted on a simplified mobility system has performance characteristics similar to and in some cases better than currently selected Mars rover technology. During the Anza Borrego field test SR2 was capable of achieving traverse rates as high as

15cm/s on Mars-like terrain. The total distance traversed by the end of the trials was over 14.8km accomplished in 22 days of driving. During a single day the rover could cover as much as 1.5km along a path of waypoints without human intervention. Distances of a similar scale were driven throughout many other days of the field work. Even with SR2's power intensive skid steering drivetrain, more power was used for computation than mobility. On average computation and sensing used 60.6W while the motors required only 23.7W throughout the 2008 field trial. Obstacle densities encountered by SR2 proved to be substantially higher than the areas explored by the Mars Exploration Rovers. SR2 came upon multiple sand laden areas that significantly affected progress due to their undetectable nature with current sensing techniques. The MER rovers have also confronted similar situations, further indicating the need for a solution.

RoverSim is capable of simulating many of the features seen in a physical model; it is a viable substitute for more rover field work. It can generate various levels of obstacle densities similar to what is visible on Mars or the Anza Borrego test site. The experimental setup generated 43,028.8km of data from 722,820 simulated paths by varying the sensor range and visibility across multiple obstacle densities. The largest source of incomplete paths are linked to the local minima detection sequence in RoverSim. However, removing the incomplete paths has not significantly influenced the results. The resulting analysis indicates neither sensor range nor visibility affect paths generated through low obstacle densities. The simulated rover was capable of achieving greater than 97% efficiency on average for all sensor configurations in low density terrain. As obstacle density increases sensor placement to improve visibility will also improve the performance for longer range sensors. Although, the practical application of improving sensor range with a high degree of visibility is severely limited. Rover path efficiency through any obstacle density under limited visibility is not significantly affected by sensor range. Only a 2.5% increase is seen between the average efficiency for simulated sensor ranges from 1m to 10m.

6.1 Future Work

The areas that require attention for better defining this research include the following:

- Analyze the quantity of heading changes with respect to sensor range.
- Determine how localized mapping of obstacle locations between rover steps affect efficiency.
- Better define the effective sensor range equation, normalize it so different rovers can be compared directly. Correct for obstacle height as it approaches sensor height, i.e. visibility should be zero.
- Increase the stability of the local minima detection for simulated paths in RoverSim.
- Include varied terrain in RoverSim for simulated paths.
- Include negative obstacles as part of the study.
- Simulate sensors to a higher degree of fidelity.

Bibliography

[1] Cogmation robotic simulator. `http://www.cogmation.com/`, August 2011.

[2] Deepak Bapna, Eric Rollins, John Murphy, Mark Maimone, William Red L. Whittaker, and David Wettergreen. The atacama desert trek: Outcomes. In *Proceedings of ICRA 1998*, volume 1, pages 597–604, May 1998.

[3] Jerome Barraquand and Jean-Claude Latombe. Robot motion planning: A distributed representation approach. *Robotics Research*, 10(6):628–649, 1991.

[4] John Bchrach, James McLurkin, and Anthony Grue. Protoswarm: A language for programming multi-robot systems using the amorphous medium abstraction. In *Autonomous Agents and Multi-Agent Systems*, volume 3 of *7*, pages 1175–1178, Estoril, Portugal, May 2008. International Conference on Autonomous Agents, International Foundation for Autonomous Agents and Multiagent Systems.

[5] Donald B. Bickler. The New Family of JPL Planetary Surface Vehicles. In *Missions, Technologies and Design of Planetary Mobile Vehicles*, pages 301–306, Toulouse, France, September 1992. CNES.

[6] Jeffrey Biesiadecki, Chris Leger, and Mark Maimone. Tradeoffs between directed and autonomous driving on the mars exploration rovers. In Sebastian Thrun, Rodney Brooks, and Hugh Durrant-Whyte, editors, *Robotics Research*, volume 28 of *Springer Tracts in Advanced Robotics*, pages 254–267. Springer Berlin / Heidelberg, California Institute of Technology Jet Propulsion Laboratory Pasadena CA USA Pasadena CA USA, 2007.

[7] J.J. Biesiadecki, E.T. Baumgartner, R.G. Bonitz, B. Cooper, F.R. Hartman, P.C. Leger, M.W. Maimone, S.A. Maxwell, A. Trebi-Ollennu, E.W. Tunstel, and J.R. Wright. Mars exploration rover surface operations: driving opportunity at meridiani planum. *Robotics Automation Magazine, IEEE*, 13(2):63–71, 2006.

[8] J.J. Biesiadecki, E.T. Baumgartner, R.G. Bonitz, B.K. Cooper, F.R. Hartman, P.C. Leger, M.W. Maimone, S.A. Maxwell, A. Trebi-Ollenu, E.W. Tunstel, and J.R. Wright. Mars exploration rover surface operations: driving opportunity at meridiani planum. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 2, pages 1823–1830. IEEE, 2005.

[9] Lisa Billingsley. Analysis of requirements of a mars rover mission to active gullies. Masters thesis, University of Oklahoma, Norman, OK, August 2010.

[10] Francois Blais. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, pages 231–240, January 2004.

[11] Yang Cheng, Mark Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 1, pages 903–910. IEEE, October 2005.

[12] Paul R. Cohen. *Empirical Methods for Artifical Intelligence.* MIT press, Cambridge, MA, 1995.

[13] Erwin Coumans. Bullet physics library. http://bulletphysics.org/wordpress/, 2011.

[14] A. Cuerva, A. Sanz-Andres, and R. D. Lorenz. Sonic anemometry of planetary atmospheres. In A. Wilson, editor, *International Workshop Planetary Probe Atmospheric Entry and Descent Trajectory Analysis and Science*, volume 6 of *SP-544*, pages 295–302, Lisbon, Portugal, October 2003. ESA Publications Division.

[15] Guilherme N. DeSouza and Avinash C. Kak. Vision for mobile robot navigation: A survey. In *Transactions on Pattern Analysis and Machine Intelligence*, volume 24, pages 237–267. IEEE, February 2002.

[16] Richard J. Doyle. Aspects of the rover problem. Artificial Intelligence Laboratory Working Papers WP-231, MIT, 1982.

[17] Gregory Dudek and Michael Jenkin. *Computational Principles of Mobile Robotics*. Cambridge University Press, New York, NY, second edition, 2000.

[18] Nokia Qt Development Frameworks. Qt framework. `http://qt.nokia.com/downloads`, August 2011.

[19] Hoa g. Nguyen and Michael R. Blackburn. A simple method for range finding via laser triangulation. Technical Report 2734, Naval Command, Control and Ocean Surveillance Center, San Diego, CA, January 1995.

[20] Brian Gerkey, Richard T. Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *In Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal, June 2003. ICAR.

[21] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimal cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[22] Wesley H. Huang and Eric P. Krotkov. Optimal stereo mast configuration for mobile robots. In *International Conference on Robotics and Automation*, Albuquerque, New Mexico, April 1997. IEEE.

[23] Louis Hugues and Nicolas Bredeche. Simbad: an autonomous robot simulation package for education and research. In *In Proceedings of, The Ninth International*

Conference on the Simulation of Adaptive Behavior, Artificial Intelligence series, pages 831–842, Roma, Italy, 2006. Springer's lecture Notes in Computer Sciences.

[24] Jon Klein. BREVE: A 3D Environment for the Simultaion of Decentralized Systems and Artificial Life. In *The 8th International Conference on the Simulation and Synthesis of Living Systems*. Artificial Live VIII, MIT press.

[25] Eric Krotkov, Martial Hebert, and Reid Simmons. Stereo perception and dead reckoning for a prototype lunar rover. *Autonomous Robots*, pages 313–331, 1995.

[26] NASA Jet Propulsion Lab. Mars exploration rover mission. `http://marsrovers.jpl.nasa.gov/home/`, June 2010.

[27] Sharon L. Laubach, Joel Burdick, and Larry Matthies. An autonomous path planner implemented on the rocky7 prototype microrover. In *International Conference on Robotics and Automation*, pages 292–297, Leuven,Belgium, May 1998. IEEE.

[28] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U. K., 2006.

[29] P.C. Leger, R.G. Deen, and R.G. Bonitz. Remote image analysis for mars exploration rover mobility and manipulation operations. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 1, pages 917–922. IEEE, October 2005.

[30] P.C. Leger, A. Trebi-Ollennu, J.R. Wright, S.A. Maxwell, R.G. Bonitz, J.J. Biesiadecki, F.R. Hartman, B.K. Cooper, E.T. Baumgartner, and M.W. Maimone. Mars exploration rover surface operations: driving spirit at gusev crater. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 2, pages 1815–1822. IEEE, October 2005.

[31] Richard LeGrand, Kyle Machulis, David P. Miller, Randy Sargent, and Anne Wright. The XBC: a modern low-cost mobile robot controller. In *Proceedings of IROS 2005*. IEEE Press, August 2005.

[32] R. A. Lindemann, D. B. Bickler, B. D. Harrington, G. M. Oritz, and C. J. Voorhees. Mars exploration rover mobility development - mechanical mobility design, development, and testing. *Robotics & Automation Magazine, IEEE*, 13(2):19–26, 2006.

[33] Tomas Lozano-Perez. Spatial planning: A configuration space approach. In *IEEE Transactions on Computers*, volume C-32, pages 108–120. IEEE, February 1983.

[34] Mark Maimone. International planetary rover efforts. `http://www-2.cs.cmu.edu/ mwm/rover/`, 1997.

[35] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars elploration rovers. *Journal of Field Robotics*, 24(3):169–186, March 2007.

[36] Mark Maimone, Andrew Johnson, Yang Cheng, Reg Willson, and Larry Matthies. Autonomous navigation results from the mars exploration rover (mer) mission. In Marcelo Ang and Oussama Khatib, editors, *Experimental Robotics IX*, volume 21 of *Springer Tracts in Advanced Robotics*, pages 3–13. Springer Berlin / Heidelberg, Jet Propulsion Laboratory, California Institute of Technology 4800 Oak Grove Drive, Pasadena, CA, 91109 USA USA, 2006.

[37] M. C. Malin and K. S. Edgett. Mars global surveyor mars orbiter camera: Interplanetary cruise through primary mission. *Geophys. Research*, 106(E10):23,429–23,570, 2001.

[38] Larry Matthies. Stereo vision for planetary rovers: Stochastic modeling to near

real-time implementation. *International Journal of Computer Vision*, 8(1):71–91, February 1992.

[39] Larry Matthies, Mark Maimone, Andrew Johnson, Yang Cheng, Reg Willson, Carlos Villalpando, Steve Goldberg, Andres Huertas, Andrew Stein, and Anelia Angelova. Computer vision on mars. *International Journal of Computer Vision*, 75(1):67–92, 2007.

[40] Larry Matthies and Steven Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, RA-3(1):239–250, June 1987.

[41] James McLurkin and Jennifer Smith. Distributed algorithms for dispersion in indoor environments and using a swarm of autonomous mobile robots. In *7th international Symposium on Distributed Autonomous Robotic Systems*. DARS, June 2007.

[42] M.Golombek, T. Parker, T. Schofield, and D. Kass et.all. Preliminary engineering constraints and potential landing sites for the mars exploration rovers. In *Lunar and Planetary Science XXXII*, pages 1234–1235, 2001.

[43] Olivier Michel. Webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.

[44] D. P. Miller, T. Hunt, M. Roman, S. Swindell, L. Tan, and A. Winterholler. Experiments with a long-range planetary rover. In *Proceedings of the The 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Nara, Japan, May 2003. ISAS, NAL, NASA.

[45] David P. Miller and Michael Ravine. Semi-autonomous rover operations: An integrated hardware and software approach for more capable mars rover missions. In *NASA Science Technology Conference*. NASA, June 2007.

[46] Andrew Mishkin. *Sojourner: An insider's view of the Mars Pathfinder Mission.* Berkley books, 2003.

[47] Stewart Moorehead, Reid Simmons, Dimitrios Apostolopoulos, and William Red Whittaker. Autonomous navigation field results of a planetary analog robot in antarctica. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, June 1999.

[48] Robin R. Murphy. *Introduction to AI Robotics.* MIT press, Cambridge, MA, 2002.

[49] NASA. Nasa image. `http://marsrover.nasa.gov/home/index.html`.

[50] I.R. Nourbakhsh, D. Andre, C. Tomasi, and M. Genesereth. Obstacle avoidance via depth from focus. In *ARPA Image Understanding Workshop*, pages 1339–1344, Palm Springs, CA, USA, February 1996.

[51] Shervin Nouyan. *Path Formation and Goal Search in Swarm Robotics.* phdthesis, University of Brussels, Belgium, August 2004.

[52] Mike Ravine. Mars scout class rover proposal. Personal e-mail, July 2005.

[53] Matthew J. Roman. Design and analysis of a four wheeled planetary rover. Masters thesis, University of Oklahoma, Norman, OK, May 2005.

[54] Matthew J. Roman and David P. Miller. Four is enough. In *Proceedings of Space 2006*, pages AIAA–2006–7399. AIAA, September 2006.

[55] Benjamin Shamah, Michael D. Wagner, Stewart Moorehead, James Teza, David Wettergreen, and William (Red) L. Whittaker. Steering and control of a passively articulated robot. In *SPIE, Sensor Fusion and Decentralized Control in Robotic Systems IV*, pages 4571–4577, October 2001.

[56] Samuel Shapiro and Martin Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, 1965.

[57] Marc G. Slack. *Situationallly Driven Local Navigation for Mobile Robots.* PhD thesis, Virginia Tech, Pasadena, California, April 1990.

[58] Anthony Stentz. Optimal and efficient path planning for partially-known environments. In *International Conference on Robotics and Automation*, volume 4, pages 3310–3317. IEEE, May 1994.

[59] Roger Stettner, Howard Bailey, and Steven Silverman. Large format time-of-flight focal plane detector development. *Proceedings of SPIE, International Society for Optical Engineering*, pages 288–292, April 2005.

[60] Roger Stettner, Howard Bailey, and Steven Silverman. Three dimensional flash ladar focal plans and time dependent imaging. April 2008.

[61] E. Tunstel, M. Maimone, A. Trebi-Ollennu, J. Yen, R. Petras, and R. Willson. Mars exploration rover mobility and robotic arm operational performance. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 2, pages 1807–1814. IEEE, October 2005.

[62] Martin J. L. Turner. *Expedition Mars.* Praxis, Chichester, UK, 2004.

[63] Christopher Urmson, M Bernardine Dias, and Reid Simmons. Stereo vision based navigation for sun-synchronous exploration. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '02)*, volume 1, pages 805–810. IEEE, September 2002.

[64] David W. Way, Richard W. Powell, Allen Chen, Adam D. Steltzner, A. Miguel San Martin, P. Daniel Burkhart, and Gavin F. Mendeck. Mars sci-

ence laboratory: Entry, decent, and landing system performance. In *Aerospace Conference*, pages 1–19. IEEE, March 2007.

[65] David Wettergreen. Life in the Atacama. `http://www.frc.ri.cmu.edu/atacama/`, 2005.

[66] David Wettergreen, Deepak Bapna, Mark Maimone, and H. Thomas. Developing nomad for robotic exploration of the atacama desert. *Robotics and Autonomous Systems*, 26(2-3):127–148, February 1999.

[67] David Wettergreen, Nathalie Cabrol, Vijayakumar Baskaran, Francisco Calderon, Stuart Heys, Dominic Jonak, Rolf Allan Luders, David Pane, Trey Smith, James Teza, Paul Tompkins, Daniel Villa, Chris Williams, and Michael D Wagner. Second experiments in the robotic investigation of life in the atacama desert of chile. In *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, volume 603, September 2005.

[68] David Wettergreen, Nathalie Cabrol, James Teza, Paul Tompkins, Christopher Urmson, Vandi Verma, Michael D Wagner, and William Red L. Whittaker. First experiments in the robotic investigation of life in the atacama desert of chile. In *International Conference on Robotics and Automation*. IEEE, April 2005.

[69] David Wettergreen, M Bernardine Dias, Benjamin Shamah, James Teza, Paul Tompkins, Christopher Urmson, Michael D. Wagner, and William (Red) L. Whittaker. First experiment in sun-synchronous exploration. In *International Conference on Robotics and Automation*, pages 3501–3507, May 2002.

[70] David Wettergreen, Michael D. Wagner, Dominic Jonak, Vijayakumar Baskaran, Matthew Deans, Stuart Heys, David Pane, Trey Smith, James Teza, David R. Thompson, Paul Tompkins, and Chris Williams. Long-distance autonomous survey and mapping in the robotic investigation of life in the atacama desert. In

International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS), February 2008.

[71] Zac White, David P. Miller, Matt Roman, Michael Ravine, Daniel Flippo, Brandon Mills, Brian Nixon, Eldar Noe, and Michael Malin. Control and operations for a long duration solar powered mars rover. In *WWDC 2006 Scientific Development Poster Session*. Apple Computer Corp., 2006.

[72] Brian Wilcox. Robotic vehicle system engineering. NASA Presentation, April 2003.

Appendix A

SR2 Field Data

2006 (June) Field test data					
Day	Start	End	elapsed	distance	GPS
14	10:37	11:33	0:55:28	7.85m	
15	12:36	13:28	0:51:40	80.55m	
16	12:21	15:42	3:20:20	280.93m	274m
17	11:49	15:1	3:12:20	196.55m	138m
18	10:42	15:12	4:29:26	789.57m	918m
19	11:30	15:38	4:7:57	943.17m	909m
20	11:0	14:38	3:37:41	560.09m	502m
21	10:2	14:48	4:46:0	759.11m	822m
22	9:47	14:35	4:48:3	1003.21m	922m
23	9:41	12:53	3:12:48	922.76m	889m
24	8:27	10:1	1:34:41	100.88m	
		total	15:47:39	5644.67m	

Table A.1: SR2 field test data from September 2006.

2007 (June) Field test data					
Day	Start	End	elapsed	distance	GPS
13	12:30	13:11	0:40:52	29.72m	
14	12:06	15:04	2:57:31	724.18m	753m
16	10:34	16:18	5:43:54	836.93m	654m
17	9:46	16:26	6:39:59	1561.98m	1592m
18	9:39	15:35	5:55:35	1323.63m	1268m
		total	11:17:56	4476.44m	

Table A.2: SR2 field test data from September 2007.

[1] Opportunity readings as of May 24, 2011

[2] Stopped receiving data from Spirit on March 22, 2010 (Sol 2210). Mission end May 24,2011

2008 (September) Field test data					
Day	Start	End	elapsed	distance	GPS
22	14:31	14:39	0:7:55	64.85m	
23	11:10	15:18	4:8:35	653.36m	512m
24	10:50	15:18	4:28:20	962.51m	948m
25	10:40	15:11	4:30:11	954.06m	949m
26	10:26	14:47	4:20:48	1322.80m	1338m
27	10:9	12:56	2:46:34	756.58m	710m
		total	9:4:22	4714.17m	

Table A.3: SR2 field test data from September 2008.

Test Year	Drive Time	Distance
2006	15h : 47m : 39s	5644.7m
2007	11h : 17m : 56s	4476.4m
2008	9h : 04m : 22s	4714.2m
Total	36h : 9m : 57s	14835.3m

Table A.4: Time and distance results from Anza Borrego field trials with SR2.

	Opportunity		Spirit	
Year	Odometer	Δ Distance	Odometer	Δ Distance
2004	1,998m	1,998m	3,958	3,958
2005	6,502m	4,504m	5,673m	1,715m
2006	9,758m	3,256m	6,887m	1,214m
2007	11,591m	1,833m	7,527m	640m
2008	13,617m	2,026m	7,530m	3m
2009	18,928m	5,310m	7,730m	200m
2010	26,506m	7,578m	7,730m	0.5m
2011	29,908m[1]	3,403m	7,730m[2]	0m

Table A.5: Mars exploration rovers yearly distances traveled.