

# **Navigation Regimes for Off-Road Autonomy**

Christopher Urmson

CMU-RI-TR-05-23

*Submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy in Robotics*

The Robotics Institute  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, Pennsylvania 15213

May, 2005

© 2005 by Christopher Urmson

Thesis Committee:  
William "Red" Whittaker, Co-chair  
Reid Simmons, Co-chair  
Alonzo Kelly  
Steven Lavalley, University of Illinois



---

## Abstract

---

A new approach to high-speed, off-road navigation is presented. The interrelatedness of sensing horizon, prior map resolution, speed and efficiency is investigated over the space of off-road navigation. From this analysis, the space of off-road navigation is partitioned into three regimes (efficiency-limited, stop-limited and swerve-limited). Safeguarding through pre-planning and swerving emerges as an approach to achieve greater performance than possible with panic-stopping.

The swerve-limited regime is characterized by high speeds that are achieved using plans derived in part from prior maps. Within the swerve-limited regime, sensing horizon is linearly proportional to robot speed. Thus an autonomous robot can operate at speeds beyond the point at which panic stopping is viable if it is safe to avoid obstacles by swerving. Navigating safely within this regime therefore explicitly requires some prior knowledge or assumption.

The efficiency-limited regime is characterized by low speeds. Onboard sensing horizon is fixed given a specified prior map resolution and desired navigation efficiency. Typical applications include planetary exploration and mine-mapping

For operation between the efficiency-limited and swerve-limited regimes, robot stopping distance limits speeds. In this stop-limited regime, sensing horizons increase quadratically as speed rises to ensure robot safety. Algorithms designed for navigation within the stop-limited regime generally require some cognizance of vehicle dynamics.

Two implemented approaches to off-road navigation, one relevant for high-speed, long duration driving, and the other for planetary exploration, are contrasted and described in detail. These particular implementations exemplify the efficiency-limited and swerve-limited regimes and represent the state-of-the-art in planetary exploration and high-speed navigation.





---

# Table of Contents

---

<i>Abstract</i> .....	<i>i</i>
<i>Table of Contents</i> .....	<i>iii</i>
<i>List of Figures</i> .....	<i>vii</i>
<i>List of Tables</i> .....	<i>xi</i>
<i>Acknowledgments</i> .....	<i>xiii</i>
<b>1 Introduction</b> .....	<b>1</b>
Motivation.....	2
Applications .....	2
Problem Statement.....	5
Thesis Statement .....	5
Preview .....	5
Outline.....	9
<b>2 Background</b> .....	<b>10</b>
Regimes in Autonomy .....	10
Low-Speed Off-Road Navigation.....	12
High-Speed Off-Road Navigation.....	14
<b>3 Methodology</b> .....	<b>17</b>
Analysis .....	17
Simulation.....	18
Testing.....	22
<b>4 Stopping, Swerving &amp; Sensing</b> .....	<b>23</b>
Stopping Distance.....	23
Swerve Distance.....	24
Sensing at Range .....	28
Sensing at Speed .....	32
<b>5 Sensing Horizon and Prior Maps</b> .....	<b>36</b>
Sensor Horizon, Map Resolution and Navigation Efficiency .....	36
Sensor Horizon, Map Resolution and Navigation Efficiency with Optimistic Prior Maps .....	38
Sensor Horizon, Map Resolution and Navigation Efficiency with Pessimistic Prior Maps .....	40
Sensor Horizon, Map Resolution and Navigation Efficiency on Synthetic Terrain.....	43

	Sensor Horizon and Navigation Efficiency without Prior Maps .....	46
	Implications .....	47
<b>6</b>	<b><i>Navigation Regimes</i></b> .....	<b>49</b>
	Efficiency-Limited Regime .....	50
	Stop-limited Regime.....	51
	Swerve-Limited Regime.....	52
	Efficiency/Stop-Limited Transition.....	53
	Stop/Swerve-Limited Transition.....	53
	Discussion.....	55
<b>7</b>	<b><i>Contrasting Navigation Regimes: Efficiency-Limited and Swerve-Limited</i></b> .....	<b>56</b>
	Speed .....	57
	Prior Map Resolution .....	58
	Sensors .....	59
	Terrainability .....	60
	Pre-Planning.....	60
	Real-Time Planning .....	61
	Summary.....	61
<b>8</b>	<b><i>Representative Implementation: Swerve-Limited Regime</i></b> .....	<b>63</b>
	Context .....	63
	Mapping & Pre-Planning .....	65
	Platform .....	75
	Navigation Software.....	85
	Field Experiments .....	96
	Summary.....	106
<b>9</b>	<b><i>Representative Implementation: Efficiency-Limited Regime</i></b> .....	<b>107</b>
	Context .....	107
	Pre-Planning.....	109
	Platform .....	110
	Navigation Software.....	114
	Field Experiments .....	122
	Summary.....	130
<b>10</b>	<b><i>Conclusions</i></b> .....	<b>132</b>
	Summary.....	132
	Contributions.....	133

<b>Future Work.....</b>	<b>133</b>
<b>Cultural Lessons.....</b>	<b>135</b>
<b>Closing Remark.....</b>	<b>136</b>
<b><i>Appendix A: Mojave Desert Cost Maps .....</i></b>	<b><i>137</i></b>
<b><i>Glossary .....</i></b>	<b><i>143</i></b>
<b><i>References .....</i></b>	<b><i>146</i></b>



---

## List of Figures

---

Figure 1. Fixed, quadratic and linear sensing horizons. ....	7
Figure 2. Composite of sensing horizon curves. ....	8
Figure 3. A comparison of the digital elevation model and cost map. ....	21
Figure 4. Representative fractally generated terrain. ....	21
Figure 5. Summary surface: navigation efficiency vs. prior map resolution and sensing horizon. ....	22
Figure 6. Stopping distance as a function of speed. ....	24
Figure 7. Swerve maneuver. ....	24
Figure 8. Swerving distance for 1 to 10m swerve offsets. ....	27
Figure 9. Laser sensor geometry. ....	30
Figure 10. Sensor error as function of range. ....	31
Figure 11. Required sensor pointing accuracy as a function of range. ....	32
Figure 12. Maximum speed for stereo vision example. ....	34
Figure 13. Maximum speed for LIDAR example. ....	35
Figure 14. Prior maps with decreasing resolution (10, 25, 100 pixel sampling). ....	37
Figure 15. Composite performance cost over six terrains. ....	38
Figure 16. Summary surface for optimistic prior maps. ....	39
Figure 17. Route path with path segment that doubles back highlighted. ....	40
Figure 18. Summary surface for pessimistic prior maps. ....	41
Figure 19. Summary surface for three non dog-leg terrains. ....	42
Figure 20. Ratio between nominal and pessimistic values in prior maps. ....	42
Figure 21. Summary surface for low frequency terrain. ....	44
Figure 22. Summary surface for high frequency terrain. ....	44
Figure 23. Ratio between fit and actual data for low frequency terrain. ....	45
Figure 24. Ratio between fit and actual data for high frequency terrain. ....	45
Figure 25. Summary curve for low frequency synthetic terrain. ....	46
Figure 26. Summary curve for high frequency synthetic terrain. ....	47
Figure 27. A division of the autonomous navigation space. ....	49
Figure 28. Navigation regimes. ....	50
Figure 29. Example of efficiency limited regime tradeoff. ....	51
Figure 30. Stop/Swerve-limited regime transition example. ....	54

Figure 31. Sandstorm. ....	64
Figure 32. A comparison of USGS and 5m postings DEM data quality. ....	66
Figure 33. A comparison of USGS and sub-meter resolution image quality. ....	66
Figure 34. Operation of the global path planner.....	70
Figure 35. Three views of data available to route editors.....	72
Figure 36. Spline construction for the route editing tool. ....	74
Figure 37. Shock isolation components that support the electronics box.....	77
Figure 38. Overlapping fields of view of Sandstorm's sensors.....	79
Figure 39. Sandstorm with sensor placement indicated.....	80
Figure 40. The gimbal. ....	84
Figure 41. On-board software architecture. ....	85
Figure 42. Configurations where non-pointed sensors perform poorly.....	87
Figure 43. A sensor view of an opening in a fence. ....	90
Figure 44. A plot showing the response of Sandstorm's velocity control loop....	95
Figure 45. Sandstorm testing in Nevada.....	96
Figure 46. Test facilities at the Nevada Automotive Test Center. ....	98
Figure 47. Histogram of run length distribution. ....	98
Figure 48. Scatter plot of run length versus speed. ....	99
Figure 49. Aerial photograph of Simpson Road canyon.....	100
Figure 50. Digital elevation model of Simpson Road canyon.....	101
Figure 51. Gimbal pointing performance.....	102
Figure 52. Sandstorm's Primary LIDAR sensing horizon. ....	103
Figure 53. A sensor view of the trail in the Hairpin.....	104
Figure 54. Pre-planned route around the hairpin. ....	105
Figure 55. Sandstorm high-centered.....	106
Figure 56. Hyperion in Arctic Canada. ....	108
Figure 57. Hyperion. ....	111
Figure 58. Hyperion power distribution system. ....	113
Figure 59. Modules operational by navigation mode.....	115
Figure 60. Safeguard operational mode.....	116
Figure 61. Autonomous navigation mode. ....	117
Figure 62. Hyperion's stereo vision based terrain evaluation.....	121
Figure 63. The location of Haughton Crater.....	122

Figure 64. The Von Braun Planitia on Devon Island. ....	123
Figure 65. Obstacle avoidance display. ....	126
Figure 66. Relative occurrence of heading error. ....	127
Figure 67. First sun-synchronous mission plan. ....	128
Figure 68. Second sun-synchronous mission plan. ....	130
Figure 69. Mojave Desert Cost Map 1. ....	137
Figure 70. Mojave Desert Cost Map 2. ....	138
Figure 71. Mojave Desert Cost Map 3. ....	139
Figure 72. Mojave Desert Cost Map 4. ....	140
Figure 73. Mojave Desert Cost Map 5. ....	141
Figure 74. Mojave Desert Cost Map 6. ....	142





---

## List of Tables

---

Table 1. Stereo vision characteristics for example implementation. ....	33
Table 2. A summary comparison of archetypical robots. ....	57
Table 3. HMMWV terrainability characteristics.....	76
Table 4. Gimbal performance characteristics.....	85
Table 5. Navigation Cycle. ....	121



---

## Acknowledgments

---

Through my time at the Robotics Institute, I have had the great fortune to work with a number of gifted and inspired teams. The Nomad team taught me a lot about developing software and hardware fast, and how to do it right. The Skyworker team gave me my first taste of technical leadership and inspired me with what a small team of dedicated students can do. The Sun-Synchronous Exploration and the Life in the Atacama teams taught me how a good field campaign is run and how to build a robot well. Finally, the Red Team let me try something really crazy and showed me what can happen if you dare. There are too many great people involved with these teams to mention them all, but the ideas and implementations presented here would not have been possible without them. Thank you.

Though the work is exciting and interesting, it would not have been possible to get through my PhD without great collaborators and friends. Bernardine, Sarjoun, Paul, Vandi, Nick R. and Mike M. have been great friends and sounding boards. I haven't given a worthwhile talk that one of them didn't help to refine. Nick M. and Matt were key to the success of Sandstorm, and key to keeping me sane. Kevin is my software development partner for Sandstorm and friend through testing during a frigid Pittsburgh winter and a hectic Nevada spring. Mike W. provided much of the soundtrack to which this work was performed (I may still have the devil's haircut, in my mind) and is a tremendous colleague and friend.

David has had a significant impact on my development as a researcher. It was his robot that brought me to CMU. His leadership and guidance have played a significant role in steering my research and his friendship has made it fun and worthwhile. The FRC faculty have been a tremendous resource and I'm lucky to count them among my colleagues and friends.

Reid and Red have been a great, complementary pair of advisors. Reid's down to earth demeanor and cutting insight have helped make sure that my work is "PhD worthy", and has helped improve my ability to think critically. Red has shown me that achieving a vision has more to do with the person than anything else. The numerous multi-hour discussions we've had about life, career and research are a valuable seed from which to grow my own wisdoms.

Lastly, I thank my family, son, and wife. My parents and brothers have always believed in me, and encouraged me for as long as I can remember. Ethan has helped to reawaken my sense of wonder. Jen always finds a way to convince me I'm never as dumb (or as smart) as I think I am. She has encouraged me to follow my dreams, even when they take me away from home for weeks or months at a time. Without her, I would certainly have stumbled, if not fallen along the path.



---

# 1 Introduction

---

The vision of reliable, off-road autonomous driving is yet to be fulfilled. While robots regularly traverse rough terrain at low speeds [Goldberg02, Wettergreen02, Kelly04] and at higher speeds in simple off-road scenarios [Bellutta00, Coombs00], autonomous off-road vehicles fall short of performance ambitions. Despite advances, autonomous ground vehicles are incapable and too unreliable to be used broadly. As robots have been developed for higher speeds, the basic form of navigation algorithms remains unchanged, with only incremental improvements to take some vehicle dynamics into account. These approaches are fundamentally limited by the sensor information available for decision making and the overly constraining requirement for absolute safety.

While driving, humans necessarily make assumptions about their world (e.g., the road will not end without warning, road curvatures will be reasonable for anticipated travel speeds, roads are free of obstructions). When these assumptions are violated (e.g., the trail around a blind corner is blocked by a fallen tree), the brittleness of the system becomes apparent, even human drivers crash. Even so, drivers make these assumptions because high-speed driving is not feasible otherwise. Research on autonomous highway vehicles incorporates many of the same assumptions, for the same reason: to make the driving problem tractable [Thorpe90, Thorpe97]. In contrast, there has been little or no work in effectively using prior models of the world (i.e., assumptions) in off-road autonomous navigation.

This thesis explores the space of off-road navigation and provides a novel and useful division of the off-road navigation problem into three regimes (efficiency-limited, stop-limited and swerve-limited), differentiated by the relationship between sensor range and robot speed. The swerve-limited regime enables a new level of performance in high-speed off-road navigation. The requirements

and implications associated with operation within these regimes will guide research to improve robotic performance.

## **Motivation**

---

Robotic vehicles have long been limited by their inability to perceive terrain at a range sufficient to allow for safeguarding when moving at high speeds. One approach is to use prior maps to alleviate sensing requirements, but conventional wisdom holds that such maps are too coarse for an off-road mobile robot to utilize, and even if they were sufficiently detailed, pose estimation is too inaccurate for this information to be readily usable.

In many application domains, conventional wisdom no longer pertains. Satellite and airborne imaging now yield maps of exceptional quality. Imagery with sub-meter accuracy and resolution can now be acquired at relatively low cost. Advances in data storage and communication mean that this information can be made available to robots before, and while, they operate.

Advances spurred by the consumer market for position sensing are generating high-accuracy, low-cost position estimation devices. The combination of GPS receivers with inertial measurement units is generating a new breed of highly robust, accurate and precise pose estimation systems [Scherzinger00] that are more than sufficient for registration with high-resolution maps.

This thesis demonstrates that new approaches that take advantage of advances in mapping and localization can lead to robots with unprecedented autonomous navigation capability.

## **Applications**

---

Fast and reliable autonomous navigation is an enabling technology for robotic applications in agriculture, transportation, military and exploration. There is a broad variation in requirements for speed and the degree to which risk associated with navigation errors can be accepted or mitigated.

## ***Agriculture***

Tilling, harvesting and spraying are all areas where autonomous farm equipment could decrease labor cost while increasing productivity. Precision farming (the combination of global positioning information, satellite imagery and computer controlled spraying) is becoming more prominent, but still requires a human in the loop. This infrastructure provides a great initial set of capabilities. Robots are beginning to see use in farming, but though farm vehicles move at relatively low speeds, they push the limit of state-of-the-art navigation performance. An improvement in both navigation speed and robustness is essential for the widespread adoption of autonomous farming equipment.

Given sufficient navigation capability, it will be possible to remove operators and farmers from the fields during robotic operation, decreasing the risk associated with any navigation error in the autonomy system.

## ***Transportation***

In forestry and mining, it is often necessary to transport materials cross-country or over poorly prepared roads. Long hours and difficult operating conditions can result in operator fatigue and error. Higher speed navigation directly translates into increased productivity. Furthermore, in remote locations the risks and liabilities associated with automation can be minimized by removing people from these hazardous environments.

## ***Military***

The military need for autonomous vehicles is broad and self-evident. During deployments, supply lines can stretch for tens to hundreds of miles, both on and off-road. Supply lines are difficult to defend and the vehicle operators are at high risk of enemy action.

There is a further desire to limit the need for human operators in positions where they would be exposed to direct fire. Existing technology does not allow autonomous vehicles to travel at speeds necessary to replace the capability

provided by manned vehicles. Advances in unmanned aerial vehicles (UAVs) provide a method for generating up-to-the-minute maps of operational areas. By effectively utilizing maps and other information available from UAVs, a tremendous improvement in unmanned ground vehicle performance can be realized.

In a military operation, speed is often of the essence and significant risk may be acceptable.

### ***Exploration***

The exploration of remote locations on Earth and other planets benefits from the ability to move around. A lander at a fixed location has a limited sensing horizon and has limited ability to take a “closer look” at an interesting rock or specimen. The benefits of a mobile exploration base are thus widely recognized. For example, had the 2003 Mars Exploration Rover missions deployed landers instead of mobile rovers, the conclusive signs of surface water would probably have been missed.

Future missions to explore, study and gather resources from other planets require greater robotic reach and thus compel higher speeds. Even so, the tremendous cost of undertaking these missions will require a very safe, risk averse navigation strategy.

On planets and moons, robots may employ a sun-synchronous navigation technique to acquire the solar energy necessary to sustain exploration for extended periods of time [Whittaker00, Wettergreen01]. There are other potential benefits, such as ensuring visibility and adequate lighting, and moderating thermal extremes by planning solar exposure. Sun-synchronous navigation could enable a persistent presence to explore, dwell and develop regions of planets and moons. With sufficient navigation performance, these missions would provide tremendous opportunities for exploration and science.



While autonomous navigation is a common requirement for each of these application domains, it is hard to imagine a single navigation strategy working well across all of them.

## **Problem Statement**

---

The interplay between onboard sensing and a priori information is not rigorously understood. This lack of understanding limits the ability of researchers to engineer sensors and algorithms appropriate to the challenges of robot performance. The thesis addresses is how sensing requirements change with speed and the availability of prior knowledge.

## **Thesis Statement**

---

This thesis asserts that off-road navigation regimes may be distinguished by the relationship between robot speed and required sensing horizon for safe operation. Furthermore, within a particular regime, it is postulated that particular formulations of navigation algorithms will be preferred.

For the purposes of this thesis:

Off-road is defined as terrain including unpaved roads, trails and areas with no discernable track.

Navigation regimes are defined as subspaces of the navigation space with consistent characteristics

Sensing horizon is defined as the maximum distance at which it is possible to perceive an obstacle.

Safe is defined by the requirements of the application.

## **Preview**

---

This research applies a methodology of simulation, analysis, testing and implementation to distinguish, characterize and confirm navigation regimes.

This research explores some of the critical relationships that drive the design of autonomous off-road robots. In particular this research explores how robots detect obstacles, avoid them and navigate efficiently.

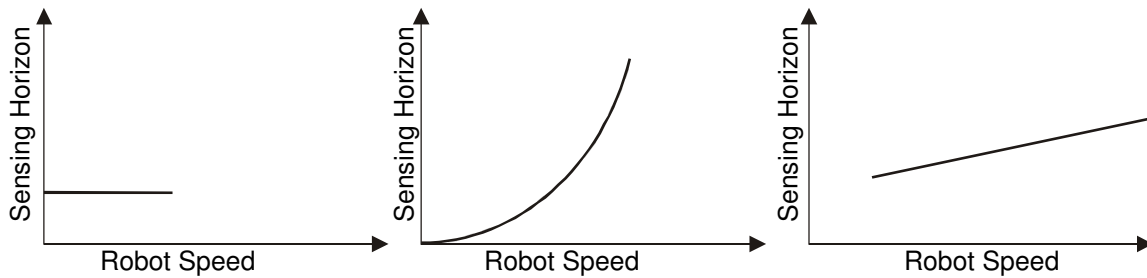
Mobile robot navigation aims to detect obstacles and avoid them while moving efficiently between locations. Detecting obstacles involves combining sensing and prior knowledge to understand what portions of terrain are not traversable. As sensing horizons increase, the difficulty of building sufficiently accurate world models also increases. Difficulties associated with increasing sensing horizons include the inability to measure range accurately and the inability to point sensors in a direction that is useful in building models. An understanding of these relationships is critical in describing the nature of off-road navigation.

A central concept in autonomous navigation is that of *panic stopping*. A principal element of high-speed navigation philosophy is that robots must utilize sensing horizons that allow for stopping whenever some unexpected obstacle is detected. An alternative is to avoid obstacles while moving by swerving. Generally, this does not alleviate the need to maintain a sensing horizon that is always greater than the stopping distance, since a robot may be trapped in a situation where no avoidance maneuver is possible.

In cases where prior maps and domain knowledge may be utilized a robot may plan around existing obstacles proactively and use an understanding of the expected size of unforeseen obstacles to set a travel that is both faster than the speed required by stopping distance constraints but slow enough that worst obstacle avoidance is possible.

A fundamental goal of navigation is to get somewhere efficiently. Much research has focused on information-optimal approaches to planning and replanning as new sensor measurements become available. One question that has remained only partially answered is how far should sensors point to produce the best paths? The form of this relationship is also an unknown. Confusing the issue further is how prior information impacts the selection of sensing horizon.

These three aspects of navigation -detecting obstacles, avoiding them, and navigating efficiently- lead to simulations and analyses that, when combined, argue for three navigation regimes. The navigation regimes are distinguished by the relationship between required sensing horizon and speed. Terrain and perception capabilities determine the transition between regimes. Within a regime certain algorithmic approaches are arguably more effective for achieving navigation performance.



**Figure 1. Fixed, quadratic and linear sensing horizons.**

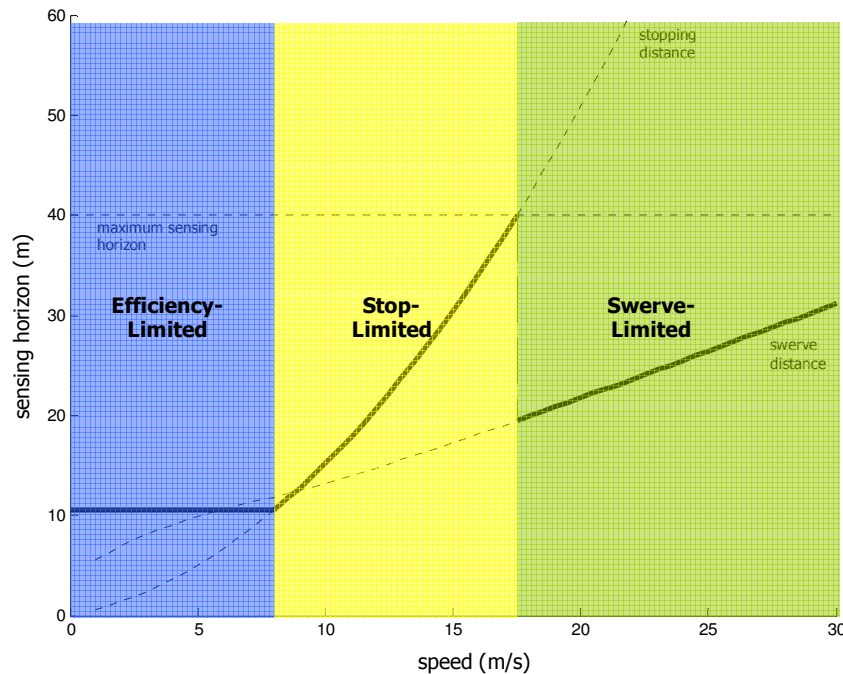
In the ***efficiency-limited regime***, required sensing horizon is a function only of the terrain and desired path optimality. It will be shown that a fixed sensing horizon can be determined as a function of terrain. This result is a key observation of this research. Planetary exploration rovers operate almost exclusively within the efficiency-limited regime. Because of the extremely low navigation speeds, the sensing horizon is always much larger than the stopping distance is primarily driven by the desire for efficient and safe navigation.

In the ***stop-limited regime***, sensing horizon is determined principally by stopping distance requirements. To maintain the ability to panic stop for obstacles the sensor horizon must increase quadratically with speed.

In the ***swerve-limited regime***, high performance is achieved by swerving instead of stopping, allowing the sensing horizons to be shorter. It will be shown that using swerving as an avoidance maneuver results in a linear growth in sensor range, and thus dramatically increases the speeds possible for autonomously navigating mobile robots.

Figure 1 illustrates the three relationships between sensing horizon and speed. Figure 2 shows the composite of these curves with the regimes labeled. The details and implications of these regimes, including the implications for navigation algorithms, are discussed throughout this document.

While the regimes are distinct, there are design freedoms that influence the location of the transitions between regimes. The transition to the swerve-limited regime can be manipulated by the selection of sensors and prior map resolution. For instance, by increasing the quality of a priori data (in terms of resolution, timeliness and accuracy) the size of unexpected obstacles in the world decreases. As the size of unexpected obstacles decreases, the distance at which they must be perceived in order for the robot to avoid them also decreases, shifting the transition point between the swerve and stop limited regimes.



**Figure 2. Composite of sensing horizon curves.**

## Outline

---

Some regimes in autonomous navigation and related work in the field of off-road navigation are introduced in the next chapter. A chapter on methodology outlines the approach to simulation, analysis and experimentation used in performing this research.

The next sections focus on elucidating the underlying relationships that distinguish separate regimes. A mathematical description of stopping, swerving and sensing suggests swerving instead of stopping as a strategy for safeguarding. The chapter on sensing horizons and prior maps reports the results of simulation experiments that lead to a relationship between these two factors and navigation efficiency. Navigation regimes are then distinguished and described in detail.

The efficiency-limited and swerve-limited regimes are next contrasted through two prototypical robotic implementations. This is followed by a description of the approaches, algorithms and mechanisms used in each approach.

The thesis concludes with a summary of the viability of navigation at speeds greater than those dictated by stopping distance constraints and the non-linear relationship between sensing horizon, prior maps and navigation efficiency.

---

## 2 Background

---

Within the field of robotics many point approaches to navigation have succeeded, but general regimes of navigation have not yet been codified. There have been few examples of high-speed, off-road navigation. This chapter presents some of the past attempts at partitioning the space of robotic navigation and provides a context for the research, technology and autonomous robots described in this document.

### Regimes in Autonomy

---

A common distinction is made between indoor and outdoor mobile robotics. Indoor robotics is generally characterized by discrete obstacles, flat terrain and simplified sensing conditions. Outdoor robotics is characterized by a continuous range of terrain costs, irregular terrain and challenging sensing conditions. Understanding which of these regimes robots are to operate in has direct implications to the viability of sensing approaches, world model representations and mechanisms. Even though the region labels seem clear and descriptive, there is no set of formulae or numerical ranges that may be used to clearly define indoor vs. outdoor robotics. For example, designing a robot to operate in a building with large windows (which provide a large dynamic range of lighting conditions) or irregular, curving walls (which complicates map based representations) blurs the boundaries between these regimes.

Similarly, autonomous on-road driving literature provides a distinction between highway and urban regimes [Aubert90]. The highway regime is marked by high speeds and predictable, near steady state environment. The urban regime is characterized by lower speeds, unpredictable vehicle motions and a highly variable environment. Approaches to highway driving utilize knowledge of the regime to reduce state [Pomerleau95 Dickmanns86]. In most highway driving examples, the state of the system can be reduced to the current speed, lateral offset within the lane and distance to any object in the lane in front of the vehicle.

In contrast, more state must be maintained in the urban regime. It is necessary to track the position and intent of vehicles and pedestrians in the environment, to predict and avoid collisions along with other state necessary to navigate through a city road grid. The urban regime has had few if any robotic successes, with the current research focus on understanding as opposed to acting within these environments [Thorpe02]. Like the indoor/outdoor regime distinction, the definition of highway and urban driving cannot be crisply defined.

While the previous examples provide useful short form to help describe requirements, they lack crisp definitions of their regions of applicability. There are explicitly defined regimes within the field of robotics. For example, there are regimes of kinematic and dynamic turning [Kelly95]. In this case, the regimes are distinguished based on turning stability. In the kinematic turning regime, robots are safely able to turn at their kinematic limits. In the dynamic turning regime slipping or roll-over limit safely achievable curvatures.

A second distinction is made between dynamic and kinematic braking regimes. When speeds are low and reaction times are large, robots operate within the kinematic braking regime; characterized by near linear braking distances. High speeds and short reaction times characterize the dynamic braking regime with its quadratic braking distance. Each of these relationships can be exactly defined by considering physical characteristics of the robot and terrain it is to operate on, and can be used during the navigation process to determine viable actions, but do not provide broad insight into the design of a robot.

Regimes have been used to select control modes. For example, loose classifications of terrain types have been used to select between algorithms for navigation [Lacroix02]. In this example, terrains are classified as easy or rough. Easy terrains have widely spaced obstacles and smooth load bearing surfaces. Uneven terrain, where the limits of rover terrainability determine an obstacle, is considered rough [Hait96, Hait99]. While these classifications can be considered

regimes, they do not explicitly consider speed, and do not provide tools for assisting in the design of a mobile robot.

While regimes like laminar and turbulent flow, and static and dynamic friction are common in science, prior to this work no concerted effort has been put forth to determine regimes that have broad implications within the realm of off-road navigation. This thesis describes a set of regimes that are crisply defined and describe the interplay between speed, perception capability and prior knowledge.

## **Low-Speed Off-Road Navigation**

---

Off-road navigation is part of the space of outdoor navigation where robots operate on trails and terrain with less structure than is common in on-road navigation. Off road navigation encompasses driving on dirt roads, trails and untouched wilderness. In general, the problem encompasses large scales, sensing is difficult, and the terrain has wide variations in traversability properties.

There are many approaches to solving these problems. One approach is to ignore the complexity of terrain and apply conventional flat world planning approaches. Point-to-point kinematic algorithms plan between two positions or poses (positions and orientations). Algorithms of this form generally use no a priori route, and are only required to remain within a particular (generally arbitrarily defined) region containing the start and goal locations. These algorithms operate with a state vector that does not include any reference to velocity. There are many examples of point-to-point kinematic algorithms [Latombe91]. One provably correct and complete approach combines straight-line motions toward a goal with tangent graph based obstacle avoidance. One implementation, RoverBug [Laubach99], navigates testbed planetary exploration rovers at cm/s speeds through unknown terrain.

Fuzzy logic has been applied to the off-road navigation problem [MartinAlvarez99, Seraji99, Seraji01]. While there have been some successes at low speeds, the applicability to high-speed navigation is questionable. The



inability to circumvent local minima further limits the applicability of these approaches.

Another family of algorithms vote among arc sets [Stentz95a]. Arcs are used to capture some of the non-holonomic constraints imposed on certain classes of robots. One example approach uses dense point cloud data to perform statistical plane-fits which are used to generate terrain evaluation in terms of roughness and slope [Simmons95, Singh00]. This algorithm, named Morphin, has been used with numerous robots and has been adopted and improved for operations on the Mars Exploration Rovers [Biesiadecki01, Goldberg02]. While these approaches work well at low speeds, they utilize a simple model of speed and steering latency and thus have limited applicability to high-speed navigation.

While such algorithms are well suited for avoiding local obstacles, global plans are generally developed utilizing efficient, A\*-like graph search approaches. Several methods have been developed to allow for the efficient reuse of the graph search tree when new information becomes available. D\* and its variations are one family of these approaches [Stentz94, Stentz95b]. A limitation of the standard implementation of the D\* algorithm is its inability to represent the non-holonomic constraints imposed by many vehicles. A recent implementation greatly reduces this limitation by using a lattice of poses as a search space instead of a regular grid [Kelly04].

On challenging terrain, it becomes important to consider both kinematics and dynamics (or kinodynamics) to ensure safe traverses. There are several examples of approaches that use physics based planning [Amar93, Chancelou96b, Shiller91]. One approach utilizes a hybrid particle/rigid body dynamics model to describe a robot [Cherif99a]. The model captures the inertial properties necessary to understand how a robot behaves as it traverses complicated, varied terrain. Limitations of this approach include the need for complete knowledge of the terrain and that obstacles must be explicitly

represented. The complexity of the model currently makes the planning cycle too slow for real-time operation.

Another approach is to use genetic algorithms in combination with detailed robot models to generate feasible trajectories over terrain [Farritor98]. This approach is able to use articulated linkages to balance a robot while traversing complicated terrain. While well suited to slow speed traversal, planning time make it inappropriate for high-speed navigation. The approach also makes no guarantees that a solution will be generated, a problem for time sensitive requirements imposed by high-speed navigation.

Randomization is another approach to generating feasible paths with kinodynamic constraints [Barraquand90, Kavraki96, Brock01]. Rapidly exploring random trees [LaValle00, LaValle01, Urmson03] provide an efficient method for generating feasible trajectories through the action space which describes the potential motions of the robot. These planners are probabilistically complete and can converge rapidly if a bias is used to direct the search. It may be possible to achieve the efficient search characteristics of the RRT without randomization [Branicky02].

## **High-Speed Off-Road Navigation**

---

Many of the low-speed approaches have been adapted for high-speed navigation. As an example, arc search methods can be extended to account for the steering dynamics and delays associated with a real vehicle [Kelly95]. Instead of planning over arcs, clothoids are utilized to model robot paths with finite steering slew rates. This approach has been shown to drive an autonomous HMMWV at moderate speeds off-road.

One of the early, important steps towards high-speed off-road navigation was the use of path tracking algorithms to blindly drive pre-recorded routes. These algorithms are unable to correct for errors in the pre-recorded routes and are limited in their performance by the accuracy of the onboard position estimation

system. The control law nature of these approaches allows for analysis that can produce error bounds and allow for detailed characterization. Path following with simple safeguards has been utilized to drive large mining dump trucks at high speeds in simple, controlled environments [Singh91].

One common approach to path following are pure pursuit algorithms [Coulter92, Amidi90]. Given a route to follow, the algorithm calculates a look ahead point at a speed appropriate distance in front of the robot. A circular arc is then calculated to connect the current vehicle pose to the look ahead point. Convergence is dependant on iterative, partial execution of these arcs and there are various ways that control “gains” can be applied to adjust stability and fidelity.

Within the pure-pursuit framework, there are a variety of different approaches that have been used to generate instantaneous actions to reach the path. Beyond fixed curvature arcs, one approach is to generate higher order polynomial paths [Singh89].

Pre-computed clothoid trajectory sets can be used to achieve high-speed navigation in some terrains [Coombs00]. A large set of the available driving maneuvers are tested against obstacles perceived in the terrain. Any trajectories that intersect obstacles are then vetoed and the best remaining trajectory is selected. This approach has driven a HMMWV at approximately 10m/s through simple off-road terrain. This approach is only viable in smooth, benign terrain with large, obvious obstacles.

A new approach is to use statistical models of vehicle/terrain interaction, enabling high-speed off-road navigation [Spenko04]. Instead of modeling robot interaction with terrain at execution time, a statistical model of viable speeds and curvatures, called an admissible trajectory space, is generated a priori. At execution time, a high-level pre-planner is used to generate a nominal trajectory through terrain. The speeds and curvatures resulting from this planning trajectory are compared with the viable speeds and curvatures for the sensed terrain. If the nominal trajectory is viable, it is executed; otherwise a secondary search is performed to

generate a viable trajectory. One potential difficulty with this approach is generating the a priori statistical model for general vehicles. Furthermore, the sensing requirements necessary to build the terrain models used in this approach are daunting. This approach has only been demonstrated in limited simulation.

---

## 3 Methodology

---

Classical approaches to distinguishing regimes include empirical observation and methods for algebraic analysis. Methodologies like classification and clustering are not yet appropriate for distinguishing navigation regimes due to challenges in acquiring statistically significant quantities of performance data under controlled conditions.

The lack of a viable algebraic description of how a robot navigates and perceives off-road terrain prevents a purely algebraic analysis. Therefore this research undertakes an empirical methodology incorporating analysis, simulation and testing.

In turn we will consider each part of the navigation problem: Obstacle avoidance, obstacle detection and navigation efficiency. An analytic consideration of robot motion and sensing provides the boundaries of the swerve and stop limited regimes. Simulation is used to investigate navigation efficiency, characterizing its relationship with sensing horizon and prior map resolution. The combination of simulation and analysis yields the three regimes. Testing and observation of two experimental robotic implementations showcase ideas within the swerve-limited and efficiency-limited regime.

This chapter lays out the underlying assumptions used in performing the algebraic analysis of obstacle detection and avoidance and then turns to the process by which simulation was used to investigate navigation efficiency.

### Analysis

---

The characteristics of stopping, swerving and sensing can be modeled algebraically. For clarity, several simplifying assumptions are incorporated into the analyses. These assumptions include:

- Flat earth – all of the relationships assume that the robot is operating a plane perpendicular to the gravity vector. This assumption removes many trigonometric terms, simplifying the presentation here.
- Bicycle steering model – vehicle motion is approximated by a steered bicycle model. The motion of many wheeled vehicles can be approximated by this model, and hence the results of this work can be used broadly.
- Rigid suspension- the complexities of suspension loading in dynamic maneuvers are beyond the scope of this work. In single swerve maneuvers suspension dynamics have a limited effect on vehicle motion [Wong78] and are hence ignored here.

The mechanics of stopping, swerving, and sensing are described and the relationships and governing equations are incorporated into the analysis that leads to the distinction of regimes.

## **Simulation**

---

The relationship between sensing horizon, prior map resolution and navigation efficiency is previously not formalized. To learn this relationship, a simulation methodology is applied. Simulation is a viable means for generating data that characterizes off-road navigation. Simulation also provides controls not readily isolatable with real world experiments. For example, terrain can be generated with prescribed characteristics or sensors can be simulated as ideal.

Throughout the simulation experiments an abstract sensor model is used. Sensors measure the cost of traversing a piece of terrain directly. By sensing cost directly, simulation models the best possible robot response. Any decrease in the accuracy of sensing will result in less accurate data in the map, which can only cause the navigation algorithm to perform less optimally. In this way, the results presented here provide an upper bound on the expected performance of

a robot, given a particular combination of sensing horizon and prior map resolution.

Through simulation, parameters such as sensing horizon and prior map resolution are varied to evaluate their effects on navigation performance. Robots are modeled as occupying a point in a cellular grid and may move to adjacent cells. Robot motion is simulated with a sense and plan cycle between each step. This simulation is relevant to low speed robot motion as non-holonomic constraints and dynamics are not modeled. Simulations are performed over significant distances of thousands of cells.

### ***Scope***

Simulations are utilized to characterize the relationship between sensing horizon and navigation efficiency with and without prior maps. This relationship is verified under a variety of conditions.

In this research, simulation is used to study the impact of types of prior maps (pessimistic, optimistic or nominal) on performance data. Pessimistic prior maps consistently over estimate the cost of traversing a region. Optimistic prior maps underestimate costs, while average prior maps represent the nominal expected costs of the terrain. Real terrain is used in these experiments to ensure that the results are representative of behavior seen in the real world. Simulation on synthetic terrain is used to provide a larger number of samples to confirm conclusions.

### ***Cost Maps***

Cost maps are a regular cellular decomposition of terrain where each cell contains an abstract “cost of traversal”. Figure 3 shows a representative cost map generated from synthetic terrain. In this research, cost maps are used both as the world and as the model of the world utilized by a navigating robot. This simplification abstracts the sensing process. When used, prior maps are generated by down-sampling the full resolution cost map. The method by which

the maps are down-sampled is dependant on the experiment being performed, and is described later.

### ***Terrain Processing***

In several simulations, cost maps were generated from real world digital elevation models. Six different maps from the Mojave Desert are used as the basis to generate real world cost maps (see Appendix A). To convert from a digital elevation model to a cost map, the map is convolved with an average slope filter:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & 1 & \frac{1}{\sqrt{2}} \\ 1 & -4 - \frac{4}{\sqrt{2}} & 1 \\ \frac{1}{\sqrt{2}} & 1 & \frac{1}{\sqrt{2}} \end{bmatrix} \left(\frac{1}{9}\right)$$

This matrix calculates the average over the slopes in each direction. The slope for each cell is then converted to an angle, using the arctangent function. Cost is then scaled between 0 and 255 based on a maximum traversable slope (defined to be 25°). The resulting cost map is smoothed using a 3x3 averaging function to reduce the impact of discrete contour steps in the elevation maps. This method of generating costs is used as it mimics the mapping of slope to cost used by several navigation algorithms [Simmons95, Tompkins01].

Figure 3 shows a digital elevation model and the resulting cost map generated from that model. In the left figure, height is represented by gray scale intensity with dark gray representing low elevations while light gray and white regions represent high elevations. A valley cuts diagonally across the left half of the terrain. The cost map is shown in the right image. Here, black cells represent terrain with a low slope and hence cost of traversal. Brighter cells represent areas where it would be difficult (high cost) for a robot to traverse.



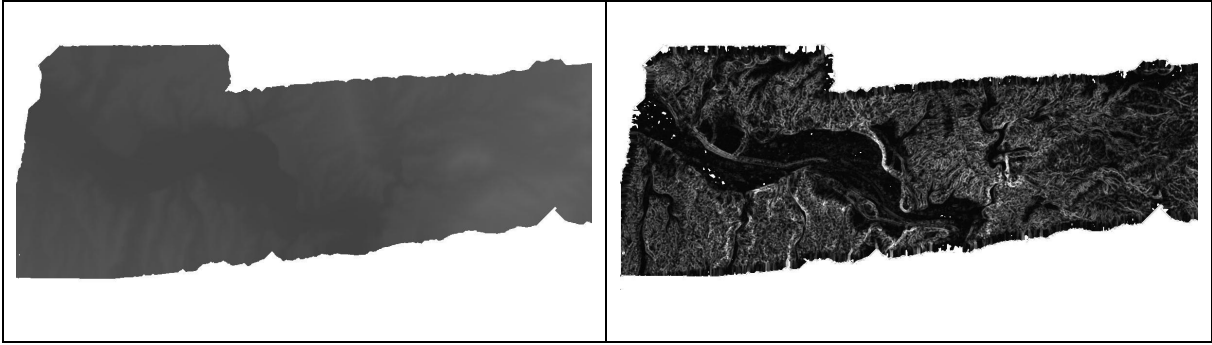


Figure 3. A comparison of the digital elevation model and cost map.

### ***Synthetic Terrain***

To provide statistically significant results, and to confirm findings from a small set of natural terrains, simulations were performed on a large number of fractally generated terrains. These terrains are generated using the diamond-square technique commonly used in generating synthetic digital elevation models [Martz96]. The diamond-square approach provides a parameter that effectively controls the frequency of the terrain. Figure 4 shows a representative terrain generated with this approach.

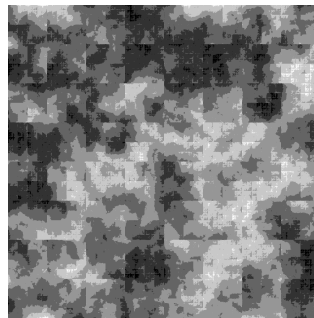
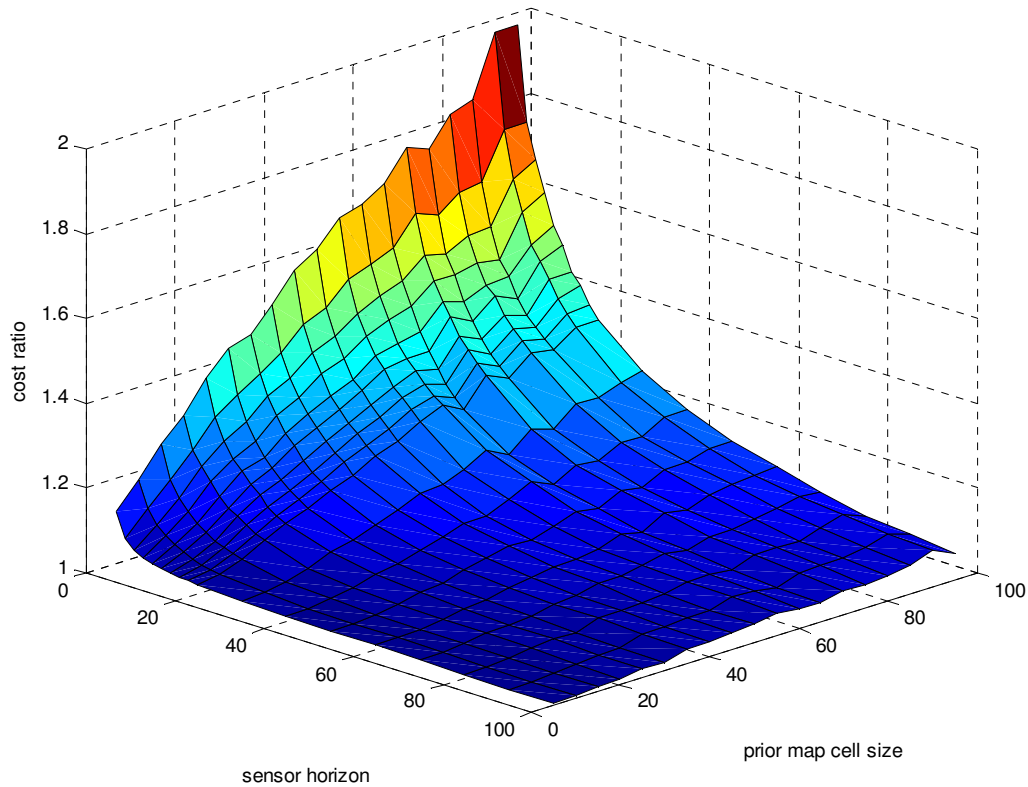


Figure 4. Representative fractally generated terrain.

### ***Summary Surfaces***

To present the results of these simulations, a standard plot relating performance to prior map resolution and sensing horizon is used. This plot will be referred to as a *Summary Surface*. The height of the surface represents the ratio between the path generated with a given configuration (of sensing and prior map data) and the cost of the optimal path. Configurations that perform well will have a

ratio approaching or equal to one, while poor configurations have larger values. The use of a ratio is important, since it allows results on different terrains to be combined and averaged with equal weighting. Figure 5 shows an example summary surface.



**Figure 5. Summary surface: navigation efficiency vs. prior map resolution and sensing horizon.**

## Testing

An example robot implementation from both the efficiency-limited and swerve-limited regimes is explored in depth to provide concrete examples of these regimes. These two implementations are suited to very different applications, highlighting the role of the different regimes.

With the process discussed, it is now time to turn to the relationships that describe how a robot moves and perceives the world.

---

## 4 Stopping, Swerving & Sensing

---

This chapter presents the relationships between speed, stopping distance, swerving, and sensing errors. Stopping is the most basic obstacle avoidance maneuver. The prevailing navigation philosophy is that for safeguarding, robots must sense at or beyond their stopping distance. This requirement quickly becomes difficult as robot speeds increase since as sensing range increases, error in measurements, spatial resolution and pointing sensors in the correct direction make building usable models difficult.

The requirement to sense only far enough to swerve around obstacles reduces the difficulty of the sensing problem, but cannot be used in isolation of other safeguarding. Without some prior knowledge of the operational environment, it is impossible to make guarantees about the amount of swerving necessary to avoid obstacles and thus a swerve-only strategy would be unsafe.

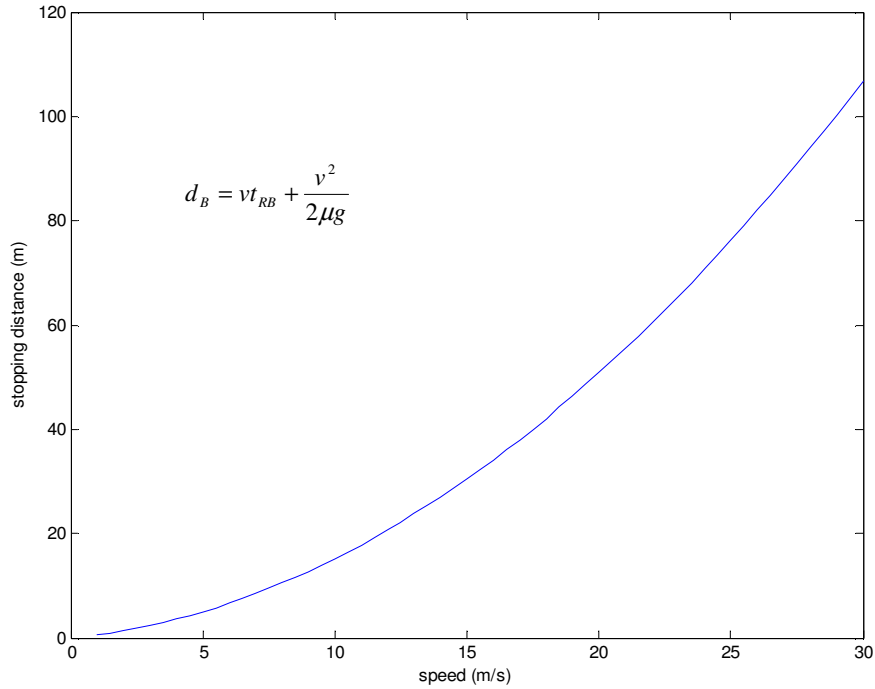
### Stopping Distance

---

At a minimum, robots must be able to detect obstacles and stop before reaching them. Stopping distance is principally a function of speed ( $v$ ), braking reaction time ( $t_{RB}$ ) and the coefficient of friction ( $\mu$ ). Stopping distance can be expressed as:

$$d_B = vt_{RB} + \frac{v^2}{2\mu g}$$

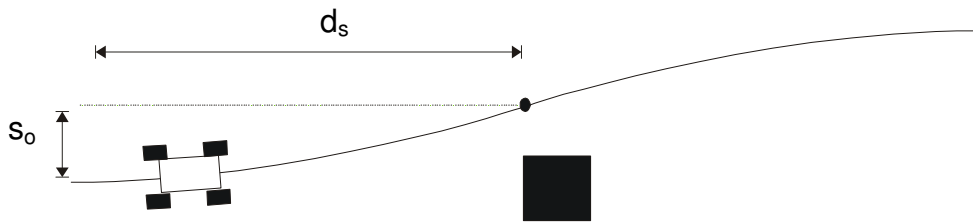
With modern computation and actuation, reaction time is generally on the order of 0.5s, and thus at relatively low speeds (i.e. 5m/s) the dynamic braking term begins to dominate. Figure 6 shows the stopping distance for coefficient of friction of 0.5.



**Figure 6. Stopping distance as a function of speed.**

## Swerve Distance

As an alternative to stopping, it is possible to swerve around many obstacles, and as will be shown, it is preferable in terms of required sensing range, particularly at higher speeds.



**Figure 7. Swerve maneuver.**

Swerving is a double steer maneuver used to avoid obstacles. The robot begins and ends the maneuver with the same heading but with a lateral offset from its initial position. Figure 7 illustrates this maneuver. Given a required swerve offset ( $s_0$ ) the required swerve distance ( $d_s$ ) can be determined as a function of speed and the vehicle's constraints on steering. To determine this swerving distance, the equations of motion are reviewed for a mobile robot with position  $(x, y)$  and

heading ( $\psi$ ) traveling along a path with instantaneous curvature ( $\kappa$ ) and velocity  $v$ .

$$\dot{\psi} = \kappa v$$

$$\psi = \psi_0 + \int v(t) \kappa(t) dt$$

$$x = x_0 + \int v(t) \cos(\psi(t)) dt$$

$$y = y_0 + \int v(t) \sin(\psi(t)) dt$$

For most mobile robots, the steering slew rate is limited to some  $\dot{\kappa}_{\max}$ , while the steering actuator acceleration is generally fast enough to be considered infinite. For this analysis, we assume that the robot will operate at a constant velocity, so the  $v(t)$  term can be moved outside of the integral. Furthermore, the mathematics can be simplified by considering the case where the robot is traveling in a straight line, so the initial curvature is zero. This leads to an expression for yaw as a function of time:

$$\psi = \psi_0 + v \int \dot{\kappa}_{\max} t dt$$

$$\psi = \psi_0 + \frac{v \dot{\kappa}_{\max}}{2} t^2$$

With this expression for heading, it is possible to consider the equations for motion. For the purposes of this analysis, the robot will begin at the origin, with an initial heading of zero.

$$x = v \int \cos\left(\frac{v \dot{\kappa}_{\max}}{2} t^2\right) dt$$

$$y = v \int \sin\left(\frac{v \dot{\kappa}_{\max}}{2} t^2\right) dt$$

These equations can be numerically integrated to generate a forward model for robot motion.

A swerve consists of two phases. In the first phase, the steering is slewed at the maximum rate until it is time to reverse the curve, or until increasing the curvature would be unsafe due to sliding or roll over. After achieving the necessary swerve offset, the steering is reversed and slewed back to zero curvature. The curvature limits are important in determining the look ahead distance required for a safe swerve.

### ***Curvature Bounds***

Curvature bounds limit the rate of heading change per distance traveled. To ensure that a path is safely drivable, robots must not exceed speeds where centripetal force equals lateral frictional force. Given a speed, the path is limited to a curvature inversely proportional to the speed squared:

$$\kappa_{slip} = \frac{\mu g}{v^2}$$

A second concern is roll over. In this case, the moment generated by the reactive centripetal friction force acting at distance ( $h$ ) below the center of gravity, must be less than the stabilizing moment provided by gravity acting at half of the vehicle cross track ( $T$ ). At the instant of lift off, these two moments are balanced:

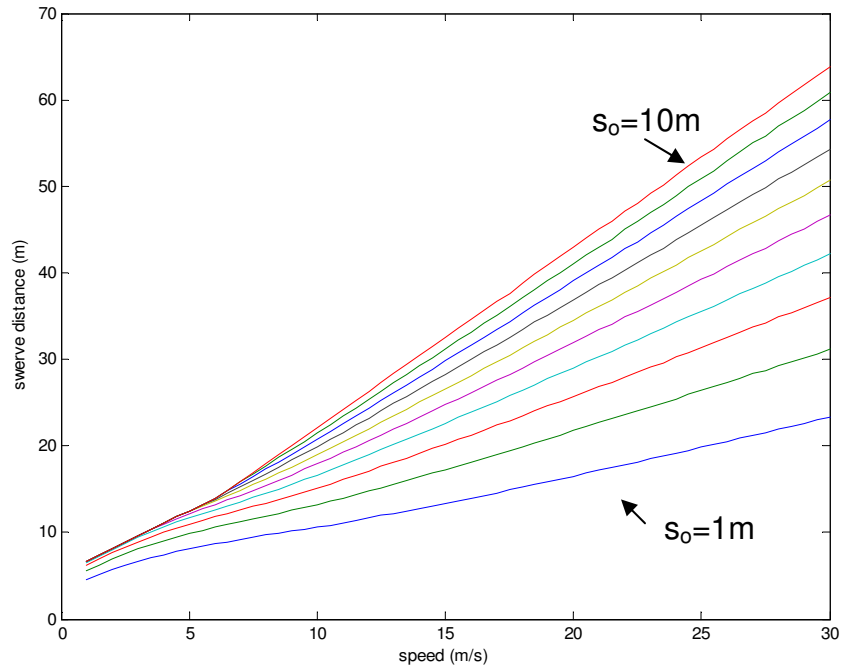
$$\frac{T}{2} g = h \kappa_{roll} v^2$$

After rearrangement, this yields an expression for the rollover curvature at a given speed:

$$\kappa_{roll} = \frac{T}{2h} \frac{g}{v^2}$$

Hence, the maximum curvature is limited by the lesser of  $\mu$  and the stability ratio  $(\frac{T}{2h})$ . For most robots on dirt trails, the coefficient of friction will be less than the stability ratio and thus the no-slip constraint is dominant.

With the curvature limits calculated, it is possible to determine swerving distances for a variety of obstacle widths. Figure 8 shows these curves for a robot operating with a coefficient of friction of 0.5, kinematic curvature limit of  $0.2 \text{ m}^{-1}$ , a lock-to-lock steering time (the time it takes to change from maximum curvature in one direction to maximum curvature in the other direction) of six seconds and a 0.1 second reaction time. This family of curves represents swerves around obstacles that require one to ten meters of swerve offset. An important characteristic of these curves is that they are roughly linear, as opposed to the stopping distance curves, which are quadratic. Therefore, if it is possible to assume that a robot will always be able to swerve around obstacles, and that guarantees can be made about the size of obstacles, it is possible to dramatically reduce the required sensing range.



**Figure 8. Swerving distance for 1 to 10m swerve offsets.**

With an understanding of possible avoidance maneuvers in hand, the discussion now turns to the detection of obstacles and the implications of speed on sensing performance.

## **Sensing at Range**

---

Increasing sensing range results in a decrease in the quality of data available for decision making. In this section, error sources in generating terrain models at range are described.

### ***Stereo Vision***

Stereo vision systems compare two or more images from roughly parallel cameras to compute a dense range field. Stereo algorithms consider the disparity (measured in the number of pixels) between images taken in each of the cameras. When pixel size and distance between cameras is known, disparity can be converted into a range to target. As range to target increases, range resolution decreases. For a pair of parallel cameras, with only an offset between them, the equation for range resolution ( $\Delta r$ ) is:

$$\Delta r = \frac{r^2}{bf} p \Delta d$$

In this equation, the range to the target is  $r$ , the cameras are separated by a baseline distance ( $b$ ) and have a specified focal length ( $f$ ) and imager pixel size ( $p$ ). The precision of the disparity measurement ( $\Delta d$ ) is a function of the stereo matching algorithm. If only whole pixel disparities can be measured then  $\Delta d=1$ , while if there is sub-pixel capability,  $\Delta d$  will be a fractional value between 0 and 1. Similarly, if the stereo algorithm operates on a decimated image, the value of  $\Delta d$  will be an integer greater than 1.

The footprint of each pixel spreads out linearly in each dimension, yielding a quadratic increase in overall size. The projection of the spherical expansion onto the ground plane results in a cubic increase in spot size. This cubic increase is



due to the combination of a linear increase in cross range pixel size and a quadratic increase in pixel size in the downrange direction.

### ***Line Scan LIDAR***

Unlike stereo vision range measurements, the accuracy of LIDAR range measurements do not vary significantly with range. The projection of the spot size onto a ground plane grows cubically with range, but in general, the spot size is so small that it is not a significant source of error. Due to the measurement techniques employed, range resolution is generally constant over the operating range of the sensor.

Within a horizontal line scan, the points spread linearly with range. Between scans, lines are separated by the product of vehicle speed and the period of the LIDAR scan rate. This yields a minimum scan rate, at a given speed, necessary to achieve a sufficient fidelity reconstruction of terrain

### ***Sensor Motion***

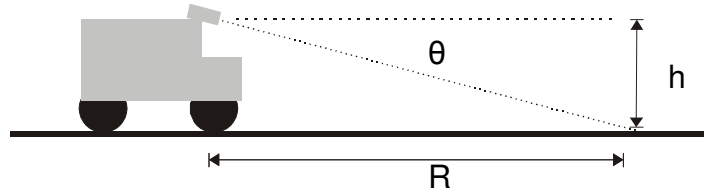
Sensor motion complicates the interpretation of range measurement data. While Line Scan LIDAR has many benefits over stereo vision, a major complication is the co-registration of laser stripes taken from different view points. While each stereo frame produces hundreds of co-registered scan lines, the transformation between each individual laser scan must be recovered to build a usable map. A major problem is ensuring that the spacing between line scans is sufficiently small to allow a reasonable fidelity reconstruction of the underlying terrain.

A constant look ahead is important since in combination with a fixed vehicle speed it provides a method to uniformly sample the terrain with a sensor (e.g. with a LIDAR line scanner). Uniform sampling is essential for detecting holes and other negative obstacles. In quasi-static terrain evaluation approaches, it may be sufficient to sample the terrain at only half vehicle lengths in the direction of travel [Kelly95]. This sampling allows for the evaluation of vehicle pitch, which is a key factor in vehicle stability. At high-speeds off-road, the terrain must be

sampled more frequently. A suggested sampling density is wheel radius ( $r_w$ ) [Spenko04]. To achieve this requirement, the terrain must be sampled at the nyquist rate and hence, the relationship between speed and required scanning frequency ( $f_s$ ) is:

$$f_s = \frac{2v}{r_w}$$

At this spacing, much of the wheel motion may be recoverable. Determining the correct sampling frequency is beyond the scope of this work. The focus instead is on the difficulty of maintaining a constant look ahead distance.



**Figure 9. Laser sensor geometry.**

Figure 9 illustrates the geometry of a laser sensor measuring level terrain. The relationship between look ahead distance ( $R$ ), height ( $h$ ) and sensor pointing angle ( $\theta$ ) is:

$$R = \frac{h}{\tan(\theta)}$$

In holding a sensor steady, the interesting relationship is the rate of change of look ahead distance with respect to changes in sensor pointing angle:

$$\frac{dR}{d\theta} = \frac{-h \sec^2(\theta)}{\tan^2(\theta)} = \frac{-h(\tan^2(\theta) + 1)}{\tan^2(\theta)} = -h\left(1 + \frac{1}{\tan^2(\theta)}\right)$$

The negative sign can be disregarded, since for this analysis, only the magnitude of the error is relevant. Rearranging and converting to a difference equation

generates the relationship between sensor pointing angle motion and change in range:

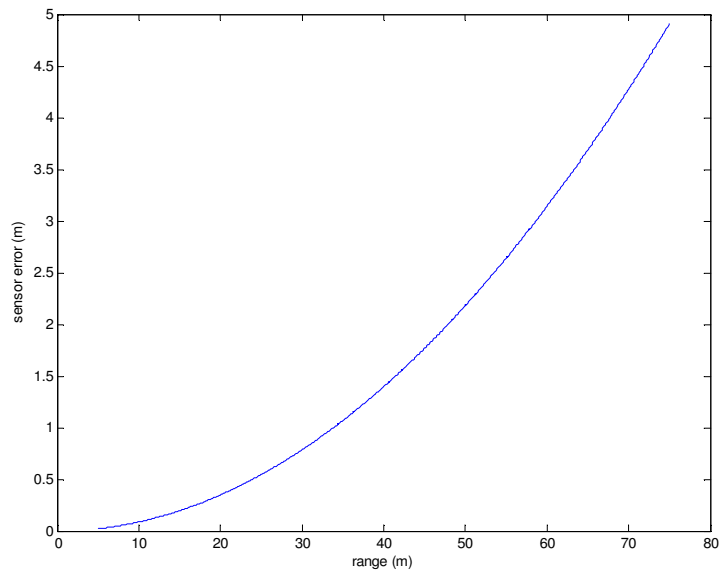
$$\Delta R = h(1 + \frac{1}{\tan^2(\theta)})\Delta\theta$$

Substitution is used to transform this equation to the relationship between sensor pointing error and look ahead range error:

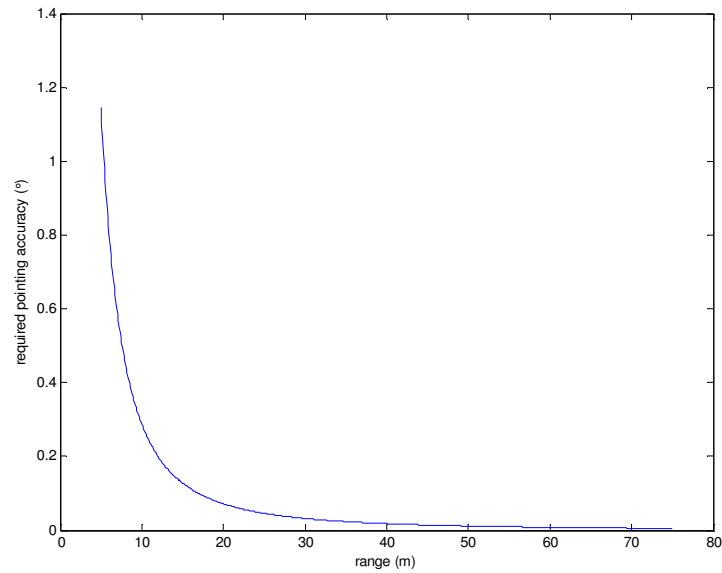
$$\Delta R = h(1 + \frac{R^2}{h^2})\Delta\theta \approx \frac{R^2}{h}\Delta\theta$$

Figure 10 shows a plot of this relationship for a perturbation of 0.1 degrees with a sensor height of two meters. Figure 11 shows the converse relationship; given a requirement of a maximum inter-scan spacing of 0.25m, the plot shows allowable sensor movement.

Though LIDAR scanners are immune to quadratic error growth in range measurement accuracy, the need to build maps with a fixed resolution introduces a different quadratic constraint.



**Figure 10. Sensor error as function of range.**



**Figure 11. Required sensor pointing accuracy as a function of range.**

## Sensing at Speed

The bounding sensing relationships (range resolution for stereo, and pointing accuracy for LIDAR line scanners) are quadratic in distance. When considered in relation to speed, variable substitution yields that these relationships are both quartic if sensing horizon is to match look ahead stopping distance. If instead, sensing horizon is increased only to meet the requirements for swerving, these relationships decrease to quadratic with speed. To highlight this idea, two examples are presented to illustrate how swerving instead of stopping can increase performance.

### ***Moderate Speed Robot with Stereo Vision***

As an example, consider a robot that uses stereo vision as its primary navigation sensor. Assume that its perception system requires range measurements with 10cm or less range error. The characteristics of the vision system are listed in Table 1.

System Characteristic	Value
Baseline	50cm
Focal length	11mm
CCD pixel size	7μm
Sub-pixel disparity	¼ pixel

**Table 1. Stereo vision characteristics for example implementation.**

From this description, the maximum useful range of the sensor can be calculated by:

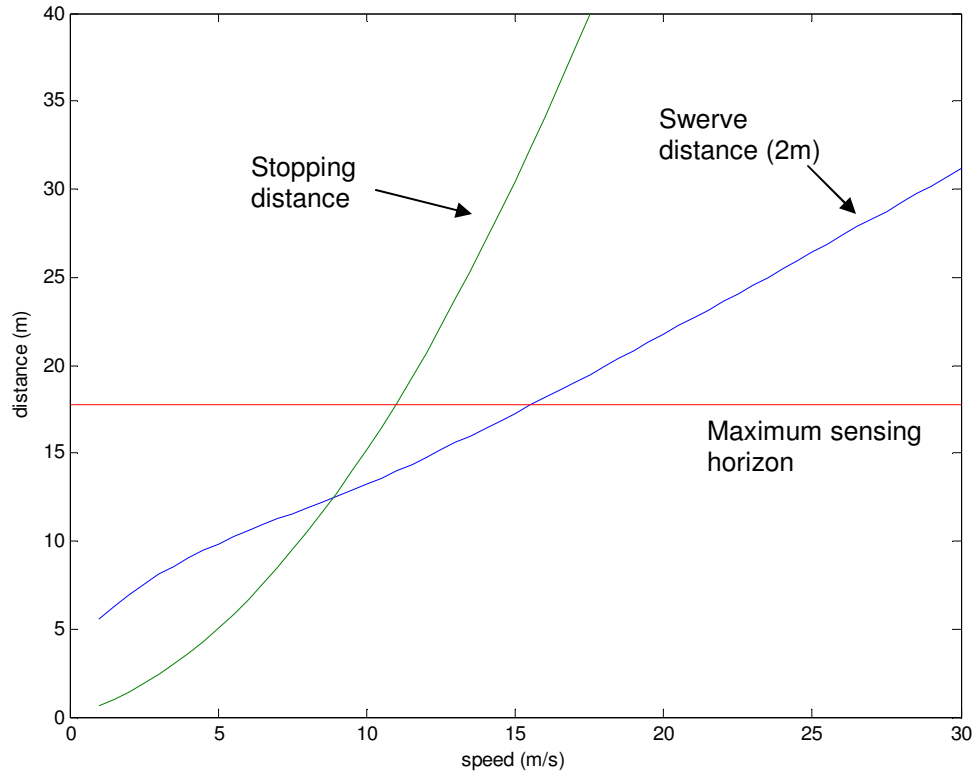
$$r_{\max} = \sqrt{\frac{bf\Delta r}{p\Delta d}}$$

For this configuration, the maximum viable sensing range is 17.7m. Operating within the stop-limited regime, performance is bound by the stopping distance equation. Applying the quadratic formula to the stopping distance equation yields a formula for maximum speed:

$$v_{\max} = \left( -t_{RB} + \sqrt{t_{RB}^2 + \frac{2d_B}{\mu g}} \right) \mu g$$

With a braking reaction time of 0.5s and a coefficient of friction of 0.5, this yields a top speed of approximately 11 m/s.

If there is sufficient prior information to allow swerving instead of stopping, and the largest swerving maneuver will have a swerve offset of 2m, the robot can safely attain speeds of approximately 16m/s. Figure 12 shows the relationship. In this example implementation, the maximum attainable speed would be increased by 45%.



**Figure 12. Maximum speed for stereo vision example.**

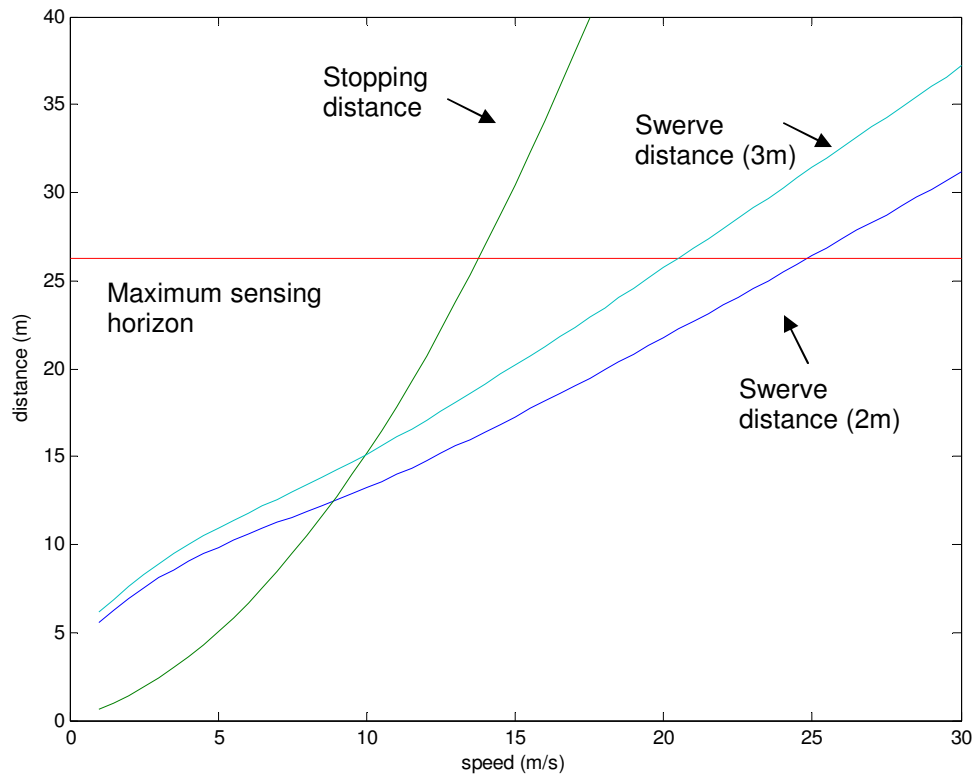
### ***High-speed Robot with Stabilized LIDAR***

As a second we consider a robot using a stabilized LIDAR system. This case is considered as an example where interline spacing is the dominant constraint, not range resolution. For this example, assume the stabilization system is capable of holding a pointing angle to within 0.05 degrees, and that the required inter-scan spacing for safe navigation is 30cm. Rearranging the sensor stability equation yields:

$$R_{\max} = \sqrt{\frac{\Delta R}{\Delta \theta}} h$$

The resulting maximum viable sensing range is approximately 26.2m, which results in a maximum speed of approximately 13.75 m/s with stopping distance based safeguarding. If instead, it is possible to operate in the swerve-limited

regime, it is possible to achieve speeds of 25m/s if the largest expected swerve offset is less than 2m or 20m/s if the largest expected swerve offset is 3m. Figure 13 illustrates this point.



**Figure 13. Maximum speed for LIDAR example.**

This significant difference between stopping and swerving is the key to enabling high-speed off-road navigation. In the next chapter the interplay between sensing horizons, prior map resolution and navigation efficiency will be explored.

---

## 5 Sensing Horizon and Prior Maps

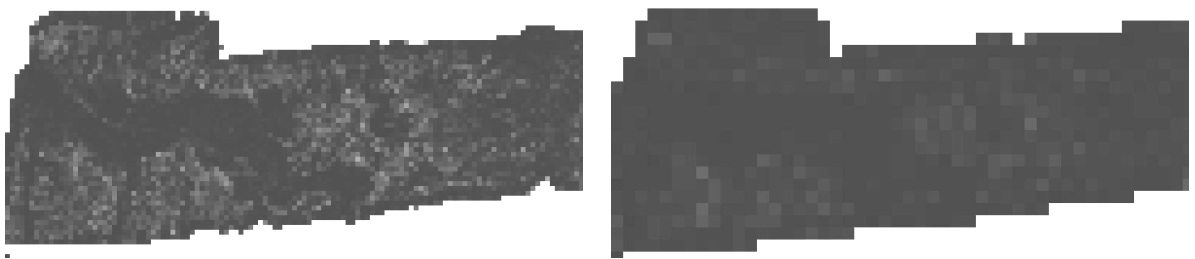
---

Mobile robots have two ways to understand the world: what is known ahead of time and what can be perceived while performing a task. It is widely accepted that by providing higher resolution prior maps and longer range sensing, performance will improve. In this chapter, these assumptions are evaluated, to better understand the relationship between prior information sources and navigation performance. The simulations presented here will show that as sensing horizons are increased the relative benefit (in terms of path optimality) diminishes.

### Sensor Horizon, Map Resolution and Navigation Efficiency

---

In this experiment, simulation is used to evaluate the performance benefits of increasing sensors horizons and increasing prior map resolutions. Each low resolution prior map cell is computed as the average of the underlying terrain. Figure 14 shows multi-resolution cost maps of a terrain used in this experiment. In these figures, cost is represented by brightness.





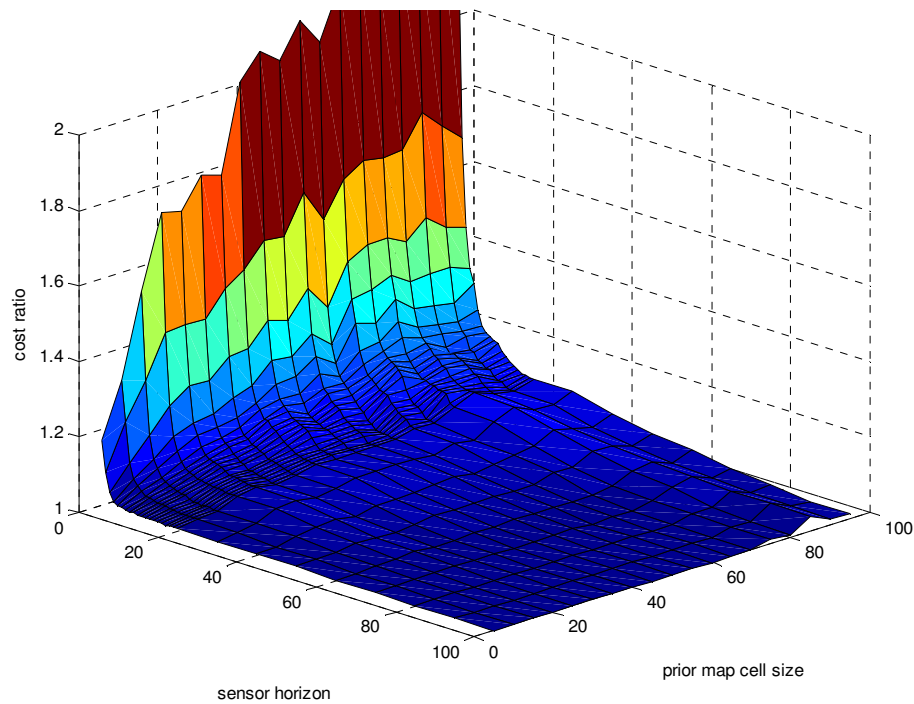


**Figure 14. Priori maps with decreasing resolution (10, 25, 100 pixel sampling).**

Over the six terrains from the Mojave Desert, there is a very clear, non-linear diminishing return relationship between sensor horizon and quality of path. Figure 15 shows a composite summary surface of results over these six terrains. The solution quality approaches the optimal marginally faster as prior map resolution increases.

Averaged over a significant number of related terrains, summary surfaces are smooth and decay at a rate approximately inversely proportional to the square root of the sensing horizon for a given prior map resolution. If generated for a single run, or a small number of runs, a summary surface may have valleys and steps. These steps represent sampling artifacts in prior map data. For example, a narrow path way, providing an efficient short cut, may be invisible below some resolution, and then become visible when higher resolution data is utilized.

The sharpness of the knee in this curve indicates that there is a point of sharp diminishing returns with regards to sensor range. This non-linear return is a key observation of this research and a key phenomenon of off-road navigation. Returns diminish so dramatically there comes a point where increasing sensor range is near irrelevant. This result is significant since sensing at long ranges is difficult in many applications. The remainder of this chapter expounds upon this point considering different types of prior map data, and experimenting on a number of relevant synthetically generated terrains.

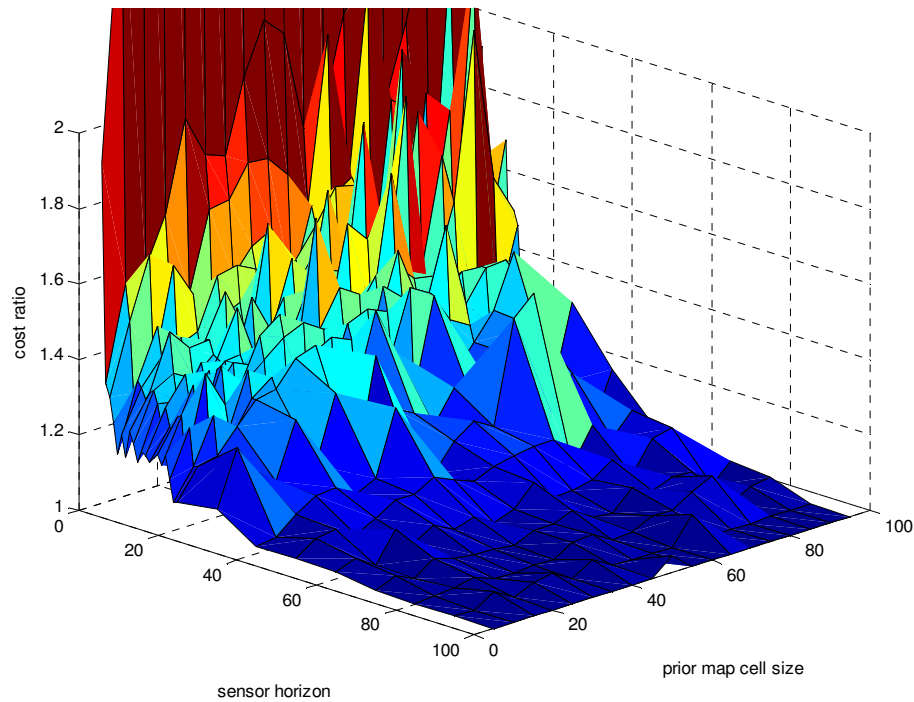


**Figure 15. Composite performance cost over six terrains.**

In the next pair of experiments, the impact of the type of prior map data (optimistic, pessimistic or nominal) is considered. This is an important consideration

## **Sensor Horizon, Map Resolution and Navigation Efficiency with Optimistic Prior Maps**

In this experiment, each pixel in the same six maps of a priori data is set to the value of the lowest cost pixel within its footprint. This results in a terrain model that underestimates the path cost through any portion of the terrain. Figure 16 shows a summary surface for this simulation. Note that the cost ratio axis is clipped to show surface detail.



**Figure 16. Summary surface for optimistic prior maps.**

The summary plot for this experiment is similar in shape to that of the previous experiment but is noisier and has a considerably longer tail. The spikes in this surface are statistical noise that will disappear with more samples leaving a summary surface that has consistently higher costs than the nominal cost summary surface. The increase in cost is due to two factors. The first is path retracing, where the robot is lured into cul-de-sacs and then “bumps” along the edge of high cost regions. The second is small scale inefficiencies. The minimum function masks higher cost cells within each prior map pixel, drawing the robot through regions that on average have high cost, but happen to have a single low cost pixel. Figure 17 illustrates a portion of a route where the robot is lured into a particularly large cul-de-sac.

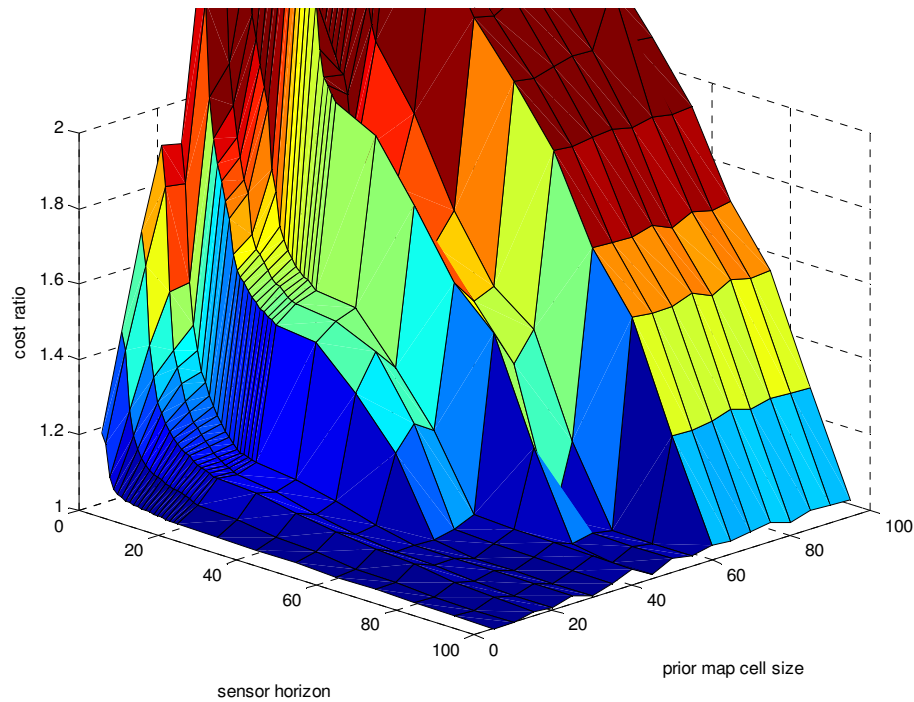


**Figure 17. Route path with path segment that doubles back highlighted.**

This experiment demonstrates that prior maps that represent the minimum costs per cell, as opposed to the nominal cost per cell, decrease the efficiency of planners.

### **Sensor Horizon, Map Resolution and Navigation Efficiency with Pessimistic Prior Maps**

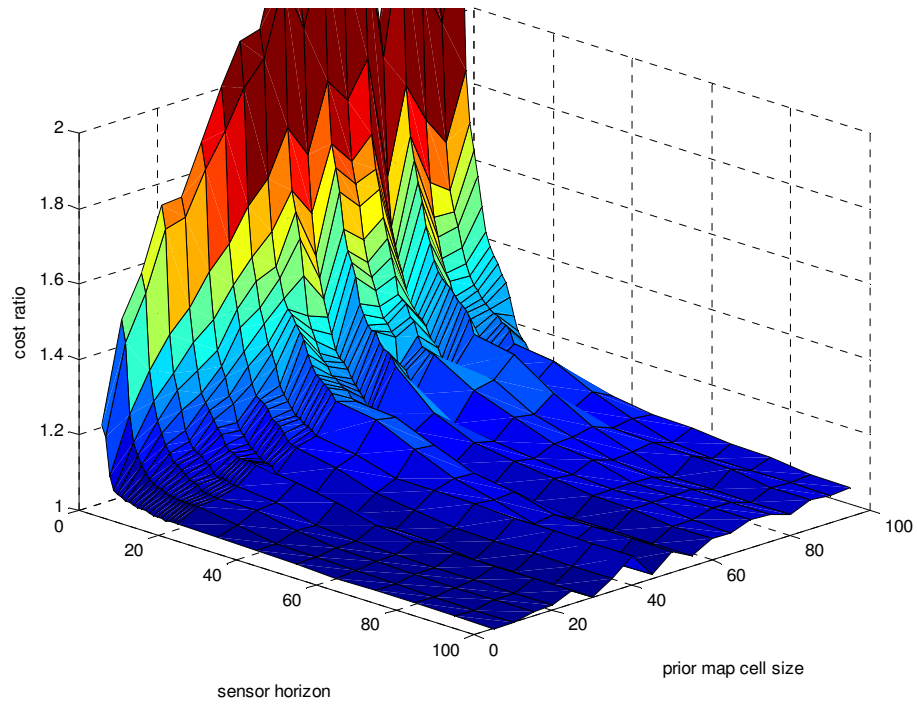
In this experiment, each pixel in the a priori map is set to the value of the highest cost pixel within its footprint. This results in a terrain model that overestimates the path cost. Figure 18 shows the resulting summary surface. For small prior map cells, the surface looks very similar to the nominal cost map, but as the prior map cells become coarser, the curve shape changes dramatically. This change in curve shape is due exclusively to the a priori terrain representation. On each of the terrains where the path is “dog-leg” in shape (see Figure 71, Figure 72, Figure 74 in appendix A), the planner cuts the corner, through the high cost area, since there is no obvious low cost route.



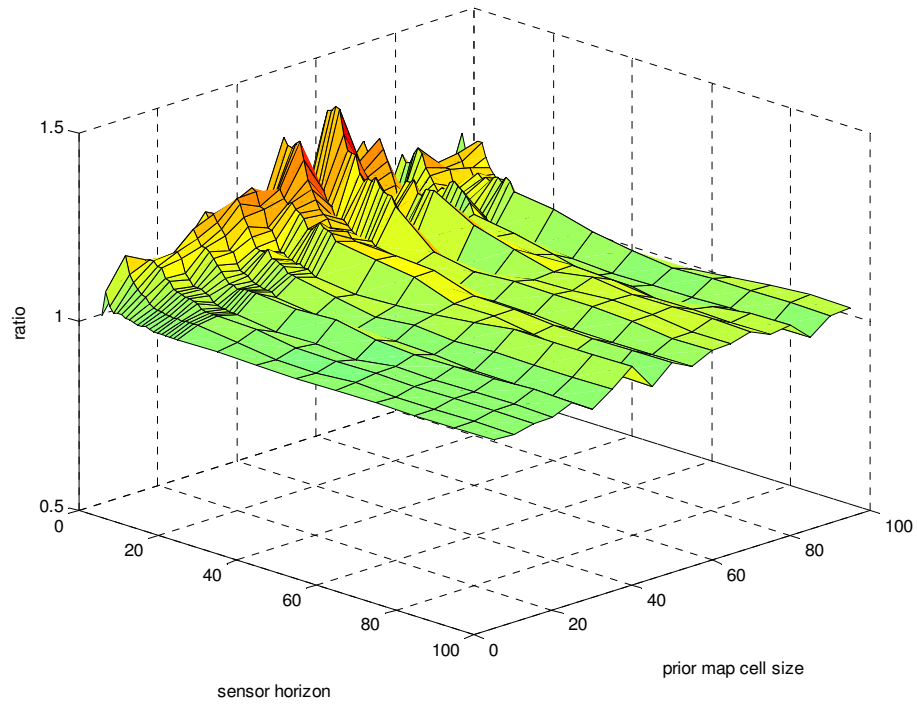
**Figure 18. Summary surface for pessimistic prior maps.**

By removing all of the “dog-leg” terrains, a new summary surface can be generated that illustrates the performance of the prior-map/sensor range data in the remaining terrains (Figure 19). In this case, the performance with the pessimistic prior map approaches performance with the nominal prior map but in general is worse. The ratio of these surfaces can be seen in Figure 20.

These experiments indicate that a terrain model that abstracts by averaging the underlying terrain will generally yield better performance than a pessimistic model.



**Figure 19. Summary surface for three non dog-leg terrains.**



**Figure 20. Ratio between nominal and pessimistic values in prior maps.**

## Sensor Horizon, Map Resolution and Navigation Efficiency on Synthetic Terrain

---

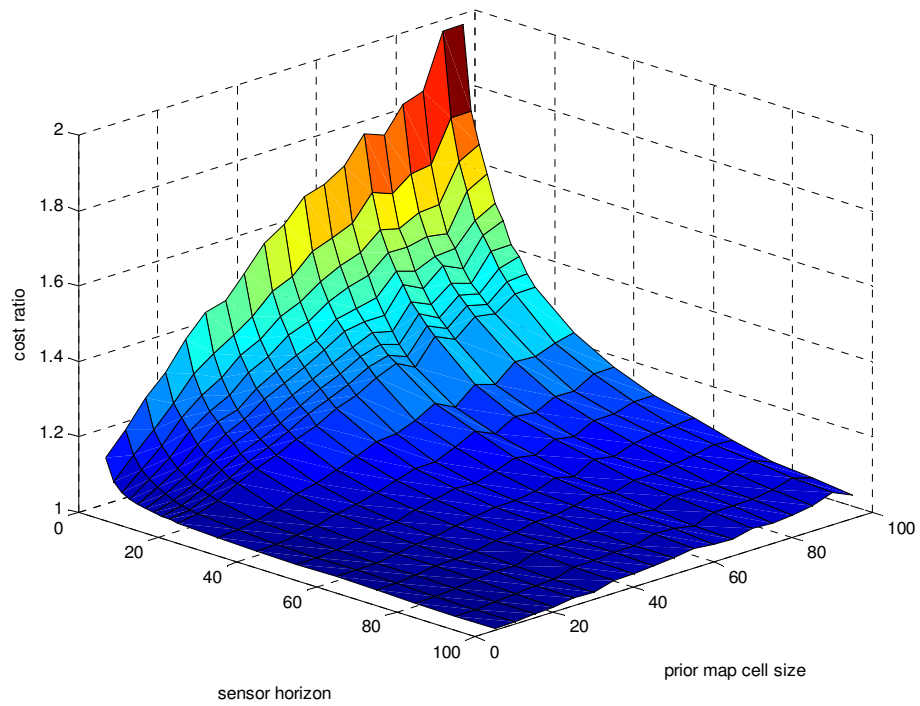
In order to provide statistically significant results, experiments were performed on a broader number of synthetic terrains. Two different sets of 90 terrains each are used. In the first set cost can vary 5 times more quickly over a given distance than in the second set. This corresponds to a more rugged and varied terrain. Figure 21 shows the summary surface for low frequency terrains while Figure 22 shows the summary surface for high frequency terrains. In both cases the surface has the same characteristic decay shape as in the previous experiments.

The characteristic shape of these surfaces can be approximately related to the sensing horizon ( $r_s$ ) and prior map cell size ( $d_c$ ) by the equation:

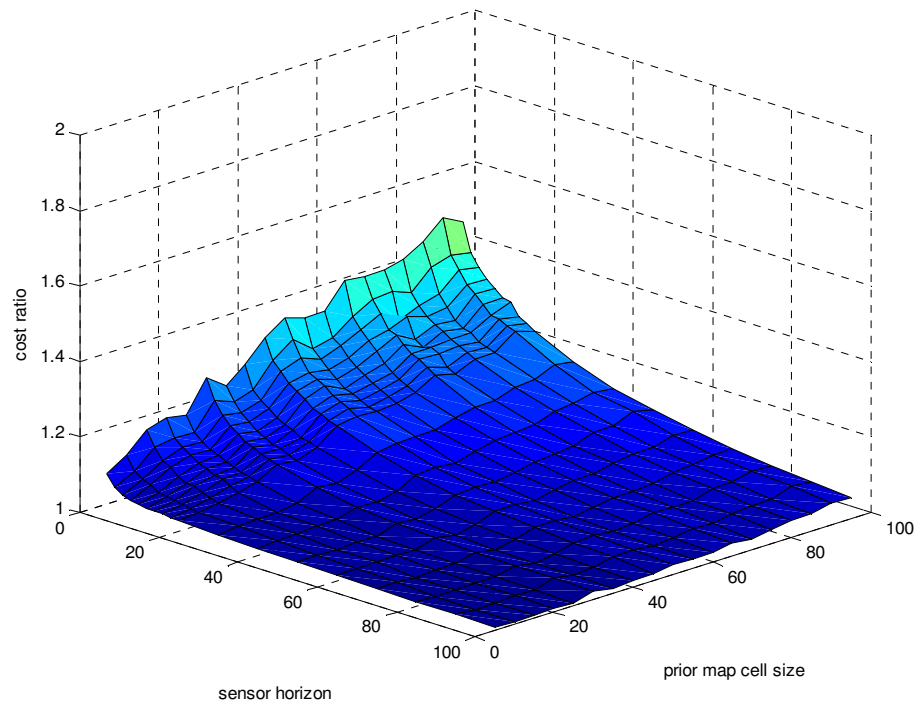
$$1 + \frac{kd_c}{\sqrt{r_s}}$$

In this equation,  $k$  is some constant based on the characteristics of the terrain. A least squares fit of this relationship to the summary surface shows good agreement. Figure 23 and Figure 24 show the ratio between the best least squares fit of the above equation and the summary surface for the two terrain sets. Note that the fit error in both cases is less than 5%.

The high frequency terrain shows a smaller ratio between the optimal path and the experimental paths. Since the terrain cost can vary quickly spatially, only small deviations from a course plan are necessary to achieve good navigation performance, and hence there is little benefit to a large sensing horizon or detailed prior maps.

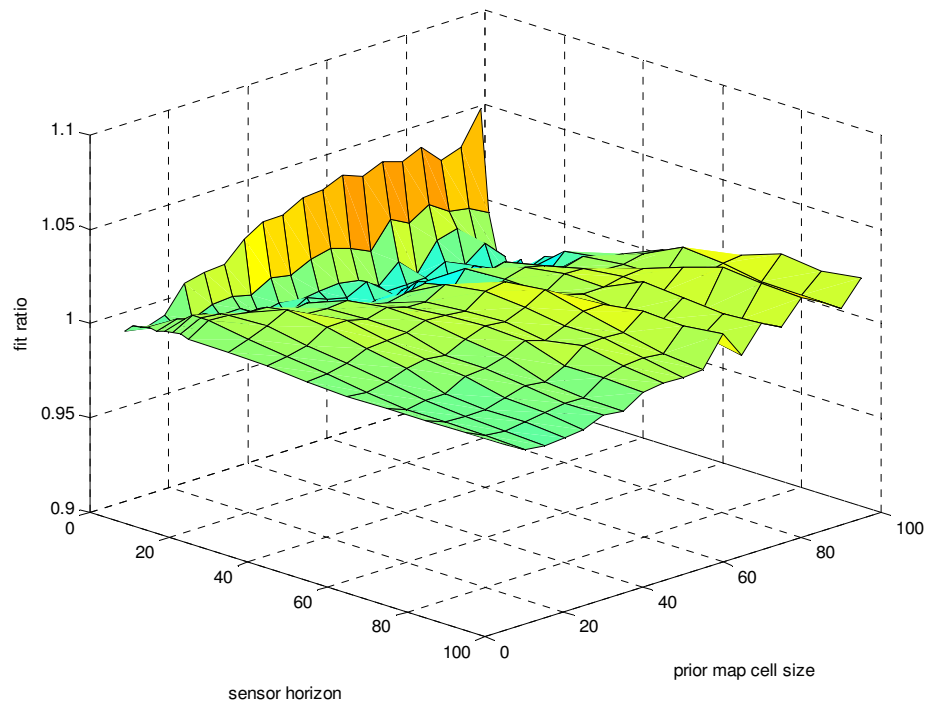


**Figure 21. Summary surface for low frequency terrain.**

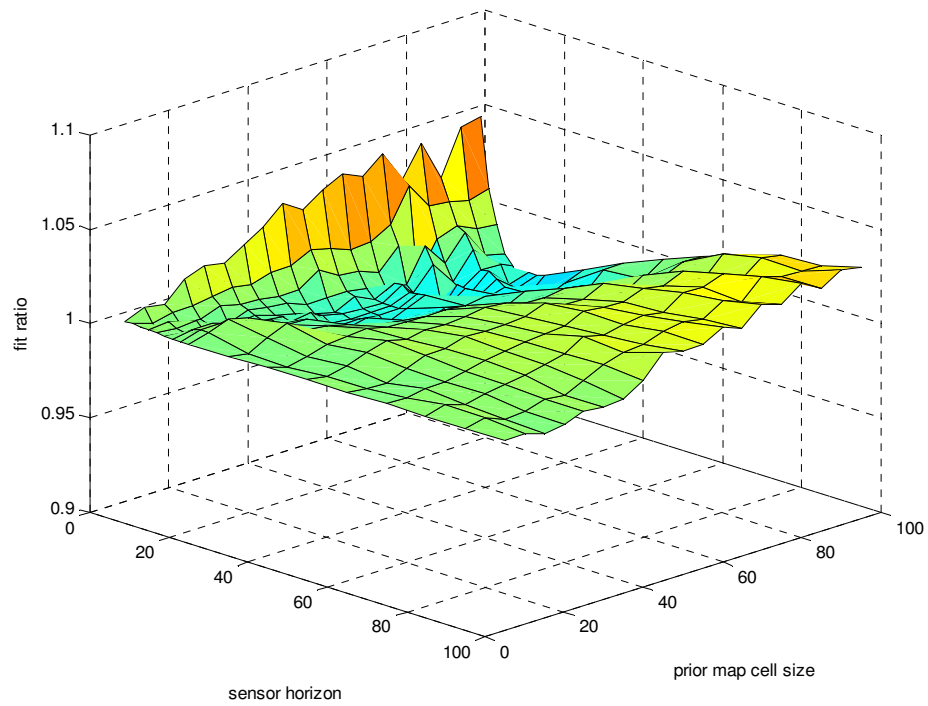


**Figure 22. Summary surface for high frequency terrain.**





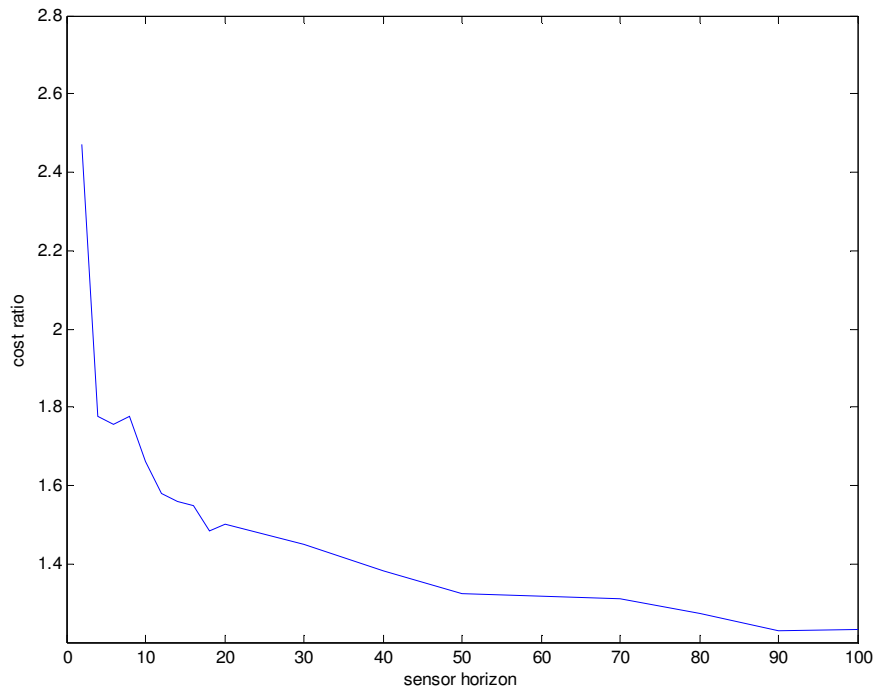
**Figure 23. Ratio between fit and actual data for low frequency terrain.**



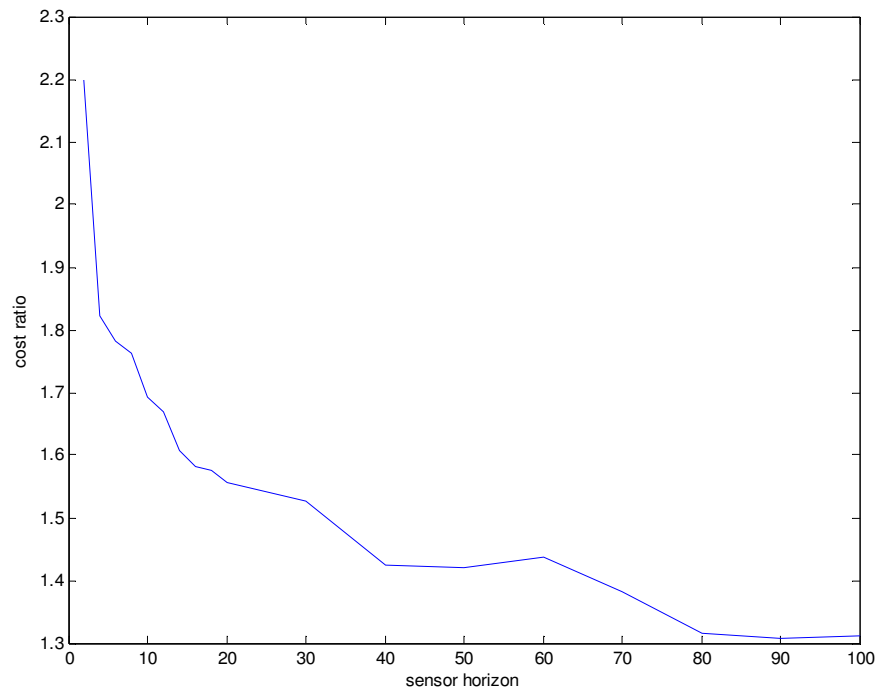
**Figure 24. Ratio between fit and actual data for high frequency terrain.**

## Sensor Horizon and Navigation Efficiency without Prior Maps

This experiment explores how navigation efficiency varies with sensing horizon if prior map data is unavailable. The resulting curve maintains the same basic shape as the summary surfaces for both low (Figure 25) and high (Figure 26) frequency terrains. Hence, even if no prior data is available, a fixed sensor horizon can be utilized to achieve certain levels of performance. Without any prior information, performance is significantly decreased, even compared to the coarse prior maps. Clearly, even limited pre-knowledge of the terrain can provide significant benefits.



**Figure 25. Summary curve for low frequency synthetic terrain.**



**Figure 26. Summary curve for high frequency synthetic terrain.**

## Implications

These experiments show that there is a clear point at which increasing sensor range does not lead to significant increase in navigation performance. These observations have implications for the design of robots:

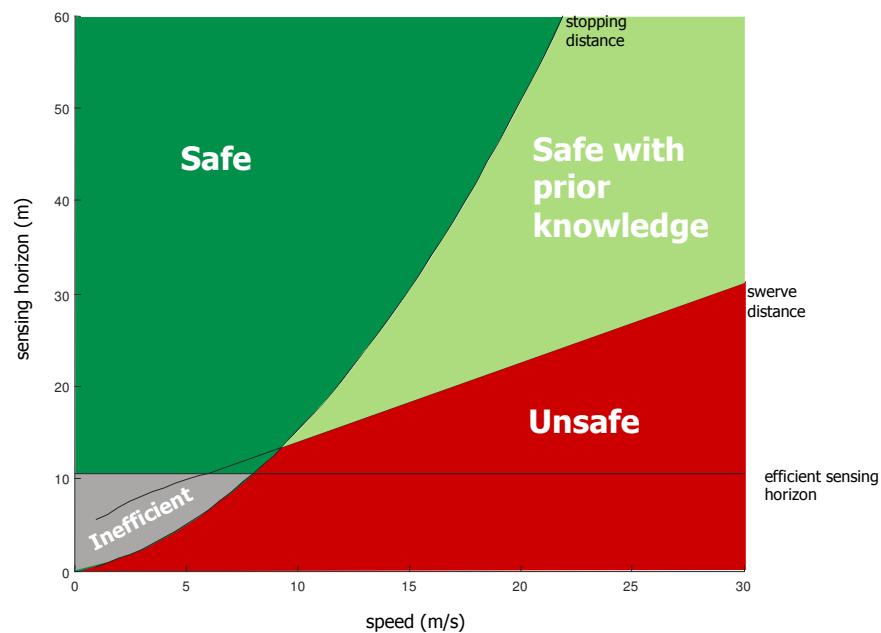
- **Reduced resolution data is useful-** Performance decreases with prior map resolution, but slowly and roughly linearly (over the resolutions considered). Should onboard storage be at a premium (or, should higher resolution data come at a premium cost), it is possible to reduce the resolution of the prior map data without significant performance loss.
- **Average terrain estimates outperform optimistic or pessimistic estimates-** Care should be taken in the preparation of prior map data. There is a significant loss in the predictability of performance as prior map terrain data strays from the average cost to either optimistic or pessimistic costs.

- **Sensor horizon is an important tradeoff to be considered-** For a given terrain type there is a point at which increasing sensor horizon has limited, but near zero, benefit. By understanding ahead of time the operating environment ahead of time, a sufficient sensor range can be selected that is not wasteful of development or processing resources. Pre-characterization of representative operational environments would be useful in determining this range. This might be done by performing a statistically significant number of simulations, or it may be possible to relate the terrain's fractal dimension to the relationships described in this chapter.

## 6 Navigation Regimes



In isolation, the relationships between speed, prior map resolution, sensing horizon and navigation efficiency fall short of characterizing the space of off-road navigation. These relationships can be used to divide the space of autonomous navigation into four sectors, as seen in Figure 27. Every point below both the swerve distance and stopping distance curves is unsafe, as no avoidance maneuver can be performed. The region bounded by the efficiency determined fixed sensing horizon and stopping distance is safe, but inefficient. All points above the stopping distance curve are guaranteed safe. All points between the swerve distance and stopping distance curves are also safe if sufficient prior map information is available. This is the region of the space that will be exploited by swerve-limited navigation.



**Figure 27. A division of the autonomous navigation space.**

From these relationships the three navigation regimes are also distinguished (Figure 28). Regimes are subspaces of off-road navigation distinguished by the relationship between required sensing horizon and speed. There are several design freedoms that influence the location of transition points between regimes.

The remainder of the chapter discusses these regimes, and the transitions between them, in detail.

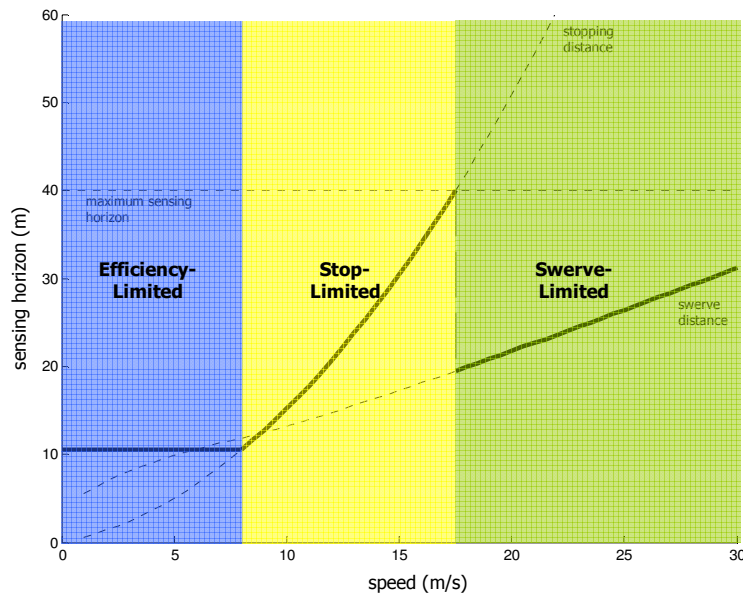
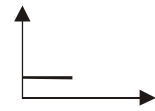


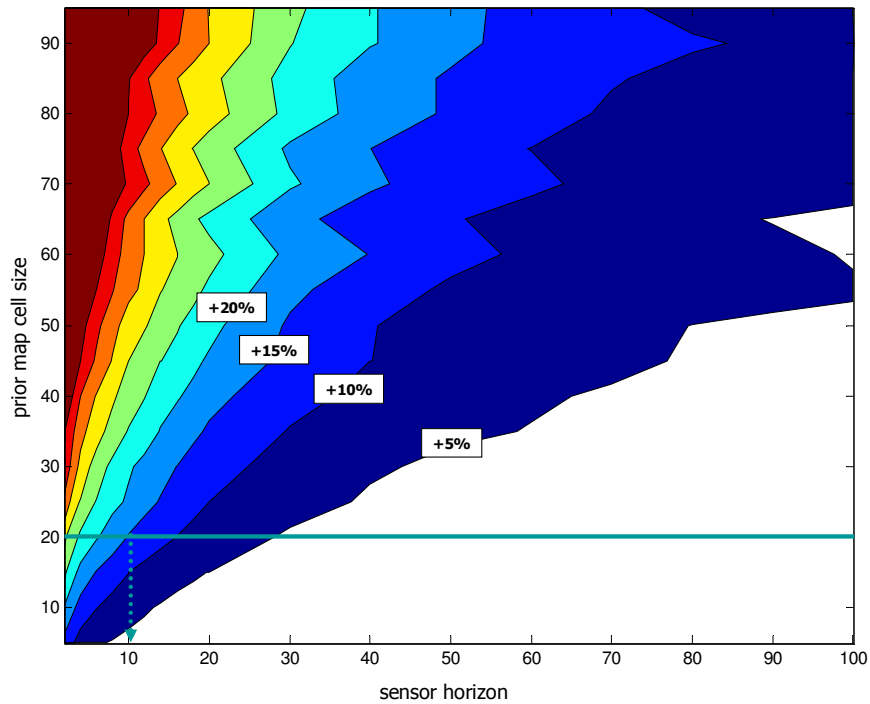
Figure 28. Navigation regimes.

## Efficiency-Limited Regime



The efficiency-limited regime is characterized by low speeds and no requirement for prior maps. The sensing horizon requirement is constant, independent of speed. The relationships between prior map resolution, sensing horizon and navigation efficiency show that for a given terrain type, increasing sensor range yields diminishing returns. While prior maps are not required, the availability of a priori data can be used to increase navigation efficiency.

Given a desired average optimality for a particular terrain, it is possible to select an appropriate combination of sensing horizon and prior map resolution that best achieves desired performance. Hence, there is design freedom in both sensing horizon and prior map resolution. As an example, Figure 29 shows the contours of a summary surface. If a navigation efficiency of within 15% of the optimal is required, and the best resolution prior maps available are 20 units on a side, then the sensor horizon must be 10 units or better.



**Figure 29. Example of efficiency limited regime tradeoff.**

Within this regime, there is often little benefit to detailed pre-plans since sensor based navigation rapidly approaches optimal results in nominal terrains. Large-scale pre-planning may be useful in avoiding large local minima (e.g. selecting the correct valley to traverse). Scheduling and planning for resource utilization may also be relevant in specific applications [Tompkins01].

## Stop-limited Regime



The stop-limited regime is characterized by moderate speeds, and no requirement for prior maps. The relationship between required sensing horizon and speed is quadratic. This relationship is necessitated by the requirement to sense to the stopping distance. Robots operating within the stop-limited regime have the guarantee that they can stop for any detected obstacles. As the sensing horizon required for safeguarding increases, the ability of sensing to detect threatening obstacles at range fails. Again, while prior maps are not required, they do provide an improvement in navigation performance.

Within the stop-limited regime, the speeds involved generally require that navigation algorithms account for some dynamics. Most prior work in moderate-speed and high-speed off-road autonomy resides within this regime.

## Swerve-Limited Regime



The swerve-limited regime is characterized by high speeds and a reliance on prior maps. Required sensing horizon increases linearly as a result of swerving around instead of stopping for obstacles.

In the swerve-limited regime, prior maps must be of sufficiently high quality that it is possible to accurately determine the maximum size obstacle a robot will need to avoid. Furthermore, it is necessary to confirm that there is a corridor of sufficient width to allow for maneuvering around any detected obstacles. Because of the reliance on prior data, safety is closely related to prior map accuracy.

Modern aerial imagery and pose estimation approach a level of prior map resolution where the number of occurrences of unexpected obstacles is low. Large swaths of terrain can be mapped quickly at sub-meter resolution. While this resolution is impressive, it is still insufficient in its raw form for determining traversable routes since terrain features of this scale represent obstacles to many vehicles.

While the raw data may be insufficient, a semantic understanding of the map may make it usable. For example, even though a quarter meter resolution image of a road does not guarantee that it is traversable, the understanding that the road is in fact *a road*, and is intended for traveling on can be used to infer much more information about the terrain; e.g. that it is flat, has little vertical variation and is traversable. Thus even with very simple semantics (road vs. trail vs. not road/trail) it is possible to gain great benefit from these maps.

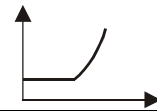
Within the swerve-limited regime, applicable algorithms must incorporate prior data into the navigation activity. Since map data is available prior to operation, it



is possible to reduce the onboard computational load by pre-computing the expected trajectory.

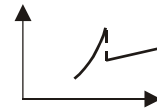
The requirement for prior maps allows for a preprocessing step whereby the local minima in the space can generally be removed by planning around them a priori. The requirement for real-time planning is also removed for pre-planners, easing computational complexity problems. Potential fields and other similar approaches that have the benefit of generating smooth paths may prove particularly relevant as speeds increase.

### **Efficiency/Stop-Limited Transition**



At some speed, stopping distance becomes greater than the fixed sensing horizon specified by performance requirements of the efficiency-limited regime. At speeds less than this, robots are able to sense far enough ahead to stop for detected obstacles. Hence, at speeds lower than this critical speed, the constraint for navigation optimality is dominant. Beyond this speed, constraints for safeguarding dominate sensing horizon.

### **Stop/Swerve-Limited Transition**

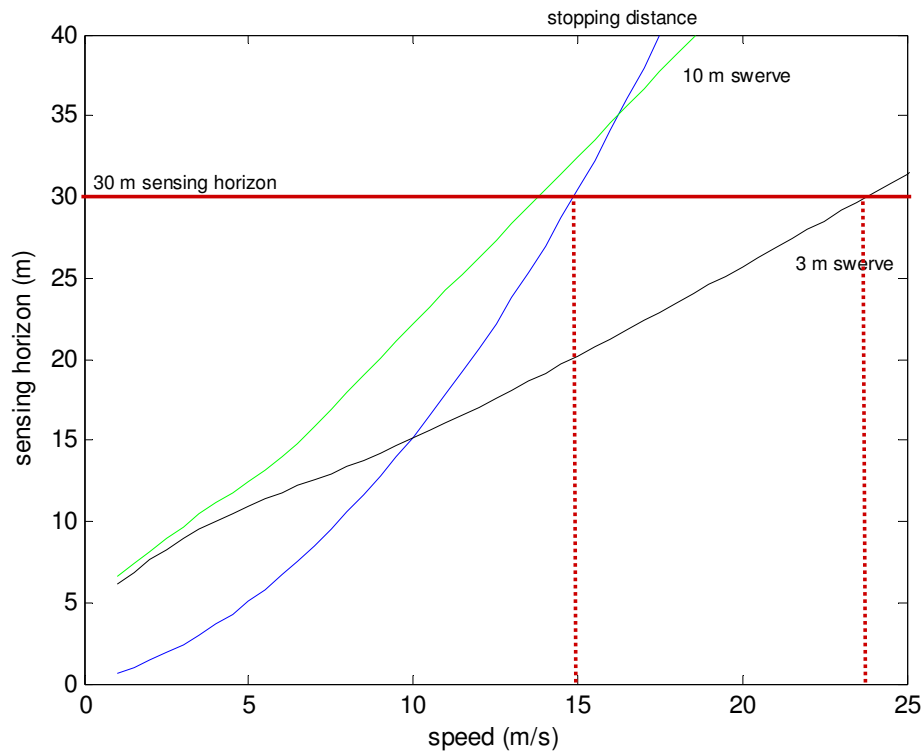


As speed requirements increase it is necessary to reconsider how robots are safeguarded. Sensing horizons required to guarantee stopping distances quickly become practical problems.

An unacceptable option would be to give up safety. The linear growth in sensing horizon associated with swerving, rather than quadratic with stopping, provides an obvious method to ease sensing requirements. Without prior maps, swerving provides no safety guarantees and thus is unviable. By incorporating available prior maps, it is possible to provide a reasonable guarantee of safety. By removing the guarantee that robots are able to stop for an unexpected obstacle, safety maybe somewhat decreased since obstacles that are unexpectedly large would be unavoidable. While this is a serious consideration, the risks are akin to those taken daily by highway drivers. In operating within the swerve-limited

regime, this decrease in safety guarantees must be acceptable within the application domain.

Attributes including sensing horizon, prior map resolution and dynamic response govern when the transition between regimes occurs. For example, consider how a lack of high resolution maps may necessitate operation within the stop-limited regime. Figure 30 illustrates this example. The plot shows that a robot with a 30m sensing horizon would be able to operate in the swerve regime at 24m/s if it has maps that were of sufficient quality that it's maximum swerve offset would be 3m (e.g. enough to drive around a parked car). If instead, the maps available were only sufficient to limit the swerve offset to 10m, the robot would gain no benefit from operating in the swerve regime (it would actually have to travel more slowly to swerve that far), and thus would be limited to the stop-limited regime and speeds up to 15m/s.



**Figure 30. Stop/Swerve-limited regime transition example.**

## Discussion

---

The swerve-limited regime insures robot safety within the constraints of the data provided by the prior map. Navigation within the swerve-limited regime is more susceptible to failure since there are more assumptions to be violated.

To achieve the best performance, mobile robots benefit from adopting algorithms and sensing ranges appropriate for the regime within which they are currently operating. For example, a robot capable of operating at high speeds may spend a majority of its time operating within the swerve-limited regime but may at times have to slow to speeds where it can operate within the stop-limited regime. When traveling at slower speeds, consideration of the panic stop maneuver may decrease risk. Furthermore, by decreasing the required maximum sensing range, it may be possible to increase the fidelity of a generated world model.

The recognition that there are discrete regimes in the off-road navigation problem should yield a set of approaches that achieve high-speed off-road navigation while maintaining reasonable robustness. In the following chapters, a detailed comparison is presented of two robots, and their associated algorithms, from the efficiency-limited and swerve-limited regimes.



---

## **7 Contrasting Navigation Regimes: Efficiency-Limited and Swerve-Limited**

---

The spectrum of off-road navigation exhibits highest performance in the swerve-limited regime and is most deliberative in the efficiency-limited regime. Example implementations, representative of the differences between the regimes, are highlighted in Table 2. Quantitative differences include orders of magnitude differences in speed, prior map resolution and sensor range while there are qualitative differences in the approach to pre-planning and onboard navigation.

These two robots (Hyperion and Sandstorm) represent the physical embodiment of an approach to the off-road navigation problem. Hyperion is the culmination of research focused on novel approaches to planetary exploration. In contrast, Sandstorm evolved from research into high-speed, autonomous navigation over long distances. Both research areas focus on operation in barren, desert like terrains and are representative of the state-of-art for their application areas. They are representative of approaches in the efficiency-limited and swerve-limited regimes.

	Swerve-Limited Regime	Efficiency-Limited Regime
Archetypical Robot	 Sandstorm	 Hyperion
Speed	10 – 20 m/s	0.1 – 0.3 m/s
Prior Maps	0.5 – 1 m pixel size	50 – 100 m pixel size
Sensors	15 – 50 m range stabilized active sensing	5-7 m range unstabilized passive sensing
Mass	3200 kg	156 kg
Approximate Power Consumption	145,000 W	200 W
Terrainability	30° slope 30 cm obstacle	15° slope 15 cm obstacle
Pre-Planning	Fine	Coarse
Real-Time Planning	Constrained path-shifting	Unconstrained arc search

**Table 2. A summary comparison of archetypical robots.**

## Speed

The speed at which a robot operates is associated with the application. In planetary exploration, a common characteristic is low specific power. Planetary rovers typically operate at a specific power on order of 1W/kg or less. In comparison to existing planetary rovers, Hyperion is actually power rich, with a specific power of approximately 1.25 W/kg, primarily because of algorithms used to optimize the amount of solar power collected as it navigates through its mission goals. The limited specific power directly impacts the amount of power

available for locomotion and thus limits navigation speed. At speeds of 10's centimeters per second, the stopping distance is effectively zero, and thus is not a driver for determining sensor range.

Sandstorm is not constrained by planetary relevance and, instead of solar power, employs a turbo-charged diesel engine as a power source. Sandstorm's specific power is roughly 40 W/kg and hence there is much more power available for locomotion. The robot is capable of speeds exceeding 35m/s but has never exceeded 20 m/s under autonomous control. At 20 m/s, the stopping distance is roughly 50 m, in optimistic conditions. Sensing at this range (and beyond), particularly while moving over rough terrain, is a challenge. The desire to travel at these speeds, and faster, necessitates Sandstorm's operation within the swerve-limited regime.

## **Prior Map Resolution**

---

Hyperion uses low resolution prior map data to maintain planetary exploration relevance. Prior maps are limited to the resolution available from interplanetary reconnaissance satellites. Therefore, the maps available to the rover a priori have approximately 50x50 m pixels. Because of the vast variation in terrain possible over this scale of pixel, this data provides limited benefit to the onboard local planner.

Since sandstorm operates within the swerve-limited regime, its performance benefits greatly from high quality prior map information. Sandstorm's task predominantly consists of driving on trails and secondary roads. In these situations the semantic understanding of the prior map allows sub-meter resolution inferences to be drawn about the terrain. For example, any regions of the prior map known to be black-top road can be inferred to have a plane like surface with variation of less than a few centimeters. Thus meter resolution data can be interpreted at close to centimeter level resolution in specific situations.

## Sensors

---

Hyperion need only have a short sensing horizon for safeguarding. To maximize planetary relevance, Hyperion uses a stereo camera pair to sense terrain. This sensor provides useful geometric data from two to seven meters in front of the robot. At the time of implementation, the relationship between sensing range and path optimality was not well understood. Thus the sensing range was not specified to optimize this criterion. Instead, sensing range was selected based upon *reasonable* coverage that balanced sensor height and camera baseline with mechanical engineering constraints. In Arctic field experiments, navigation was efficient and robust.

In later experiments in more challenging terrain in the Atacama Desert, navigation performance was less efficient. This deficiency in performance was characterized by Hyperion approaching too close to small cul-de-sac obstacles and then being unable to avoid them. This evidence implies that in the Arctic Hyperion was operating within the efficiency-limited regime with sufficient sensing, while in the Atacama it had an insufficient sensing horizon for the terrains it was traversing.

Sandstorm's high speeds require commensurately large sensing ranges to maintain safety constraints. The primary navigation sensor is a line scanning LIDAR. By utilizing a LIDAR, the quadratic range error problem is avoided as are the difficulties in making a stereo vision system work reliably at high speeds in an outdoor environment. Instead, the quadratic error associated with sensor pointing is introduced. While challenging, this problem was believed to be easier to solve. A two stage shock isolation system is used to maintain the pointing accuracy required by the navigation and perception software. Even with the isolation system, achieving sufficient quality and uniformity of terrain evaluation data out to stopping distances is unachievable at high speeds.

Through operation consistent with the swerve-limited regime, Sandstorm is able to achieve reliable cross country navigation on trails and secondary roads with limited risk.

## **Terrainability**

---

Mobility plays a major role in determining the sufficiency of prior map data sources. As robot terrainability increases, the required quality of perception and prior map data decreases. In the case of Hyperion, the quality of prior map data was determined by the availability of data sources. It was not feasible to acquire better than 30m resolution data.

In contrast, there were many sources available to provide high resolution data for Sandstorm's pre-planning system. Sandstorm's terrainability helps determine the relevant prior map resolution. While it is not feasible to navigate open-loop using only a high resolution prior map, a combination of resolution and image understanding make it possible to make reasonable safety guarantees about the pre-planned routes Sandstorm follows.

## **Pre-Planning**

---

Hyperion utilizes only a low resolution pre-plan. The onboard navigation system is required to move from one waypoint, marked by a rectangular goal region, to the next over a distance of approximately 50m. The pre-planning algorithm is able to understand the movement of shadows and sunlight over the terrain, maximizing the amount of solar energy received by the robot. In this way, the pre-plan is critical to the success of the mission but provides limited input to the onboard navigator.

Sandstorm requires a fine resolution pre-plan. The path to be taken is specified by waypoints at one meter intervals along the route. The pre-planner uses a combination of an autonomous planner and human experts to generate routes. A high resolution cost map is generated from semantic data describing the location of roads and trails. This cost map is then passed through an automated



planning system to generate an initial route. Upon completion, the route is verified and improved by a team of human experts that use their understanding of the high resolution imagery to generate a drivable route.

The pre-planning for Sandstorm provides much more direct input to the onboard navigation system. Each waypoint is tagged with a corridor width that provides guidance to the onboard navigation software as to how far it may stray from the plan. Each waypoint is also assigned an estimated best speed, which is used to ensure that the robot is traveling at an appropriate speed given the pre-knowledge of the terrain and environment.

## **Real-Time Planning**

---

Hyperion uses stereo vision data to generate a statistical evaluation of terrain. A planner then selects arcs that minimize the integrated cost of traveling over terrain to a goal. Because of the low speeds and a relatively fast steering slew-rate, arcs are evaluated as true arcs and not clothoid paths. The onboard planner is information optimal but is able to wander anywhere within a large, arbitrarily specified rectangle containing adjacent waypoints in the mission plan.

Sandstorm performs an evaluation of LIDAR line scan data to adjust the pre-existing waypoints, centering the path on the smoothest portions of terrain. This adjustment is necessary to account for terrain features invisible in prior maps and to deal with registration errors between the estimated pose of the robot and the map. Sandstorm is unable to deviate from the pre-planned route boundaries for two reasons: it is explicitly disallowed by the requirements of the application, and it would move the robot out of the understood pre-planned corridor into a region without pre-plan data where Sandstorm would be unable to safely operate at swerve-limited regime speeds.

## **Summary**

---

Hyperion exemplifies the efficiency-limited regime. It is able to traverse as needed to achieve a goal, without a detailed pre-plan. In contrast, the desire for

high-speed autonomy compels Sandstorm's operation within the swerve-limited regime and thus the need for detailed pre-planning and constrained navigation. The following two chapters provide a detailed technical description of each of these robots.

---

## **8 Representative Implementation:**

### **Swerve-Limited Regime**

---

A robot representative of approaches viable in the swerve-limited regime was developed and exercised to demonstrate these capabilities. Sandstorm and its associated pre-planning algorithms achieve state-of-the-art high-speed autonomous off-road navigation representative of the swerve-limited regime. Performance is beyond that capable within the constraints of conventional requirements for safeguarding.

This work is notable for achieving high-speed, long distance traverses. A key to success was the incorporation of detailed prior mapping and pre-planning. A novel approach to the onboard navigation task adjusts pre-planned paths rather than operating with complete freedom. The details of this approach are presented to further contrast the efficiency-limited and swerve-limited regimes.

#### **Context**

---

Sandstorm (Figure 31) was developed to advance the capabilities of high-speed off-road autonomous vehicles, and to compete and win the DARPA Grand Challenge. The Grand Challenge is a driverless desert race. The winner is a robot that completes an unrehearsed course in the least amount of time and does so within 10 hours. The 2004 Grand Challenge was a 143 mile race across the Mojave Desert from Barstow California to Primm Nevada. Prior to competing in the race, each robot is required to demonstrate safety and navigation capabilities in a qualification and demonstration event.

After successfully completing qualifications in 2004 at the California Motor Speedway, invited teams transitioned to the race start area near Barstow California. Roughly 3 hours prior to the race start, the organizers provided each team with a CD describing the route. This description consisted of 2586 waypoints that marked the corridor within which the robots were required to

travel. The waypoints were spaced 90m apart on average, and had a typical corridor width of 8m. With each waypoint, the organizers provided a maximum speed limit and a lateral offset that defined this corridor. Once past the starting line, robots were required to be completely autonomous. The only communication to the robots was a remote safety kill and telemetry system used by race personnel to ensure safety.



**Figure 31. Sandstorm.**

To maximize performance, Sandstorm combines autonomous and human pre-planning to optimize routes. Once underway, the robot uses onboard sensors to adjust routes, to account for obstacles that are not represented in the pre-planning map and to correct for errors in its position estimate.

Sandstorm made notable progress in addressing the Grand Challenge off-road navigation problem and demonstrated a novel approach to the navigation of off-road terrain, but neither Sandstorm nor competitors succeeded in the challenge. Sandstorm traveled a total of 11.9 km along the route at a pace that would have

finished the race within the required time, averaging 24 km/h (6.7m/s). During the race, Sandstorm achieved a sustained peak speed of 58 km/h (16.1m/s). Sandstorm was the only robot to meet qualification performance standards based on the original specification for the demonstration and went the farthest and fastest on race day.

## **Mapping & Pre-Planning**

---

The mapping and pre-planning system is designed to provide Sandstorm with a sufficiently high quality plan that it will be able to safely navigate the Grand Challenge route. To achieve this objective, the mapping and pre-planning system incorporates a large database of terrain data. Once the required route has been provided, this information is used by automated and human planners to generate the list of one meter spaced waypoints that Sandstorm will use to plan its route across the desert.

### ***Mapping***

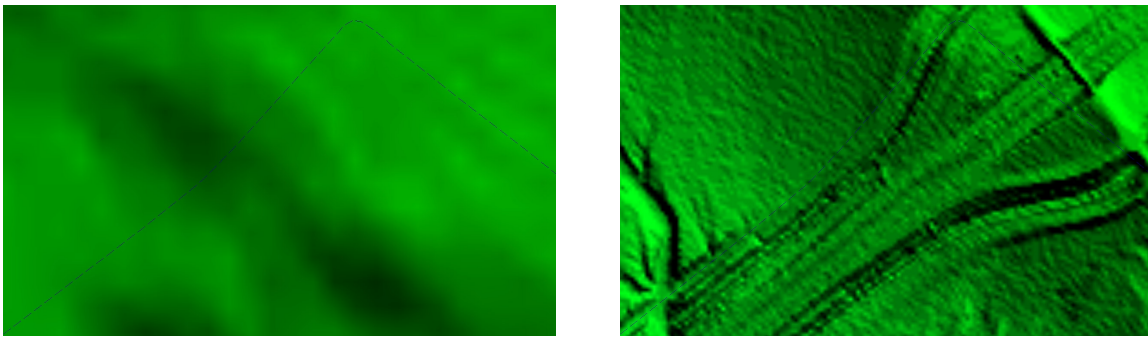
The prior map database synthesizes high resolution, high accuracy maps of the race region to enable autonomous planning and human refinement of a navigation route. In building a prior map database for the 2004 Grand Challenge, the team collected roughly 650GB of cartographic data. The bulk of this data was acquired from freely available data sources originating from the United States Geological Survey (USGS). This data was in the form of:

**Digital Line Graphs (DLG):** Vector representations of various layers such as road networks, rail roads or lakes and rivers.

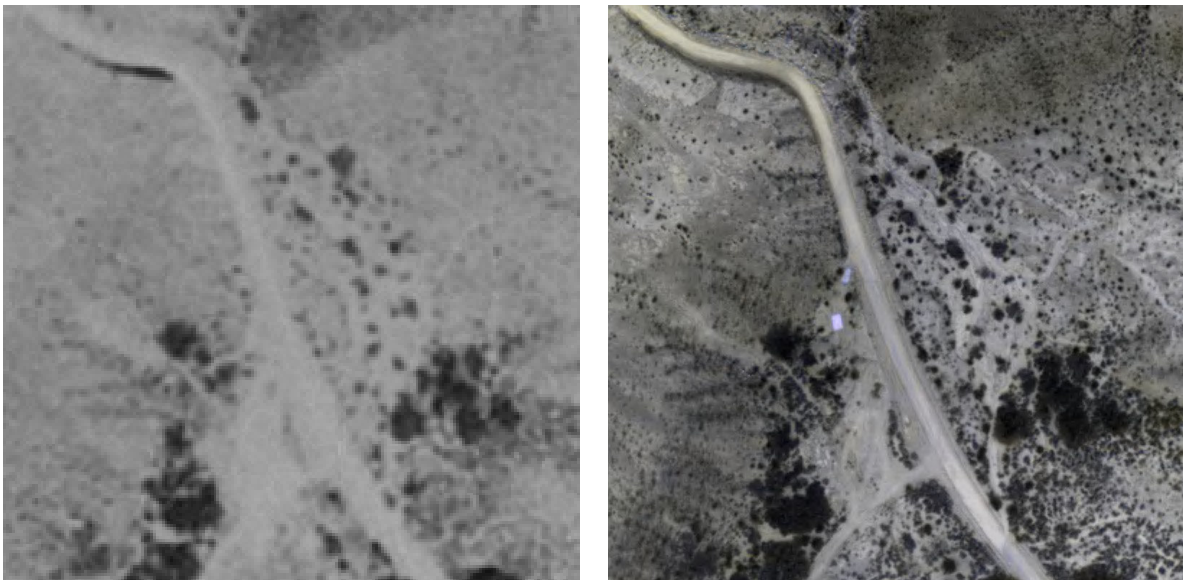
**Digital Elevation Models (DEM):** A regular grid of elevation samplings available in a geo-referenced raster format called GeoTIFF.

**Digital Orthographic Quarter Quads (DOQQ):** Overhead imagery also in the GeoTIFF format.

USGS data is combined with high resolution maps. New aerial reconnaissance and mapping technology [Visual02] provides narrow swaths of ultra high resolution/accuracy imagery (approximately 20-40cm pixel size) and digital elevation models with 5m postings (i.e. 5m spacing between elevation samples). Figure 32 shows a comparison between standard USGS data (which has 30m postings) and 5m posting elevation data. Figure 33 shows a comparison of standard USGS data and sub-meter resolution image quality on a different desert scene.



**Figure 32. A comparison of USGS and 5m postings DEM data quality.**



**Figure 33. A comparison of USGS and sub-meter resolution image quality.**

Ground reconnaissance of the potential race routes is also incorporated into the pre-map database. Many of the trails and roads in the region that were probable candidates for the race route were mapped to sub-meter accuracy. Since this data set provides very accurate, ground-truth location of traversable routes, it is trusted by the planning system more than any of the other data types.

The road network DLGs form the backbone of the map, but are insufficient in themselves as they are not held to specific standards of accuracy. In contrast, DOQQ's are held to national map accuracy standards and are commonly used to correct DLG's. Using the available DOQQ imagery, some DLG data throughout the ~129,500 sq. kilometer region surrounding the Barstow to Vegas corridor was manually corrected. Approximately 1000 man hours were invested in this effort, and although large regions were revised, the coverage of the revisions was not comprehensive enough, nor of sufficient quality to be trusted because of the rate of human errors in correcting the data.

The inability to correct these data sources over large regions led to a new "just-in-time" approach to map correction. Rules for the Grand Challenge indicated that the route would be revealed two hours prior to race start. This time is sufficient to correct the narrow ribbon of the data that comprises the navigation corridor. A pessimistic estimate of the area of this ribbon is 23 sq kilometers, or approximately 0.02% of the race region.

The route revision process begins with an automated planner that processes the route and then uploads waypoints specifying where Sandstorm should drive within a route corridor. The server then assigns equal length groups of segments of the route to each of the cluster machines. As each of the 14 human route editors progresses through their assigned segment, the route editing software uploads the revised work units to the server which maintains the latest composite route. In regions of the route where adjacent segments meet, the route editors coordinate to ensure their waypoints are viable.

Periodically, a central server generates an estimate for the race completion time. This estimate allows a supervising team to understand how challenging the route is and allows them to formulate an overall speed strategy. Given the completion time estimate, the supervisory team is able to adjust route speeds to achieve a target elapsed time for navigating the route. Software that performs the scaling maintains speed limits defined in the route definition file or imposed internally by the team.

### ***Infrastructure***

To store and transport the vast amounts of geographic data between the central repository and the route editing workstations, a high performance file server is used in combination with a gigabit network. This system is implemented using a two terabyte RAID5 array attached to a dual Xeon processor-based server.

The massive volume of data passed between the server and cluster machines is sufficient to saturate a gigabit network, if implemented naïvely. To prevent this, a local image cache is implemented on each client machine. Due to the intensity of pre-race training, most of the client machines already had local copies of the relevant imagery required for editing their portions of the route. Thus greatly reducing the network burden.

To access the massive data set, a standardized file organization is utilized and the various data formats are aggregated into a Geographic Information Systems (GIS) aware database. An XMLRPC server provides a front end for the client cluster, allowing the route editing software to query the database for the availability of a particular form of cartographic data in a region. The database then replies with the standard location for the relevant files in the archive. Clients check their local cache for a file, resorting to the network server only when necessary.



## ***Automatic planning***

An automatic planner uses the route definition file provided by the race organizers in conjunction with terrain database to output an optimized route with respect to travel time and implicit safety. The various data sources are sampled to a regular 1m grid which creates to simplify the pre-planning problem. DLG data, combined with path traces generated from reconnaissance efforts are used to provide six base tiers of cost values. From lowest to highest cost, the tiers are:

Highway or wide road **with** reconnaissance trace

Narrow road **with** reconnaissance trace

Off-road **with** reconnaissance trace

Highway **without** reconnaissance trace

Narrow road **without** reconnaissance trace

Off-road **without** reconnaissance trace

This tiering explicitly represents a higher confidence in areas that have been previously traveled by a ground reconnaissance team. The system makes no distinction between paved and unpaved roads, and trails are considered as narrow roads. This simplification was made since the onboard planner does not understand the significance of different road surfaces.

To guide Sandstorm close to the center of the corridor, cost is increased as a function of the distance from the center of the corridor. Similarly, in order to avoid unsafe areas of the terrain, cost is increased as a function of the slope of the terrain. The tiers bracket how much the cost can be increased, such that the final cost still remains within the same cost tier. For example, the planner will favor a path with a reconnaissance trace on a sloped road over a path on unknown terrain, even if it appears “flat”.

## Planning the path

Routes are planned using a modified wave-front planner. Wave-front planning is utilized since it has low memory requirements and because, in this application, the search space is frequently restricted to corridors which are long and narrow, reducing the benefit of a heuristically guided search.

Because the route is defined by corridors that the planner must stay within, rather than waypoints that need to be reached, the waypoints along the path should not be used as goals, as they may not fall on the best plan. Instead the planner generates paths for groups of three waypoints at a time. If there are four consecutive waypoints  $P_1$ ,  $P_2$ ,  $P_3$ , and  $P_4$  with associated corridor widths, the planner first plans a path from  $P_1$  to  $P_3$ . This path must remain within the corridors between  $P_1$  and  $P_2$  and between  $P_2$  and  $P_3$ . A new point is now selected,  $P'_2$ , which is the point on the path closest to  $P_2$ , as illustrated in Figure 34. The segment of the path from  $P'_2$  to  $P_3$  is then discarded and the planner is then run again from  $P'_2$  to  $P_4$ , with the constraints that the path must remain within the corridor from  $P_2$  to  $P_3$  and from  $P_3$  to  $P_4$ . By iterating this process over all of the waypoints in the route, an efficient path is generated for the entire route.

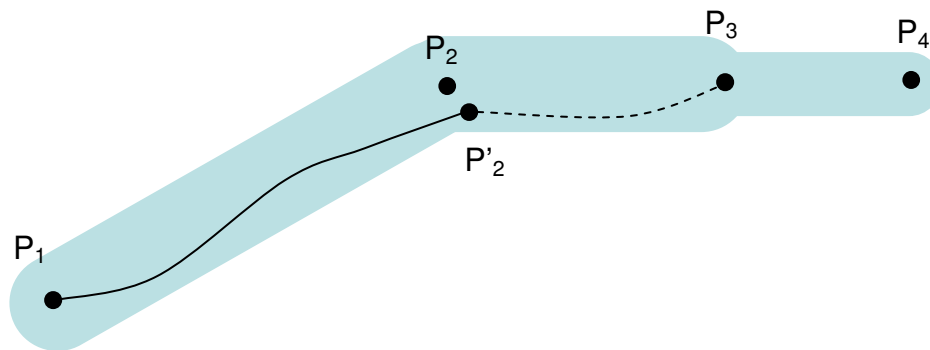


Figure 34. Operation of the global path planner.

The planner incorporates a modification that breaks ties between neighboring cells of equal cost by preferring those that are closer to the line connecting the start to the goal. In the race relevant planning domain this helps reduce problems associated with 8-connectivity when planning in open areas. Since the cost

values used for planning the path represent the time required to travel each unit of area, the resulting path is a path that optimizes the expected travel time.

### ***Vectorizing the Path***

The path calculated by the wave front planner is a sequence of points on a regular one meter grid. This dense path representation is not easy for human editors to verify or modify. Additionally, the resolution of the reconnaissance traces used as input for the planner may be on the order of 10 cm; resolution that would be lost if it were sampled to a one meter grid. To avoid these problems, the path is vectorized, allowing a sparse representation of segments where possible (enabling efficient editing) and a finer representation where necessary.

To perform this data reduction, consider the points that make up the race route. If a group of consecutive points can be represented by a straight line with a mean square error of less than 2 meters, then the group of points is replaced by the end points of the line. For example, in a one kilometer straight road this change of representation reduces the number of points from 1000 to two. The vectorized representation is more suitable for human editing, since manipulating the endpoints of the resulting vector implicitly changes the location of the thousand points in the original path.

To use the full accuracy of ground truth GPS traces, the vectorization algorithm considers groups of points that lie on or near the GPS traces. These groups of points are then replaced by the original GPS trace over regions where the pre-planned path is within one meter. Since the GPS traces can also have a large number of redundant points, they are also vectorized, but with a tighter error tolerance of 20 cm. The resulting path is a vector path, containing few points in straight segments of the road, fine detail in curves and preserves the resolution of the original GPS traces.

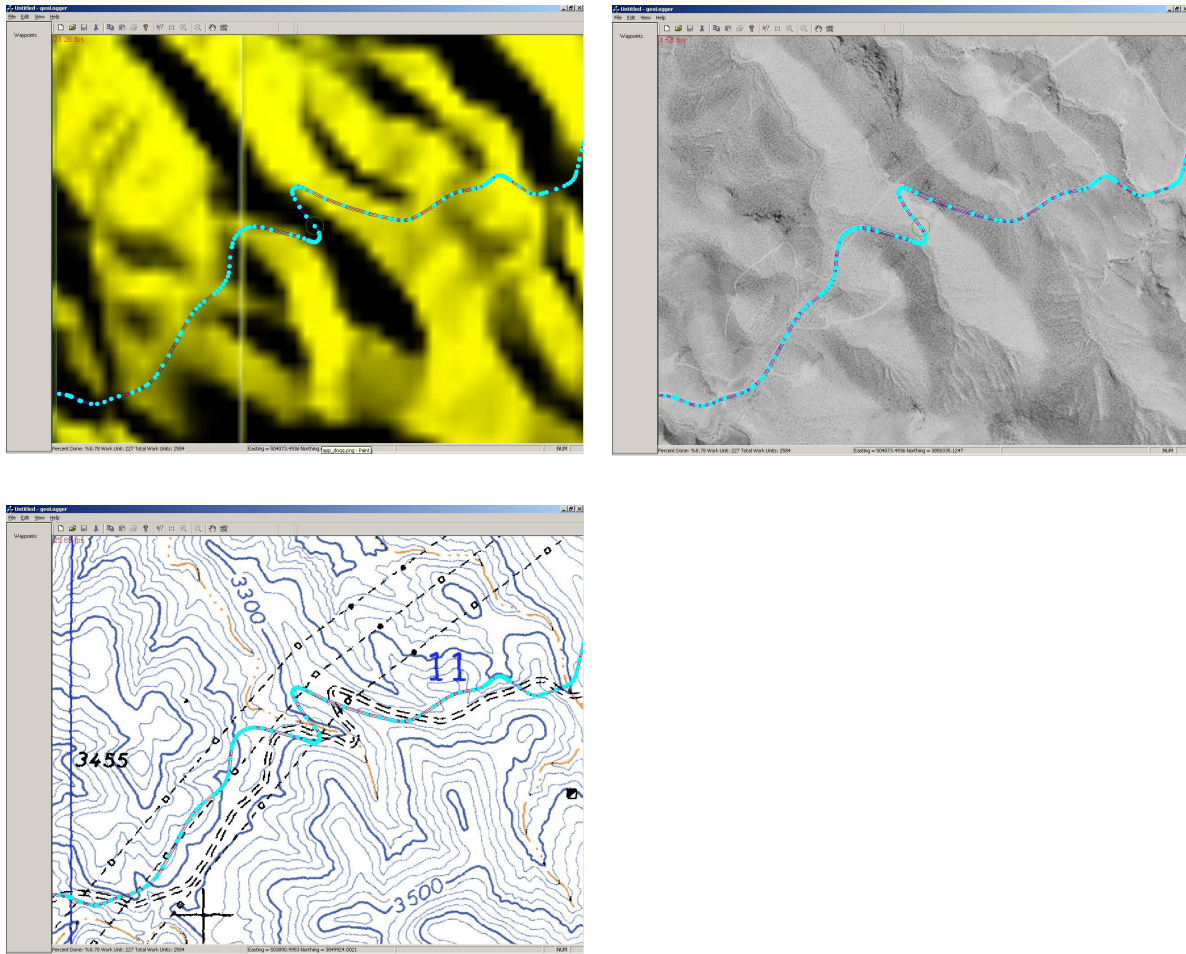


Figure 35. Three views of data available to route editors.

## ***Route Editing Tool***

The autonomously planned route may be unsafe or inefficient due to the limited cost representation used by the planner. In contrast, human editors are generally able to extract subtle cues regarding the terrain traversability from the various image data formats. This understanding is used to improve the pre-planned route. Human editors use a custom developed route editing tool to perform this task.

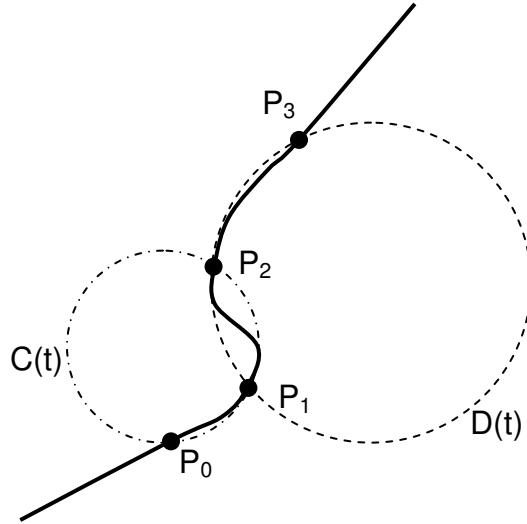
The route editing tool (RET) was derived from an existing GIS interface tool [Drewes03]. The original software provided much of the low level capabilities required to interface with GIS data sources. The RET interface is simple: it

provides a mechanism to view the route in the context of all of the cartographic information, and then change the route as necessary. It connects to the central database server and downloads any pre-planned routes and map-data associated with the current segment of the route under consideration. Figure 35 shows three of the cartographic data sources in the RET. The top left image shows an overlay of a route over DEM data. The top right image again shows the same route, but overlaid on the broadly available satellite imagery. The bottom left image shows the route again, but overlaid on a DLG based data product. Note that though the route (solid line) is intended along the path indicated in the data source (dashed double line), there is a significant offset between the route and the path due to registration errors in the data source.

Routes are represented graphically in two ways: as a set of linear segments and as a smooth spline. The linear segments show the operator the connectivity between waypoints while the spline illustrates the true path that will be transferred to Sandstorm. In addition to this information, the application shows the bounding corridor constraints, a safety corridor (which is 2 meters narrower than the course corridor) and any GPS traces that were collected in the area. Each data layer can be toggled on or off, and each data type is displayed in a customizable color.

To create executable routes for Sandstorm, it is desirable to create paths that are smoothly varying. A specialized spline is utilized to generate smooth paths with an intuitive user interface. To be “intuitive”, the spline:

- Intersects all of its control points.
- Localizes the effect of moving a control point.



**Figure 36. Spline construction for the route editing tool.**

The spline that best met these criteria is a weighted interpolation of arcs. Given an ordered set of 4 points, there are two sets of three consecutive points which can define a pair of circles, as shown Figure 36. The equations of these circles are  $C(t)$  and  $D(t)$  with the constraints that:

$$\begin{aligned} C(0) &= P_1 \\ C(T) &= P_2 \\ D(0) &= P_1 \\ D(T) &= P_2 \end{aligned}$$

For the segment where the two sets of points overlap, the spline is calculated as the weighted sum of the two curves:

$$S(t) = \frac{t}{T} D(t) + \frac{T-t}{T} C(t) \quad (0 \leq t \leq T)$$

In the degenerate case at each end of a list of waypoints, the second circle is replaced by a linear fit between the remaining two points.

To edit a path, a user either selects and moves control points (i.e.: the endpoints of the linear segments), or clicks somewhere along a linear segment to insert a new control point. Inserting a control point, automatically creates an associated

spline between the neighboring waypoints. A feature, called “Snap to GPS” moves the currently selected points onto the closest GPS trace. Associated with each point is an editable speed that represents the maximum travel speed at that point. Also displayed for each point is the required speed limit for that segment of the route.

The “Snap to GPS” implementation calculates the closest path segment to the point being edited. To achieve an efficient search through the 200,000+ segments that make up a Grand Challenge relevant route, the application utilizes an axis-aligned bounding box (AABB) tree algorithm. The AABB approach provides a fast search and is easy to implement. An AABB is simply a box defined by the minimum and maximum extents of the contents it represents. To build the AABB tree, the GPS trace is broken into sets of 100 adjacent nodes. Each set is a leaf in the AABB tree. The second layer of the tree contains the pairs of adjacent leaf nodes; each additional layer is built by recursively combining pairs of adjacent nodes. Searching this tree is very efficient, a point is recursively compared to the nodes in the tree, if the point falls in the bounding box defined by a node, the search descends that branch, otherwise a comparison is made to the second branch, if the point is in neither box, the search terminates, indicating no node is nearby. At the leaves of the tree, the closest node is found by linearly searching through the 100 segments.

This hardware and software combined with an intensively trained team of editors provides the a priori knowledge essential for Sandstorm to achieve high levels of navigation performance. In less than two hours, the team of 20 editors was able to generate a safe and viable route.

## **Platform**

---

The best mobility platform for high-speed off-road navigation would combine reliable, robust terrainability with agility and sufficient payload capacity to support large amounts of sensing and computing hardware. Toward these ideals, Sandstorm is developed by specializing around a military High Mobility Multi-

purpose Wheeled Vehicle (HMMWV). This foundation provides a very capable platform able to traverse a wide variety of off-road terrain. Table 3 highlights some of these capabilities.

Property	Value
Ground Clearance	40 cm
Approach Angle	72°
Departure Angle	45°
Breakover Angle	32.5°
Sideslope Capability	40%
Gradability	60%
Turning Radius	7.4m
Width	2.20m
Length	4.70m
Payload Capability	> 1350 kg

**Table 3. HMMWV terrainability characteristics.**

Actuators and enhancements were incorporated into this platform to make it amenable to computer control and to improve upon its basic capabilities.

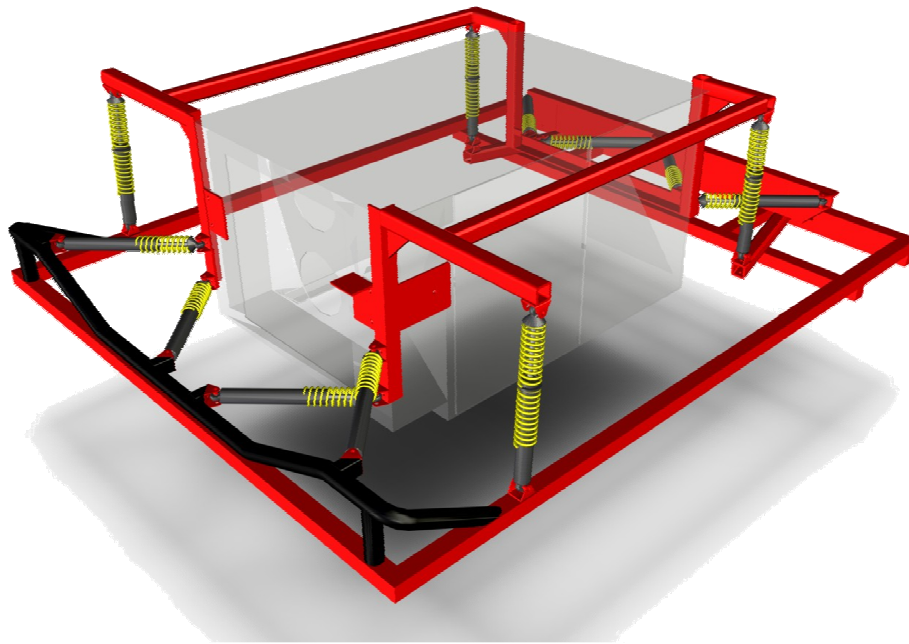
### ***Electronics Box***

The electronics box encompasses all of the computing and navigation that commands Sandstorm and acts as the common inertial reference frame for onboard sensors. The box minimizes the connections between the navigation “brain” and the host vehicle. By containing all of the sensing, algorithms and power distribution in a single unit, a replacement HMMWV could be rapidly automated should the original vehicle become irreparably damaged or should a better platform become available.

The electronics box uses an open, forced air system for cooling. Investigation of the expected operating environment indicated that forced air would be sufficient for cooling. Initial analysis predicted a 4000W thermal load that would need to be



removed from the box. A brief thermal analysis showed that, given a 24°C ambient air temperature, this heat could be transferred to the air with roughly a 5°C temperature increase if 1100 cubic feet/minute of air is moved through the box. This specification allowed the computers to operate within their nominal temperature range under the expected race conditions. Additional fans are used to overpressure the electronics enclosure to reduce the intake of dust and other contaminants.



**Figure 37. Shock isolation components that support the electronics box.**

The electronics box is suspended atop the vehicle platform on twelve coil over damper struts. Suspension lowers the natural frequency of the electronics enclosure so that sensors can be rigidly mounted and still provide useful measurements of the environment. It also minimizes the shock dose to the computers and electronics so they are protected from severe accelerations. Figure 37 shows the configuration of shock isolation components that support the electronics box. The orthogonal configuration of the shocks decouples the vertical and horizontal forces, simplifying the design and tuning of spring and dampening constants. It also allows translations in all directions without inducing

any rotations. The suspension configuration also leaves the driver cockpit area open, which is important for logistical reasons.

## ***Computation***

Sandstorm's computing system was designed to ensure that computational power would not limit performance. Sandstorm's onboard computing includes a quad processor Itanium II, three dual processor Xeon rack mount computers, and four Pentium III class PC-104 computers. The Itanium II and Xeon computers provide the high performance computing necessary to perceive and navigate at high speeds but provide little support for interfacing to sensors and actuators. In contrast, the PC-104 computers provide sufficient processing to act as sensor and actuator interfaces and have the necessary inputs and outputs to control actuators and interact with sensors.

The Itanium II processor was intended to perform faster than real-time dynamic simulation of Sandstorm's motion over terrain. The simulation was to be used as the feed-forward portion of a randomized dynamics cognizant planning algorithm [Urmson03]. This planning scheme was not completed so the Itanium II was not used in the race day configuration.

The dual Xeon computers interface with Sandstorm's sensor suite, and provide processed data products to be used by the terrain evaluation and path shifter process(es). The various sensor interfacing processes were distributed across these machines to balance I/O and processor bound tasks where possible.

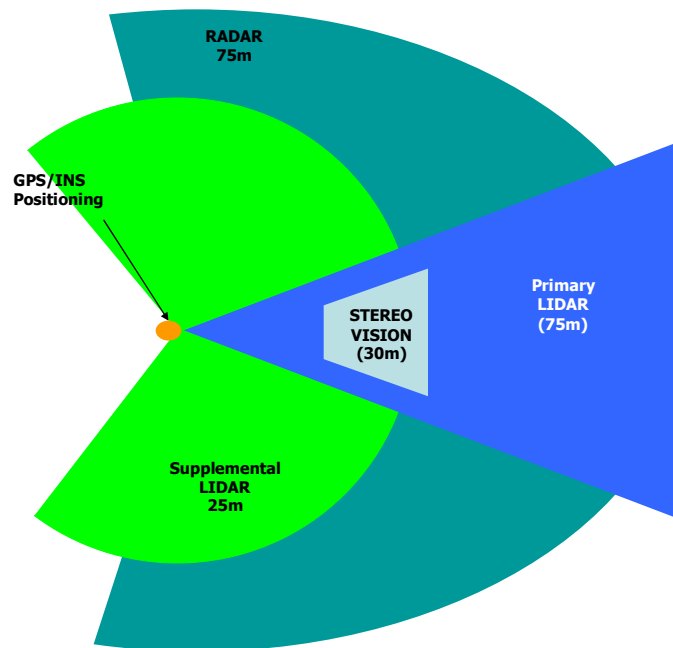
The Pentium III class PC-104 computers are used to perform embedded control tasks throughout Sandstorm and as an interface to the high-speed stereo vision system. For each of the embedded control tasks, the same set of processor and input/output boards were used. This facilitated the reuse of code and the sharing of software implementation knowledge among systems.

These eight computers and the various Ethernet interfaced sensors are connected using bridged, gigabit Ethernet. By utilizing a gigabit Ethernet in a

bridged, star configuration, any point-to-point high bandwidth communication has minimal effect on throughput or latency between other nodes on the network since the network traffic is only seen by the computers involved in the interaction. In this way, the throughput of the network is maximized allowing for the efficient transfer of information between processes in the system.

## ***Sensors***

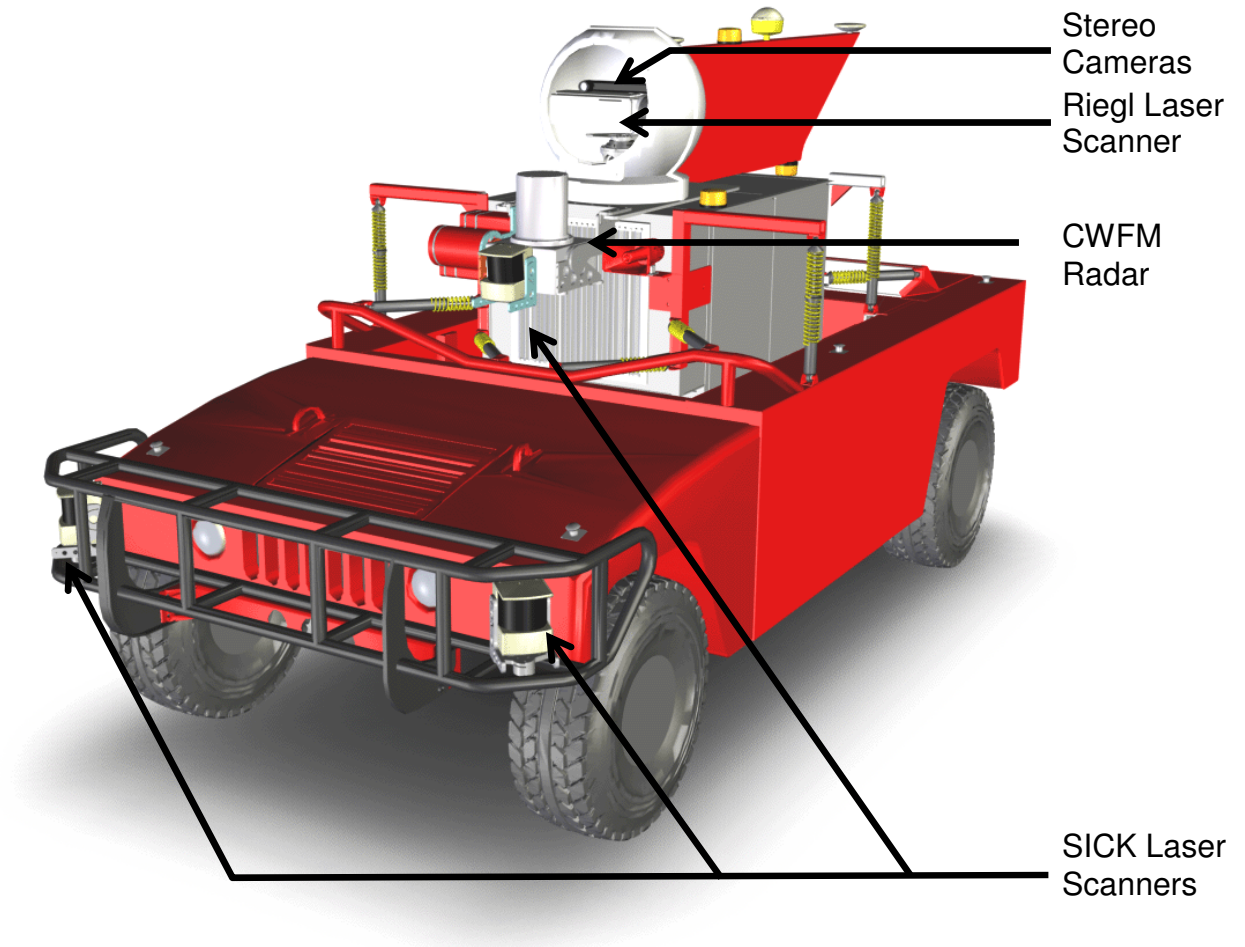
Sandstorm was designed to provide all weather sensing and driving capabilities. To achieve this goal, the onboard sensor suite incorporates a variety of sensing modes with overlapping ranges and capabilities. Sandstorm was designed to use a combination of LIDAR sensors, stereo vision, RADAR, and a combined inertial/GPS sensor for pose estimation. In practice, Sandstorm used only the primary LIDAR.



**Figure 38. Overlapping fields of view of Sandstorm's sensors**

Figure 38 illustrates the overlapping fields of view of each of these sensors. The Riegl LIDAR is used to profile terrain at long range, while the stereo system was intended to provide short to middle range dense terrain modeling. Short range

SICK laser scanners are used to provide secondary obstacle detection, and to assist in modeling underpasses and nearby walls in constrained operations. A RADAR was included in the sensing package to provide sensing in dusty conditions and to assist in the detection of robots and other moving vehicles. Figure 39 shows the mounting locations of these sensors on Sandstorm.



**Figure 39. Sandstorm with sensor placement indicated.**

## **LIDAR**

A Riegl Q140i scanning laser range finder is the primary navigation sensor. It provides a model of the terrain at relatively long ranges, is easy to integrate and has only a few, well understood failures modes. A limitation of scanned LIDAR is that it is generally possible to collect dense point data only in a single plane.

Flash LIDAR does not exhibit this problem, but operates over insufficient range to be useful at speed. Two axis mechanically scanned LIDARs have reasonable range, but cannot scan rapidly in both axes and thus do not provide significant benefit over single axis scanners for this application. Sandstorm's Riegl LMS Q140i Airborne line-scanner operates with a 60-degree field of view, 30KHz pixel rate, and has a specified line-scan period of 20ms (50Hz).

Three SICK LMS laser scanners are used to provide short range supplemental sensing. Two are mounted in the front bumper, providing low, horizontal scans over a 270° wedge centered in front of the robot. These sensors can detect obvious, large positive obstacles. A third SICK LMS laser scanner is mounted to provide profiling of over hanging obstacles, and to detect airborne dust but was not used during the race.

## **Stereo Vision**

To complement the low density, long range LIDAR system, Sandstorm incorporates a high-speed stereo vision system. The stereo system utilizes the DeepSea Stereo engine which is a hardware implementation of the CENSUS disparity matching algorithm [Woodfill97]. The system is capable of achieving frame rates of ~120Hz on 512x512 images. The entire stereo vision processing unit comprising of a Pentium III based PC104 stack and the stereo engine are mounted in a ruggedized aluminum casing on the electronics box.

The stereo vision system processes images generated by a stereo camera pair mounted on the stabilized gimbal. The stereo system is therefore pointable and can be directed to look at particular areas of interest. Difficulties in generating dense, accurate terrain models under varying lighting conditions prevented the stereo system from being fully integrated with the navigation system.

## **RADAR**

Both Stereo vision and LIDAR can have difficulties sensing in dusty environments. RADAR operates at a wavelength that penetrates through dust

and other visual obscurants but provides data that is more difficult to interpret. Objects are detected by finding amplitude peaks in the frequency shifted return signal. The amplitude of the signal can be used to estimate the size, and thus significance, of an object. This process can be confounded due to surface properties of the object and the orientation of the object relative to the transceiver.

A NavTech DS2000 Continuous Wave Frequency Modulated (CWFM) radar is used as a complementary sensor. The sensor is capable of 360° scanning, at 2.5 Hz and provides an Ethernet interface to the RADAR range measurements. Our mounting of the sensor limited its field of view to approximately 220°. The radar utilized on Sandstorm has a specially designed antenna that generates a beam that is tilted down 3.22°. This down angle was selected to cause the 4° vertical beam width to cover a range from roughly 24-100m from the antenna. The RADAR was not integrated with the navigation software due to difficulties extracting noise free data.

## **Pose Estimation**

Reliable and robust position sensing is essential for pose based navigation. Sandstorm uses an Applanix POS-LV position estimation system that fuses inertial and differential GPS position estimates through a Kalman filter [Scherzinger00]. The output estimate maintains sub meter accuracies, even during extended periods of GPS dropout. The POS-LV system also provides high accuracy angular information by combining techniques for carrier wave differencing with inertial sensor measurements. The POS-LV system outputs a pose estimate over a high-speed serial link at a rate of 200 Hz. This constant stream of low-latency pose information simplifies the task of integrating the various terrain sensor data sources.

## ***The Gimbal***

The ability to interpret data from any sensor can be severely hampered by pitching and rolling induced by robot motion over terrain. In particular, the

performance of a single axis scanning LIDAR is particularly affected by mechanical excitation in the pitch axis (see Chapter 5). When sensing at reasonably long ranges, even small pointing errors can result in a dramatic change in the location where the sensor's beam intersects the terrain. Because of this, range data associated with small obstacles or terrain details at distance become difficult to interpret, and the overall perception performance is severely degraded.

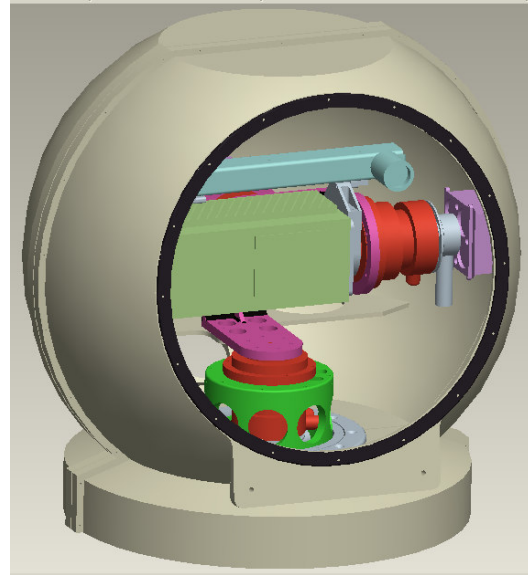
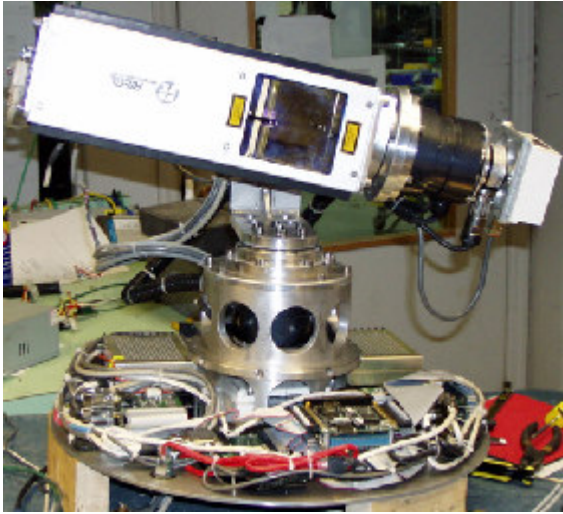
The passive stabilization system discussed previously smoothes these motions and reduce the principal frequency of terrain inputs, but is unable to completely remove the disturbances. Sandstorm incorporates an actively stabilized gimbal to further reduce these effects.

## **Implementation**

The relatively low-mass active LIDAR and passive stereo vision sensors are co-mounted to provide a common field of view and reference frame. The sensor mounting design aligns the LIDAR's optical aperture with the center of the gimbal and balances the mass distribution around the rotational center. Each axis includes minimal-mass components and low complexity gimbal support structure with the design goal of minimizing the moment of inertia. By minimizing the moment of inertia, the overall responsiveness of the gimbal is increased, yielding better perception data.

Each axis use common components and design elements including a harmonic drive (HD) actuator, incremental and absolute position encoders, a fiber-optic gyro (FOG), and an optical switch based limit detection sensor. Each gimbal axis assembly is designed for electrical and mechanical simplicity.

The entire gimbal mechanism is enclosed within a protective shell to prevent water and dust from damaging the mechanism and electronics. The sensors operate through a front window with specially coated optical glass. The fully assembled gimbal is shown in Figure 40. Table 4 outlines the gimbal performance capabilities and summarizes its interfaces.



**Figure 40. The gimbal.**

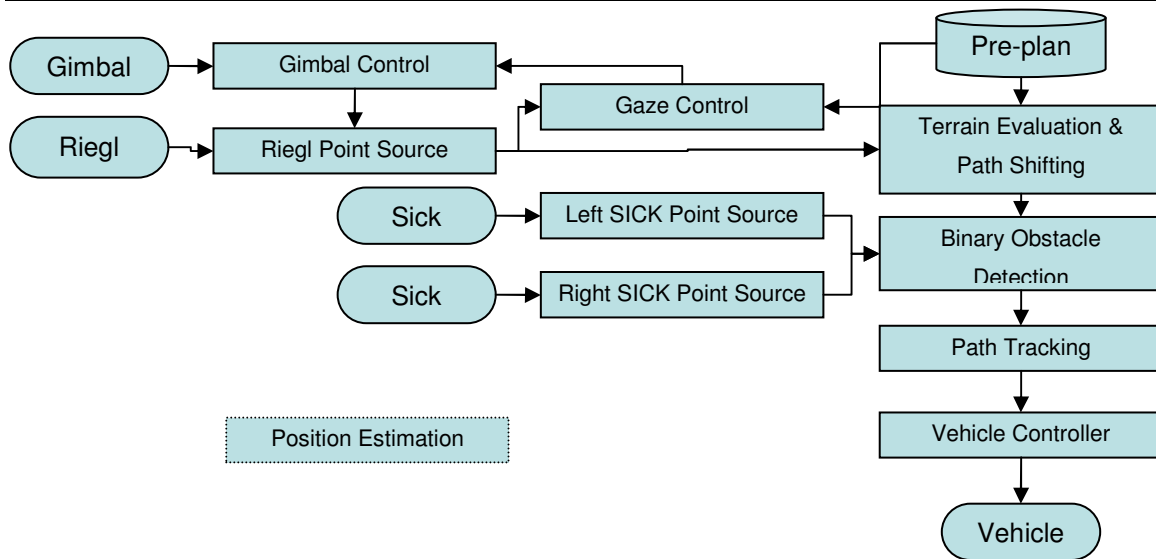
Property	Value
Payload Complement	High Resolution LIDAR line scanner and stereo vision head
Payload Dimensions	24mm x 25mm x 500mm
Payload Mass	12+ Kg
Gimbal Mass	< 25 Kg
Computation	Pentium III PC104 stack
Communication Interface	100Base-T Ethernet
Power Consumption	400W
Enclosure	Water resistant lightweight shell with optical window with pass for 300–900nm
<b>Pitch</b>	
Range of Motion	$\pm 40^\circ$
Angular Velocity	6.28 Rad/s



Angular Acceleration	863.63 Rad/s <sup>2</sup>
<b>Roll</b>	
Range of Motion	$\pm 40^\circ$
Angular Velocity	6.28 Rad/s
Angular Acceleration	73.89 Rad/s <sup>2</sup>
<b>Yaw</b>	
Range of Motion	$\pm 90^\circ$
Angular Velocity	6.28 Rad/s
Angular Acceleration	25.59 Rad/s <sup>2</sup>

**Table 4. Gimbal performance characteristics.**

## Navigation Software



**Figure 41. On-board software architecture.**

## Overview

Navigation software drives Sandstorm by combining incoming sensor data with the pre-planned path. The pre-planned path is loaded into the terrain evaluation and path shifting module prior to execution (see Figure 41) . The path is also passed to the gaze controller which monitors the vehicle's position and

independently commands the gimbal controller to point the laser scanner in the direction of anticipated driving. The Riegl point source communicates with the Riegl laser range finder, receiving laser line scans at between 15 and 20 Hz. This range data is then transformed into a vehicle coordinate frame referenced point cloud. Data is passed to the terrain evaluation and path shifting algorithm which makes adjustments to the path Sandstorm is following.

Once the terrain has been evaluated and the pre-planned path has been shifted, the modified path is passed to the binary obstacle detection process. This process uses the bumper mounted short range laser scanners to detect if any object in the environment is an immanent threat. If an obstacle is detected, Sandstorm is commanded to stop. This is characteristic of traditional safeguarding within the stop-limited regime. Racing calls for aggressive driving at speeds beyond the stopping distance constraint. Hence the stopping behavior is only intended to be run during testing and the qualification phase of the race. The algorithms can potentially produce false positives, which command stopping, not avoidance of detected obstacles at very low speeds.

Once obstacles have been taken into account, the resultant path is passed to a pure pursuit path tracker. Curvature and velocity commands are then passed from the path tracker to the vehicle controller which interfaces with the robot hardware. Figure 41 illustrates this data flow. The following sections describe these components in more detail.

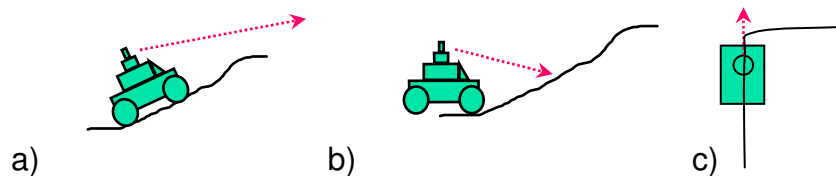
## ***Infrastructure***

Sandstorm uses architectural and communications tools that enable algorithm developers to view the system through abstract, reconfigurable interfaces. During initial development and ongoing debugging the interfaces for an algorithm can be configured to read data from time-tagged files using a common set of data access tools. As an algorithm matures, interfaces are reconfigured to use a common set of interprocess communication tools which integrate the individual algorithm into the larger system.

## ***Gaze Control***

The gaze control module controls the gimbal to point the primary sensors to provide information about the world that a non-pointed sensor would not be able to provide. A non-pointed sensor loses sight of the road when moved over hills and around corners. A pointed sensor can be pitched up, down, left, and right to fill in data that would otherwise be missed.

As shown in Figure 42a, as the vehicle crests a hill, a non-pointed sensor that is aimed to see flat ground at a reasonable distance will be looking out into the sky. Similarly, as shown in Figure 42b, as the vehicle approaches the bottom of a hill a non-pointed sensor will repeatedly sense the bottom of the hill rather than looking ahead up the hill. Immediately before rounding a corner, non-pointed sensors look out beyond the road rather than in the direction the robot is about to travel (Figure 42c).



**Figure 42. Configurations where non-pointed sensors perform poorly.**

Inputs to the gaze control module include the current state of the vehicle, the current pointing of the gimbal, the desired path ahead of the vehicle, and data from the primary navigation sensor. Roll and pitch are stabilized while yaw is pointed. High frequency pitching and rolling is attenuated by the gimbal control software. High frequency yawing does not occur due to the mass of the vehicle relative to the magnitude of input from the ground.

To simplify control of the pointed sensor, control of the pitch and yaw axes are decoupled. Roll is not pointed. This separation generally works well, as it is rare to see hills that are very steep combined with roads that turn sharply enough to

prevent this mode of control from working. Paths are fed to the gaze control module as a series of  $x,y$  pairs in the global coordinate frame. A “look-ahead distance,” for directing sensor attention is determined by the current robot speed.

### ***Gimbal Control***

The gimbal control software accepts vector pointing commands and also accepts commands for how fast the mechanism should servo to this vector. Commands can be issued at up to 10Hz without disrupting operation. By default, the gimbal control software stabilizes all three pointing axes but controlling software can disable this functionality on each of the axes independently. To eliminate gyro drift, the controller incorporates data from the onboard absolute pose estimation system.

### ***Terrain Evaluation and Path Shifting***

The terrain evaluation and path shifting software is the heart of the onboard navigation system. It was developed after the disclosure that a majority of the Grand Challenge route would be on trails or poor dirt roads. The module is designed to smoothly switch between sensor based navigation and blind navigation as onboard sensor data is determined to be valid or invalid, respectively. The pre-planned path is adjusted by a control-law-like navigation algorithm to ensure a smooth output path.

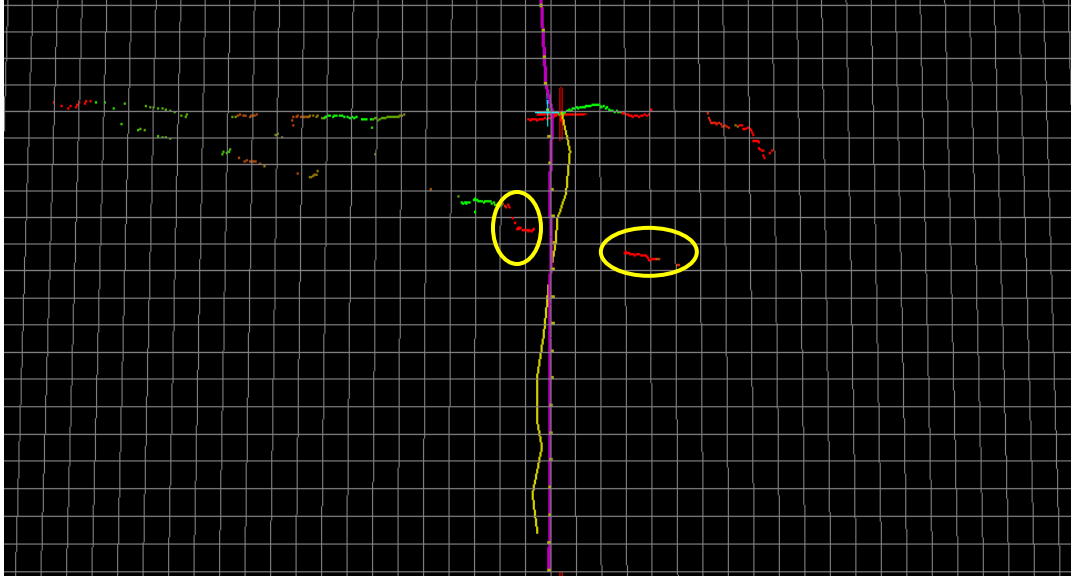
#### **Terrain Evaluation**

Terrain evaluation uses a statistical evaluation technique akin to that used in the Morphin/Gestalt algorithms [Simmons95, Singh00, Urmson02, Goldberg02]. For each laser line scan, the evaluation returns a vector of traversability scores, one for each point in the scan, indicating how safe it would be for Sandstorm to drive over that portion of the terrain.

The analysis is performed over vehicle width point sets centered at every point in the line scan. Using linear regression, a line is fit to this set of points. The slope of this line with respect to the gravity vector is used as one part of the cost

measure (the steeper the slope, the less desirable the terrain is for driving). A second measurement of the cost of the terrain comes from the residual between the point set and the line fit; if the residual is large, the terrain is irregular, and likely unsafe for driving. Finally, a measure of the slope of the line relative to a vector in the direction of travel is calculated. If the gimbal is pointed in the direction of travel, this measure can be used to detect vertical relief in the terrain. Unfortunately, this measure will consider terrain seen with the laser pointed in a direction other than along the direction of travel as unsafe. A better approach would be to consider the slopes of various windows of the line scan relative to the slope of the line generated by intersecting the plane of the laser scan with the nominal ground plane; this second approach would avoid the off-pointing problem.

The three cost measures (two slopes and the residual) are scaled to have a value between zero and one based on the capabilities of the vehicle. A cost of zero represents safe, and a cost of one represents unsafe. The worst cost value is returned as the cost for the vehicle to travel with its center over a particular laser point. Figure 43 shows an example of this analysis. The set of roughly horizontal lines in the figure represents a laser scan intersected with the terrain. Green regions are considered safe for Sandstorm to traverse while the red regions are considered to be unsafe. The circles mark two detected fence posts.



**Figure 43. A sensor view of an opening in a fence.**

Once a line scan has been evaluated, the costs are clustered into regions. The clustering uses the inherently ordered nature of the line scanner, so that very few distance calculations need to be performed. An initial pass through the line scan finds the lowest cost. If this cost is below a floor of acceptable terrain traversability, the floor cost is used to clip regions; this floor cost is set to be the expected limit of “easily traversable” terrain. If no point in the array has a cost less than the floor cost, the lowest cost in the array is used as a seed to clip regions. Regions are also clipped to remain within the route corridor specified in the pre-plan.

Though similar to the Morphin algorithm, this single scan approach has several distinct advantages and disadvantages. Since this process only requires a single scan of range data, not a two dimensional field of data, the latency in the terrain evaluation is much less than if Morphin were run over a series of line scans. Furthermore, performing the analysis within a single scan reduces the required accuracy of inter-scan registration. The single scan evaluation is unable to detect obstacle with flat faces perpendicular to the direction of the LIDAR scanner. This problem could be corrected by using multiple line scanners with different viewing angles, and then combining the result in a composite map. The cost evaluation is also dependent on the yaw angle of the sensor, resulting in

invalid terrain analyses when the sensor is pointed away from the robots direction of travel.

## Path Shifting

The path shifting module is a corrected version of the preplanned path. The path shifter incorporates sensing data from the terrain evaluation module to change the path so that it is safe. This is done by applying a control law like algorithm to adjust the path offset at each way point. By control law approach, it is obvious how to adjust parameters to adjust the navigation performance

The navigable region of each scan line is determined by the terrain evaluation algorithm, and the signed perpendicular distance from the pre-planned path to the center of this region is used as the input to a controller servoing the path:

$$o_{k+1} = o_k + K_p d_s + K_d \dot{d}_s$$

For a waypoint in the path, its offset ( $o_{k+1}$ ) from the preplanned path is calculated as the offset of the previous point in the path ( $o_k$ ) plus the distance to the center of the navigable region provided by the sensing system ( $d_s$ ) and the change in this measurement since the last waypoint was processed ( $\dot{d}_s$ ), adjusted by proportional and differential gains. By adjusting the control gains, , it is possible to dampen response to noise sensor readings, and to thus achieve stable driving.

If the pre-planned path has smooth curves, the output of the path shifter will also be smooth. During the initial testing of the path shifter algorithm, the gimbal control code and pointing code were not yet reliable, so a pair of simple filters were used to throw out laser data should the gimbal be pointed incorrectly. The first filter calculates the distance between the center point of the current navigable region and the center of the navigable region used from the last scan. Data is discarded if it is not within a window (one to three meters). The second filter checks to ensure that the current region center is within a cone with its apex

at the center of the previous region. This second filter insures that the laser is not excessively off-pointed.

Referring to Figure 43, the dots along the vertical purple curve show the preplanned path. The purple curve represents the path that is passed from the path shifter to the path tracker. The yellow curve represents the time history of the center of the safe region closest to the pre-planned path. Note that after it passes the current laser scan, the route fades back towards the pre-planned route. This combination of the pre-planned and sensor planned route is required so that in instances where there is sharp acceleration and the pure-pursuit path tracker exceeds the laser look-ahead distance momentarily, there is some path for that software to track.

### ***Binary Obstacle Detection***

The binary obstacle detection module detects moving obstacles. The obstacle detection algorithm builds an occupancy grid model of the world using data from a single pair of scans from the forward looking SICK LIDAR scanners. If any cell in the grid receives more than a specified number of hits, it is considered to be occupied, and thus an obstacle. The algorithm then checks to see if any of the obstacle cells are within half a vehicle's width of the adjusted path and are within some threshold distance of the robot. If these conditions are both true, then the desired velocity for the output path is set to zero. This causes the robot to stop, preventing a collision. Since no state is maintained between laser scans, if the obstacle is removed, Sandstorm will detect an open path and continue.

This simple algorithm will incorrectly classify gently sloping hills as obstacles. Thus, during normal operation this module is removed from the data stream, and the adjusted path is passed directly from the terrain evaluation and path shifter module to the path tracking module. However, in tightly constrained conditions such as those associated with the Grand Challenge qualification course, the algorithm works reliably.



## ***Path Tracking***

A pure-pursuit waypoint tracking algorithm [Amidi90, Coulter92] executes the planned route. The pure-pursuit algorithm works by commanding a steering arc that will intersect the planned path at some look ahead distance in front of the robot. The look a head distance of the tracker is adjusted dynamically based on speed. A fixed set of control gains are configured to provide a balance between good performance at both low speed in tight maneuvering situations, and at high speeds on straight-aways and soft corners.

## ***Vehicle Controller***

All of the software that interfaces with the motion components of the vehicle is contained within a single PC-104 stack. This stack has four functions, gear shifting, steering, velocity control and emergency stop.

All control loops operate within a custom scheduler running on a stock Redhat 9 kernel. This choice was justified since the amount of jitter and scheduling delay is generally insignificant when compared to the mechanical time constants associated with the vehicle. The software is structured so that descendant classes may increase the capabilities of the drive by wire system. Each descendant class overrides the “loop” function, to provide a capability. This software structure allows for a logical decoupling of each of the control functionalities.

## ***Gear Selection***

The HMMWV’s automatic transmission greatly simplifies gear selection. In practice the onboard software selects between 3 gears: drive, neutral and reverse. Testing showed that the velocity control loop functions equally well in any of the automatically selected gears so there is no need to actively constrain which gears should be selected by the transmission (via shifting to 1<sup>st</sup> or 2<sup>nd</sup> gear directly). Testing was not exhaustive over all possible terrains, road surfaces or

speeds, so further experimentation is necessary to understand the range over which “gear-ambivalence” is viable.

The electrical interface to the linear actuator controlling the shifter consists of a pair of computer controlled relays. Since the smoothness of the control activity is unimportant, it is possible to avoid adding other control electronics. The position of the actuator is provided via a potentiometer. This interface lends itself to a bang-bang control approach. The control loop servos a linear actuator until it is positioned at a set of pre-calibrated locations that represent the detents for each of the gear shifter positions.

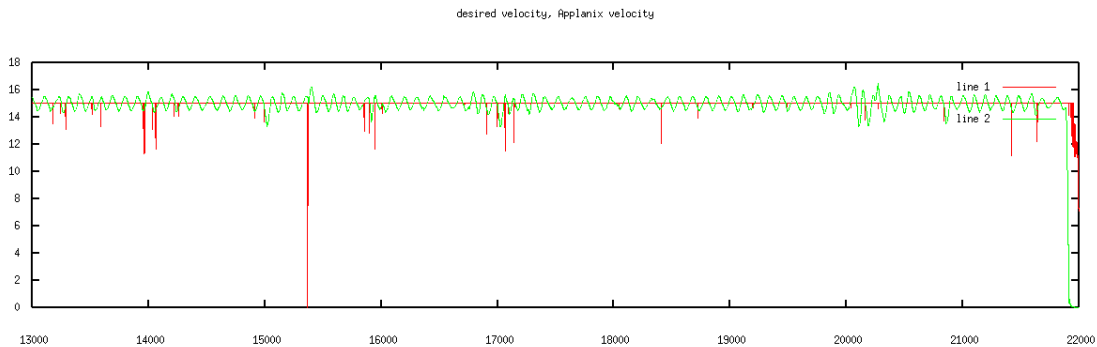
### **Steering Control**

The electrical interface for the steering control system uses an off the shelf motor controller (an AMC-DR100EE series) to drive a DC harmonic drive actuator mounted to the steering column. The motor controller provides servo velocity loop that is interfaced to the drive-by-wire computer through an analog control line. Position control feedback is provided by a rotary variable differential transformer (RVDT) mounted on the output of the power steering box. The measured angular position is then used in a classical PD control loop, operating at 20 Hz, to set the speed and direction of rotation of the steering actuator.

### **Velocity Control**

To control vehicle velocity, a single actuator is used to either press the brake pedal or pull on the throttle cable. The actuator is driven by a DC amplifier controlled by a differential signal from the drive-by-wire PC-104 stack. The Applanix POS-LV system provides an accurate measure of velocity that is used as the control input signal. Velocity control is implemented using a PID like servo loop and provides accurate speed control ( $\pm 0.5$  m/s with an oscillation period of 0.1 Hz) with relatively short rise and drop times, over a range of velocities ranging from 5-22 m/s (11mph- 49mph). Figure 44 shows the controller’s response on a cross country run during testing in Nevada. The vertical axis in

this figure indicates velocity in meters per second, while the horizontal axis units are controller ticks, which are approximately equal to 0.1 seconds.



**Figure 44. A plot showing the response of Sandstorm's velocity control loop.**

## Emergency Stop

The emergency stop system incorporates a variety of safeguards to ensure that, should sub-components of Sandstorm fail, the robot can be reliably stopped. The throttle control system has an integral heart beat circuit that causes the vehicle service brakes to be fully applied should the drive-by-wire computer system fail to maintain real time performance. The engine ignition and fuel pump are also wired into the kill system such that should computing and/or actuator power fail, the HMMWV engine is turned off and fuel flow stopped. Furthermore, this system also causes the HMMWV parking brake to engage, causing the robot to stop faster than relying on rolling friction alone.

If Sandstorm needs to be stopped or paused during operations where the computing system remains operable, any operator requests for a kill are understood by the drive-by-wire system. These E-stops result in the service brakes being applied. In this state, the onboard drive by wire computer disregards other drive commands until the e-stop signal is cleared.



**Figure 45. Sandstorm testing in Nevada.**

## **Field Experiments**

---

Sandstorm was tested and demonstrated in the Mojave Desert (Figure 45). Testing and shake-down took place at the Nevada Automotive test center where there is an abundance of trails and off-road terrain with significant pre-existing testing infrastructure.

### ***Desert Testing***

Our Nevada test facility provides a wide range of relevant test trails(Figure 46). The paved and unpaved ovals provide diverse conditions for initial debugging and development, as well as nominal road testing. The calibrated test courses provide a variety of rugged terrain conditions for testing vehicle stability, clearance and sensor stabilization. The sand serpentine courses represent challenging, high curvature routes with significant elevation changes. These routes have particular relevance to performance testing for navigation over hilly terrain where switchbacks are used to limit slopes on trails. The center also

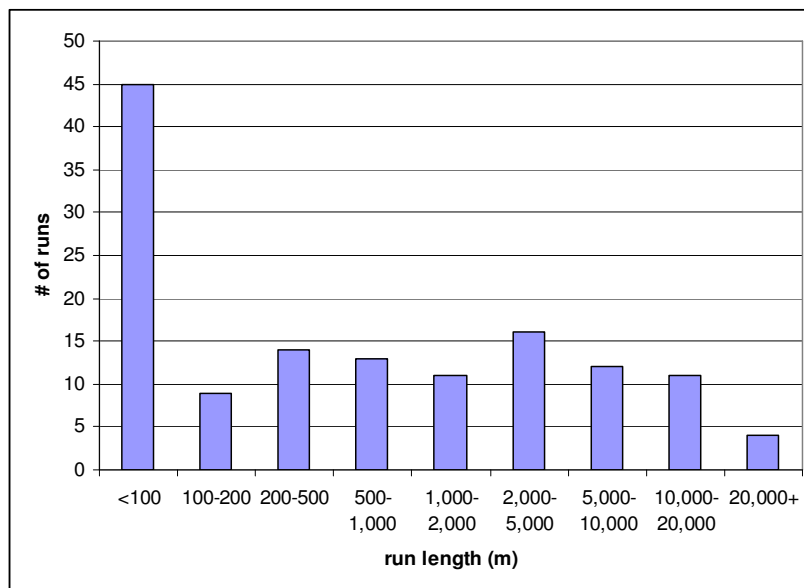
enables testing access to nearby public land which is riddled with relevant back roads and trails.

During a month of testing Sandstorm performed over 440 km of autonomous testing at an average speed of 8.87m/s. Figure 47 is a histogram of autonomous run lengths. Figure 48 shows a scatter plot of range versus speed. For the purposes of these plots, a run may be punctuated by remote e-stops or pauses, which count against the average speed for that run. For a run to be counted as continuous, the navigation algorithms must not have been restarted no other human intervention (other than the e-stop release) is allowed.

While both of the distributions are weighted towards slow, short distance traverses, this is not unusual during developmental testing. The interesting characteristic of these distributions is the significant number of high-speed long duration runs.

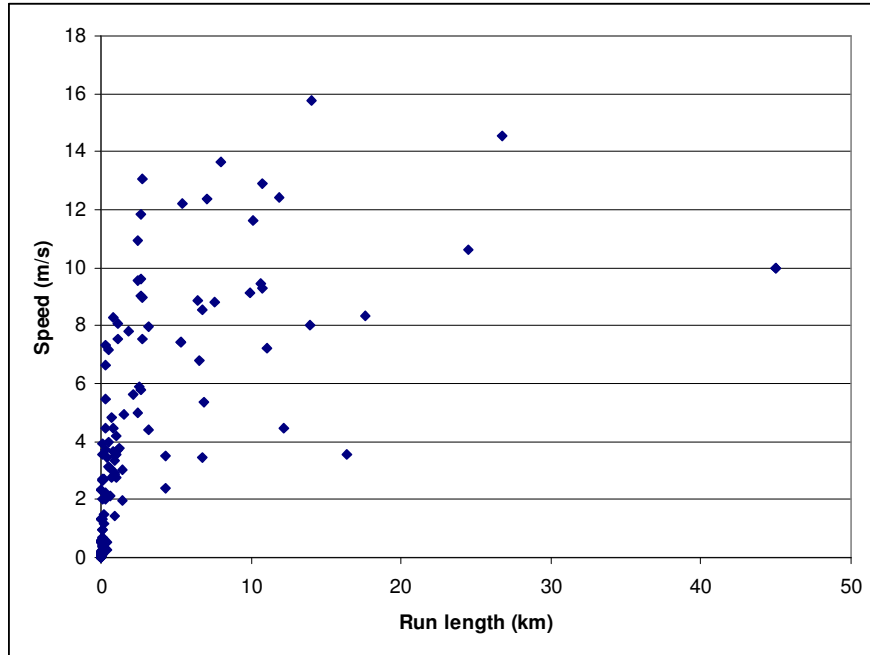


**Figure 46. Test facilities at the Nevada Automotive Test Center.**



**Figure 47. Histogram of run length distribution.**





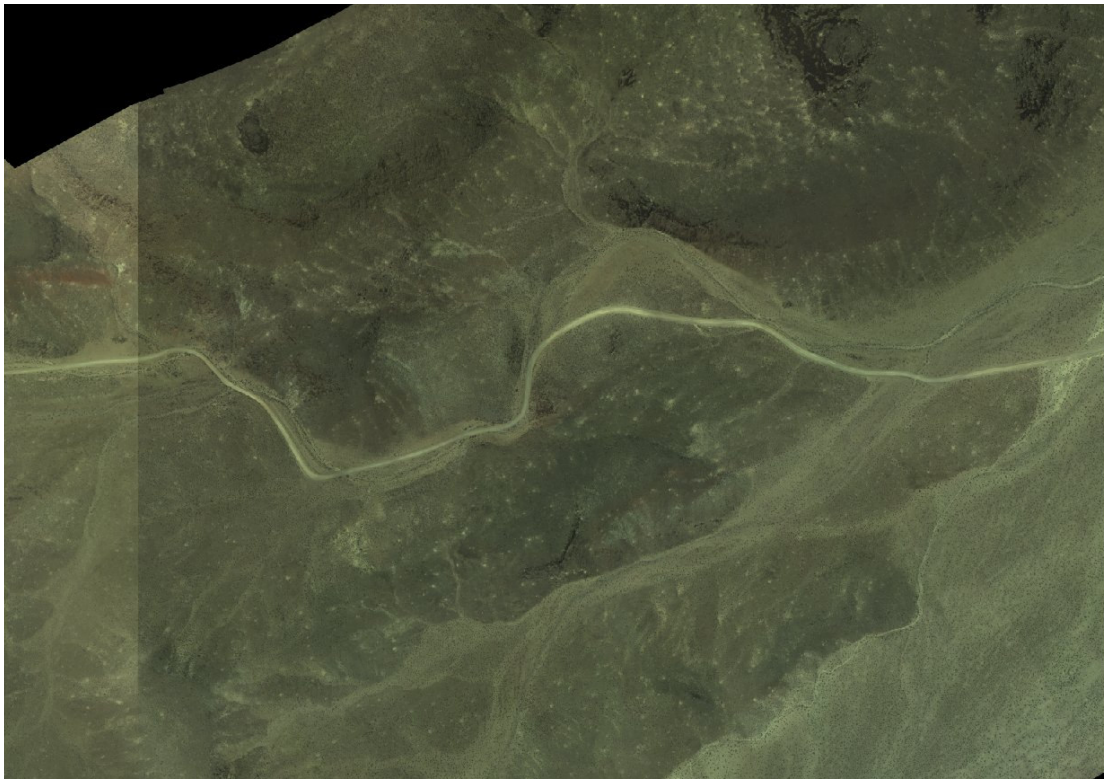
**Figure 48. Scatter plot of run length versus speed.**

Several of the long runs were achieved on Simpson road, a nearby public access way that contains sections of varied surface ranging from dirt road to trail. Simpson road was selected as a test environment because of the variety of road surfaces and interesting topography. The road ranges from a soft chalk bed to crushed shale and coarse gravel. These different materials provide varied sensing and control environments that test a broad range of Sandstorm’s capabilities.

The route cuts through varied terrain ranging from softly rolling hills to a narrow canyon. In places, there are severe drop-offs within a meter of the road. The canyons (see Figure 49 and Figure 50) provide both a complex sensing environment (due to both geometry and lighting) and a multi-path environment to verify drop out performance of GPS-based pose estimation.

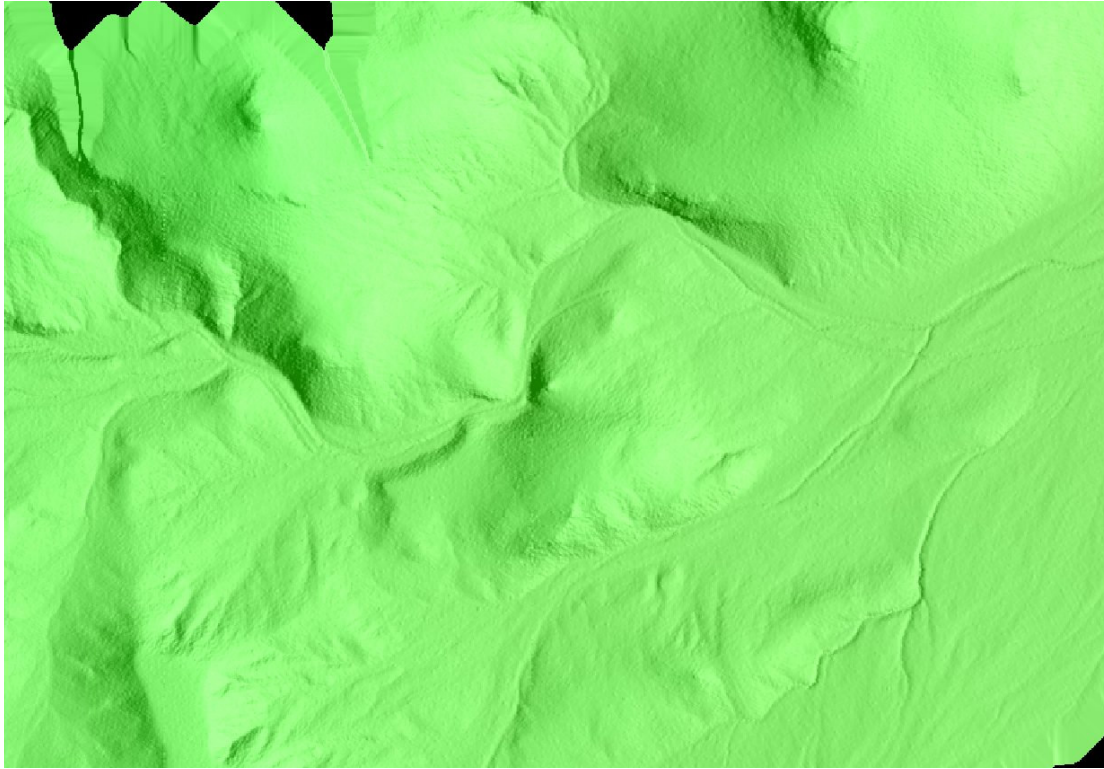
The two longest runs of the desert trials occurred on Simpson road. Sandstorm performed an “out-and-back” run, 45km each way. These experimental runs were performed with a small number of emergency stops. The return leg

averaged close to 8.3m/s. In both cases, the runs were terminated when Sandstorm reached the end of the trail. While these runs represent significant traverses, they did not achieve the distance required to complete the Grand Challenge event. Due to a lack of technological readiness and time, Sandstorm was never tested over durations equal to that required by the Grand Challenge.



**Figure 49. Aerial photograph of Simpson Road canyon.**

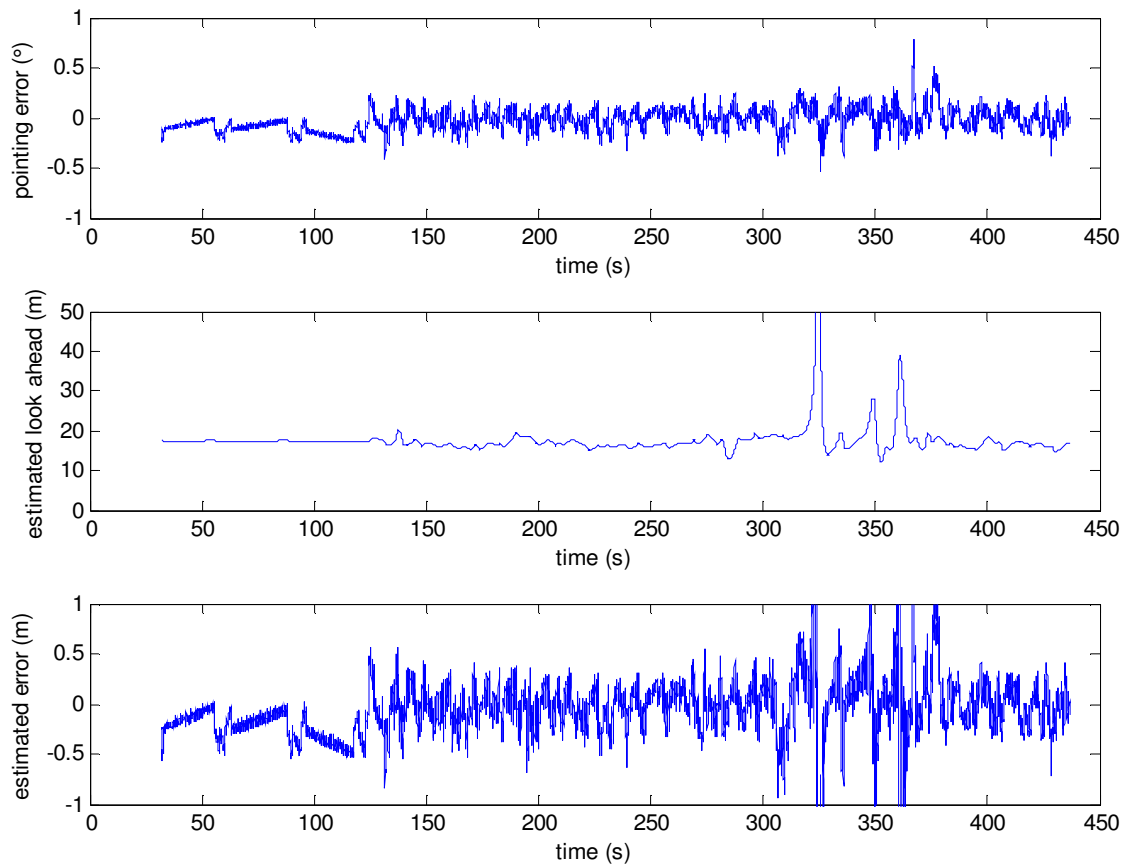




**Figure 50. Digital elevation model of Simpson Road canyon.**

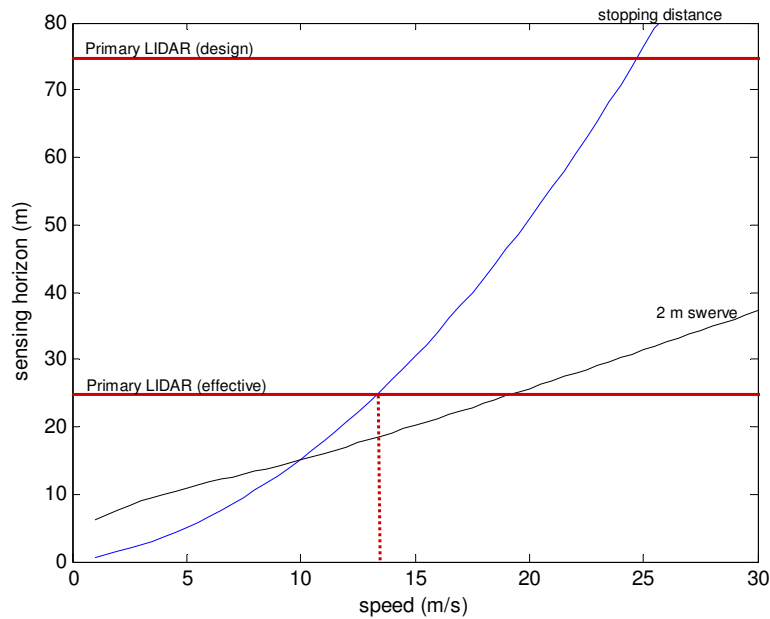
During desert testing, the stabilized gimbal was under constant development and evaluation. Figure 51 shows representative pointing performance for a test run along Simpson Road on February 23, 2004.

At reasonable short sensing ranges (corresponding to low speeds), the estimate of the error in the sensing range exceeds 40cm. For the navigation system described here, this pointing error was not a significant problem. This error would represent a significant problem for navigation algorithms that utilize a high fidelity world model.



**Figure 51. Gimbal pointing performance.**

Sandstorm's sensors were originally selected to enable high speed driving with stopping distance safeguarding (as can be seen in Figure 52). Technical problems and development schedule resulted in Sandstorm operating with only its primary LIDAR, at a relatively short effective range of 25m. This short effective range meant that in many situations Sandstorm was operating within the swerve-limited regime. Figure 48, shows a number of runs where sandstorm was operating for tens of kilometers at average speeds in excess of 12m/s in natural, unrehearsed terrain. These data points demonstrate the viability of swerve-regime navigation.



**Figure 52. Sandstorm's Primary LIDAR sensing horizon.**

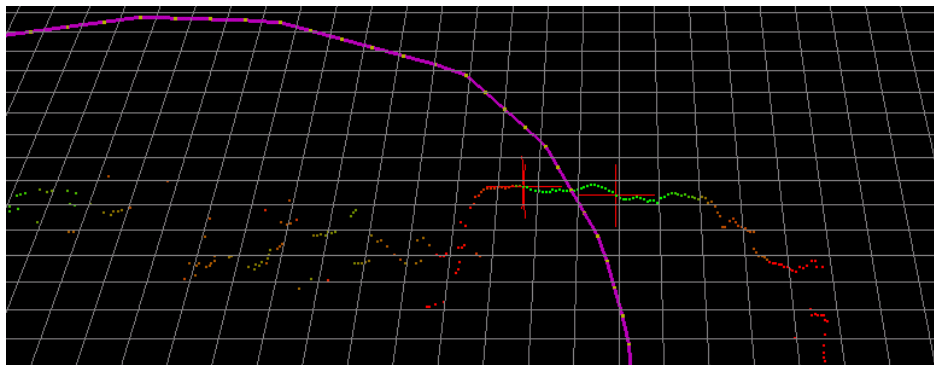
## ***Grand Challenge Performance***

Sandstorm did not complete the Grand Challenge, traveling only 11.9 km along the route, but at a pace that would have finished the race within the required time. Sandstorm averaged over 24 km/h (6.7m/s), obeyed lower speed limits where specified, and hit a sustained peak speed of approximately 58 km/h (16.1m/s) while generally demonstrating smooth, stable driving over off-road terrain. During 25 minutes of operation, Sandstorm traveled faster and farther than any other entry.

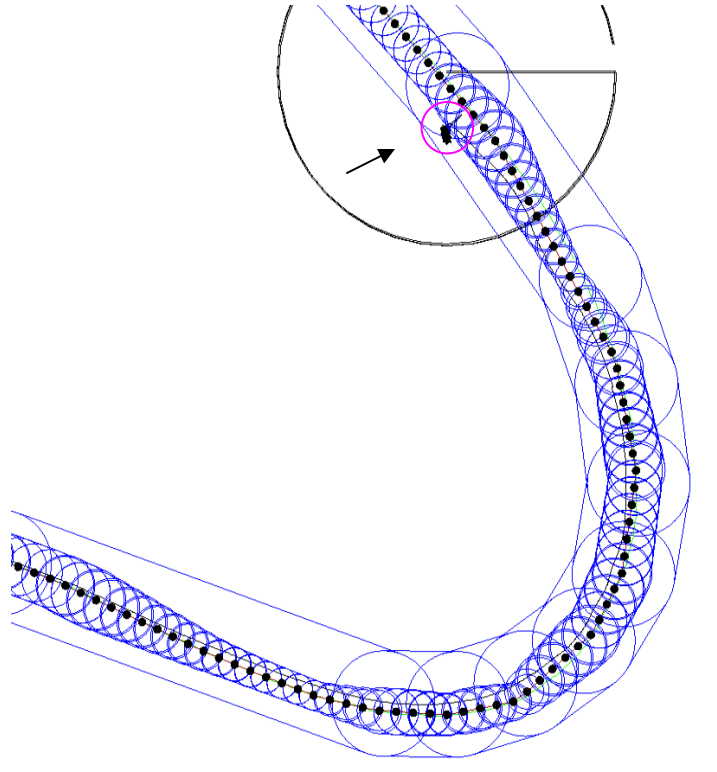
Sandstorm's run finished part way through a tight switch back. While making the turn, Sandstorm cut toward the inside of the corner and the left side wheels slipped off trail. The robot continued along the berm until it came into contact with a large rock buried in the soil. Sandstorm melted the rubber off of its front tires in a futile attempt to extract itself from the berm. Eventually the crew in the chasing control vehicle triggered a disable emergency stop, which caused the front service brakes to clamp down on the inboard side of the four half-shafts.

This sudden stop, combined with the momentum carried by the fast spinning front wheels, caused both front half-shafts to snap.

This failure was a combination of several factors. Entering the corner, the onboard navigation system began to erroneously ignore the processed laser data due to the sharpness of the corner. Figure 53 shows that even though the data was disregarded, the classification of the terrain from the laser scan was still reasonable. The plateau in the middle of this image correctly identifies the location of the trail within the sensor view.



**Figure 53. A sensor view of the trail in the Hairpin.**



**Figure 54. Pre-planned route around the hairpin.**

With the laser data filtered, the onboard planning system switched to following GPS blindly. It was at this point that Sandstorm began to cut towards the inside of the curve. Sandstorm's GPS measurement of the preplanned path was offset by a small distance towards the inside of the corner. In addition, the preplanned was incorrectly biased towards the inside of the corner. Finally, the pure-pursuit path tracking software also caused Sandstorm to attempt to cut the corner. These three effects combined to push Sandstorm roughly 1.5 to 2 meters to the left of the road center such that one wheel fell off the edge. Though Sandstorm came to rest at the location indicated by the arrow, the wheels had already slipped off the edge of the road by the middle of the corner. Figure 54 shows a plot of the pre-planned corridor (inner blue circles), pre-race reconnaissance (green) and Sandstorm's ground track (black). From this data, the path error seems to be due equally to each of the above mentioned sources.

With one wheel over the edge Sandstorm was unable to pull itself back onto the road before high-centering. Figure 55 shows Sandstorm's resting place, just after the tight hairpin corner. For scale, Sandstorm is roughly 2.25 meters wide.



**Figure 55. Sandstorm high-centered.**

## **Summary**

Sandstorm represents a new approach to high-speed off-road navigation performance. While sensors were originally specified to achieve stopping distance based look ahead, in practice it was a challenge to achieve useful sensing at this range on a moving platform. The controls based navigation algorithm in combination with the detailed pre-planning represent a first-cut implementation of a swerve-limited regime approach to off-road navigation.

Testing and demonstration illustrate the potential of a swerve-limited regime approach and realize state-of-the-art navigation performance in terms of duration and speed in unrehearsed terrain.

---

## **9 Representative Implementation: Efficiency-Limited Regime**

---

There are many implementations representative of the efficiency-limited regime [Bresina01, Goldberg02, Miller03, Schenker01]. Hyperion is selected as an archetype that is documented with state-of-the-art results in exploration navigation, pre-planning and duration. This implementation builds upon a many of the best ideas in planetary robotics and includes several novel technologies including a passively articulated steering mechanism and a temporal-spatial planner. Details are provided to help further the contrast between regimes and to guide future research.

### **Context**

---

Hyperion was developed to explore and demonstrate the concept of sun-synchronous navigation. Sun-synchronous navigation is an energy-cognizant strategy for planetary exploration [Shrounk95, Whittaker00, Wettergreen01] that utilizes knowledge of terrain, time, and rover characteristics to ensure sufficient solar energy is gathered while performing a mission.

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

Sun-synchronous navigation involves reasoning about sunlight: where the Sun is in the sky, where and when shadows will fall, and how much power can be obtained through various courses of action. It is accomplished by traveling opposite to planetary rotation, synchronized to the orbital motion of the Sun, to control exposure to sunlight [Shrounk95]. On the planets and moons, robots may employ sun-synchronous navigation techniques to acquire the solar energy



needed to sustain exploration for extended periods of time [Wettergreen01]. There are other potential benefits, such as ensuring good sensing conditions, and moderating thermal extremes by planning solar exposure. Sun-synchronous navigation is a potential means to provide for persistent, in some cases perpetual, presence to explore, dwell in, and develop regions of planets and moons.

Hyperion (Figure 56), a solar powered robot, was developed to demonstrate the idea of sun-synchronous navigation on Earth. In July 2001, Hyperion was deployed to Haughton Crater on Devon Island, above the Arctic Circle, to perform experiments in sun-synchrony. The culmination of these experiments was a 6+km, 24-hour sun-synchronous route, which Hyperion planned and executed successfully, traveling at speeds between 5 and 25 cm/s.



**Figure 56. Hyperion in Arctic Canada.**

After the successful mission to Haughton Crater, Hyperion was re-tasked to search for life in the Atacama Desert in Chile. In this mission, planetary



scientists at NASA Ames remotely commanded the robot to search for signs of life. The scientists sent coarsely specified waypoints and requested science activities (i.e. spectrometer samples, high resolution imagery, panoramic imagery). While this mission was not sun-synchronous, Hyperion did reason about terrain and solar energy.

Hyperion utilizes a combination of off-board pre-planning and a stereo vision based on-board navigation algorithms to autonomously traverse desert terrain.

## **Pre-Planning**

---

Hyperion is capable of performing on-board mission pre-planning. A mission planner generates paths that enable a robot with a fixed solar array to gather sufficient solar energy to complete its mission. In the case of a sun-synchronous mission, these plans take the shape of closed circuits over long periods of time, clocking with the sun. The planner reasons about the complex interactions between motion of the Sun in the sky, terrain, shadows, solar array pointing, energy consumption and finite battery capacity [Tompkins01]. Since Hyperion's solar array is body-fixed, the robot must synchronize its route timing with the position of the Sun to achieve array pointing that yields adequate energy for driving and other loads. To point the solar array at the sun, the orientation of a path must continuously change to match the angular rate of Earth rotation. On an idealized planar surface without obstacles, optimal paths are circular. In practice, terrain, shadows and intermediate goals (specifically science targets) typically prevent circular path solutions. Because non-circular routes prevent the solar array from always pointing directly at the sun, the specific choice of route and timing must be planned to maintain sun-synchrony and appropriate power levels.

The mission planner, called TEMPEST, searches through a space of position, time and battery energy to find sun-synchronous paths [Tompkins02]. It defines the sun-synchronous navigation domain in terms of a number of models that aid in computing costs for gross robot actions. At its core, TEMPEST uses the Incremental Search Engine (ISE) to determine the sequence of actions, and

hence trajectory in the 4-D search space, that is optimal in terms of navigation and energy criteria. ISE is a graph-theory based search algorithm optimized for planning and re-planning in high-dimensional spaces under global constraints. Resulting plans comprise a queue of time-sequenced position-time-energy waypoints that are transferred to the Navigator for execution.

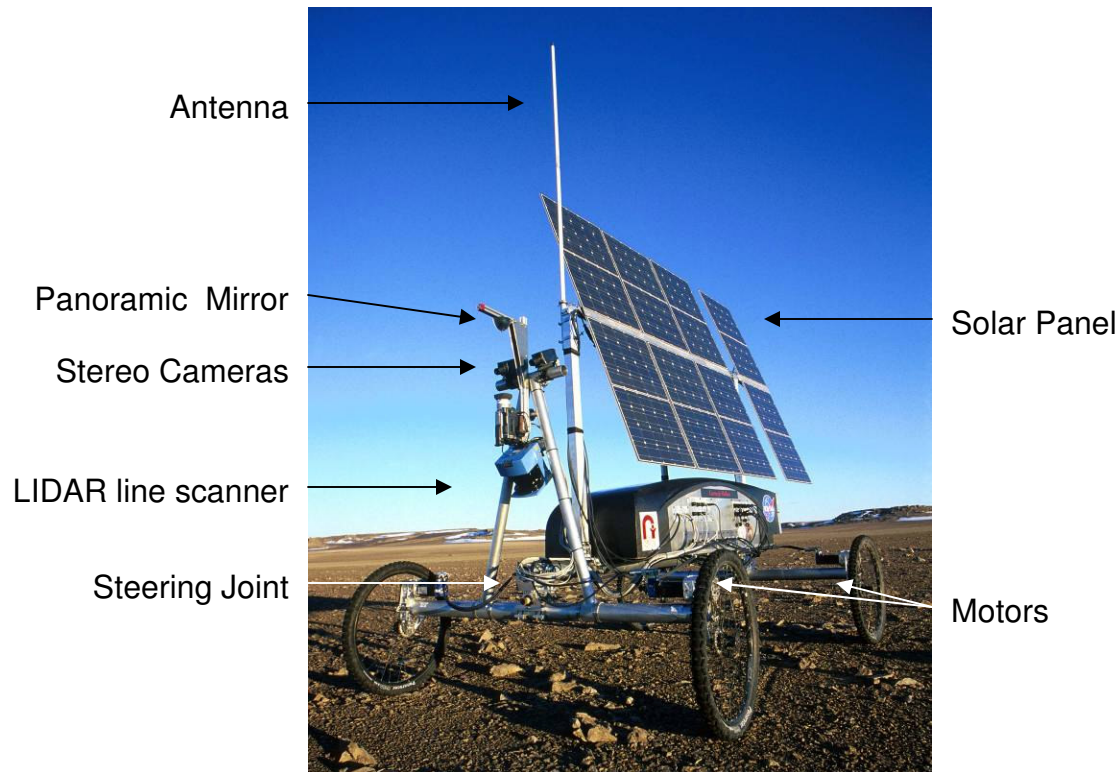
The Mission Planner does not mandate optimal solar array pointing, but rather seeks a balance between navigational requirements to reach all the intermediate goals, and energy requirements to satisfy operational costs. This means the rover can, and sometimes does, travel with its panel sub-optimally pointed or through shadowed terrain but it has always planned for sufficient energy to complete its traverse.

## **Platform**

---

The robot Hyperion was created to exploit the advantages of sun-synchrony. Hyperion represents a class of polar robot notable for reduced complexity, mass and power, and solar panels oriented vertically to catch the low-angle polar sunlight.

The configuration of Hyperion stresses mechanical simplicity in order to reduce mass and thus reduce necessary locomotion power. To obtain sufficient solar energy for sustained locomotion in a terrestrial polar environment requires a careful balance between the amount of energy needed for locomotion and the amount collected by a given solar array area. Hyperion's chassis is composed of three tubes and clamping brackets, all of identical design, that can be easily assembled in the field. The rear "T" frame is joined to the front axle by a passive steering joint (Figure 57).



**Figure 57. Hyperion.**

The passively articulated steering joint is composed of two free rotations. The first is about a vertical axis, which allows the front axle to yaw relative to the rover body, in order to steer the vehicle. The second rotation is a roll motion of the front axle which allows all four wheels to maintain contact with the ground on uneven terrain. An advantage of the steered front axle is that by fixing perception sensors to the axle their view is always directed along the driven path. The steered front axle carries a sensing mast with stereo cameras, a laser range finder, and a panoramic camera. Mounting the sensor mast on the steered front axle ensures that they are always pointed in the direction of travel.

Hyperion is designed for natural terrain. It is capable of surmounting obstacles up to 20 cm in height using four-wheel independent drive. It operates at a nominal speed of 0.25 m/s, slightly under 1 km/hr.

An important innovation is that Hyperion uses differential velocities on all four wheels to steer the front axle relative to the vehicle. This strategy drives the

vehicle smoothly and efficiently along a desired path without a steering actuator; four drive motors are all that is needed. This configuration avoids the mass of high-ratio steering actuation and offers the mechanical simplicity of skid steering while maintaining the power efficiencies of articulated-steering vehicles. Velocity control on all four wheels eliminates slipping and skidding thus improving locomotion drive power. The absence of steering actuators reduces mass but increases control complexity. Rather than performing position control on a steering motor, the front axle angle is controlled by commanding drive motor velocities appropriately. This approach is effective even when traveling over uneven terrain [Shamah01].

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

### ***Polar Solar-Power System***

Hyperion employs two parallel power distribution systems each consisting of a solar array, maximum power-point trackers (MPPT) and battery pack (Figure 58). Hyperion's solar arrays consist of 8 paired modules, visible in Figure 57, fabricated from 12.8% efficient Silicon cells. The array is tilted and fixed perpendicular to the average sun elevation at the approximate latitude of operation. For the case of experiments on Devon Island, the panel was set to an angle of  $21^\circ$ . The total area of solar cells is  $3.45\text{m}^2$  which can provide 220W of electrical power given isolation of  $500\text{ W/m}^2$ . A power control and distribution system each of the arrays operate with maximum efficiency for various insolation and thermal conditions and support two separate battery buses at a nominal 24 volts. Two sets of lead acid gel-cell batteries power the robot during conditions of shadowing or peak power consumption, such as during slope climbing. The two battery buses also supply power to a main bus for the computing, sensing, control and communication subsystems. Without input energy, the batteries have the capacity to power Hyperion for approximately 2 hours under typical operating

conditions. Power system parameters, such as bus voltage, battery and load currents, are sensed and monitored continuously.

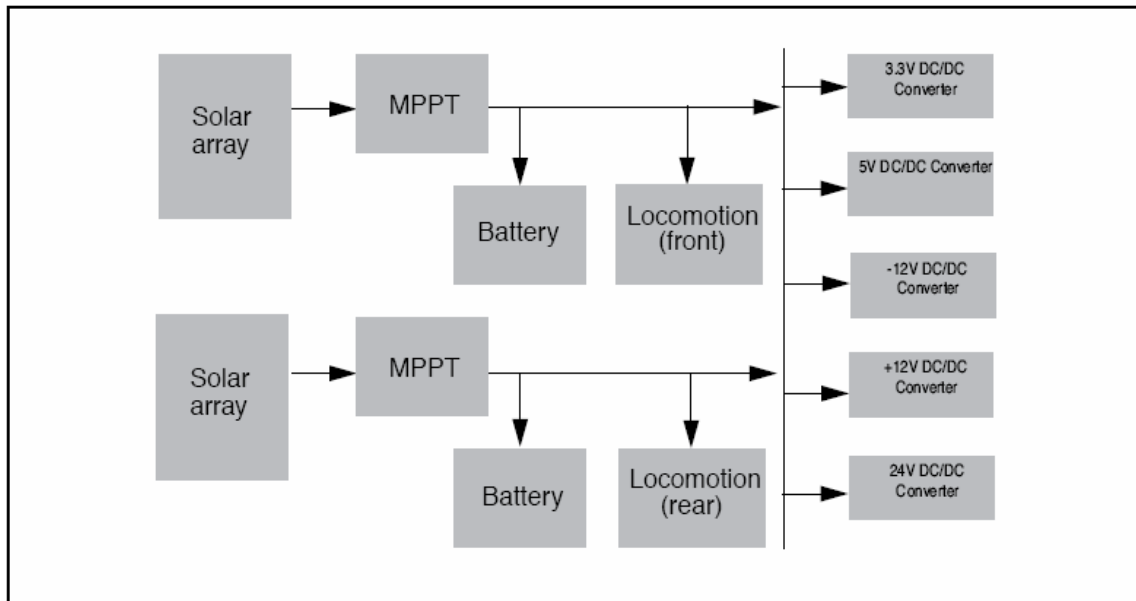


Figure 58. Hyperion power distribution system.

## ***Computing and Sensing Systems***

The computing hardware is designed to support onboard autonomy, including high-throughput components like stereo vision and local navigation, as well as a diverse set of processes that require fewer computational resources. The computational complexity of robot autonomy can make full use of high-performance processors. To meet competing demands for energy efficiency and processing power, the main computing system uses a single 500MHz Pentium III. The onboard operating system is Red Hat Linux 7.1, kernel 2.4.2. The soft real-time requirements of this event-driven system are handled by the standard (non-real-time) Linux operating system. Hard real-time control is handled by dedicated motion control hardware (Galil DMC-2142).

The computing system communicates over IEEE-1394 buses with devices including two digital cameras (Sony DFW-V500) and a digital video camcorder. RS-232 is used to communicate with the motion controller, laser rangefinder

(SICK LM20), tilt/roll sensor (Crossbow CXTILT02E), and the power management and distribution micro-controller.

Hyperion's real-time drive and steering motion control is accomplished with a Galil multi-axis controller that provides PID motor control using encoder feedback. An independent motion controller increases reliability since driving, and more importantly stopping, maneuvers do not require Hyperion's main computer. This strategy provides the accuracy needed by the navigation software, the maneuverability to avoid obstacles, and the reliability to prevent chassis damage [Shamah01].

## **Navigation Software**

---

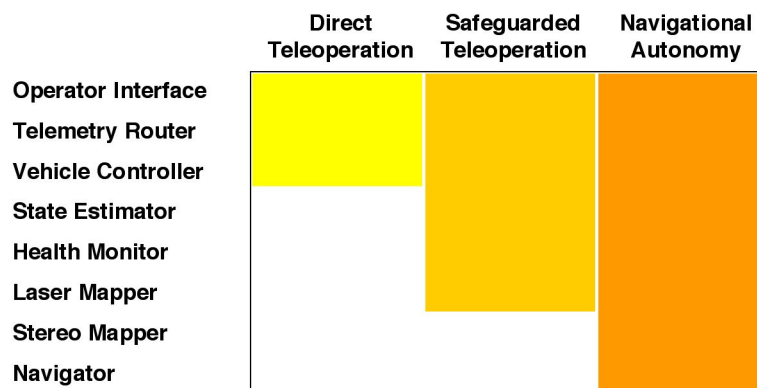
To operate sun-synchronously, Hyperion optimizes the orientation of its solar panel with respect to the sun. By maintaining its orientation within  $15^\circ$  of the sun, it receives more than 90% of the available solar energy. This poses significant constraints for the navigation software; it must go beyond avoiding obstacles and reaching goal locations to also maintaining orientation relative to the sun. Hence, Hyperion can run into difficulty not just from obstructed paths but also from improper orientation or being off schedule.

Architecture is emerging in planetary exploration research as a crucial challenge for rovers that have complex skills and true autonomy [Volpe01]. This motivated the development of an architecture that enables rigorous fault detection and flexibility in the command structure to facilitate fault recovery either automatically or through human assistance.

Communication onboard Hyperion is implemented using the IPC protocol [Simmons01]. IPC is a publish/subscribe, message-based protocol for which a message dictionary was defined specific to Hyperion's software design. A Telemetry Router process filters and records message traffic among modules. The resulting telemetry can be replayed to analyze experimental performance.

## Operational Modes

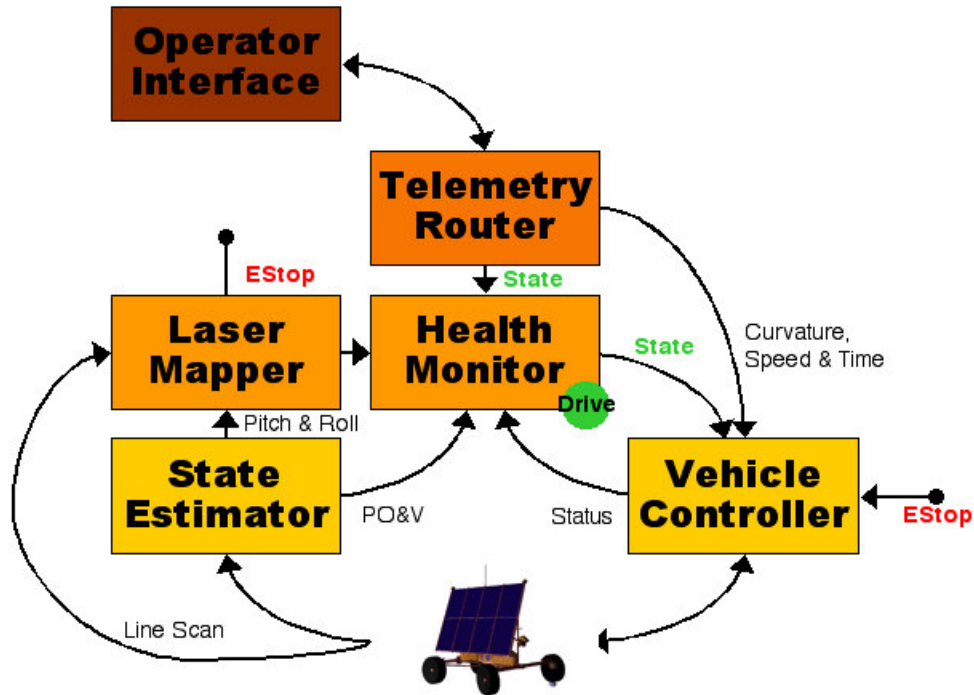
Hyperion's software architecture exhibits a property of sliding autonomy such that the current conditions dictate the operational mode. This concept and architecture has evolved through many planetary rovers [Bares99]. The operator can interact with the robot by directly teleoperating its actions, by allowing it to safeguard operator commands, or by enabling it to navigate autonomously. Figure 59 shows the software components active during each of the operational modes.



**Figure 59. Modules operational by navigation mode.**

Direct teleoperation allows direct low-level commands to the robot and provides minimal safety—safety monitors warn the operator of possible faults but they are superseded by operator commands. An operator is able to move Hyperion by appropriately setting individual wheel speeds, which can be tedious, but is rarely needed.

In the safeguarded teleoperation mode, the operator guides the robot with coordinated motion commands (e.g. drive with some speed and curvature). The State Estimator integrates sensor information including position, orientation and speed. The Laser Mapper detects nearby obstacles and signals the Health Monitor of potential collision. When the Health Monitor detects this or other anomalous conditions, motion is stopped and the fault is signaled to the operator. Figure 60 diagrams the modules in this operational mode.

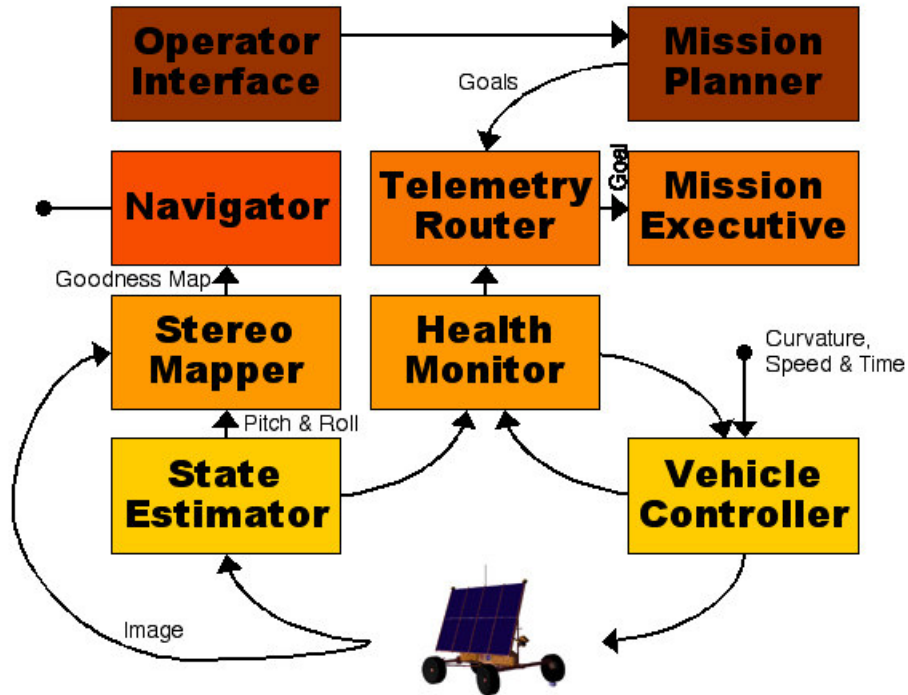


**Figure 60. Safeguard operational mode.**

In its autonomous mode the Stereo Mapper classifies terrain, generating a traversability map from stereo imagery. The Navigator evaluates the map and selects a path that best leads the robot to the next goal dictated by the mission plan. If the local goal cannot be achieved or if the time to reach the goal jeopardizes sun-synchrony, the Mission Executive signals the Mission Planner to replan the route. Figure 61 show the additional modules active in this operational mode.

In the Hyperion software system, the Health Monitor samples approximately 100 state variables including temperatures, voltages, currents, positions, orientations, and velocities. It also monitors software state, like process activity and uncertainties in stereo correlation and terrain modeling. This information is used to detect faults in the system, which are signaled to the operator [Verma01].





**Figure 61. Autonomous navigation mode.**

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

### ***State Estimation***

Hyperion utilizes odometry, inertial sensing, and carrier phase differential GPS to track its location relative to a satellite-derived digital elevation model. Its dGPS (Novatel Beeline) provides accurate heading, pitch and location information using two receiver antennae, but is subject to drop-out. By filtering the dGPS and

incorporating odometry and separate inertial measurement of roll and pitch, stable and accurate pose information is obtained. Accurate localization, within 10cm and 1°, is important for collecting ground truth data on robot performance for analysis, but is not critical to the navigation algorithms which operate within a local reference frame. Foregoing GPS and applying a less accurate positioning system, such as localization from inertial sensing and visual features, would be sufficient for an actual planetary mission, and would be necessary beyond Earth.

### ***Goal Regions***

Waypoints, generated by the mission planner, are specified as goal regions whose dimensions and orientation are described at runtime. In general, the navigation map is oriented along the direction in which the robot should travel to optimize sun-angle. By orienting the map along the direction of travel, and aligning goal regions perpendicular to the preferred direction of travel, common artifacts caused by planning on an 8-connected grid are minimized.

### ***Perception and Navigation***

Hyperion uses a combination of stereo camera imaging and laser ranging for terrain perception. The relatively benign arctic terrain encourages the adoption of an “optimistic/pessimistic” local navigation strategy. Terrain that is unseen by stereo vision, for example due to occlusion or distance, is optimistically assumed to be traversable. The laser operates as a “virtual bumper” that stops the robot prior to collision when detecting obstacles missed by the optimistic evaluation, thus it acts as a pessimistic safeguard. This allows long range measurements that have an associated large error in range measurement accuracy, to be ignored. This decreases the number of false positive obstacles, increasing navigation efficiency. In practice, most of the terrain is eventually sensed by stereo vision and all is swept by the laser before it is traversed. This combination of optimistic and pessimistic strategies allows for efficient navigation with neither undue risk nor the need for perfect terrain knowledge.

The onboard software serializes the obstacle avoidance and local navigation algorithms and utilizes complementary sensors to consider terrain [Urmson02]. Previously, the output of local obstacle detection software had been combined in parallel with global planning information through an arbiter [Singh00]. Steering commands are selected by the Navigator using the obstacle avoidance and global information in serial, avoiding potential conflicts between global and local path optimization.

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

### ***Obstacle Avoidance***

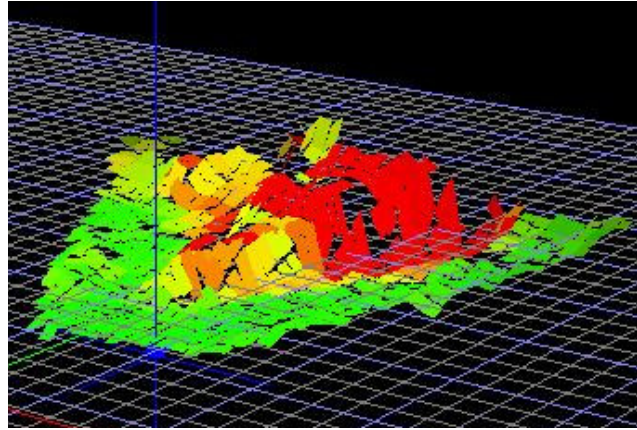
The specific characteristics of the intended terrain are critical to the design of perception and navigation algorithms. The high arctic where Hyperion is designed to operate is typical of a terrestrial polar environment and shares many properties with regions of Mars and the Moon. The terrain is gently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

Optimizing the characteristics of a stereo vision system involves trading between range accuracy, minimum viewing distance, stereo rate, and field of view. To increase the effective field of view of the stereo system, the cameras can be actively pointed. In the case of Hyperion, having the stereo camera mast attached to the steered front axle this allows the use of relatively narrow (60°) field of view cameras that are effectively made wider by the swept motion. Without the steered stereo vision system, the cameras would need to be much

wider field of view, which make the stereo matching process less reliable, making the navigation less efficient.

Hyperion utilizes a 20cm stereo baseline that yields a range resolution of approximately 15cm at 7m. Since Hyperion travels at only 30cm/s, and has a stopping distance of less than 5cm, it is clearly operating within the efficiency-limited domain. Limiting the number of disparities in the stereo correlation dramatically reduces the computing resources required to process each frame, but also limits the minimum distance at which range data can be acquired. Due to this effect, Hyperion is unable to reliably recover accurate range information closer than 1m from the cameras. However, since the cameras are mounted 1.5m above the ground, this arrangement provides a workable trade-off between accuracy at range and minimum sensing distance.

Traversability is determined from range information by extracting three metrics: the slope, roughness, and discontinuity of the terrain in rover sized patches. Slope is the steepest of the pitch or roll the vehicle would encounter, and is calculated from a plane fit to the range information. Hyperion is designed to be able to climb slopes and navigate side slopes of up to  $15^\circ$ , thus any patch with a slope greater than this is considered impassable. Roughness is calculated as the chi-squared residual of the plane fit. The discontinuity metric is an estimate of the variance of the height of points in a patch. It detects step features in the terrain. Each metric is normalized and traversability of the rover-sized patches is considered to be the worst of the three measures. Figure 62 shows the evaluation of a region of terrain with 3-D geometry of terrain patches shaded based on traversability. Regions of the stereo field of view that contain more points usually generate a better estimate of the actual terrain. To capture this, a certainty value is computed for each patch. Certainty is calculated as a function of the number of points in a patch and the evenness of the distribution of the points over the patch.



**Figure 62. Hyperion's stereo vision based terrain evaluation.**

The navigation cycle is shown in Table 5. The Navigator autonomously guides the rover from its current location to the next goal location, or waypoint, by selecting the lowest cost combination of possible steering arcs and remaining path to the goal as determined by the D\* algorithm [Stentz95b]. The robot travels a fraction of the chosen path and then the navigation cycle is repeated—at 1.5Hz, so that a smooth transition among steering arcs is achieved.

Navigation Cycle	
1	Update the robot position via the State Estimator
2	Obtain range information from the Stereo Mapper
3	Compute evaluation metrics into the discretized terrain map (Figure 62)
4	Incorporate terrain costs into composite terrain map.
5	Update position to account for movement during computation
6	Evaluate the cost of motion along a discrete set of arcs
7	Estimate the cost of reaching the goal with each arc (using the D* algorithm)
8	Choose the arc that has the lowest value of total cost
9	Send radius, speed, and time for arc to the Controller
10	Repeat

**Table 5. Navigation Cycle.**

## Field Experiments

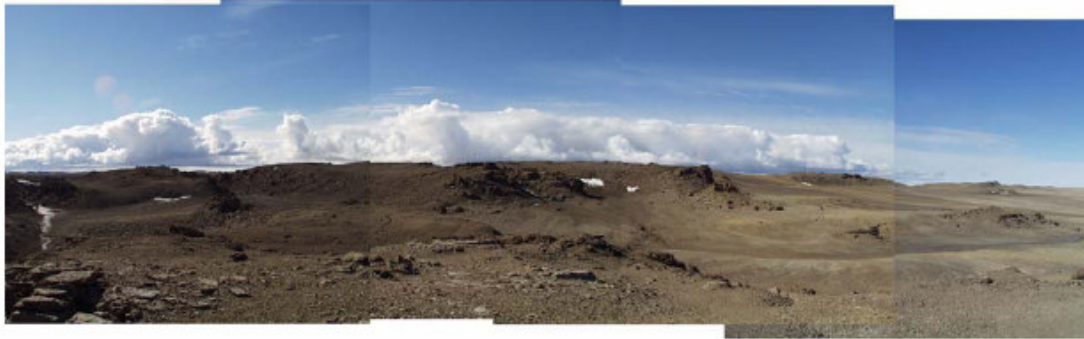
The Arctic Circle, at 66.55°N, marks the southernmost latitude of the northern polar regions at which the sun does not set on the summer solstice. Haughton Crater is located at 75.36°N, 89.68°W on Devon Island about 600 miles north of the Arctic Circle (Figure 63) where the Sun stays above the horizon for over 3 months in summer. Haughton is particularly notable for the lunar-like breccia inside the crater and Mars-like planitia to the northwest of the crater rim (Figure 64). It is the site of ongoing investigation by the NASA Haughton Mars Project (HMP) and, through the support of the HMP, was the location of our field experimentation.



**Figure 63. The location of Haughton Crater.**

The overall goal of the field experiment was to demonstrate the concept of sun-synchronous exploration by navigating a sun-synchronous circuit in 24 hours, finishing with batteries fully charged so that, in principle, the operation could be repeated indefinitely while sunlight persisted. An important aspect was to determine the practical effectiveness of synchronizing with the Sun, requiring the measurement of ground-truth power input and output, and to learn the achievable

vehicle speed and endurance. The three week field experiment in July 2001 involved numerous hardware and software component tests, and culminated in two 24-hour sun-synchronous circuits.



**Figure 64. The Von Braun Planitia on Devon Island.**

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

Given a start position, the list of intermediate goals, and a 24-hour range of allowable start times, the Mission Planner, produced the a sun-synchronous plan including the recommended start time. Beginning at this time, waypoints from the plan were sent to Hyperion individually for execution. The robot would autonomously navigate the terrain, avoiding obstacles while travelling through the defined waypoints. Operators monitored robot progress throughout the field experiment. When the robot reached a waypoint, it would wait until the designated time to execute the next waypoint, thus maintaining the schedule imposed by the plan. In effect, this also simulated the effects of conducting a

science exploration in conjunction with navigation. Mission planning was too computationally demanding to allow re-planning during a traverse. Therefore, the operations team had to enforce the navigation schedule imposed by the plan, by teleoperating to re-synchronize Hyperion with the plan in the event of operational delays.

The first 24-hour, sun-synchronous experiment was completed on July 19, 2001. Hyperion traveled 6.1km with 90% (Figure 70) of the mission completed autonomously. Teleoperation was used upon indication from Hyperion's health monitor that a fault had occurred. In this experiment, the only faults to occur were failures to find a forward path and laser obstacle detection. No intervention was required by field observers for safety or any other reason, despite winds up to 27 KPH. Teleoperation typically involved very brief instances of teleoperation to place the robot in more benign conditions from which it could continue autonomously. In several instances during the experiment, Hyperion fell behind its plan due to the difficulty of the terrain it encountered, but each time it was able to catch up to the plan when it reached more benign terrain. The first experiment successfully demonstrated the concept of sun-synchronous navigation; solar array and battery power were sufficient for continuous operation and the rover's batteries were fully charged at the completion of the circuit.

A second experiment was conducted in which more challenging terrain and greater distance was attempted. In this experiment, Hyperion traversed 9.1 km (Figure 68) with 50% autonomy and 50% safeguarded teleoperation. During this experiment a single instance of manual intervention occurred when, in driving close to a large rock, the steering axle tripped a limit switch when a rear wheel was obstructed and the rover was forced to halt. A software patch to temporarily override the fault would have resolved the problem, but in the interest of maximizing the experimental data return, this time-consuming option was not exercised. Instead, the axle was re-positioned manually and the traverse resumed. This incident occurred in safeguarded teleoperation mode.



Due primarily to communication outages resulting from the distance and line-of-sight visibility to the base camp, Hyperion fell behind its plan by three hours. This delay resulted in poor solar array sun angles, and also in stereo camera blinding from the sun directly ahead. To recover, teleoperation was needed on 50% of the path (still following the sun-synchronous navigation plan, but without delays to re-acquire the schedule). Safeguarded teleoperation was initiated to command the rover to stop and orient for stationary recharging to build battery reserves. Hyperion regained its schedule after 6 hours and, after several more hours of autonomous performance, completed the circuit with batteries fully charged. This second experiment served to illustrate how deviating from sun-synchrony can cause many operational difficulties.

In autonomous mode, driving decisions were based on perception from stereo sensors and the location of sun-synchronous mission goals. Operators were limited to two actions in autonomous mode: submitting goals and setting the nominal driving speed. An operator would generally decrease driving speed from 0.25m/s to between 0.15m/s and 0.20 m/s if rocky terrain was perceived ahead; in retrospect this conservative measure was probably unnecessary.

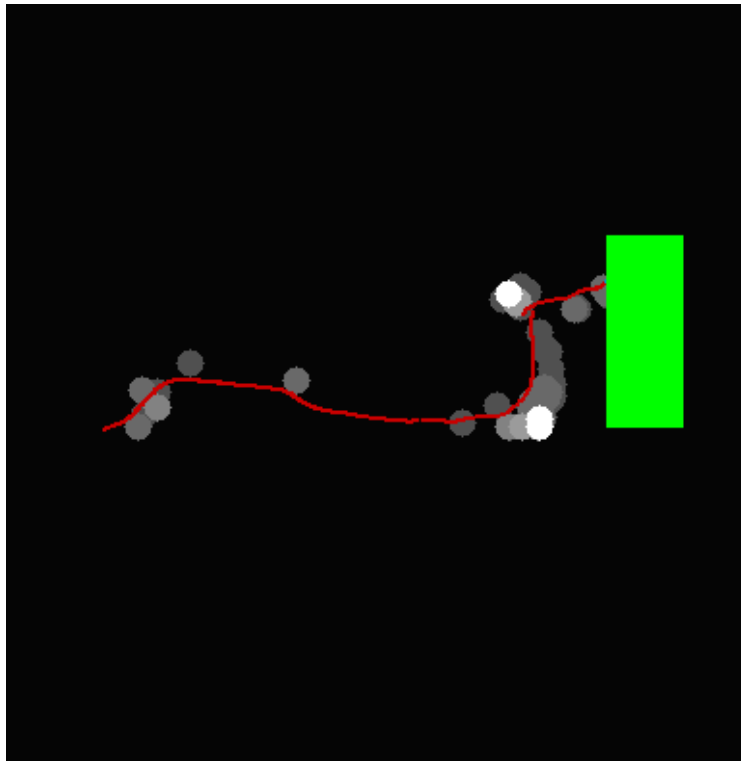


Figure 65. Obstacle avoidance display.

### ***Obstacle Avoidance***

The navigation software proved reliable and capable of avoiding obstacles in environments with obstacle density as high as 30%, meaning that 30% of the area the robot drove through was impassable. Figure 65 shows the display of detected terrain costs on the approach to a goal region (green rectangle) in one path segment. Brighter circles indicate higher cost terrain. The average absolute heading error for the robot was  $21^\circ$  while traveling through terrain with an average obstacle density of 6.9%. While  $21^\circ$  is significant, it translates to approximately a 7% increase in cost above the straight line distance between waypoints which implies that the navigation was relatively efficient. Important to note in Figure 66, which shows the occurrence of the full range of non-zero heading errors, are several instances of large heading error ( $\pm 160^\circ$ ) where Hyperion was navigating to avoid obstacles and came fully about to get out of impassible regions.

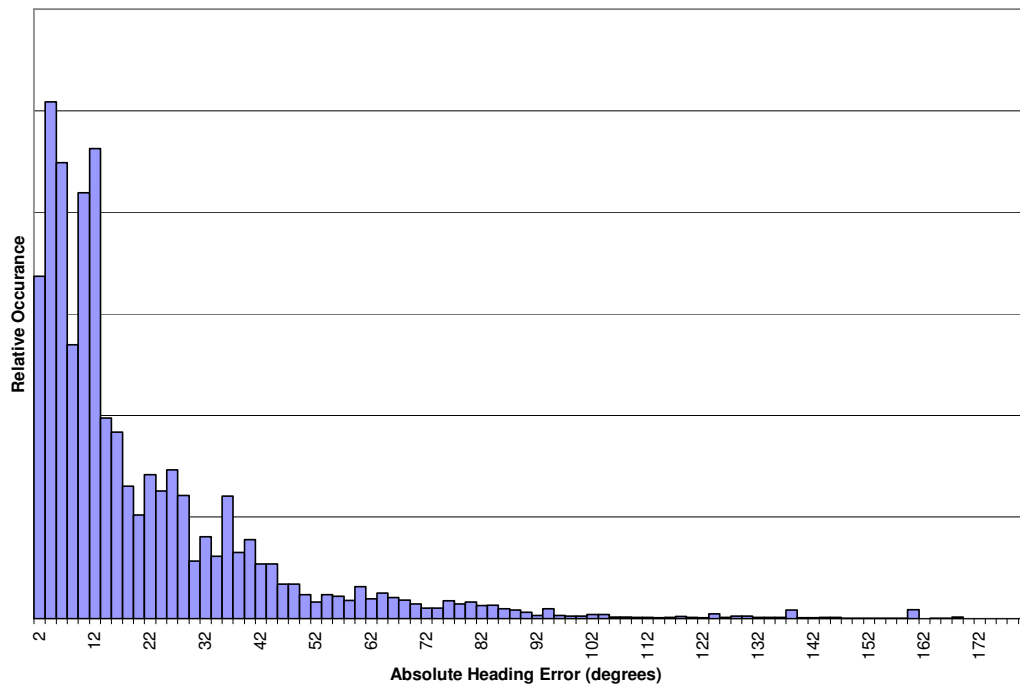
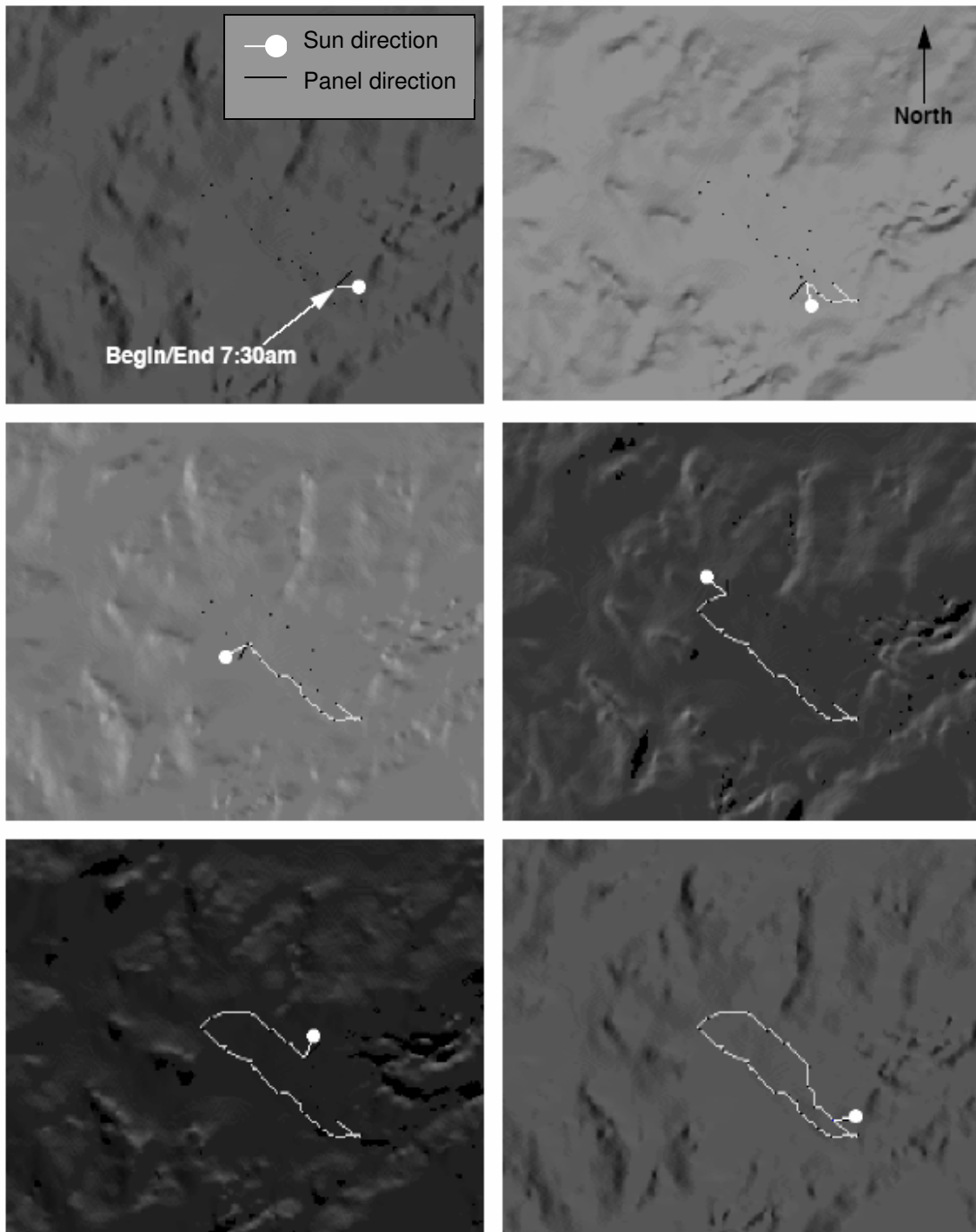


Figure 66. Relative occurrence of heading error.

### ***Sun-Synchronous Performance***

The Mission Planner was successful in generating sun-synchronous routes. Figure 67 and Figure 68 depict the plans generated for the two 24-hour sun-synchronous navigation experiments. Each figure is a sequence of six "snapshots" of the plan, running from upper left to lower right across three rows. The snapshots show the terrain lighting (grayscale background; normal incidence is white, and shadows are black), the remaining intermediate waypoints, and the path executed so far. Each pixel in the image is a 25 meter cell, corresponding to the DEM spatial resolution. To the resolution shown, the route was followed exactly by the onboard navigation system. The bottom right snapshot in each sequence shows the completed route, with the rover in the start and finish position. Note that the snapshots also indicate, from the current rover position, the vectors to the sun and of the solar panel normal. The solar panel normal appears  $90^\circ$  to the left of the driving direction, matching Hyperion's solar array configuration.



**Figure 67. First sun-synchronous mission plan.**

The non-circular shapes of the routes stem from natural terrain that constrains navigation. Streambeds ran along the outside of both diagonal legs, preventing

excursions outside those bounds. The first experimental plan stopped short of a rocky promontory to the northwest end of the route, while the second experiment, largely over the same terrain, went all the way around the promontory. The field team selected the intermediate goal points to avoid terrain hazards below the scale of the elevation map for the region, at intervals of approximately 400 m. Hence, the Mission Planner had to plan the route between these positions and, more importantly, reason about how to best time the route to minimize the inevitable solar array mispointing forced by the non-circular circuit.

Figure 67 (top right) shows early progress of the plan for the first experiment. Note the sun direction is aft of the solar array normal. Meanwhile, in Figure 67 (middle left), the sun direction is forward of the solar array normal. These aft and forward biases reflect TEMPEST's ability to schedule the path to achieve the best average sun angle over the path. The length of the northwest leg in these frames, coupled with a limitation in rover top speed, prevents an ideal sun angle over the course. Therefore, TEMPEST biases the sun aft at the beginning of the leg (Figure 67, top right) in anticipation that the sun will overtake the rover near the center of the leg (middle left), and be biased ahead of the rover by an equivalent angle at the northwest end. A similar behavior occurs on the return, southeast leg of the traverse (Figure 67, middle right, bottom left, and bottom right).

The second experiment plan (Figure 68) was far longer than the first, and circled around the rocky promontory at the northwest end of the route. The behavior here was similar to that in the first experiment, except complicated by the highly concave route shape. The mission plan for the second experiment forced the rover to turn its solar array away from the sun (for example Figure 68, middle left) more frequently than for the first experiment. These deviations, along with the greater overall plan distance, caused a greater sensitivity to the delays experienced during this plan's execution.

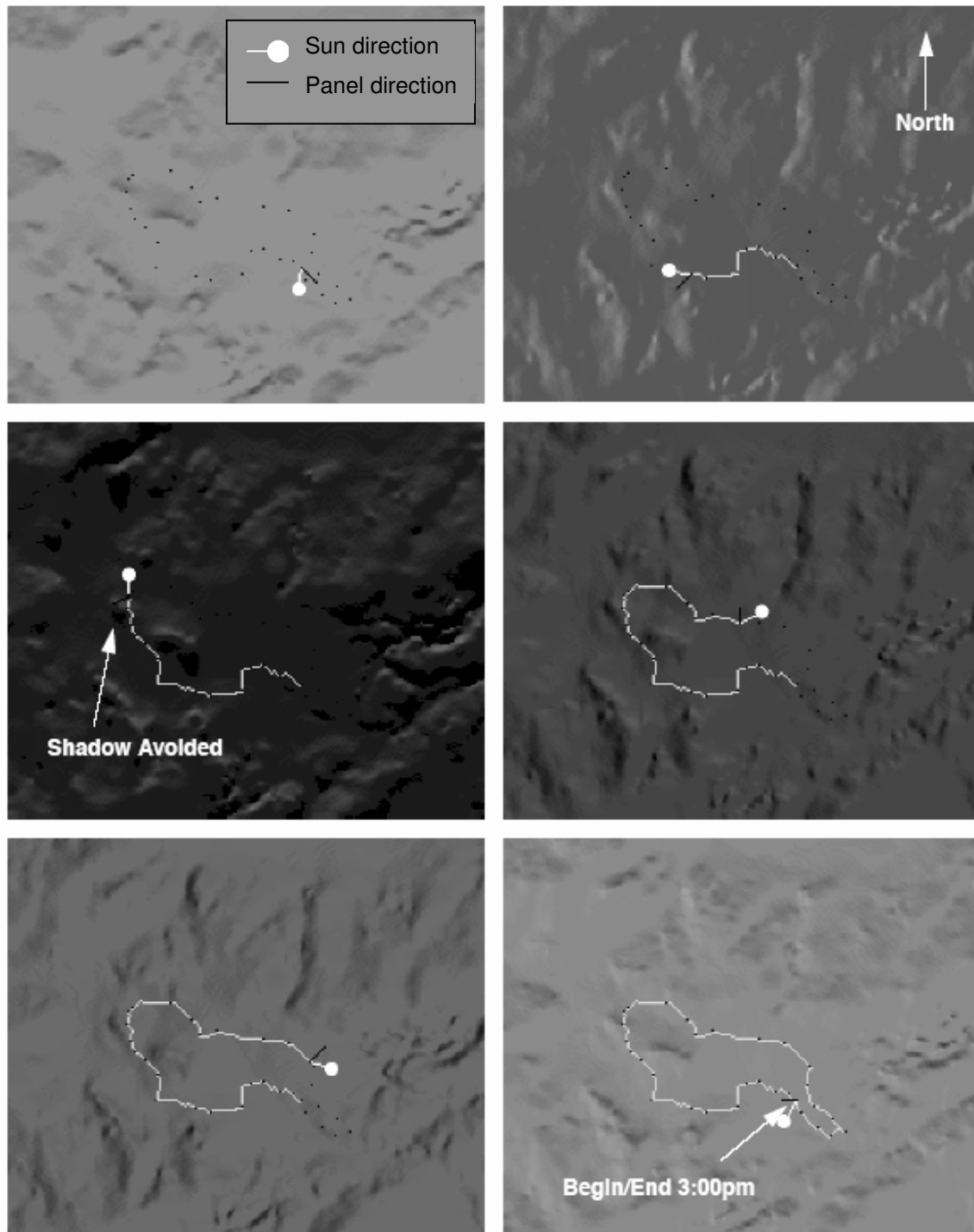


Figure 68. Second sun-synchronous mission plan.

## Summary

Hyperion combines short range navigation sensors with coarse pre-planning and low resolution prior maps to achieve efficient, low-speed navigation that is

representative of approaches that succeed in the efficiency-limited regime. While it is difficult to evaluate the efficiency of the navigation achieved, an average heading error of only  $5^\circ$  implies that plans were within a few percent of the straight-line path between objectives.

In this application domain the overall navigation performance (speed, reliability, efficiency) was representative of the state-of-the-art at the time of these experiments. This performance has only recently been surpassed by the same algorithms operating on a more capable platform. Lunar energetics associated with  $1/6$  gravity and high solar flux suggest that a ten fold multiple of speed capability will be achievable on the moon. By increasing specific power, robots may be able to explore much broader areas, and performance requirements may eventually reach the swerve-limited regime.

---

# 10 Conclusions

---

Two important conclusions from this work are:

1. Safe navigation is possible at speeds beyond constraints based on stopping distance.
2. There is an interesting, reproducible relationship between sensing horizon, prior map resolution and navigation efficiency. This research distinguishes the relationship but does not fully quantify or explain the phenomenon.

## Summary

---

Three navigation regimes are distinguished by the relationship between required sensing horizon and speed.

The efficiency-limited regime is characterized by low speeds and though it benefits from prior knowledge, makes no requirement for it. Onboard sensing horizon is fixed given a specified prior map resolution and required navigation efficiency.

Within the stop-limited regime, speed is bounded by stopping distance. To ensure robot safety sensing horizons must increase quadratically with speed,. Algorithms designed for navigation within the stop-limited regime generally require some cognizance of dynamics.

The swerve-limited regime is characterized by high speed, which is achieved through planning with reliance on prior maps. Sensing horizon increases roughly linearly with speed, reducing the difficulty of the sensing task. Obstacles must be avoided by swerving since panic stopping is not viable. Algorithms operating within this regime have an explicit requirement for pre-knowledge of the terrain they are operating in. In particular it is necessary to bound the swerve offset required in making any traverse.



## Contributions

---

The major contributions of this thesis are:

- The identification of navigation regimes. The recognition that there are distinct regimes that require algorithmic treatment in different ways will help researchers address the problem of off-road navigation more effectively. For instance, increasing sensor range to increase navigation performance will result in smaller returns than adopting strategies reliant on some prior information.
- The discovery of a rapid decay in benefit as a function of the sensing horizon. Prior to this work, there was no principled discussion of how navigation algorithms benefit from increases in sensing horizon. Evidence suggested that longer range sensing is better. While this is generally true, the rate at which benefits diminish argues that development time and resources should not be wasted attempting to increase sensor range without bound.
- A technical demonstration of capabilities and implementation necessary to achieve high-speed off-road navigation.
- A detailed comparison of robots representative of the efficiency-limited and swerve-limited regimes. Through concrete example, the differences in application and resulting differences in system requirements and performance will help provide guidance to future research.

## Future Work

---

Where from here? This work has uncovered a number of interesting avenues for future research, they include:

- The efficiency-limited regime stems from a sharp decay in benefit as sensing horizons increase. This work makes no attempt to explain this phenomenon. Understanding why this relationship exists may be a significant undertaking but could provide much insight into the navigation problem.
- The coefficient of the rate of decay of benefit in sensing horizon is shown to vary dependant upon the “type of terrain”. This work does not provide any method for determining this coefficient. An approach for measuring or estimating this value based on known methods of characterizing terrain (e.g. [Golembek97]) would be a great tool for developing autonomous vehicles for specific applications.
- No attempt to evaluate sensing horizon scaling when incomplete or uncertain data is combined over multiple sensing cycles has been made. Understanding how sensing horizons change to achieve required path optimality under these conditions would be of great value.
- As robots travel faster, and move up through the navigation regimes, the number of safety assumptions that may be violated increases, and thus the likelihood of failure increases. Research into multi-modal control approaches that switch between algorithms dependent on the current navigation regime may result in robots that are as safe as possible given their instantaneous operating speed.
- Much information encoded in aerial images cannot be extracted by current image understanding techniques. The ability to encode the semantic meaning of prior maps so that onboard planners can utilize the full implicit resolution would be a significant contribution.
- This research has only made a first cut at a path shifting algorithm; there is much room for improvement. Path shifting holds promise for high speed

navigation as it provides a natural structure for swerve-based planning algorithms which must know the corridor in which they should operate.

## Cultural Lessons

---

This research has benefited from close work with a number of research teams, each with their own culture. Some of the cultural approach of this research may benefit continued gains in high performance off-road navigation. Here is what worked:

**Implement simply-** The simplest approaches are generally the best approaches.

**Implement well-** Too often there is overwhelming pressure to hammer something out that just does the job. There are benefits to quick proof of concepts, but they need to be revisited and *done right*. Particularly in software, the nature of rapid development and testing demands rapid implementation, but also means that next time a similar problem comes up, the same code will be reused, and the author will need to support it. Do it right the first time, and the reuse will be possible with minimum difficulty.

**There are no “trivial tasks”** - Often, researchers will declare that something is “trivial”. Nothing is trivial. “Trivial tasks” often drag on much longer than anticipated, wreck schedules and derail projects. Beware of “trivial changes”, particularly as deadlines approach.

**Reliability and robustness are at odds-** The more constrained the environment, the easier it is to make something work reliably, but once removed from this constrained environment, the system will generally perform poorly in unanticipated situations. The trick is to find a correct balance between scope of scenarios handled (robustness) and the ability to “do it right” (reliability). Swerve-limited navigation is an example of exactly this trade-off. By limiting the scenario (the maximum swerve offset) it is possible to achieve reliably navigation but it comes at a cost. A robot operating in the swerve-limited regime that encounters

an obstacle larger than those anticipated will likely crash, due to its lack of robustness,

**“Hard tasks” are achievable-** “Hard” problems exist, but in most cases of implementation the loosely described problem is not the problem that really needs to be solved. By exploring the objective, rather than procrastinating and avoiding the problem, great progress can be made- perhaps not towards a general solution, but certainly towards a useful solution.

**Compartmentalized development and testing is essential-** Attempting to build a whole system as one piece is more time consuming than planning ahead and developing, testing and then integrating components. Without confidence in the individual pieces of a system it is very difficult to diagnose problems in an integrated system.

**Go boldly-** Many things “can’t be done”, and then are. Deterministic, steady progress is critical to the advancement of a field, but without bold steps, **even those that fail**, research stagnates short of potential. Breakthroughs are caused through intention and action.

## **Closing Remark**

---

As autonomous navigation speed is pushed higher, it will be necessary to change the way we treat the navigation problem. It will become necessary for robots to drive in the same way that humans do: they will seamlessly combine prior knowledge and real time sensing. This work is a first step in that direction.

---

## Appendix A: Mojave Desert Cost Maps

---

The elevation models in this appendix were generated from a combination of high resolution aerial imagery and LIDAR range data. Each pixel is approximately 1m on a side.

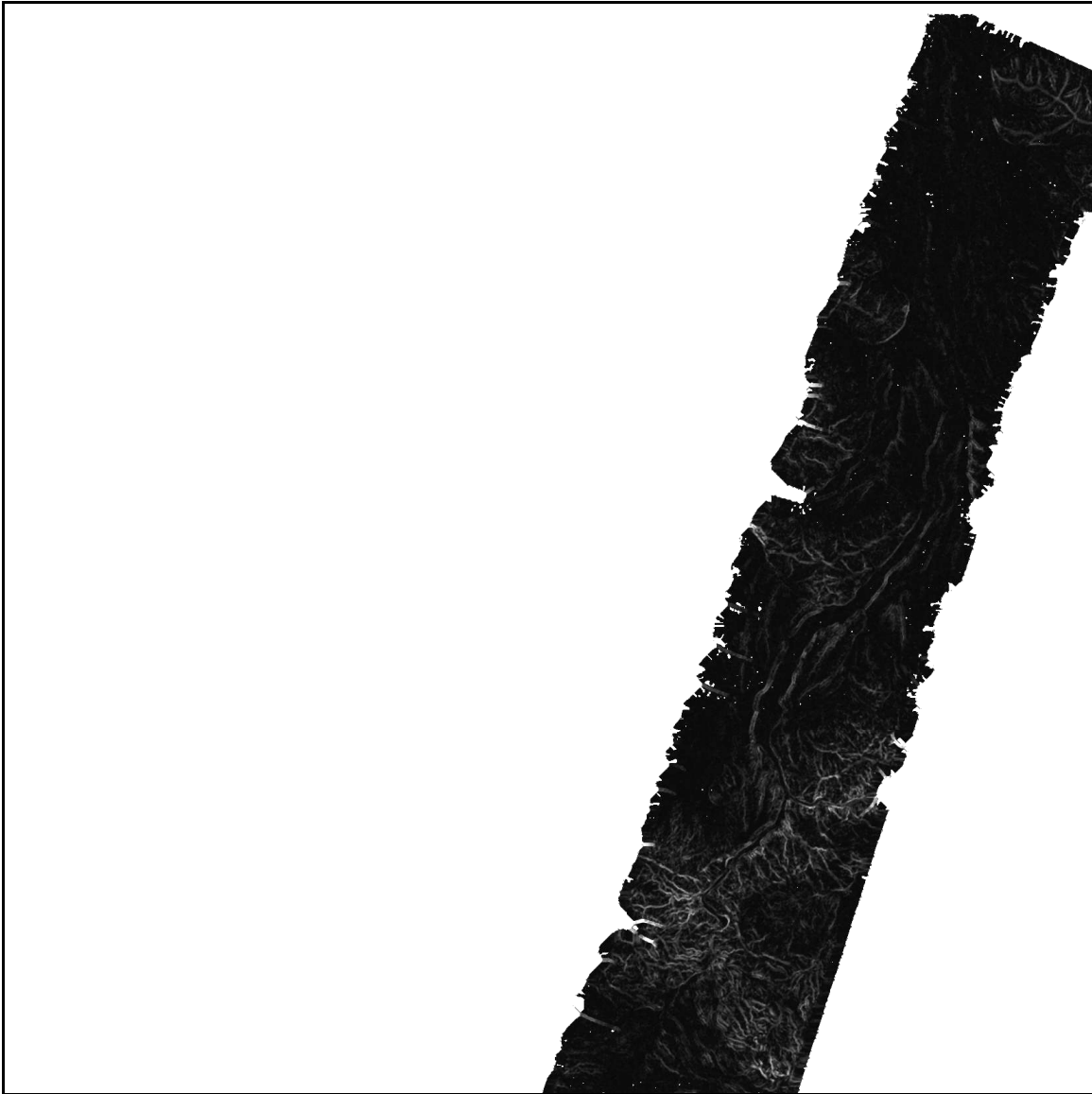
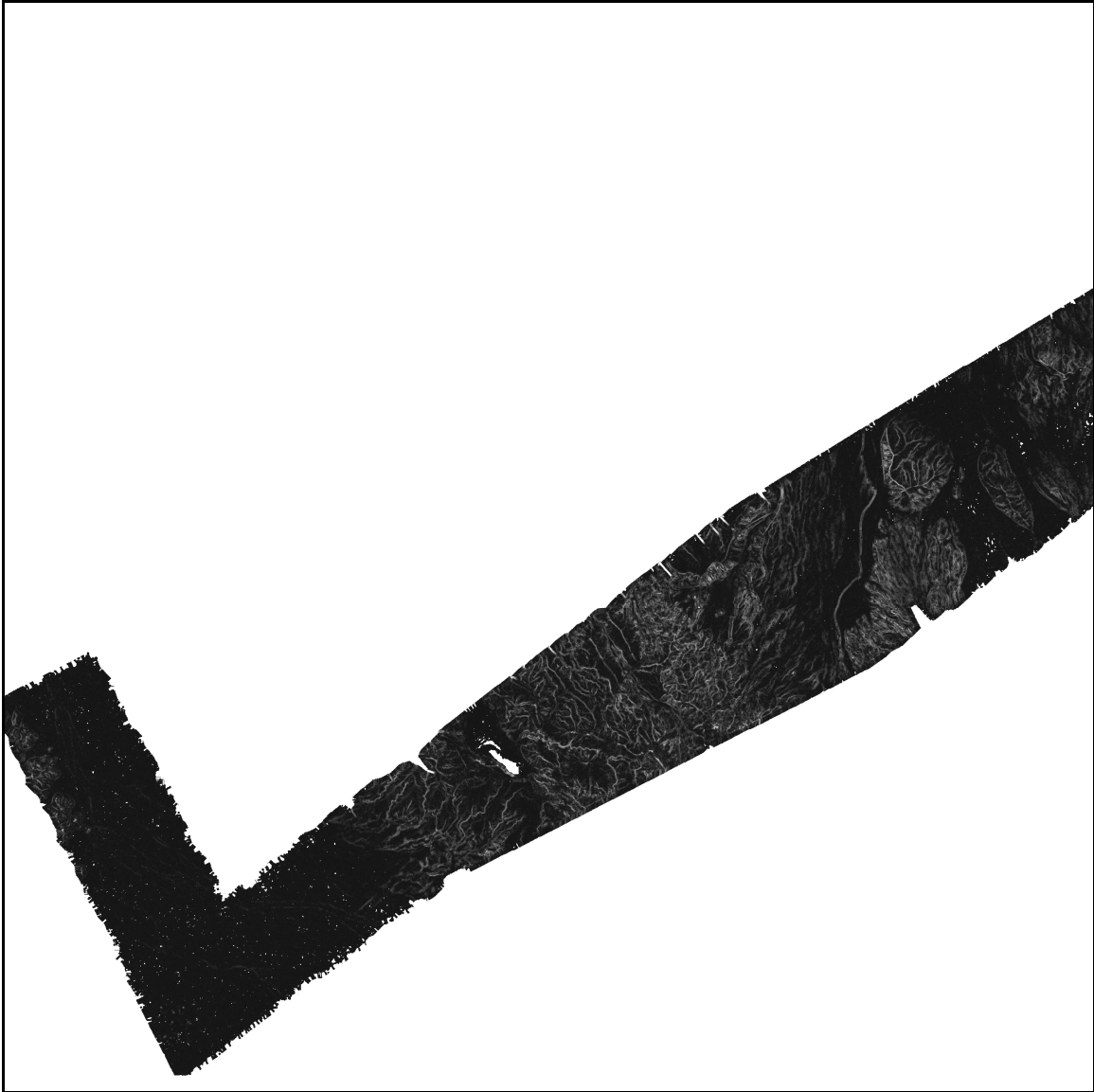


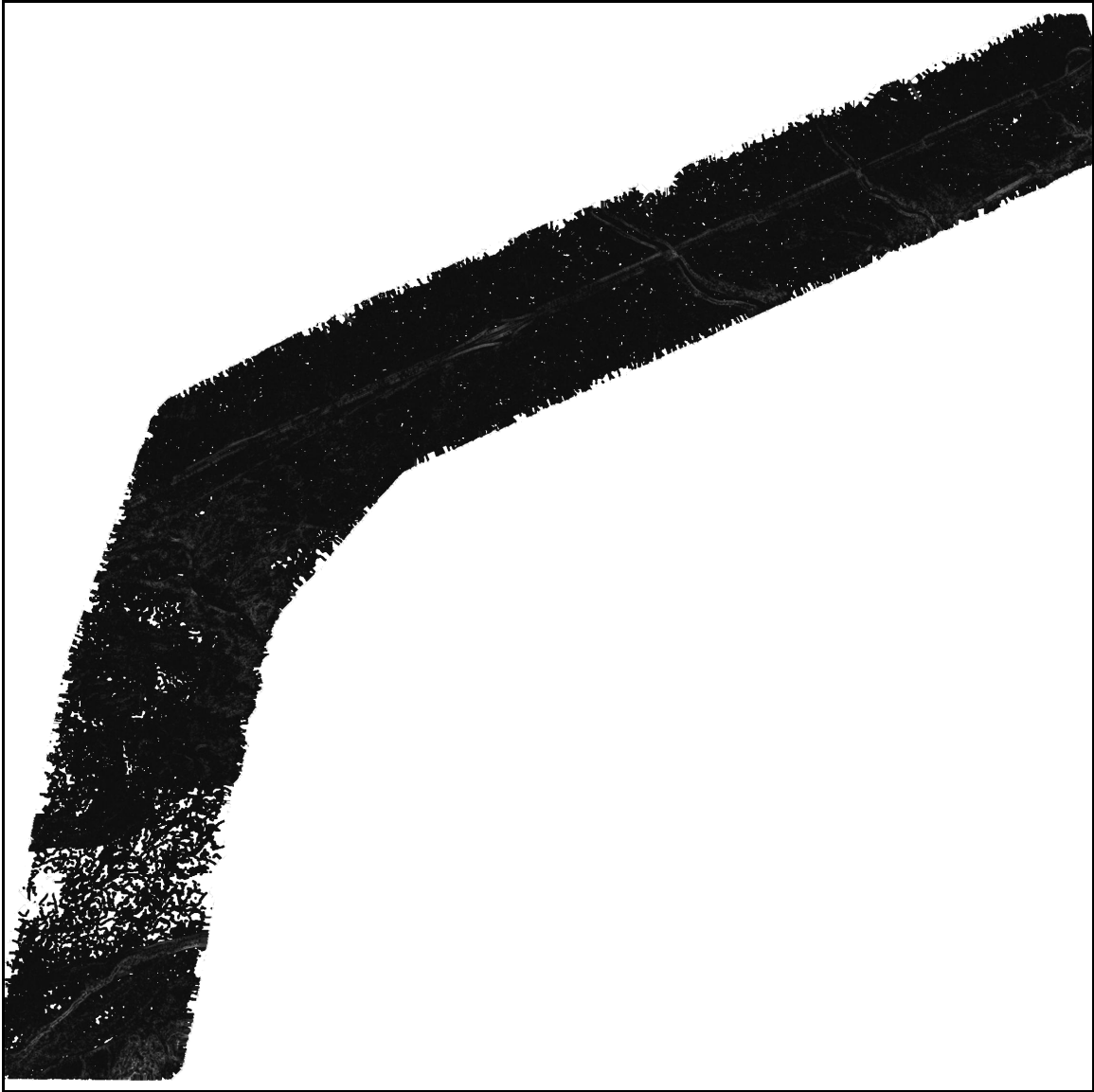
Figure 69. Mojave Desert Cost Map 1.



**Figure 70. Mojave Desert Cost Map 2.**



**Figure 71. Mojave Desert Cost Map 3.**



**Figure 72. Mojave Desert Cost Map 4.**





Figure 73. Mojave Desert Cost Map 5.



Figure 74. Mojave Desert Cost Map 6.

---

## Glossary

---

**Prior Map-** A cost map that provides information about the world as it was understood prior to the robot's operation.

**Cost Map-** A map of the world, often represented as a regular grid, where each location in the map holds the current estimate of the traversal cost. Cost may represent distance, energy required for traversal, risk, or any other relevant measure.

**Cost Ratio-** The ratio between the cost of a complete information optimal path and the cost of some other path. This ratio can be used to judge the efficiency of a particular planning and perception approach.

**DEM- Digital Elevation Model,** a digital representation of the topography of some terrain, often represented by a regular grid with an elevation for each cell.

**Dog-leg Terrain-** Any terrain that requires a robot to move predominantly away from goal prior to reaching some intermediary point at which the robot then is able to head towards the goal. So named since the route a robot takes on such terrain will generally resemble the shape of a dog's leg.

**Efficiency-limited Regime-** A navigation regime where sensing horizon is determined by desired navigation efficiency and terrain characteristics.

**Grand Challenge-** An event where autonomous robots race long distances across the Mojave Desert to advance robot technology and compete for a monetary prize. The event is sponsored by DARPA.

**LADAR/LIDAR- Laser/Light Detection and Ranging,** the use of the time of flight of a light pulse (generally laser light) to measure range.

**Line Scan LIDAR-** A special type of LIDAR where the beam used to measure range is scanned (generally using a spinning mirror) such that all of the light is emitted in a plane.

**Navigation Regime-** A range of speeds over which the requirements for sensing horizon maintain a consistent relationship.

**Path Shifting-** The local adjustment of a pre-planned route based on sensor measurements.

**Sensing Horizon-** The range out to which a robot's sensors perceive the world.

**Stop-Limited Regime-** A navigation regime where speed is bounded by stopping distance constraints and sensing horizon requirements increase quadratically.

**Stopping Distance-** The distance required for a robot to realize that it must stop and then to physically come to a complete stop. This is a function of the speed at which the robot is moving.

**Summary Surface-** Show how navigation performance varies with sensor horizon and prior map resolution.

**Sun-synchronous Navigation-** Autonomous navigation that reasons about sunlight: where the Sun is in the sky, where and when shadows will fall, and how much power can be obtained through various courses of action. It is accomplished by traveling opposite to planetary rotation, synchronized to the orbital motion of the Sun.

**Swerve Distance-** The distance required for a robot to realize that it must swerve around an obstacle and then to maneuver such that it doesn't impact the obstacle. This is a function of the speed of the robot and the size of the obstacle.

**Swerve-Limited Regime-** A navigation regime characterized high speeds and a linearly increasing sensing horizon.

**Terrainability-** The measure of a vehicle's ability to cross terrain

---

## References

---

- [Acton96]            **Ancillary Data Services of NASA's Navigation and Ancillary Information Facility**  
C. H. Acton Jr.  
*Planetary and Space Science*, 44 (1):65-70, 1996.
- [Amar93]            **On Modeling and Motion Planning of Planetary Vehicles**  
F. Amar, Ph. Bidaud & F. Ouezdou  
*Proc. IEEE/RSJ IROS*, Yokohama, Japan, 1993.
- [Amidi90]           **Integrated Mobile Robot Control**  
O. Amidi  
*Technical Report CMU-RI-TR-90-17*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 1990.
- [Andrade98]        **Modeling of Robot-Soil Interaction for Planetary Rover Motion Control**  
G. Andrade, F. Amara, P Bidaud & R. Chatila  
*Proc. IEEE/RSJ ICIRS*, Victoria, Canada, 1998.
- [Aubert90]         **Autonomous Navigation of Structured City Roads**  
D. Aubert, K. Kluge, and C. Thorpe  
*Proceedings of SPIE Mobile Robots V*, 1990.
- [Balarkisky00]     **World Modeling and Behavior Generation for Autonomous Ground Vehicle**  
S. Balarkisky & A. Lacaze  
*Proc. IEEE ICRA*, San Francisco USA, April 2000.
- [Bares99]          **Dante II: Technical Description, Results, and Lessons Learned**  
J. Bares, D. Wettergreen  
*International Journal of Robotics Research*, Sage Publications, Palo Alto, USA, vol. 18, no. 7, July 1999.
- [Barraquand90]    **A Monte-Carlo Algorithm for Path Planning with Many Degrees of Freedom**  
J. Barraquand & J-C. Latombe  
*Proc. IEEE ICRA*, 1990.
- [Barraquand91]    **Numerical Potential Field Techniques for Robot Path Planning**  
J. Barraquand & J-C. Latombe  
*Proc. IEEE ICRA*, 1991.

- [Baten98]                   **Techniques for Autonomous, Off-Road Navigation**  
S. Baten, M. Lutzeler, E. Dickmanns, R. Mandelbaum & P. Burt  
*IEEE Intelligent Systems*, pp 57-65, November/December 1998.
- [Baumgartner98]           **Mobility Performance of a Small-Body Rover**  
E. Baumgartner, B. Wilcox, R. Welch & R. Jones  
*Int'l Symposium on Robotics with Applications, World Automation Congress*, 1998.
- [Bellutta00]               **Terrain Perception for DEMO III**  
P. Bellutta, R. Manduchi, L. Matthies, K Owens & A. Rankin  
*Proc. IEEE IVS*, Dearborn, USA, October 2000.
- [Bernard99]               **Spacecraft Autonomy Flight Experience: The DS1 Remote Agent Experiment**  
D. Bernard, G. Dorais, E. Gamble, B. Kanefsky, J. Kurien, G. Man, W. Millar, N. Muscettola, P. Nayak, K. Rajan, N. Rouquette, B. Smith, W. Taylor, Y. Tung  
*Proceedings of the AIAA 1999*, Albuquerque, NM, 1999.
- [Biesiadecki01]           **The Athena SDM Rover: a Testbed for Mars Rover Mobility**  
J. Biesiadecki, M. Maimone & J. Morrison  
*Proc. I-SAIRAS 2001*, St-Hubert, Canada, June, 2001.
- [Branicky02]               **Deterministic vs. Probabilistic Roadmaps**  
M. Branicky, S. LaValle, K. Olsin & L. Yang  
2002.
- [Bresina01]               **K9 Operations in May'00 Dual-Rover Field Experiment**  
J.L. Bresina, M.G. Bualat, L.J. Edwards, R.M. Washington and A.R. Wright  
*Proc. I-SAIRAS 2001*, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18-22, 2001.
- [Brock01]                  **Decomposition-based Motion Planning: A Framework for Real-time Motion Planning in High-dimensional Configuration Spaces**  
O. Brock & L. Kavraki  
*Proc. IEEE ICRA*, Seoul, Korea, 2001.
- [Bussey99]                 **Illumination Conditions at the Lunar Poles**  
D. B. J Bussey, M. S. Robinson, P. D. Spudis  
*Lunar and Planetary Science 30*, abstract #1731, Lunar and Planetary Institute, Houston, TX, March 1999.

- [Chanclo96a]      **Global and Local Path Planning in Natural Environment by Physical Modeling**  
B. Chanclo & A. Luciani  
*Proc. IEEE IROS*, 1996.
- [Chanclo96b]      **Physical Modeling and Dynamic Simulation of Off-Road Vehicles and Natural Environments**  
B. Chanclo & A. Luciani  
*Proc. IEEE IROS*, 1996.
- [Cherif99a]        **Motion Planning for All-Terrain Vehicles: A Physical Modeling Approach for Coping with Dynamic and Contact Interaction Constraints**  
M. Cherif  
*IEEE Transactions on Robotics and Automation*, Vol. 15, No 2. 1999.
- [Cherif99b]        **Kinodynamic Motion Planning for All-Terrain Wheeled Vehicles**  
M. Cherif  
*Proc. IEEE ICRA*, Detroit USA, 1999.
- [Chien00]          **ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling**  
S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, D. Tran  
*SpaceOps 2000*, Toulouse, France, June 2000.
- [Choset95]         **Sensor Based Planning, Part I: The Generalized Voronoi Graph**  
H. Choset & J. Burdick  
*Proc. IEEE ICRA*, 1995.
- [Coombs00]        **Driving Autonomously Off-road up to 35km/h**  
D. Coombs, K. Murphy, A. Lacaze & S. Legowik  
*Proc. IEEE IVS*, Dearborn USA, 2000.
- [Coulter92]        **Implementation of the Pure Pursuit Path Tracking Algorithm**  
R. Coulter  
*Technical Report CMU-RI-TR-92-01*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 1992.
- [Dickmanns86]     **A Curvature-Based Scheme for Improving Road Vehicle Guidance by Computer Vision**  
E. Dickmanns and A. Zapp.



*Proc. SPIE Conference on Mobile Robots*, 1986.

- [Drewes03]      **Demonstration of a Systems Architecture for Live, Virtual, and Constructive UGV Operation**  
P. Drewes  
*AUVSI*, Baltimore, USA, 2003.
- [Farritor98]      **Physics-Based Planning for Planetary Exploration**  
S. Farritor, H. Hacot & S. Dubowsky  
*Proc. IEEE ICRA*, Leuven, Belgium, 1998.
- [Feddema97]      **Rapid World Modeling: Fitting Range Data to Geometric Primitives**  
J. Feddema & C. Little  
*Proc. IEEE ICRA*, Albuquerque, USA, April 1997.
- [Goldberg02]      **Stereo Vision and Rover Navigation Software for Planetary Exploration**  
S. Golberg, M. Maimone & L. Matthies  
*Proc. IEEE Aerospace Conference*, Big Sky, USA, March 2002.
- [Golembek97]      **Size-frequency distributions of rocks on Mars and Earth analog sites: Implications for future landed missions**  
M. Golembek & D. Rapp  
*Journal of Geophysical Research*, Vol. 102, No. E2, pp 4117-4129, 1997.
- [Haddad98]      **Reactive Navigation in Outdoor Environments using Potential Fields**  
H. Haddad, M. Khatib, S. Lacroix & R. Chatila  
*Proc. IEEE ICRA*, Leuven, Belgium, 1998.
- [Hait96]      **Motion Planning on Rough Terrain for an Articulated Vehicle in Presence of Uncertainties**  
A. Hait & T. Simeon  
*Proc. IEEE IROS*, 1996.
- [Hait99]      **Robust Motion Planning for Rough Terrain Navigation**  
A. Hait, T. Simeon & M. Taix  
*Proc. IEEE/RSJ IROS*, 1999.
- [Howard00]      **Real-Time Assessment of Terrain Traversability for Autonomous Rover Navigation**  
A. Howard & H. Seraji  
*Proc. IEEE/RSJ IROS*, 2000.

- [Howard01]      **An Intelligent Terrain-Based Navigation System for Planetary Rovers**  
A. Howard & H. Seraji  
*IEEE Robotics and Automation Magazine*, Vol. 8, No. 4, Dec. 2001.
- [Hsu97]      **Path Planning in Expansive Configuration Spaces**  
D. Hsu, J-C. Latombe & R. Motwani  
*Proc. IEEE ICRA*, Albuquerque, USA, 1997.
- [Iagnemma99]      **Rapid Physics-Based Rough-Terrain Rover Planning with Sensor and Control Uncertainty**  
K. Iagnemma, F. Genot & S. Dubowsky  
*Proc. IEEE ICRA*, Detroit, USA, 1999.
- [Iagnemma01]      **Planning and Control Algorithms for Enhanced Rough-Terrain Rover Mobility**  
K. Iagnemma, H. Shibly, A. Rzepniewski & S. Dubowsky.  
*Proc. I-SAIRAS 2001*, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18-22, 2001.
- [Kavraki94]      **Randomized Preprocessing of Configuration Space for Fast Planning**  
L. Kavraki & J-C. Latombe  
*Proc. IEEE ICRA*, 1994.
- [Kavraki96]      **Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces**  
L. Kavraki, P. Svestka, J-C. Latombe & M. Overmars  
*IEEE Transactions on Robotics and Automation*. Vol. 12, Issue 4, August 1996.
- [Kelly94]      **A Partial Analysis of the High Speed Autonomous Navigation Problem**  
A. Kelly  
*Technical Report CMU-RI-TR-94-16*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1994.
- [Kelly95]      **An Intelligent Predictive Control Approach to the High-Speed Cross-Country Autonomous Navigation Problem**  
A. Kelly  
*Ph.D. Thesis, CMU-RI-TR-95-33*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1995.
- [Kelly97]      **An Analysis of Requirements for Rough Terrain Autonomous Mobility**  
A. Kelly & A. Stentz

*Autonomous Robots*, Vol. 4, No. 4, December, 1997.

- [Kelly04] **Toward Reliable Off-Road Autonomous Vehicle Operating in Challenging Environments**  
A. Kelly, O. Amidi, M. Happold, H. Herman, T. Pilarski, P. Rander, A. Stentz, N. Vallidis, R. Warner  
*International Symposium on Experimental Robotics*, June, 2004, Singapore.
- [Khatib86] **Real-Time Obstacle Avoidance for Manipulators and Mobile Robots**  
O. Khatib  
*The International Journal of Robotics Research*, Vol. 5. No. 1, Spring 1986.
- [Kindel00] **Kinodynamic Motion Planning Amidst Moving Obstacles**  
R. Kindel, D. Hsu, J-C. Latombe & S. Rock  
*Proc. IEEE ICRA*, San Francisco, USA, 2000.
- [Kuffner00] **RRT-Connect: An Efficient Approach to Single-Query Path Planning**  
J.J. Kuffner & S. LaValle  
*Proc. IEEE ICRA*, San Francisco, USA, 2000.
- [Lacroix02] **Autonomous Rover Navigation on Unknown Terrains: Functions and Integration**  
S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb and R. Chatila  
*International Journal of Robotics Research*, 21(10-11), pages 917-942, 2002.
- [Latombe91] **Robot Motion Planning**  
J-C. Latombe  
Kluwer Academic Publishers, 1991.
- [Laubach99] **An Autonomous Sensor-Based Path-Planner for Planetary Microrovers**  
S. Laubach & J. Burdick  
*Proc. IEEE ICRA*, Detroit, USA, 1999.
- [LaValle00] **Rapidly-exploring random trees: Progress and prospects**  
S. LaValle & J. Kuffner  
*Proc. Workshop on Algorithmic Foundations of Robotics*, Hanover, USA, Mar. 2000.

- [LaValle01]      **Randomized Kinodynamic Planning**  
S. LaValle & J. Kuffner  
*The International Journal of Robotics Research*, Vol. 20,  
No. 5, May 2001 pp. 378-400.
- [MartinAlvarez99]      **Fuzzy Reactive Piloting for Continuous Driving of Long Range Autonomous Planetary Micro-Rovers**  
A. Martin-Alvarez, R. Volpe, S. Hayati & R. Petras  
*Proc. IEEE Aerospace Conference*, 1999.
- [Martz96]      **Generating Random Fractal Terrain**  
P. Martz  
<http://www.gameprogrammer.com/fractal.html>, 1996.
- [Miller03]      **Experiments With a Long-Range Planetary Rover**  
D. P. Miller, T. Hunt, M. Roman, S. Swindell, L. Tan & A. Winterholler  
*Proc. I-SAIRAS 2003*, Nara, Japan, May 19-23, 2003
- [Nashashibi92]      **Indoor Scene Terrain Modeling using Multiple Range Images for Autonomous Mobile Robots**  
F. Nashashibi, M. Devy, P. Fillatreau  
*Proc. IEEE ICRA*, Nice, France 1992.
- [Pomerleau95]      **RALPH: Rapidly Adapting Lateral Position Handler**  
D. Pomerleau  
*Proc. IEEE IVS*, September, 1995, pp. 506 - 511.
- [Schenker01]      **FIDO: A Field Integrated Design & Operations Rover for Mars Surface Exploration**  
P.S. Schenker, E.T. Baumgartner, P.G. Backes, H. Aghazarian, L.I. Dorsky, J.S. Norris, T.L. Huntsberger, Y. Cheng, A. Trebi-Ollennu, M.S. Garrett, B.A. Kennedy, A.J. Ganino, R.E. Arvidson, S.W. Squyres  
*Proc. I-SAIRAS 2001*, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18-22, 2001.
- [Scherzinger00]      **Precise Robust Positioning with Inertial/GPS RTK**  
B. Scherzinger  
*Proc. ION-GPS*, Salt Lake City, USA, 2000.
- [Seraji99]      **Traversability Index: A New Concept for Planetary Rovers**  
H. Seraji  
*Proc. IEEE ICRA*, Detroit USA, 1999.

- [Seraji01]           **Safe Navigation on Hazardous Terrain**  
H. Seraji, A Howard & E. Tunstel  
*Proc. IEEE ICRA*, Seoul, Korea, 2001.
- [Shamah01]           **Steering and Control of a Passively Articulated Robot**  
B. Shamah, M. D. Wagner, S. Moorehead, J. Teza, D. Wettergreen, W. Whittaker  
*SPIE, Sensor Fusion and Decentralized Control in Robotic Systems IV*, Vol. 4571, October, 2001.
- [Shiller91]           **Dynamic Motion Planning of Autonomous Vehicles**  
Z. Shiller  
*IEEE Transaction on Robotics and Automation*, Vol. 7, No. 2, April 1991.
- [Shiller99]           **Motion Planning for Mars Rover**  
Z. Shiller  
1999.
- [Shiller00a]           **Obstacle Traversal for Space Exploration**  
Z. Shiller  
*Proc. IEEE ICRA*, San Francisco, USA, 2000.
- [Shiller00b]           **Online Suboptimal Obstacle Avoidance**  
Z. Shiller  
*The International Journal of Robotics Research*, Vol. 19, No. 5, May 2000, pp. 480-497.
- [Shrounk95]           **Sun-Synchronous Operation**  
D. Shrounk  
*Workshop Discussion*, ISE Lunar Conference, 1995.
- [Simeon91]           **Motion Planning for a Non-Holonomic Mobile Robot on 3-Dimensional Terrains**  
T. Simeon  
*Proc. IEEE/RSJ IROS*, Osaka Japan, 1991.
- [Simeon93]           **A Practical Motion Planner for All-terrain Mobile Robots**  
T. Simeon & B. Dacre-Wright  
*Proc. IEEE/RSJ IROS*, Yokohama, Japan, 1993.
- [Simmons95]           **Experience with Rover Navigation for Lunar-Like Terrains**  
R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, L. Katragadda, S. Koenig, G. Krishnaswamy, Y. Shinoda, W. Whittaker, & P. Klarer  
*Proc. IEEE IROS*, 1995.

- [Simmons01] **IPC Technical Doc**  
R. Simmons  
<http://www.cs.cmu.edu/afs/cs/project/TCA/www/ipc/ipc.html>,  
May 2001.
- [Singh89] **Position Based Path Tracking for Wheeled Mobile Robots**  
S. Singh and D.H. Shin  
*Proc. IEEE IROS*, September, 1989.
- [Singh91] **A System for Fast Navigation of Autonomous Vehicles**  
S. Singh, D. Feng, P. Keller, G. Shaffer, W. Shi, D.H. Shin, J. West, and B.X. Wu  
*Technical Report CMU-RI-TR-91-20*, Robotics Institute, Carnegie Mellon University, September, 1991.
- [Singh00] **Recent Progress in Local and Global Traversability for Planetary Rovers**  
S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja & K. Schwehr  
*Proc. IEEE ICRA*, San Francisco, USA, 2000.
- [Spenko04] **High Speed Hazard Avoidance for Mobile Robots in Rough Terrain**  
Spenko, M., Iagnemma, K., and Dubowsky, S  
*Proc. SPIE Conference on Unmanned Ground Vehicle Technology VI*, Orlando, FL, Vol. 5422, April 2004.
- [Stentz94] **Optimal and Efficient Path Planning for Partially-Known Environments**  
A. Stentz  
*Proc. IEEE ICRA*, May 1994.
- [Stentz95a] **A Complete Navigation System for Goal Acquisition in Unknown Environments**  
A. Stentz & M. Hebert  
*Proc. IEEE/RSJ IROS*, August, 1995, pp. 425 - 432.
- [Stentz95b] **The Focused D\* Algorithm for Real-Time Replanning**  
A. Stentz  
*Proc. IJCAI*, August 1995.
- [Thorpe90] **Vision and Navigation: The Carnegie Mellon Navlab**  
C. Thorpe  
Kluwer Academic Publishers, 1990.

- [Thorpe97]      **The 1997 Automated Highway Free Agent Demonstration**  
C. Thorpe, T. Jochem, and D. Pomerleau  
*IEEE Conference on Intelligent Transportation Systems*,  
November, 1997, pp. 496 - 501.
- [Thorpe02]      **Driving in Traffic: Short-Range Sensing for Urban Collision Avoidance**  
C. Thorpe, D. Duggins, J. Gowdy, R. MacLachlan, C. Mertz, M. Siegel, A. Suppe, C. Wang, and T. Yata  
*Proceedings of SPIE: Unmanned Ground Vehicle Technology IV*, Vol. 4715, April, 2002.
- [Tompkins01]      **Automated Surface Mission Planning Considering Terrain, Shadows, Resources and Time**  
P. Tompkins, A. Stentz, W. Whittaker  
*Proc. I-SAIRAS 2001*, St-Hubert, Canada, June, 2001.
- [Tompkins02]      **Mission Planning for the Sun-Synchronous Navigation Field Experiment**  
P. Tompkins, A. Stentz, W. Whittaker  
*Proc. IEEE ICRA 2002*, Washington D.C., May 2002.
- [Urmson02]      **Stereo Vision Based Navigation for Sun-Synchronous Exploration**  
C. Urmson, M. Dias  
*Proc. IEEE ICRA 2002*, Washington D.C., May 2002.
- [Urmson03]      **Approaches for Heuristically Biasing RRT Growth**  
C. Urmson, R. Simmons  
*Proc. IEEE/RSJ IROS*, Las Vegas, USA, 2003.
- [Verma01]      **Non-Parametric Fault Identification for Space Rovers**  
V. Verma, J. Langford, and R. Simmons  
*Proc. I-SAIRAS 2001*, St-Hubert, Canada, June, 2001.
- [Visual02]      **AirRECON V® - The 5th Generation micropixel digitally collected color aerial photo system**  
Visual Intelligence  
<http://www.visidata.com/visisource/htm/Whitepaper.htm>
- [Volpe00a]      **Enhanced Mars Rover Navigation Techniques**  
R. Volpe, T. Estlin, S. Laubach, C. Olson & J. Balaram  
*Proc. IEEE ICRA*, San Francisco, USA, 2000.
- [Volpe00b]      **Technology Development and Testing for Enhanced Mars Rover Sample Return Operations**  
R. Volpe, E. Baumgartner, P. Schenker, S. Hayati

*Proc. IEEE Aerospace Conference, 2000.*

- [Volpe01]      **The CLARAty Architecture for Robotic Autonomy**  
R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das  
*Proc. IEEE Aerospace Conference 2001*, Big Sky MT,  
March 2001.
- [Wettergreen01]      **Robotic Planetary Exploration by Sun-Synchronous Navigation**  
D. Wettergreen, B. Shamah, P. Tompkins, W. Whittaker  
*Proc. I-SAIRAS 2001*, St-Hubert, Canada, June, 2001.
- [Wettergreen02]      **Experiments in Sun-Synchronous Navigation**  
D. Wettergreen, B. Dias, B. Shamah, J. Teza, P. Tompkins,  
C. Urmson, M. Wagner, W. Whittaker  
*Proc. IEEE ICRA*, Washington D.C., May 2002.
- [Whittaker00]      **Sun-Synchronous Planetary Exploration**  
W.L. Whittaker, G.A. Kantor, B. Shamah, and D.  
Wettergreen  
*Proc. AIAA Space 2000*, September, 2000.
- [Wong78]      **Theory of Ground Vehicles**  
Wong, J.C  
John Wiley & Sons, New York, 1978
- [Woodfill97]      **Real-Time Stereo Vision on the PARTS Reconfigurable Computer**  
J. Woodfill, B. Von Herzen  
*Proc. IEEE Symposium on Field-Programmable Custom Computing Machines*, Napa, pp. 242-250, April 1997.