NOVA: Nottingham Off-road Vehicle Architecture

Jamie Robert Strachan, BSc.

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

July 2009

ABSTRACT

This thesis describes a program of research aimed at the creation of an unmanned ground vehicle. In this research the Nottingham Off-road Vehicle Architecture (NOVA) was developed along with the ARP (Autonomous Route Proving) vehicle. NOVA is a control architecture for a vehicle with the role of autonomous route proving in natural terrain. The ARP vehicle was constructed to demonstrate this architecture.

NOVA includes all the required competence for the ARP vehicle to be deployed in unknown outdoor environments. The architecture embodies systems for vehicle localisation, autonomous navigation and obstacle avoidance.

The localisation system fuses data from absolute and relative localisation equipment. GPS provides the absolute position of the ARP vehicle. Relative position information is derived from wheel encoders and a pose sensor. NOVA uses a probabilistic technique known as a particle filter to combine the two position estimates.

NOVA maintains a local obstacle map based on range data generated by the perception sensors on the ARP vehicle. Analysis is performed on this map to find any untraversable terrain. A local path planner then selects the best path for the vehicle to follow using the map. Decisions made by the path planner are recorded to allow the vehicle to backtrack and try another path if NOVA later finds the chosen route is blocked.

NOVA has been extensively tested onboard the ARP vehicle. Results from a series of experiments are presented to validate the various parts of the architecture.

ACKNOWLEDGEMENTS

This research was made possible by funding from the Engineering and Physical Sciences Research Council.

I would like to offer my gratitude to my employer Remotec UK who have supported me throughout this research. Special thanks must go to John Spence who allowed me to use Remotec UK parts and facilities for the development of the ARP vehicle. Additionally I must thank Paul Foster, Roger Jubb & Chris King for helping me rescue and repair ARP whenever it broke down. John Ellis deserves a thank you for his assistance during the experimental phase of this research. Also thanks to Ed Gummow for helping with the proofreading of this thesis and showing amazing ability to spot obscure errors.

Thanks are due to the School of Mechanical, Materials & Manufacturing Engineering at the University of Nottingham. My supervisor Dr Phil Webb deserves particular acknowledgement for his guidance from the initial research idea through to the thesis proofreading. Also thanks to Shane Maskill & Mick Joliffe for their valuable input into the design and manufacture of the many plates and mounting brackets that were required to construct ARP.

Finally I must thank the woman of my life Dr Elizabeth Gordon [Gordon 2005] who is still talking to me despite the many hours I have put into this research. We are probably even now on the proofreading front.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS & ACRONYMS

ADC	Analogue to Digital Converter
AGV	Automated Guided Vehicle
ALV	Autonomous Land Vehicle
ARP	Autonomous Route Proving
AUV	Autonomous Underwater Vehicle
CAD	Computer Aided Design
CCD	Charge Coupled Device
CD	Compact Disc
CMU	Carnegie Mellon University
CPU	Central Processing Unit
CRASAR	Center for Robotic Assisted Search and Rescue
CSEM	Centre Suisse d'Electronique et de Microtechnique
DAMN	Distributed Architecture for Mobile Navigation
DARPA	Defense Advanced Research Projects Agency
DGPS	Differential Global Positioning System
EGNOS	European Geostationary Navigation Overlay Service
ELROB	European Land Robotics Trial
EOD	Explosive Ordinance Disposal
EPSRC	Engineering and Physical Sciences Research Council
ESA	European Space Agency
GPS	Global Positioning System
GRS80	Geodetic Reference System 1980
GSR	Ground Surveillance Robot

GUI	Graphical User Interface
HC	Hand Controller
HMMWV	High Mobility Multipurpose Wheeled Vehicle
INS	Inertial Navigation System
JPL	Jet Propulsion Laboratory
LADAR	Laser Detection and Ranging
LED	Light Emitting Diode
MDARS	Mobile Detection Assessment and Response System
MoD	Ministry of Defence
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NAMR	Nottingham Autonomous Mobile Robot
NASA	National Aeronautics and Space Administration
NAVLAB	Navigational Laboratory
NIST	National Institute for Standards and Technology
NMEA	National Marine Electronics Association
NOVA	Nottingham Off-road Vehicle Architecture
OSGB36	Ordnance Survey Great Britain 1936
POD	Polar Obstacle Density
PRIMUS	Program of Intelligent Mobile Unmanned Systems
RAM	Random Access Memory
SBAS	Space Based Augmentation Service
SIS	Sequential Importance Sampling
UGV	Unmanned Ground Vehicle
USB	Universal Serial Bus
VCU	Vehicle Control Unit

VFH	Vector Field Histogram
VFF	Vector Force Field
WGS84	World Geodetic System 1984
VMT	Vehicle Model Trajectory
WTC	World Trade Center
XUV	Experimental Unmanned Vehicle

1 INTRODUCTION

This thesis introduces the Nottingham Off-road Vehicle Architecture (NOVA) for autonomous route proving. NOVA is a modular control architecture for an unmanned ground vehicle (UGV) that provides motion control, vehicle localisation, autonomous path planning & obstacle avoidance. NOVA was created to demonstrate that the technology now exists to automate hazardous tasks such as landmine detection that put human beings in mortal danger.

NOVA has been developed under a joint project between the Robotics Research Group within the School of Mechanical, Materials and Manufacturing Engineering at the University of Nottingham and unmanned ground vehicle manufacturer Remotec UK. A large part of the funding for this work was provided by the Engineering and Physical Sciences Research Council (EPSRC).

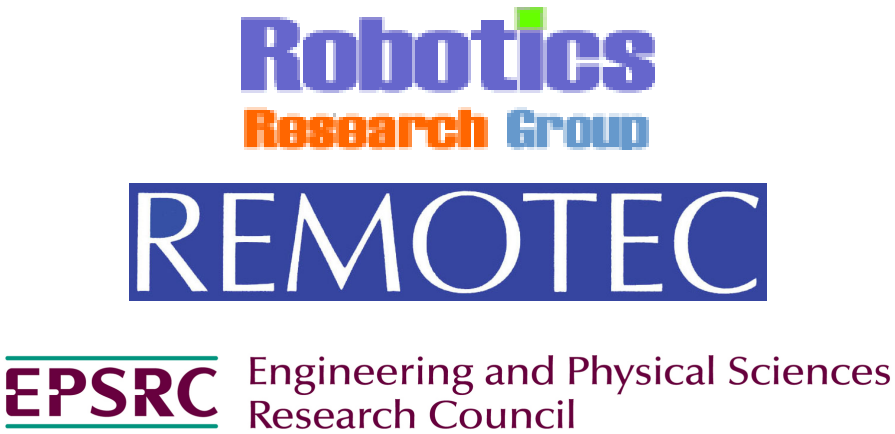

Figure 1.1 - Project Partner Logos

At present the majority of unmanned ground vehicles are used in the military realm. Tasks such as Explosive Ordinance Disposal (EOD) have employed remotely operated vehicles for over 30 years [Ryder 2006]. Most modern EOD vehicles are still entirely teleoperated with every motor drive controlled individually by a remote human operator rather than having

autonomous functions. This is the case for the Remotec Revolution EOD vehicle shown in Figure 1.2.

Figure 1.2 - Remotec Revolution

In contrast simple autonomous behaviours are beginning to appear in every day consumer items. The iRobot Roomba robotic vacuum cleaner is an example [iRobot 2007]. This device can systematically vacuum a room and even return to a charging point when the battery needs to be recharged. It features a basic motor driven platform that has been augmented with a number of inexpensive obstacle detection sensors. The control architecture is capable of a basic level of localisation, navigation, motion control and obstacle avoidance.

There has been a growing demand for unmanned vehicles in recent years. This has been coupled with a greater requirement for autonomy from the vehicle as more complex roles are envisaged. At present a great deal of research is being directed towards the realisation of autonomous unmanned vehicles.

Recent high profile events such as the US DARPA sponsored Grand Challenge and Urban Challenge have seen the development of sophisticated robotic vehicles. The aim of the Grand

Challenge was for the vehicles to autonomously complete an off-road route through the Mojave Desert. No entry managed to exceed 5% of the 142 mile course during the first Grand Challenge in March 2004. At the second Grand Challenge in October 2005 five teams completed the 132 mile course with Stanley the Stanford University entry setting the fastest time. Figure 1.3 shows Stanley during the 2005 DARPA Grand Challenge [Thurn 2006].

Figure 1.3 - Stanley Winner of the 2005 DARPA Grand Challenge

Other robotic trials are springing up all over the world as a result of the success of the DARPA challenge events. European Robotics now annually stages ELROB the European Land Robotics trials. Also in August 2008 the UK MoD held the first UK Grand Challenge for smaller robotic vehicles. The overall aim of these competitions and indeed the work presented in this thesis is summarised by the following points:

- To push forward the technology of unmanned ground vehicles
- To find opportunities to replace people in hazardous situations with robots
- To reduce the workload of the human operators of unmanned vehicles
- To find new methods to perform traditional tasks using a robot

There is a gap between the academic unmanned ground vehicle developments and those from industry. Academic projects tend to have cutting edge features but may not be robust in operation. Industry needs to supply robust systems as unmanned ground vehicles tend to be deployed in situations where a wrong move could result in disaster. Hence despite the research successes a number of open research areas still remain.

NOVA is a robust control architecture that offers solutions to a number of identified challenges. In particular NOVA includes a novel method for dealing with the local navigation problem. Additionally NOVA provides a means to fuse data from various localisation sensors that can cope with sensor error.

1.1 PREVIOUS WORK AT NOTTINGHAM

This research is part of an ongoing mobile robotics program within the University of Nottingham. Other contributions to the field have been made by members of the Robotics Research Group within the School of Mechanical, Materials and Manufacturing Engineering. In recent years the group has developed two mobile robots in projects looking at human machine interfaces and autonomous mobile robot navigation.

The first robot as shown in Figure 1.4 was based on the chassis of a Mk 7 Wheelbarrow EOD Vehicle modified to accept motor demands from a tethered computer based control system. This system was originally developed as part of an investigation into intuitive user interfaces for the operation of remote control vehicles [Oudijk 1997]. The robot featured sensors to feedback the position of actuators so the robot configuration could be displayed to the user.

Later improvements [Breitenbach 2000] introduced a control architecture following the Subsumption Architecture concept [Brooks 1986] to give the system more autonomy. Novelty for the project came from the navigation layer of the architecture that provided behaviours to

steer toward a target and to reactively avoid obstacles [Webb 2000]. The implementation used a 2D laser scanner to find the closest obstacle point then a fuzzy logic controller produced an escape vector for the robot based on the rule sets for the two behaviours. If the closest obstacle was far away or at a large angle the controller would give a higher weighting to the steer toward target behaviour. Otherwise the obstacle avoidance rules would have higher weight leading the robot away from the obstacle.

Figure 1.4 - Mk 7 Wheelbarrow with layered control architecture

The navigation layer used onboard the robot was part of a much larger navigation kernel created in simulation for efficiently exploring known, partially known and completely unknown environments [Fayad 2001]. Within this kernel a system was developed using a genetic algorithm to optimise the parameters of the fuzzy logic controller according to a fitness function that favoured shorter paths and greater obstacle clearance.

NAMR the Nottingham Autonomous Mobile Robot was the second system to be developed within the research group. This platform was based upon an Evolution Robotics Inc development kit. An embedded PC was incorporated to run the high level control software and a 2D laser scanner was used to observe the environment. NAMR is pictured in Figure 1.5.

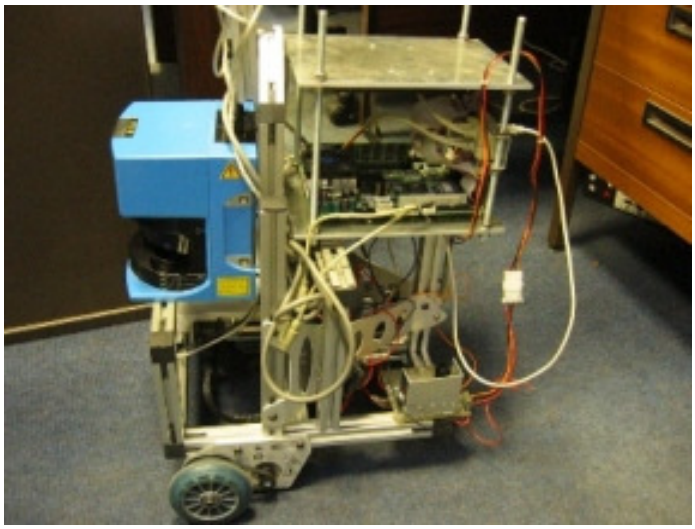

Figure 1.5 - NAMR

NAMR has been used in further research into local path planning for unknown indoor environments [Ye 2006]. In this research a new Sub Goal Seeking reactive navigation system was devised and validated both in simulation and onboard NAMR. The purpose of the Sub Goal Seeking system was to lead the robot to a sequence of visible sub goals until the final target was in sight. In each cycle the system searched for passable gaps in the observed obstacle environment. A gap was selected according to a cost function with weighted terms intended to turn the robot towards the target and also try to maintain the present heading. The final turning angle for the sub goal was set as the minimum turning angle for the robot to pass through the selected gap.

Performance of the Sub Goal Seeking navigation system was evaluated against the fuzzy logic system developed earlier within the group. The Sub Goal Seeking system was generally found to perform better. In cluttered environments the robot trajectory was typically less ballistic and could guide the robot through smaller gaps if they offered a more direct path to the target. The parameter controlling the detection range for obstacle avoidance was found to be critical in the performance of the fuzzy logic based method. Smaller values for the parameter allowed the robot to closer approach obstacles but at a greater risk of becoming trapped in a cul-de-sac.

1.2 MOTIVATION FOR PROJECT

Remotec was involved with the MINDER program funded by the UK MoD. The remit of the MINDER program was to develop a series of vehicles for the complementary tasks of landmine detection, neutralisation and route marking [Bewsher 2001]. It was envisaged that a teleoperator would drive the mine detection vehicle from the relative safety of a following command vehicle. These vehicles could then be used to prove routes in advance of a following convoy or to clear entire areas of strategic importance. Figure 1.6 shows a prototype of the mine detection vehicle.

Figure 1.6 - MINDER Prototype Mine Detection Vehicle

Unfortunately this work was suspended for reasons including a lack of research into certain aspects of the technology. The remainder of the project was due to feature a phase to identify aspects of the mine detection task that could be automated in order to reduce the workload of the teleoperator. The identification and implementation of a system to perform those autonomous behaviours thus formed the basis for this research. A key requirement for this phase of MINDER was the development of a localisation system that was GPS based but could cope if a GPS signal was unavailable temporarily or for an extended period.

1.3 STATEMENT OF OBJECTIVES

The ultimate objective of this research was to develop an unmanned ground vehicle capable of a high standard of autonomous operation in an unknown outdoor environment. It was expected that the operational environment would contain elements of untraversable terrain that should be detected and avoided. To that end it was important to recognise the challenges of building an autonomous unmanned system and appreciate the practical limitations of such a system. The vehicle platform dictates what obstacles are safe to traverse. The obstacle detection sensor configuration dictates what obstacles can be detected and so avoided.

To satisfy the main objective it was clear that a test vehicle would have to be developed featuring a layered or modular control architecture. This architecture must embody systems to localise the vehicle and also detect any untraversable terrain. It would be beneficial if the architecture could be easily adapted to accommodate new sensors and algorithms.

The localisation system would need to be GPS based if the system was to be used outdoors in natural terrain. Data from additional localisation sensors would need to be fused with the GPS solution to allow the system to cope when a GPS signal was not available.

Previous robots developed within the Robotics Research Group only considered obstacle detection in 2D using laser scanners mounted horizontal to the ground plane. Each robot was designed with the implicit assumption that they would operate only on flat ground. Navigation in natural terrain requires detection of the ground and discrimination of a traversable path between any positive and negative ground features. For this methods to create 3D range data need to be used.

As several sensors were considered for obstacle detection it was also important to create an obstacle avoidance system that was not sensor specific. In fact the possibility of accepting data from various sources and fusing the results was an attractive feature.

The application for this unmanned ground vehicle system is autonomous route proving. Route proving often utilises a vehicle carrying sensors such as metal detectors and ground penetrating radar to locate landmines. Armoured ploughs or flails are typically used to unearth and destroy buried mines. Havlík [Havlík 1998] gives an introduction to robotic demining operations and provides several example systems. No effort was made in this research to equip the development vehicle with landmine detection or neutralisation tools.

Figure 1.7 - Mine Flail

Demining vehicles are typically required to cover a search area systematically usually with a sequence of overlapping swaths. An objective was thus to create a system that could produce overlapping parallel swaths of a specified route. It was also desirable to allow the perimeter of an area to be specified so that the interior could then be proven using a sequence of swaths.

The final objective of the system was to create a route and decision recording system capable of recording the path of the UGV coupled with the path planning decisions that were taken at each step. Route recording was desired to allow the UGV to return from a demining task using a proven route over the landmine laden ground. Decision recording would allow the rationalisation of a route that the vehicle had found to be blocked. Examination of how such a system could be used to prevent an autonomous unmanned vehicle from falling foul of common local path planner problems such as becoming trapped in cul-de-sacs was warranted.

1.4 OUTLINE OF WORK CONDUCTED

During the research a number of significant tasks have been completed. These have included the implementation of NOVA as a modular UGV control architecture. A UGV equipped with a variety of sensing equipment has been developed. Finally a number of experiments were conducted with the UGV to validate NOVA.

Construction of the ARP vehicle was the initial task allowing it to be used as the development platform for the remainder of the research. A donated robot chassis was refurbished and furnished with a new control system to give a basic mobile vehicle. Sensors for localisation and obstacle detection were then added to the chassis. Next a computer system was integrated to manage the whole vehicle. The ARP Interface application was then developed to provide manual control of the vehicle and display the feedback from the onboard sensors.

NOVA was developed on a module by module basis. First the Motion Module was implemented to allow the vehicle to be driven under manual or autonomous control. The Location Module was added soon after to provide a localisation system for the vehicle. This module was developed in parts allowing first relative then absolute localisation before a suitable means to fuse the two parts was devised. Autonomous driving behaviours were then added with the Route Planner Module that provided waypoint navigation.

Next the Protection Module framework was designed as a means to process range data from obstacle detection sensors. An associated sensor independent obstacle mapping system was also created to provide storage for processed range data. This allowed trials to be undertaken with the different obstacle detection sensors that were made available. Finally a local path planning capability was added to the Route Planner Module to allow the vehicle to perform waypoint navigation avoiding obstacles en route.

When complete NOVA was extensively tested and debugged in a series of trials onboard the ARP vehicle. Experiments were then performed to test that the various parts of NOVA worked as intended. Separate experiments were conducted to demonstrate the capabilities of the localisation system, obstacle mapping, local path planning and decision recording.

1.5 THESIS STRUCTURE

The background to this research is provided by Chapter 2. First an introduction to unmanned ground vehicles is given. This is followed by a summary of the popular control architecture paradigms. The chapter concludes with a survey of sensors used for obstacle detection and localisation onboard unmanned ground vehicles.

Chapter 3 reviews the literature from related developments. This chapter is segregated into literature concerning major unmanned ground vehicle research, vehicle localisation, generation of 3D range sensor data, assessment of terrain traversability and local navigation concepts. Finally the sources of direct inspiration for this project are presented.

The development path for NOVA is described in Chapter 4. First gaps in the research literature are highlighted and the solutions provided in this research are discussed. Next a general concept for the architecture when operating an unmanned vehicle is given. Finally the framework of NOVA is presented and the modules within are detailed.

Chapter 5 introduces the ARP vehicle that was constructed as a test platform for NOVA. Hardware onboard the APR vehicle is examined with particular attention paid to perception and localisation sensors. The control interfaces developed for the vehicle are also presented in this chapter.

An analysis of the UGV localisation system is given in Chapter 6. This covers the processes for deriving the vehicle position from the absolute and relative localisation equipment. Finally the probabilistic method used to fuse data from the various sources is detailed.

The UGV obstacle avoidance system is discussed in Chapter 7. System stages include the processing of perception sensor data, obstacle mapping, analysis of terrain traversability and local path planning. Each stage of this process is detailed in turn. The use of decision recording to escape from cul-de-sacs is also introduced.

A sequence of experiments with NOVA using the ARP vehicle is detailed in Chapter 8. Experimental results are also presented and discussed. The primary aim of the experiments was to prove the correct operation of the architecture. In addition alternate strategies for certain tasks within the architecture are trialled.

Chapter 9 concludes this thesis. Here the research is summarised and the findings are presented. A list of contributions from the research is also given. As part of this the novel aspects of the research are highlighted. Finally ideas are suggested for future developments in this research.

2 BACKGROUND

The aim of this chapter is to provide general background information for the research presented in this thesis. It is intended that the information contained here should bridge the gap from the initial research idea to the novel elements of the research.

As such the chapter begins by outlining the general idea of an Unmanned Ground Vehicle (UGV). The concept of autonomy is introduced and examples are given of the various levels of autonomy that a UGV could exhibit. A number of applications for unmanned vehicles are then explored.

The remainder of the chapter considers the make up of an autonomous unmanned ground vehicle. Paradigms for the control architecture of a UGV are discussed. A survey of popular perception sensors for use in UGV applications is given. This is followed by a similar survey of UGV localisation methods.

2.1 UNMANNED GROUND VEHICLES

2.1.1 What is a UGV?

Unmanned vehicles are devices that can move around and perform tasks but do not need an operator onboard. It is normal for the operator of an unmanned vehicle to direct the behaviour of the vehicle from a remote location. This research specifically considers unmanned vehicles that operate on the ground. Unmanned vehicles are also deployed in airborne, waterborne, underwater and space environments.

An unmanned ground vehicle should be seen as a tool to perform a task that does not require the close attendance of an operator. As such the vehicle can be sent into hazardous locations in place of human beings. Additionally unmanned ground vehicles could be substituted for

people or manned vehicles that were doing laborious or repetitive tasks such as carrying loads or following a leader.

Technology has moved on rapidly from the first unmanned ground vehicles. Early vehicles deployed for explosive ordnance disposal were set to drive forwards and steered by an operator pulling ropes attached to the steering mechanism [Ryder 2006]. Today the majority of unmanned ground vehicles use radio links to receive motion demands from remote command stations. To facilitate this kind of remote control real-time video images and other sensor feedback is often returned to the operator from the vehicle. In applications where radio communication is not feasible it is not uncommon to see umbilical cords or reels of fibre optic carrying command data and audio visual feedback.

2.1.2 Functional Autonomy

Autonomy is an important topic when discussing unmanned vehicle systems and indicates the degree to which a function can be performed without need of operator intervention. Figure 2.1 depicts five levels of increasing autonomy for an unmanned vehicle function.

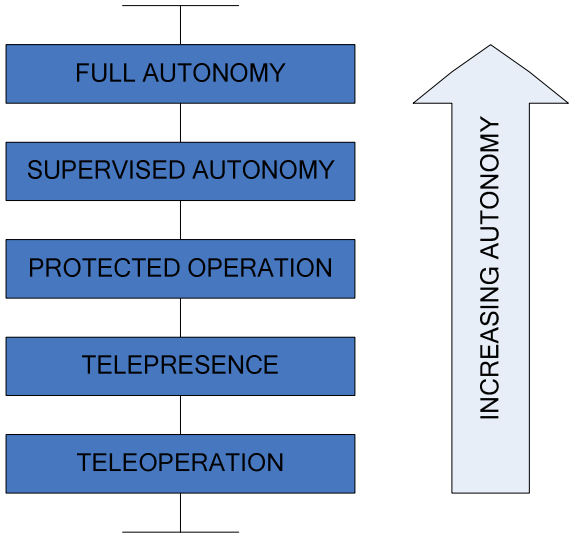

Figure 2.1 - Levels of UGV Autonomy

Teleoperation is shown as the lowest level of autonomy and so requires the greatest operator input. The unmanned vehicle operator is responsible for all low level movements of a vehicle function when teleoperating. Situational awareness for the operator is entirely dependent on what feedback is received from the vehicle or if it is within sight. A teleoperated system would typically provide video feedback from cameras onboard the unmanned vehicle to show what effect the operator input was causing. This means teleoperated vehicles are often challenging to control.

Moving to the telepresence level brings improvements in terms of the feedback for the operator. A telepresence system would include a suite of sensors fitted to the unmanned vehicle. The suite may include proprioceptive sensors to monitor the configuration of the vehicle and perceptive sensors to monitor the environment around the vehicle. At the operator console feedback from these sensors would be displayed to allow the operator to see if the vehicle had a problem or was close to an obstacle. The operator would still be responsible for low level control of the vehicle function.

Protected operation allows the vehicle to refuse motion commands that could cause part of the vehicle to collide with either another part of the vehicle or an obstacle. To facilitate this behaviour the vehicle monitors the feedback from onboard sensors, continuously checking if a vehicle component is in close proximity of anything else. These systems typically allow an override should the operator consider it safe to continue.

Supervised autonomy is the first level where the unmanned vehicle can generate and execute motion commands. The operator would issue basic task level commands such as “move to a location” or “search this area” for a vehicle with this level of autonomy. These commands would be broken down into primitive task lists by the unmanned vehicle. Appropriate motion commands would then be generated for the autonomous completion of the task. At any time

the operator may interrupt the vehicle to change the task parameters or rescue the vehicle if it is stuck. The operator essentially becomes a supervisor for the vehicle.

The final level of full autonomy is only theoretical at this time. One possible vision is for unmanned vehicles to be loaded with a mission package and left to autonomously complete the whole task. This may involve planning a route to a target avoiding any obstacles but most importantly recognising when it was stuck and finding another way to achieve the task.

The levels of autonomy discussed here are very general. For example the Remotec Revolution pictured in Figure 1.2 would meet the teleoperated level of autonomy although it does have some features of telepresence and protected operation. The research presented in this thesis aims to produce a vehicle with a navigation system at the level of supervised autonomy.

2.2 APPLICATIONS OF THE TECHNOLOGY

2.2.1 UGV applications

Many applications for unmanned ground vehicles have been identified. Example domains include search and rescue, fire fighting, nuclear decommissioning and the study of natural phenomenon on Earth or other planetary bodies. Most of these applications require the vehicle to be hardened in some respect to withstand extreme temperatures or exposure to radiation. These vehicles may also be equipped with special sensor payloads to measure the characteristics of their environment.

A robot assisted search and rescue team has been formed by the Center for Robot Assisted Search and Rescue (CRASAR) at the University of South Florida. They helped in the aftermath of the World Trade Center collapse (WTC) on September 11th 2001 [Casper 2003]. Small teleoperated vehicles were sent into the rubble to help search for survivors in places that could not be safely accessed by the rescuers.

Figure 2.2 - CRASAR robot at the site of the WTC collapse

Several robotic fire fighting platforms have been developed. For example the JCB Fire Spy was a teleoperated vehicle created for the West Yorkshire Fire Service to use in areas too dangerous for firemen to enter. Fire Spy was a mini skid loader that could be equipped with a grab to lift barrels of flammable chemicals or a fire suppressant foam cannon [Garnett 2008].

Figure 2.3 - Fire Spy

On the 26th April 1986 an explosion at the Chernobyl Nuclear Power Plant released radioactive isotopes into the atmosphere killing numerous people. At the time a shelter was hastily erected over the damaged reactor to prevent further radiation leakage. A radiation hardened inspection robot called Pioneer was later developed to address concerns about the structural integrity of the reactor shelter [Abouaf 1998]. Pioneer was sent inside the shelter controlled via an umbilical to take environmental measurements and acquire samples from the structure.

Figure 2.4 - Pioneer outside the Chernobyl Nuclear Power Plant

Volcanoes are a natural phenomenon on which surveys have been performed using unmanned ground vehicles. Dante II was used to explore the Mount Spurr Volcano in Alaska [Bares 1994]. Teleoperation was used to control Dante II for the majority of the 5 day exploration but a scheme of supervised autonomy was also trialled. Dante II was used to measure the composition of gases from several large volcanic vents.

NASA has launched a number of robotic exploration missions to the surface of Mars. Mars Pathfinder landed on the Martian surface in July 1997 and released a solar powered rover

named Sojourner [Iovine 1998]. Sojourner was teleoperated by operators on Earth. The rover had enough autonomy to prevent accidents in light of the fact that the control latency was up to 40 minutes depending on the relative positions of Earth and Mars. Stereo imaging was used for obstacle detection and the rover was also equipped with a spectrometer to determine the composition of soils. In 2003 NASA landed two more rovers Spirit and Opportunity on Mars. Through soil analysis these rovers have since provided evidence that the surface of Mars once had water [Lopez 2007b].

In the military world UGV use has become more widespread in recent years. US soldiers are adopting man portable teleoperated robots for searching buildings and confined spaces [Lopez 2007a]. Larger teleoperated vehicles equipped with weapons have also been used to flush out hostile forces. It has been suggested that the roadmap for military UGV development may take the stages of Searcher, Mule, Wingman & Hunter-Killer [NRC 2002].

Searcher type platforms are seen as essentially teleoperated vehicles deployed by infantry and these are what are being used today. The Mule is described as a small logistics vehicle carrying supplies around the battlefield. Such a vehicle would follow a route laid by a soldier or vehicle, perhaps taking the form of a sequence of electronic markers. Environmental sensing and autonomous local path planning would be required by the Mule but only to avoid obstacles on a route that had already been traversed. As part of the research presented in this thesis a system was developed that fulfils the requirements of the Mule stage and more.

The remaining stages define autonomous vehicles that have highly effective sensor systems. A Wingman is described as a vehicle requiring minimal supervision and is capable of moving in formation with a leader, performing reconnaissance tasks and launching non-lethal countermeasures if attacked. Finally the Hunter-Killer would be a platform carrying lethal weapons that could be programmed with rules of engagement for enemy encounters.

2.2.2 Applications of this Research

Of direct relevance to this research are applications that involve significant navigation challenges or require thorough coverage of an area. Future UGV applications shall require only basic route specification by an operator. Once specified the vehicle will autonomously follow the route detecting and negotiating any obstacles that are encountered.

Potential military UGV roles could include route proving, reconnaissance, perimeter surveillance & logistics convoy guidance. With each of these tasks it could be expected that a map of the area existed to allow the UGV operator to specify a basic route. These roles also offer the potential for a single operator to supervise multiple unmanned vehicles if each exhibits sufficient autonomy. With multiple vehicles precise route information could be relayed from lead vehicles to those following to supplement the basic operator route specification.

While much of the funding for UGV programs is coming from military sources the potential civilian applications should not be overlooked. Agriculture is a prime candidate for the use of unmanned vehicles. In fact GPS based guidance and data collection systems are already being integrated with new farm machinery.

The idea is that the location information can be used to increase crop yields. GPS position data is used throughout the crop cycle. First to ensure the seed is sown at regular spacing by indicating which direction the farmer should the steer the vehicle. Then as different processes are applied to the sown crop the GPS ensures the machinery is always guided along the same path. Finally when the crop is harvested data can be collected about the yield from every area of the field. This yield data allows the farmer to better target the use of fertilizers in the next year.

The step to unmanned agricultural vehicles is thus not large. In fact automated crop harvesting at speeds comparable with a human operator has been demonstrated using a GPS guided harvester [Pilarski 1999]. An autonomous mowing system has also been developed that is capable of completely covering a specified area of smooth terrain with a low obstacle density such as a golf course or sports field [Batavia 2002]. Of course inexpensive autonomous mowers are currently available for domestic use but these typically follow a haphazard path turning each time they sense a boundary wire that encloses the lawn.

Other applications of route proving behaviours could include planetary rovers or intra-company delivery vehicles. The geographical coverage techniques could be used in commercial cleaning devices or for construction tasks such as ground surveying.

2.3 UGV CONTROL ARCHITECTURES

At the heart of every autonomous UGV is a control architecture that forms the link between sensing and motion. Typically the UGV task is broken down into manageable pieces, each handled by different components of the control architecture. A number of architectural paradigms exist presenting a whole spectrum of ways to break down the UGV task. At opposite ends of this spectrum are deliberative and reactive architectures.

2.3.1 Deliberative Architecture

Figure 2.5 illustrates a simple deliberative architecture. The UGV task is split into five processes that operate in series. The perception module would gather data from the UGV sensors about the proximity of obstacles and the location of the vehicle. Next the modelling module would use this new data to update a world model that contains all of the things of interest to the UGV. A planning module would generate a high level plan for how the UGV can complete the task given the current world model. Task execution breaks the overall plan

into a list of primitive moves that the UGV should make. Finally the motor control module provides appropriate low level control of the UGV motors.

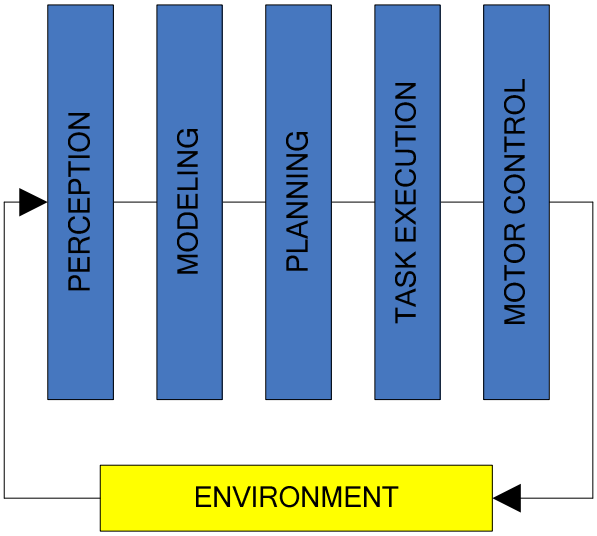

Figure 2.5 - Deliberative Architecture

A key feature of the deliberative architecture is the sequential nature of execution. All stages must run in order with each using the output of the stage before. In this way the architecture senses, models, plans then acts. Speed of execution is critical if this architecture is used in a dynamic environment. If the total time taken to perform each stage of the architecture is too large the environment may have changed. As such the executed action may no longer be appropriate.

Another characteristic of the deliberative architecture is that a world model is maintained. This typically includes any prior knowledge of the environment such as a geographical map. Goal locations for the UGV would be overlaid on this map. As the UGV interacts with the environment the world model must be updated. Any detected obstacles would be recorded in the model and the location of the vehicle would be tracked.

Storage of a world model allows for deliberation over alternative courses of action. The likely consequences of each action can be derived from the model. Optimal UGV behaviour can only be achieved if the world model is complete. As new obstacles are detected and added to the world model then the current plan would need to be revised to circumnavigate these new obstacles.

The number of possible courses of action to take the UGV to the goal will be related to the number of steps in the task. Assuming a common set of actions can be performed at each step this relationship would be exponential in nature. There are numerous ways to reduce this action search space at the cost of optimality. One could be to only consider a few steps and use a metric such as distance to the goal to score alternatives.

The DARPA Autonomous Land Vehicle (ALV) is an example of an early UGV program developed for both on-road and off-road use. The control architecture employed during demonstrations of the on-road obstacle avoidance capability in 1986 followed the deliberative paradigm [Dunlay 1987].

Within ALV a sensors subsystem acquired data from a video camera and laser scanner. Scene modelling was performed by the perception subsystem using the latest data from the sensors subsystem. The reasoning subsystem received goals from the human operator and created a plan to accomplish those goals. Then using the model created by the perception subsystem the reasoning subsystem would generate a trajectory to follow the road while avoiding detected obstacles. Finally the control subsystem would move the vehicle along the generated trajectory.

2.3.2 Reactive Architecture

An example reactive control architecture for a simple mobile robot is illustrated in Figure 2.6. Here the robot task is split into a number of behaviours that operate in parallel. At the lowest level an avoid obstacles behaviour prevents collision by turning the robot away from detected obstacles. The next level of competency makes the robot wander at random around the environment. A third level makes the robot follow any paths that are detected. The highest level guides the robot to a goal location.

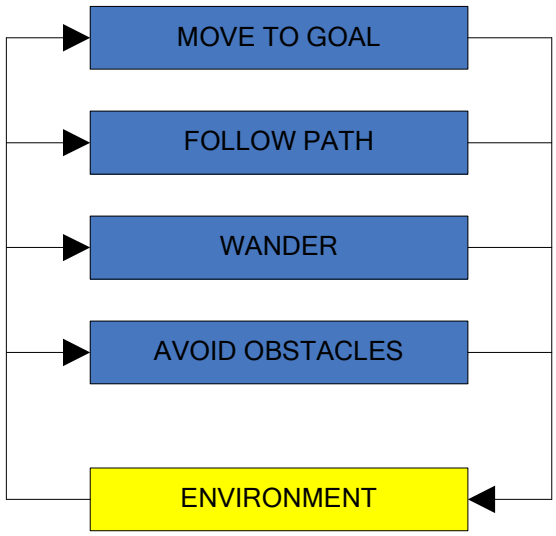

Figure 2.6 - Example Reactive Architecture

The best known reactive control architecture is the Subsumption Architecture [Brooks 1986]. Following this idea behavioural modules are arranged in a hierarchy of architectural layers. Lower layers of the architecture control the most basic robot behaviours. Higher layers are added in a way to include the competence of lower layers. If appropriate the lower layers of the architecture can provide input to the higher layers. In turn the higher layers can suppress the output of lower layers to avoid conflicts.

Another well known style of reactive control architecture is based on the concept of artificial potential fields [Khatib 1986]. Arkin developed a schema based architecture to guide the motion of a robot using potential fields [Arkin 1998]. Following this approach each behavioural layer of the reactive control architecture would be represented by perceptual schema and motor schema.

The role of the perceptual schema was to process sensor data and provide the necessary environmental data for the behaviour. Using the environmental data the motor schema then produced an action vector as output. As an example an avoid obstacle motor schema would simulate a repulsive artificial potential field around any obstacles that had been detected. This would result in an action vector leading away from the obstacle. Vector summation was used to combine the action vectors from different motor schema and produce the overall emergent robot behaviour.

Reactive control architectures allow a real time response to new sensor data. A simple transformation from sensing to motion control is provided by the behaviours. In this way the architecture provides a fixed predetermined response to a given situation. The architecture as a whole avoids maintaining a detailed model of the world and so cannot generate an overall plan to achieve a goal. Instead individual behaviours build instantaneous, behaviour specific world representations as required. Behaviours are constructed to achieve goals by reflex and have limited memory.

If multiple behaviours are triggered by the latest perception data then outputs are combined. For some reactive control architectures a hierarchy or arbiter ensures the output from the highest priority behaviour wins through. Other architectures explicitly combine outputs by summation allowing multiple goals to be pursued simultaneously. The overall behaviour emerges from the combined outputs of the architectural layers.

As part of the DARPA DEMO II UGV program a reactive control architecture was used to demonstrate formation movement of multiple unmanned HMMWV (High Mobility Multipurpose Wheeled Vehicle) platforms [Arkin 1998]. A perceptual schema was used to continuously monitor the relative positions of the vehicles. Motor schemas were then used to move the vehicles into formation and ensure that the formation was maintained. The system was capable of controlling leader referenced formations in patterns including a line, column, wedge and diamond.

2.3.3 Hybrid Architecture

A hybrid of the deliberative and reactive control architectures can offer the advantages of both paradigms. With low level control performed by a reactive system a UGV could respond rapidly to new sensor data. Higher level tasks could be governed by a deliberative system running in the background and managing the overall behaviour of the UGV. Figure 2.7 illustrates this type of architecture.

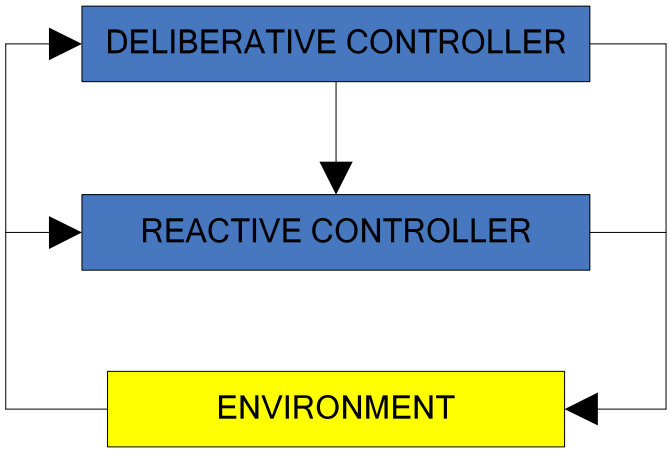

Figure 2.7 - Hybrid Deliberative / Reactive Control Architecture

A representative low level task is steering a UGV. The deliberative portion of the control architecture works out a direction for the UGV to travel. Initially turning the UGV to face the appropriate direction and maintaining the heading while in motion could be performed by

reactive control. The reactive controller needs only to know the present and desired heading of the UGV. Present heading would be available from a UGV sensor and desired heading from the deliberative controller. The reactive controller would continuously monitor these two values aiming to minimise the error between the two.

The role of the deliberative controller is much more complex and involves the use of a world model. Periodically the controller would consult the world model and search for the best path to a goal location. The output of the deliberative controller may take the form of a continuous curved trajectory from the UGV position to the goal. This trajectory could be followed by the reactive controller.

Within a hybrid architecture the reactive controller is best suited to performing a simple fast acting continuous role. The response to each possible situation is hard coded. The deliberative controller performs the more complex elements of the UGV task initiating new actions only periodically. A flexible response can be expected from the deliberative controller finding the best solution to the task given the available world model.

There can be communication between the two controllers within a hybrid control architecture with the deliberative controller providing input to the reactive controller or vice versa. In the example above the deliberative controller guides the reactive controller. Other hybrid architectures use different processes to influence the reactive controller. For example the deliberative controller configured the reactive controller within AuRA the Autonomous Robot Architecture [Arkin 1998]. AuRA selected which behaviours in a schema based reactive controller were active and how their output should be combined.

Xavier an office delivery robot developed at Carnegie Mellon University (CMU) is an example of an extensively used robot that featured a hybrid control architecture [Simmons 1997]. The

Xavier architecture had layers for obstacle avoidance, navigation, path planning and task planning. Each successive layer had a more abstract representation of the world and provided input to the adjacent layers.

Task planning was deliberative and integrated asynchronous delivery requests into the current plan. To increase efficiency pickup and drop off actions for different tasks were interleaved. The path planning layer determined the best route to follow between locations. Routes were not chosen according to the shortest distance but the shortest expected travel time. In fact route choice took into account the chances of each possible path being blocked or the robot missing a critical turn and having to backtrack.

Xavier's navigation layer calculated the best heading for the robot as it attempted to follow the path provided by the path planner. It used a probabilistic method known as Markov Localisation to estimate the current robot position based on sensor readings. The obstacle avoidance layer guided the robot trying to follow the heading set by the navigation layer. When confronted by obstacles the robot would attempt to steer around them and continue along the set heading.

2.3.4 Blackboard Systems

The analogy of a blackboard system is a group of experts working together on a problem [Corkill 1991]. The group shares ideas and each contributes their knowledge to a solution being drawn up on a blackboard. Each expert speaks when the appropriate stage of the solution is written on the blackboard. In this way the problem is solved in an incremental fashion.

In terms of the other architectures described above the concept of a blackboard system is more a style of implementation rather than an architectural paradigm. It would be possible to

implement a deliberative or hybrid control architecture as a blackboard system. As purely reactive control architectures do not use world representations a blackboard would be unnecessary.

Figure 2.8 illustrates an example UGV control architecture implemented as a blackboard system. The blackboard holds information about the geographical position of the UGV and where it is in relation to a goal location. Updates to the blackboard come from a localisation system and route planner. The output from the blackboard is used by a motion controller.

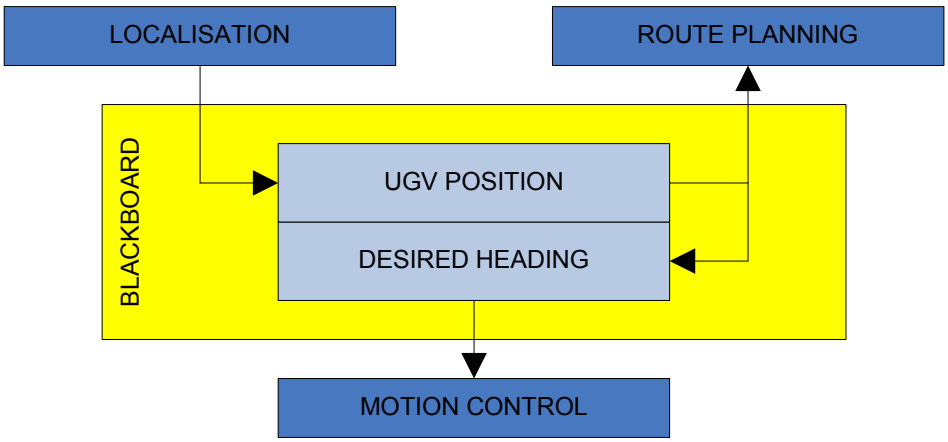

Figure 2.8 - Example Blackboard System

In the example the localisation system maintains a record of the current UGV position on the blackboard. Using the latest position record the route planner periodically generates a desired heading for the UGV. The motion controller then drives the UGV along this desired heading.

An architectural blackboard provides a flexible shared memory area for a group of processing systems. The processing systems can be thought of as being like the experts in the analogy. Processing of blackboard data is incremental and requires input from multiple processing systems. A set of triggers is required to initiate the execution of each processing system.

Appropriate triggers may be events like the arrival of new data or a system completing the processing of some data.

The blackboard concept is very useful for rapid initial prototyping of a control architecture. It allows processing systems to be coordinated in a way that is not dependent on the execution order. As the architecture evolves processing systems can easily be changed as can the order of execution. In this way each processing system could be thought of as a black box.

Blackboards can be a double edged sword. A disadvantage of the flexibility is that the system needs to be carefully specified. It would also be very easy for a blackboard implementation to grow out of hand and become unmaintainable. Access control of the blackboard is another source of problems. With multiple processing systems using the same data it must be ensured that data is ready to be used when each system is triggered.

The Ground Surveillance Robot (GSR) is an example of a UGV based on a blackboard system [Harmon 1987]. GSR was an armoured personnel carrier modified for automated control. The vehicle used six networked computers as processing subsystems. An area of shared memory was put aside on each computer to simulate the blackboard. Individual subsystem roles included controlling vehicle locomotion, navigation, path planning, mapping local obstacles, modelling distant terrain & response to a human interface.

Unfortunately the GSR program was concluded before the control architecture was completely implemented. The final state of the vehicle was capable of obstacle avoidance using a number of ultrasonic range finders. Complexity of the blackboard system was cited as one of the stumbling blocks for the project.

2.4 UGV PERCEPTION EQUIPMENT

Autonomous vehicle navigation in natural terrain poses additional problems over the indoor autonomous navigation task. Indoor navigation is often based on the implicit assumption that the vehicle will operate in a world with a flat ground plane. Operating such a vehicle on uneven ground could lead to strange behaviour. If for instance the vehicle drove into a rut then sensors may detect the ground as an obstacle and the corresponding obstacle avoidance behaviour would cause the vehicle to avoid a phantom obstruction.

Successful autonomous navigation in natural terrain requires detection of the ground and discrimination of a traversable path from the various positive and negative ground features. A 3D model of the terrain must be constructed as the UGV is in motion. This implies that sensors have to perceive points in the terrain that are then converted to 3D coordinates relative to the UGV.

Many technologies have been applied to the task of perception for robotic vehicles. Simple range measuring devices can be systematically swept along the ground in front of the UGV thus generating a 3D model using the motion of the vehicle. Alternatively a device that directly produces 3D range data could be used.

Typical sensors measure range based on the propagation of an acoustic or electromagnetic wave. Range to an obstacle is inferred using techniques such as triangulation, time of flight or reflected signal strength. The following sections describe some of the available sensor technologies.

2.4.1 Ultrasonic Sensors

Ultrasonic transducers are popular sensors for mobile robot applications due to their low cost. They are also commonly found embedded within the rear bumper of cars to warn the driver of

close obstacles when reversing. Figure 2.9 shows a Polaroid ultrasonic transducer and the associated electronic module used to calculate obstacle range.

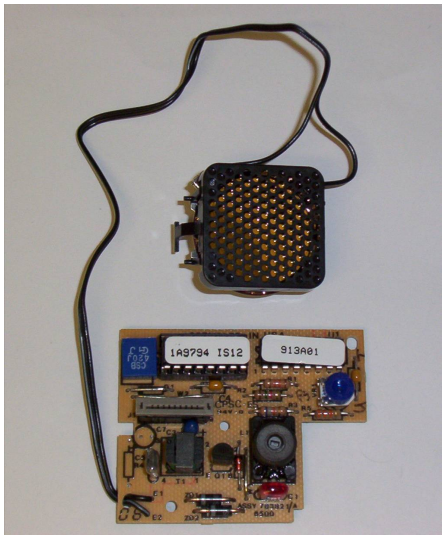

Figure 2.9 - Polaroid Ultrasonic Transducer and Ranging Module

These devices periodically emit an acoustic pulse in the ultrasonic frequency range (~50 kHz). If the pulse reflects off an obstacle in the environment and returns to the transducer the obstacle proximity can be calculated. The speed of sound is known so the distance travelled by the pulse can be calculated from the time between the emission and receipt of the reflected pulse. This is known as time of flight ranging.

A typical ultrasonic ranging module such as the Polaroid 6500 shown in Figure 2.9 can measure distances from a few centimetres up to about 10m. Distance measurements can be made with an accuracy of around 10cm. However precise discrimination of the location of a reflecting obstacle is difficult. This is because ultrasonic transducers tend to have a wide field of view. The Polaroid 7000 transducer pictured above has an effective 17° field of view.

2.4.2 Infrared Sensors

Of equal providence in the mobile robot world is the infrared sensor. Basic sensors observe the strength of signal reflected back to a phototransistor from an infrared LED. Receipt of a reflected signal indicates the close proximity of an obstacle and signal strength can be used as a crude judge of distance. Higher quality infrared sensors and some laser range finders use triangulation to determine the distance to the reflecting surface. The Sharp GP2D12 sensor shown in Figure 2.10 is an example of an infrared sensor that uses triangulation.

Figure 2.10 - Sharp GP2D12 Infrared Sensor

On one side of an infrared triangulation sensor is an infrared LED. This emits a narrow beam of infrared light (~850nm wavelength) in the forward direction. On the other side of the sensor is a linear CCD array. This array consists of a row of pixels each capable of measuring the intensity of infrared light that falls within their area. Infrared light from the LED can be reflected back from a distant surface onto the CCD array. Figure 2.11 demonstrates that the nearer the reflecting surface the shallower the angle that the light strikes the CCD array.

Within the infrared sensor pictured above the incident of reflected infrared on the CCD array is processed electronically to determine the distance to the reflecting surface. The separation between the reflected light striking the CCD array and the centre line of the corresponding lens is inversely proportional to the distance between the sensor and the reflective surface.

Mathematically the distance to the reflecting surface can be calculated if the separation of the infrared emitter and receiver is known along with the focal length of the lens in front of the receiver.

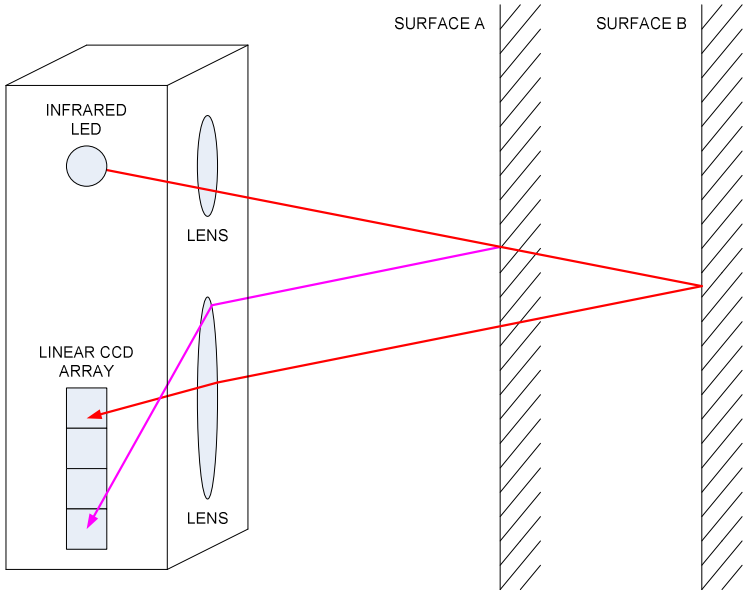

Figure 2.11 - Infrared Range Measurement Using Triangulation

Neither ultrasonic nor infrared sensors are particularly suitable for sweeping across terrain to produce a high resolution model. Although the beam of the Sharp infrared sensor is much narrower at 0.8m from the sensor the beam width is around 100mm. The range of these sensors is also limited to around 2m. Greater precision and range was required to produce terrain models for the UGV in this research.

2.4.3 Laser Range Finders

Many types of laser range finder also emit light from the infrared part of the electromagnetic spectrum. These devices use a more collimated beam of light to determine distance than an infrared sensor. A common technique for laser range finding is to use time of flight similar to ultrasonic sensors. As light travels much faster than sound more accurate timing is required.

There are several techniques used for measuring laser time of flight. The first is to pulse the laser and monitor the interval before the pulse is returned to a detector. An alternative is to use a continuous modulated beam. Modulating the amplitude or frequency of the electromagnetic wave adds a non-uniform pattern that can be detected. By comparing the phase difference between the transmitted and received wave modulation the time of flight can be determined. The different waveforms used in these methods are illustrated in Figure 2.12.

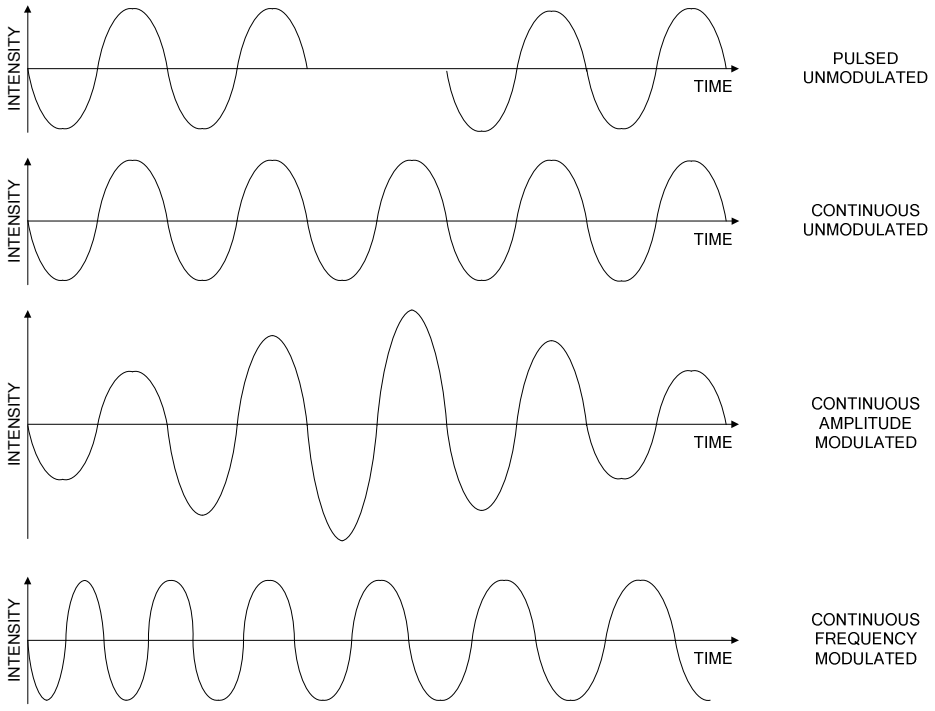

Figure 2.12 - Pulsed, Continuous and Modulated Waveforms

Scanning laser range finders are often used on robotic vehicles to build an instantaneous 2D map of the environment. One realisation uses a rotating mirror to sweep the laser beam as illustrated in Figure 2.13. An encoder on the mirror allows location of the reflective surface to be calculated from the mirror angle and time of flight range value. 3D laser scanners are also produced using a pair of rotating mirrors to sweep the laser beam raster fashion. It is common to see scanning laser range finders referred to as LADAR (Laser Detection and Ranging).

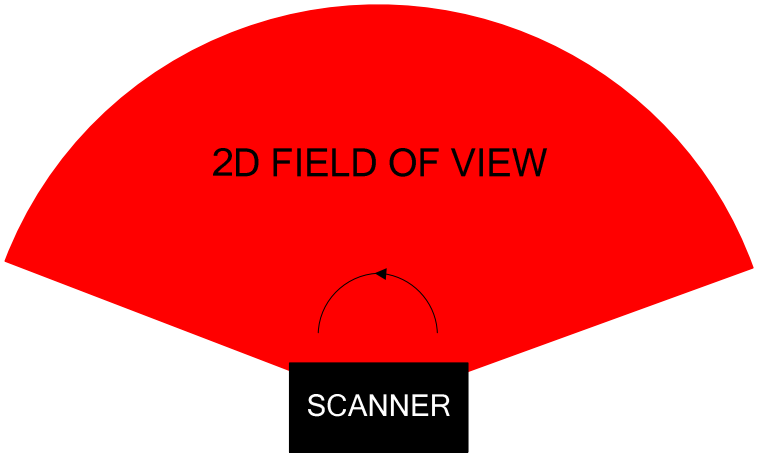

Figure 2.13 - 2D Scanning Laser Range Finder

2.4.4 3D Time of Flight Range Cameras

Alternative 3D range finding techniques are presently being developed that use infrared light but do not rely on scanning mechanisms. These devices promise to be smaller, more reliable and offer higher refresh rates than traditional scanning range finders. In the literature this type of sensor is often referred to as a Flash LADAR.

Currently available models typically feature a camera with an infrared light source alongside [Oggier 2004]. Within the camera is a 2D array of pixels that are sensitive to infrared light. The light source is used to illuminate the camera scene while the image sensor generates both intensity (greyscale) and range images.

For this kind of range camera the light source is amplitude modulated with a continuous sinusoidal wave. As the modulated light is reflected from the scene it is captured by the image sensor. The phase of the captured light signal is then correlated with the transmitted illumination signal. This allows determination of the distance between the camera and the reflecting object. The process is performed separately for each of the pixels in the image sensor to produce a complete range image.

2.4.5 Stereo Vision

Another technology that offers sufficient range and precision for terrain analysis is stereo vision. Unlike the other sensing technologies mentioned here stereo vision functions without emitting acoustic or electromagnetic waves. This has made stereo attractive for military applications where electromagnetic emissions could make the UGV an easy target. With stereo vision the 3D structure of a scene is inferred from two camera images taken from different viewpoints. Objects will appear in different positions in each image. The size of the position disparity is related to the depth of the object in the scene.

Key to the success of stereo vision is finding corresponding pixels in each camera image. This is often done by extracting and matching features such as edges from both images. Other methods include correlating the colour or intensity patterns from the two images. As such stereo vision will not work in environments that have sparse features or little contrast. Once a pair of pixels is matched the angle from the centre line of each camera to the pixel in the corresponding image can be calculated. Triangulation is then used to derive the depth of the pixel contents in the scene.

2.4.6 Sources of Error

For each of the active sensor types mentioned above flat surfaces perpendicular to the device will produce the most accurate range measurements. If an ultrasonic or infrared beam strikes a surface at an acute angle it may be reflected away from the sensor. With this kind of specular reflection the sensor would report the space containing the reflecting object as clear. It is also possible for a beam to reflect back to the sensor after one or more specular reflections providing an erroneous time of flight. Specular reflection is the biggest source of error with ultrasonic sensors and laser range finders.

Another common source of error with active sensors is the so called mixed pixel problem [Hebert 1992]. This is when a single sensor (or pixel of a sensor array) receives a reflection from two surfaces separated by a distance. Figure 2.14 illustrates the problem. The range value yielded by the sensor will be somewhere between the range of the two surfaces. This effect is often observed at the edge of objects in a scene.

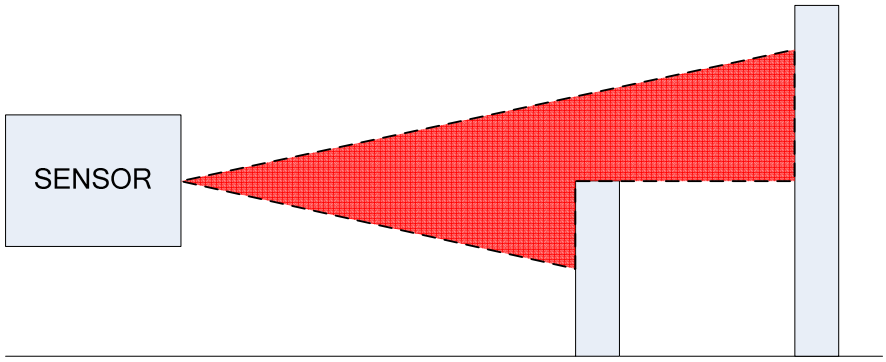

Figure 2.14 - Mixed Pixel Problem

For sensors that use continuous modulated beams or illumination the interval of ambiguity is also a potential problem. These devices measure time of flight by comparing the transmitted and reflected waves to determine the phase difference. This measure is ambiguous as it does not consider the possibility that the phase difference may be greater than one cycle of the modulating wave. So it is possible for highly reflective surfaces just beyond the non-ambiguity range to be interpreted as being close to the sensor.

2.5 UGV LOCALISATION EQUIPMENT

Localisation is an important task for an autonomous vehicle. If the vehicle does not know where it is in relation to a target location it would be impossible to navigate effectively. Localisation equipment onboard a UGV can be used to calculate the vehicle position with respect to some coordinate system and is generally grouped into the categories of absolute and relative.

Absolute localisation calculates the present UGV position with respect to a set of markers or beacons with known location. GPS is the most ubiquitous form of absolute localisation today and uses a network of satellites as beacons. Relative localisation calculates the present UGV position with respect to the UGV position at some point in the past. Examples of relative localisation equipment include inertial navigation systems and wheel odometry.

An objective for this research was to create a UGV localisation system that was GPS based but had the ability to maintain an accurate sense of the UGV location if a GPS solution was not available. The decision was made that any auxiliary equipment should provide relative localisation. This was because the placement of markers or beacons in the environment was not feasible.

2.5.1 GPS

The NAVSTAR Global Positioning System (GPS) is a satellite based position multilateration service [Jong 2002]. This US system uses a constellation of satellites orbiting Earth as depicted in Figure 2.15. The satellites transmit radio signals that can be used to calculate the position of a compatible receiver in three dimensions. GPS can be used anywhere in the world and operates continuously. Global coverage is achieved with four satellites in each of six orbital planes inclined at 55° to the equator.

GPS satellites continuously transmit a pseudo random code modulated within a carrier wave. Earth based GPS receivers monitor the code received from each satellite and compare it with their own internally generated representation of the code. The synchronisation difference between the received and internally generated versions of the code is found by correlation. This gives an estimate of the time of flight for the signal and so distance from the satellite to receiver.

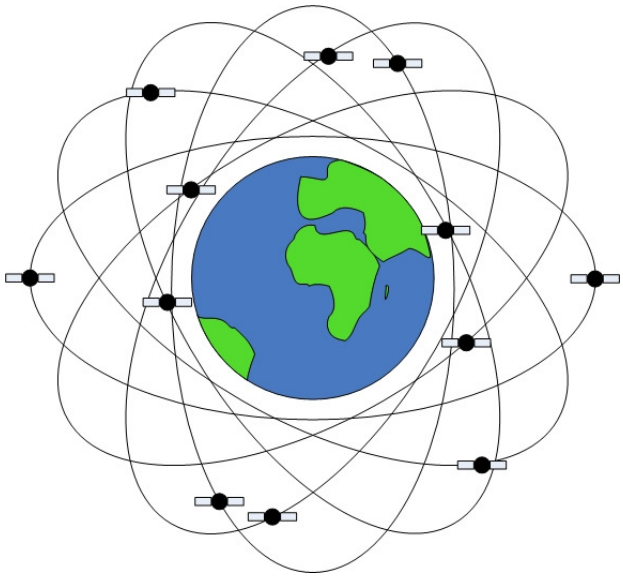

Figure 2.15 - GPS Satellite Constellation

Superimposed in the pseudo random code each GPS satellite periodically transmits data encoding the orbital path of all of the satellites within the constellation [Borenstein 1996]. From this information the receiver derives the position of each satellite. Then using the estimated distance to each observed satellite the receiver position can be established. Simultaneous observation of a least four satellites is required to determine the receiver position in three dimensions [Planas 2004].

Three satellites are enough to calculate a position in two dimensions (longitude and latitude). A fourth is required for an accurate height fix. This is because the accuracy of the clock within a typical GPS receiver is much lower than the atomic clocks used onboard the satellites. Observation of a fourth satellite allows the receiver to establish the relationship between the satellite and receiver clocks. Knowing this relationship the receiver can hone the satellite distance estimates and derive a three dimensional position solution. Good quality receivers monitor up to 12 satellites to further refine the accuracy of the calculated position.

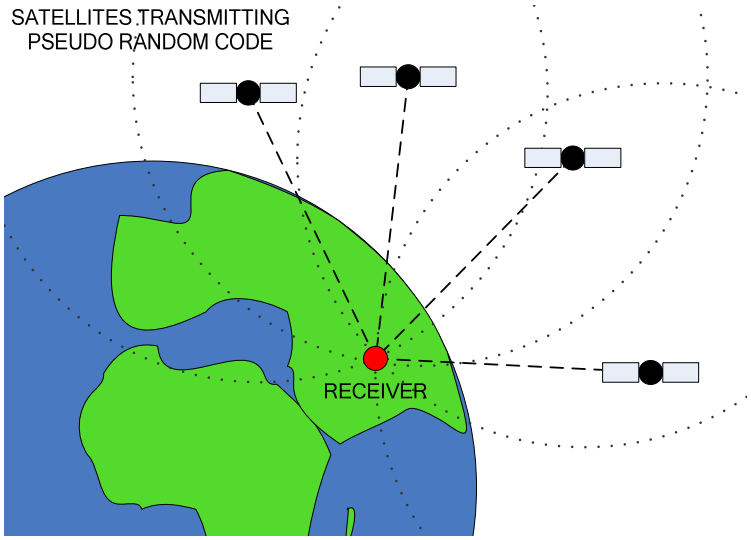

Figure 2.16 - GPS in Operation

When a radio transmitter and receiver are moving with respect to each other the frequency of the received signal will differ from the transmitted signal. This observation is known as the Doppler Effect. Sophisticated GPS receivers provide even higher accuracy by monitoring the frequency shift of the carrier wave from each satellite [Jong 2002]. Within the receiver the frequency of each received carrier wave is compared against an internally generated version of the transmitted wave. For each carrier an expected rate of change of the satellite to receiver distance can be calculated from the observed frequency shift. This can then be used to smooth the position of the receiver as calculated from the pseudo random code by monitoring the changing distance over short periods of time.

Calculated receiver positions are always subject to error. Horizontal position accuracy is measured in terms of a radius from the receiver and a probability of the reported position falling within that radius. For a basic GPS receiver if you drew a circle of radius 20m around the position of the receiver then that receiver would report a position within the circle 95% of the time [ESA 2008a]. However around 50% of the time the receiver would report a position in a circle of 10m radius from the receiver.

A number of error sources affect the accuracy of GPS. Direct line of sight is required between satellite and receiver. This means that GPS does not work indoors. Furthermore GPS use outdoors is affected by tree cover and structures that occlude satellites. Error is also introduced through atmospheric refraction, multipath reflection, satellite clock and orbital path discrepancies [Borenstein 1996].

As the US Government operates GPS they have control over the accuracy and availability of the position information. A more accurate version of GPS is available to approved military customers. Civilian GPS can also be deliberately degraded by the insertion of random errors in the codes transmitted by the satellites. This degradation is known as Selective Availability and reduces the horizontal position accuracy to 100m 95% of the time [CSI 2005]. Selective Availability has not been used since May 2000.

2.5.2 Differential GPS

Raw GPS is good enough for people to fix their position using a map but does not have sufficient accuracy for UGV navigation. Differential GPS (DGPS) is a technique that can be used to reduce the effects of the GPS error sources mentioned above. The system relies on the use of two or more GPS receivers, one that is mobile and others that are static and in known positions. Static receivers calculate the offset between the known and GPS calculated position then transmit appropriate corrections to the mobile receiver. Figure 2.17 shows the idea.

Use of DGPS requires either purchase of a site surveyed static GPS receiver or subscription to a commercial Differential GPS service such as OmniSTAR. OmniSTAR has a network of GPS receivers around the world that broadcast differential corrections. The increase in accuracy is dependent on the distance of the mobile receiver from the static receiver. However horizontal position accuracy of better than 1m 95% of the time can be achieved. This is sufficient accuracy for UGV navigation.

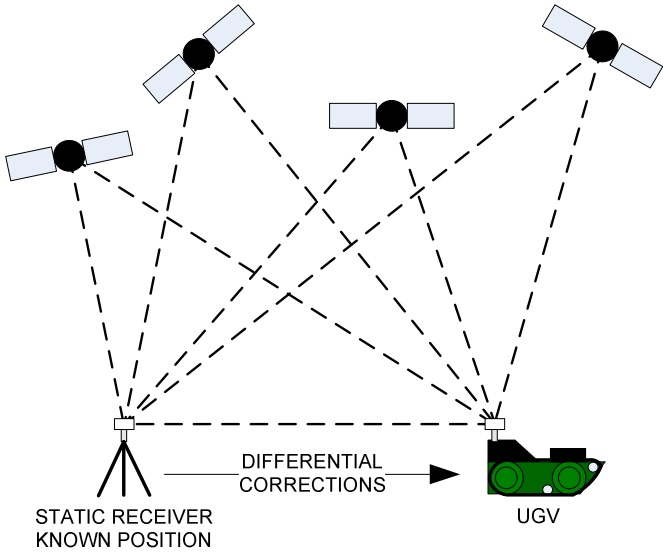

Figure 2.17 - DGPS Concept

A more economic way to achieve this order of position accuracy is to use a GPS receiver that is Space Based Augmentation Service (SBAS) compatible. EGNOS is the SBAS for Europe. Other similar systems are being developed in the USA, Japan and India.

2.5.3 EGNOS

A new service created by the European Space Agency (ESA) was used in this research. The European Geostationary Navigation Overlay Service (EGNOS) uses a system of geostationary satellites and ground stations to provide corrections to GPS information [ESA 2008b]. A compatible GPS receiver will augment the raw GPS solution with EGNOS corrections to produce more accurate position estimates similar to Differential GPS.

EGNOS transmits GPS style signals from three satellites. Encoded within the signal is differential correction data from a network of monitoring stations. This provides information about local signal disturbances within the atmosphere and independent verification of the orbital paths followed by the GPS satellites.

The ESA reports horizontal position accuracy for EGNOS augmented GPS of better than 3m 95% of the time. EGNOS is free to use but requires a compatible receiver. The majority of new GPS receivers in the UK are EGNOS compatible.

2.5.4 Other Absolute Localisation Systems

Since the advent of GPS the use of other forms of absolute localisation has declined for outdoor applications. The relatively low cost and simplicity of the GPS equipment has made it ubiquitous.

Historically people have used natural landmarks as localisation aids. Seeing familiar geographic features reassures travellers that they are headed in the correct direction and signals the progress they have made with their journey. Artificial landmarks such as signposts or markers perform the same role in helping people localise themselves and navigate from A to B in unfamiliar environments.

One class of unmanned ground vehicle that can use artificial landmarks to navigate are Automated Guided Vehicles (AGVs). These vehicles are typically found performing logistics roles in factories such as the movement of components between a warehouse and production line. Many AGV systems use path following techniques to navigate [Bostelman 2001]. For these vehicles a reflective or magnetic path is integrated with the factory floor and the AGV uses proximity sensing to track the path. AGV movement is thus restricted to the path.

Position triangulation using a vehicle mounted scanning laser is another common localisation technique for an AGV. Laser reflective markers are placed in known locations around the AGV environment. The AGV measures the angles from the vehicle heading to each marker. Position and orientation of the AGV can then be calculated using a variety of algorithms [Hu 2000].

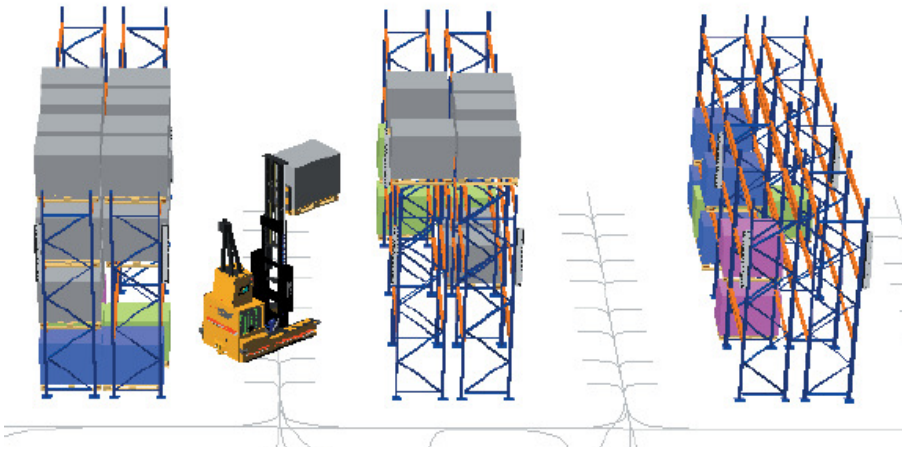

Figure 2.18 - Path following AGV operating in warehouse

Artificial Landmarks used in this way are often referred to as passive beacon systems. Passive beacon systems normally have a network of reflectors around the perimeter of an area where a mobile transceiver is used. By contrast an active beacon system would have a network of transmitters and a mobile receiver. GPS is an example of an active beacon system.

Another active beacon system that has been used for UGV localisation outdoors is the MTI CONAC. The system uses a spinning laser emitter that encodes the angle of the emitter within the beam. Receivers report the angle of the emitter whenever the beam is swept past. CONAC was used for the localisation of a cooperative team of low cost landmine clearing robots [Pook 1998]. For that project two receivers were mounted onboard each robot to allow robot position and orientation to be calculated.

There are numerous examples of similar robot localisations systems using different technologies. A radio and ultrasonic trilateration system was developed at the Carnegie Mellon University Robotics Institute for multi-robot localisation [Navarro-Serment 1999]. Visual landmark triangulation has been demonstrated in a number of research programs [Krotkov 1989, Madsen 1998]. The development of smart floors containing a network of RFID tags for robot localisation is ongoing [Dae-Sung 2005].

An entirely different approach to the localisation problem is to observe the whole operational environment. This can be done by positioning a camera above the work area of the vehicle. Localisation is then achieved by detecting the vehicle within the camera image. A demonstration of this technique has been developed for the demining task [Colon 2002]. In this project a coloured ball was mounted on a UGV that was tracked by a remote camera. A hue tracker was used to follow the ball in the camera image with the camera panning and tilting to stay centred on the ball.

These alternative absolute localisation methods all require equipment to be set up around the operational area of the vehicle. This localisation equipment would need to be surveyed so the relative positions of transmitters and receivers were known. None of these systems are as flexible as GPS although greater accuracy could potentially be achieved for small coverage areas.

Map matching is an absolute localisation technique that can be used in known or partially known environments without the need for auxiliary equipment. This method is most analogous to the way people navigate using natural landmarks. Initially the robotic vehicle is given a map of the operating environment. This map may be in the form of a CAD model or generated by the robot observing the environment from known locations. As the robot subsequently moves through the environment it compares proximity sensor data with map dimensions to narrow down the possible set of locations the robot could occupy.

A popular method for map matching is called Markov Localisation. This is a Bayesian approach to estimating the probability that the robot is at a particular location. A probability distribution is generated for a grid of possible robot locations. The probability of the robot being in a particular grid cell is based on the current proximity sensor feedback and the chance that robot arrived at that location since the last sensor feedback [Burgard 1996].

2.5.5 Relative Localisation Systems

There is a wide variety of equipment for the relative localisation of a UGV. Wheel rotation can be measured to estimate how far and how fast the UGV has travelled. Pose sensing can be used to supplement this by providing UGV heading information. Inertial sensors may be used to estimate errors in other sensor readings by monitoring the acceleration of the UGV chassis.

Wheel encoders offer an inexpensive means to measure rotation of the UGV drive axles. These devices typically measure the angular position or rotational speed of the encoder shaft. If encoders are fitted to the drive train on either side of a vehicle then dead reckoning can be used to calculate vehicle position. The average encoder change gives a measure of distance travelled. Relative changes between left hand and right hand encoders can be used to monitor changes in robot orientation.

Encoder measurements are subject to error and so a position calculated by dead reckoning will typically diverge from the true position as the vehicle moves. Error is induced from systematic and non-systematic sources [Borenstein 1994]. If the drive trains on either side of the vehicle are not perfectly symmetric this will be a source of systematic error. Non-systematic error sources include wheel slippage or the extra rotation of a wheel needed to overcome terrain irregularities.

Use of an electronic compass to monitor vehicle heading can reduce the non-systematic error caused by wheel-slippage. Compasses measure heading with respect to the Earth's magnetic field. The Earth is roughly a dipole magnet with a field originating near the South Pole and terminating near the North Pole. A component of this field is parallel to the Earth's surface and always points to magnetic north. The British Geographical Survey estimate magnetic north as 3.72° west of geographical north for the test area used in this research [BGS 2008].

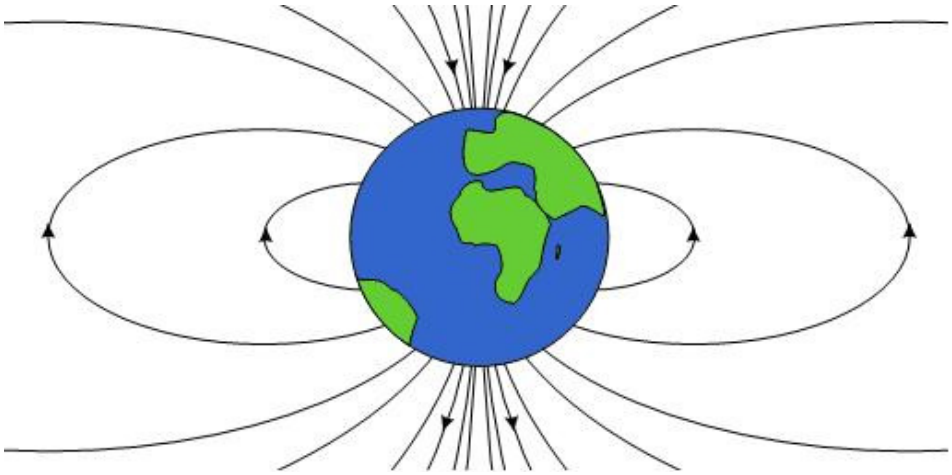

Figure 2.19 - Earth's Magnetic Field

Within an electronic compass are at least two orthogonally mounted magnetic sensors [Caruso 1999]. These sensors measure the Earth's Magnetic field along the X and Y axes of the compass. A third sensor is often used to monitor the Z axis. Output of these directional sensors is proportional to the magnetic field strength. When the X axis is facing magnetic north the X sensor will see the full horizontal component of the field and the Y sensor should see no field. As the compass is rotated the magnetic field strength seen by the sensors will follow a sine wave. Figure 2.20 shows the idea.

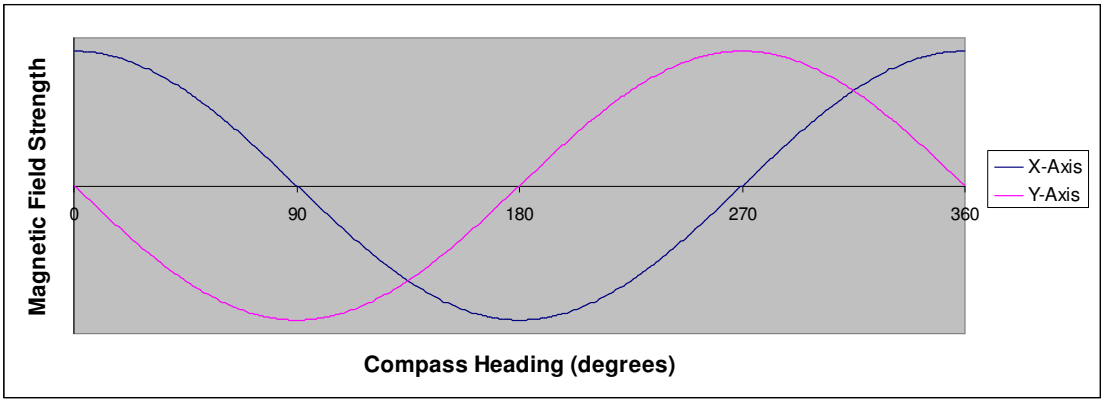

Figure 2.20 - Graph of observed field strength as compass is rotated

Given measurements of the magnetic field strength from the X and Y axis sensors the compass heading can be calculated using the following formula:

$$
\begin{aligned}
Heading &= 0^\circ \ (Y = 0, \ X > 0) \\
&= 180^\circ \ (Y = 0, \ X < 0) \\
&= 90^\circ - \left(\arctan\left(\frac{X}{Y}\right) * \frac{180^\circ}{\pi} \right) \ (Y > 0) \\
&= 270^\circ - \left(\arctan\left(\frac{X}{Y}\right) * \frac{180^\circ}{\pi} \right) \ (Y < 0)
\end{aligned}
$$

As the Earth's magnetic field dips towards the planetary poles there is also a vertical component to the field. Gradually rolling a compass from a horizontal pose will deflect the heading value [Caruso 1999]. This deflection is due to the vertical field component. In terms of the graphs shown in Figure 2.20 rolling the compass affects the relative amplitude of the sine waves.

To allow compass roll to be detected and corrected for while the vehicle is in motion a means of level detection is required. Integration of an inclinometer with a three-axis compass will allow this to be done. A bonus of using an inclinometer is that it can be used to monitor the slope of the terrain and show when the UGV is tilted to a dangerous angle.

Compasses are sensitive to other sources of magnetic field [Ojeda 2000]. This implies they may be also deflected if positioned close to ferrous metal or exposed to the electromagnetic field from say an electric motor. Most electronic compasses can be calibrated to compensate for constant strength magnetic fields but not variable fields. Thus care must be taken when

mounting the compass on a UGV. Also it should be expected that the compass will deflect if the vehicle is driven into area with a strong magnetic field.

Gyroscopes can be used to estimate heading relative to a starting pose allowing the rotation of a UGV to be tracked over time. Electronic gyroscopes measure angular rotation and often output signals encoding angular velocity. By monitoring the yaw of the UGV with a gyroscope any dead reckoning errors or electronic compass deflections can be detected. Gyroscopes themselves have well known error characteristics [Woodman 2007]. They typically suffer from drift over time. This means that even if a UGV is stationary an onboard gyroscope may report a small angular velocity. Other pose sensors are required to periodically reinitialise a gyroscope based heading estimate.

Another popular inertial sensor is the accelerometer. Accelerometers can be used to monitor UGV acceleration and provide a relative estimate of UGV position. When used in conjunction with wheel encoders they can be used to isolate dead reckoning errors caused by wheel slip and terrain irregularities. Purely accelerometer based position estimates are known to exhibit extensive drift [Borenstein 1996].

All of the vehicles that completed the 2005 DARPA Grand Challenge were equipped with an Inertial Navigation System (INS). These devices typically combine a number of gyroscopes and accelerometers to measure platform rotation and acceleration in three dimensions. It is not uncommon to see differential GPS integrated with an INS and wheel speed sensing [Braid 2006]. This combination allows highly accurate measurement of UGV position, velocity and orientation. Also errors in the measurements from one type of sensor can be detected and corrected using measurements from the other sensors. An INS was not used in this research because the cost was prohibitive.

3 LITERATURE REVIEW

Unmanned ground vehicles are complex systems. In order for a UGV to travel autonomously through the world it must have a means of localisation and methods to detect and avoid obstacles. Obstacle avoidance requires the vehicle to first acquire data from perception sensors. Then obstacles must be detected within the acquired data. Finally the best path must be chosen for the vehicle to follow through the terrain.

In this research a complete UGV architecture has been developed that encompasses all of these requirements. Many other research projects have devised full UGV architectures or just different approaches for the individual architectural components. As such there is a wealth of literature concerning UGV system design. This chapter reviews some of these projects.

This literature review is organised into six separate sections. The first provides a brief history of UGV research. Following this is a review of the methods for accurately localising unmanned ground vehicles. The next three sections examine elements of the obstacle detection and avoidance process. These discuss techniques for generating 3D range data, analysing the traversability of terrain and planning a route through the terrain. The chapter concludes by identifying the sources of inspiration for this research.

3.1 HISTORY OF UGV RESEARCH

Very few robotic vehicles have been produced specifically for the task of demining. However such vehicles would have to overcome the same challenges as any other UGV operating in natural terrain. This section introduces some of the well known UGV projects. After this developments in the field of unmanned demining vehicles are surveyed. Finally state of the art for a UGV is discussed.

From around 1980 research has been conducted to develop unmanned vehicles that can operate autonomously in natural terrain. In the past few years the technology has matured to the extent that long range autonomous cross country navigation is now a reality. Much of the UGV research has been driven by a small number of military sponsored projects. A selection of the better known UGV projects is displayed on a timeline in Figure 3.1.

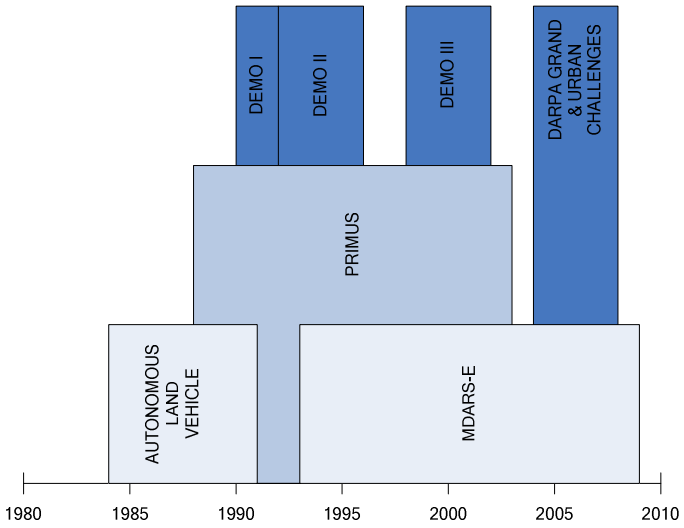

Figure 3.1 - UGV Timeline

3.1.1 Autonomous Land Vehicle

The DARPA sponsored Autonomous Land Vehicle (ALV) program was launched in the early 1980s [Gage 1995]. ALV was based on a custom eight wheeled all terrain chassis. The four wheels on each side were linked by chains and skid steering was used to manoeuvre the vehicle. A fibreglass body shell enclosed a number of networked computers that processed sensor data. ALV is pictured in Figure 3.2.

Perception for the vehicle came from a colour video camera and a two axis scanning laser range finder. Both of these sensors were mounted at the front of the vehicle. The video camera was used exclusively during the road following trials to track road edges. The laser

range finder was used throughout the project to detect obstacles. Reports suggest that positive obstacles of at least 0.5m in height could be reliably detected [NRC 2002].

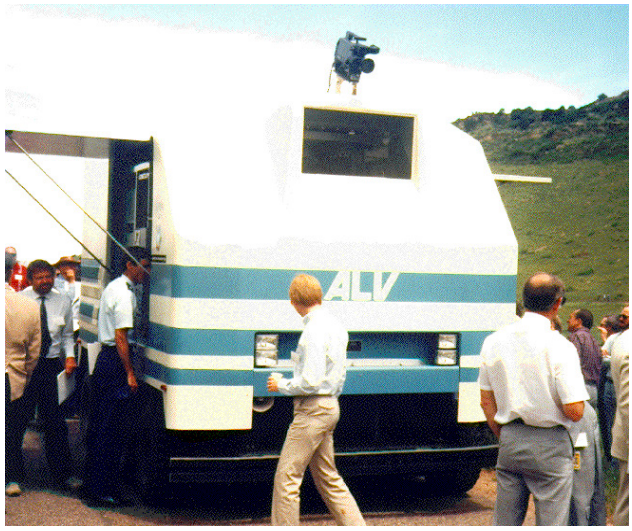

Figure 3.2 - Autonomous Land Vehicle

Localisation for ALV was based on a land navigation system that could provide vehicle heading with wheel odometry providing distance travelled. Ultrasonic sensors were used to measure the vehicle pose with respect to the ground. Finally Doppler radar was used to measure ground speed and correct for wheel slip in the odometry.

In 1987 ALV became the first vehicle to drive autonomously through a natural environment containing steep slopes, ditches, rocks and trees [Olin 1991]. During experiments, speeds of up to 3.5km/h were recorded with a longest autonomous navigation of 735m. Sensor processing time was cited as the limiting factor for the top speed of the vehicle.

3.1.2 DEMO I

As the ALV project drew to a conclusion the Army Research Lab began another UGV program dubbed DEMO I that ran from 1990 till 1992. The focus of this was the development of technology to enhance vehicle teleoperation. As part of the research effort the Robotic Systems

Division of the National Institute of Standards and Technology (NIST) developed a teleoperated HMMWV (High Mobility Multipurpose Wheeled Vehicle) [Murphy 1993].

NIST demonstrated a process they called retrotraverse with the HMMWV [Murphy 1996]. During teleoperation the system recorded the path that was driven as a sequence of points at metre intervals along with the speed of the vehicle at that point. When the HMMWV had reached the final position the teleoperator could command a retrotraverse. This would cause the vehicle to autonomously turn around and drive back along the recorded route.

For DEMO I the retrotraverse was based solely on the output of an inertial dead reckoning system. Later the HMMWV was equipped with a differential GPS receiver. A Kalman filter was used to combine the outputs of two localisation systems to improve performance. This fusion of systems provided fast updates while maintaining the accuracy of the differential GPS.

3.1.3 DEMO II

DARPA then sponsored another UGV program known as DEMO II that ran from 1992 till 1996 [Firschein 1997]. The aim of DEMO II was to create a team of autonomous vehicles that could perform reconnaissance missions with minimum human supervision. Figure 3.3 shows two of the four computer controlled HMMWV platforms that were constructed by Lockheed-Martin for DEMO II.

Carnegie Mellon University (CMU) developed a suite of cross country navigation software on their NAVLAB II vehicle. This went on to become the core software used on each of the DEMO II vehicles. The final suite of CMU software included a number of behavioural modes. These modes allowed for low bandwidth teleoperation or autonomous operation of the vehicles. The autonomous modes included road following, waypoint navigation and obstacle

avoidance using a local model of sensed range points [Arkin 1998]. An arbiter was used to coordinate the outputs of active behaviours and dictate UGV motion.

Figure 3.3 - DEMO II HMMWV

Overall control for DEMO II used mission plans specified by a remote human operator and downloaded to each vehicle. Plans would be overlaid on a digital terrain map of the environment. Each mission plan would dictate a path for the UGV and where mode changes should take place. A single operator console fitted in another HMMWV was used to control multiple vehicles and coordinate their actions.

The primary obstacle detection sensors on the DEMO II vehicles were two pairs of forward pointing stereo cameras. One pair had a wide field of view to cover the possible turning radius of the vehicle. A second pair had a narrower field of view giving better resolution at longer distances. Localisation of each vehicle used a combination of an inertial navigation system, odometry and differential GPS.

It has been reported [NRC 2002] that the DEMO II vehicles could detect positive obstacles 0.5m in height from a range of approximately 15m. This allowed autonomous cross country

navigation at speeds of around 10km/h. The stereo vision system struggled with the detection of negative obstacles [Shoemaker 1998].

3.1.4 DEMO III

To address limitations identified during DEMO II the DEMO III program (1998-2002) was conceived. The aim was to develop autonomous vehicles that could be deployed by day or night in conjunction with manned units. From the outset obstacle detection was attacked on two fronts by NASA Jet Propulsion Laboratory (JPL) and NIST. General Dynamics were contracted to construct the XUV (Experimental Unmanned Vehicle) to be used as a platform for the project. A DEMO III XUV is pictured in Figure 3.4.

Figure 3.4 - DEMO III XUV

NASA JPL continued to develop the stereo vision systems that had been used at DEMO II looking also at thermal infrared cameras for night deployment. NIST evaluated obstacle detection using a two axis scanning LADAR. All of these technologies were integrated into the final XUV. Localisation for the XUV used differential GPS backed up by an inertial navigation system and odometry [Hong 2002].

A great deal of the obstacle avoidance system used on the DEMO III XUV was implemented at NIST. They report that the system was capable of driving their own HMMWV test platform at speeds up to 35km/h over a rolling meadow with relatively large obstacles [Coombs 2000]. The vehicle was commanded to follow sequences of GPS coordinates. As the vehicle drove it repeatedly evaluated a set of possible curved trajectories that would lead it towards the next coordinate.

As part of the DEMO III program research was also conducted into classifying terrain. Algorithms were developed by NIST and NASA JPL to analyse LADAR and stereo range imagery [Hong 2002]. Obstacles in LADAR data were classified according to the observed density of the obstacle. If the variation between adjacent range points was high the object was likely to be vegetation. Colour was used to classify obstacles in the stereo imagery.

3.1.5 PRIMUS

During the timeframe of the DEMO I-III programs the German MoD sponsored their own autonomous vehicle program called PRIMUS (Program for Intelligent Mobile Unmanned Systems). EADS-Dornier was prime contractor for this project that ran from 1988 till 2003 [Schaub 2004]. In that period an autonomous demonstrator vehicle was constructed. It was based on the chassis of a Rheinmetall Landsysteme Wiesel 2 with a second Wiesel 2 used as a command vehicle. PRIMUS is shown in Figure 3.5.

For obstacle detection PRIMUS was fitted with a two axis scanning LADAR. A CCD camera was used to track roads, terrain contours or objects. Visual feedback for teleoperation was provided by a second operator controlled CCD camera. The localisation system was GPS based with optical gyroscopes to track vehicle pose changes.

Figure 3.5 - PRIMUS

PRIMUS had several control modes. It could maintain a heading set by an operator or follow a sequence of waypoints drawn on a digital terrain map. Additionally it could track objects or follow either roads or terrain contours. In teleoperation mode the system marked potential obstacles in the video feedback and could adapt the vehicle speed to prevent collision.

3.1.6 MDARS-E

Many of the programs above have been aimed at the task of military reconnaissance. The Mobile Detection Assessment Response System – Exterior (MDARS-E) was developed as robotic security system. This system recently completed a year long operational assessment at a US ARMY depot [Shoop 2006]. Figure 3.6 shows the MDARS-E platform.

MDARS-E could perform surveillance, intruder detection, inventory checks and ensure that storage structures are secured. Operator input was only required if an abnormal situation was detected. Ordinarily vehicles operated according to a schedule of patrol routes programmed into the operator console. Routes were made up of map waypoints that the UGV should autonomously follow.

Figure 3.6 - MDARS-E

MDARS-E has been designed primarily for on-road operation but could be used in moderately rough terrain [Carroll 2005]. Vehicle localisation was based on differential GPS with the back up of an encoder on each wheel and gyroscopes. The obstacle avoidance system used a combination of two-axis scanning LADAR and stereo vision. Additionally wide field of view horizontal scanning LADAR were mounted at the front and rear of the vehicle to help with manoeuvring in tight situations. A 360° radar system was used to detect intruders. Colour and thermal imaging cameras were also used to track the movement of any one intruder. Finally each vehicle was fitted with a RFID tag reader for inventory checking.

During the operational assessment MDARS-E vehicles operated at night and weekends patrolling a depot area of around 30 square miles. In each 12 hour operational period a vehicle would autonomously cover around 100km. The vehicles typically drove around on the depot road network in amongst the normal traffic.

3.1.7 DARPA Grand / Urban Challenge

Arguably the greatest push of UGV technology in recent years has come from the DARPA sponsored Grand Challenge and Urban Challenge events. These have led to numerous teams

developing novel vehicles, sensing techniques and software systems in an attempt to win the multimillion dollar prize money. The 2005 Grand Challenge saw five autonomous vehicles complete a 132 mile desert course. In 2007 six autonomous vehicles completed the 60 mile Urban Challenge course.

The 2005 Grand Challenge route consisted of a corridor specified by a set of GPS waypoints with associated widths and speed limits for each segment. The width of the corridor generally tracked the width of the road [Thurn 2006]. As such no global planning was required by the vehicles just the capability to follow the corridor and avoid obstacles. Waypoints were placed close together in turns so corners were not cut.

LADAR was the most common perception technology. This was used on nearly all of the competing vehicles. The winning vehicle Stanley (Stanford Racing Team) used five single axis scanning LADAR to sweep a wide patch of ground in front of the vehicle. Stanley's steering decisions were based solely on data from these LADAR. A camera was also used to classify terrain beyond the range of the LADAR according to colour. Based on an obstacle free area detected by the LADAR and shown in the camera image the rest of the image frame would be classified. The extent of the region classified as drivable was used to limit the speed of Stanley. A maximum speed of 38mph was recorded by Stanley during the event.

Sandstorm and H1ghlander were vehicles entered by the Red Team from CMU that finished second and third in the 2005 Grand Challenge. These vehicles used a combination of short range LADAR, long range LADAR and RADAR for obstacle detection [Urmson 2006]. A gimballed platform was used to track the GPS corridor ahead of the vehicle with the long range LADAR mapping the ground about to be traversed. The RADAR provided a means to detect large obstacles at long range.

Of the successful teams in the Grand Challenge the selection of localisation equipment was relatively similar. All used GPS receivers with the Omnistar differential correction service for the highest possible accuracy. Inertial navigation systems with wheel speed monitoring were also used in each of the vehicles to back up the GPS.

The 2007 Urban Challenge posed a different set of problems for the autonomous vehicles [Perry 2008]. Multiple autonomous vehicles were driving around the course at the same time. The vehicles needed to avoid each other and around 50 manned vehicles. This introduced a dynamic element to the world that was not present during the Grand Challenge. Rules also stated that the autonomous vehicles should follow Californian road laws. This implied that the vehicles must handle road intersections appropriately and keep their distance from other traffic. DARPA again provided competitors with a tightly constrained route to follow as with the Grand Challenge.

The winning vehicle named Boss was fielded by the Tartan Racing Team from CMU. For localisation the familiar combination of GPS, inertial navigation system and wheel speed monitoring was employed. The perception system was a combination of LADAR and RADAR. Multiple sensing horizons were used to allow the vehicle to reach a maximum speed of 30mph in clear space.

3.1.8 Demining Robots

A number of research projects have looked specifically at the use of robots for landmine detection and neutralisation. In Chapter 2 one of the following projects was mentioned briefly in the survey of absolute localisation techniques.

Staff at ERA Technology and Alvis Logistics (now Remotec UK) collaborated to test the deployment of a ground penetrating radar on a Wheelbarrow EOD vehicle [Mallinson 1996].

A future vision cited by this research was the use a robotic vehicle to systematically search an area of ground and produce a location registered scan.

At the Robotics Centre of the Belgian Royal Military Academy a series of robots were developed to sweep mine detection sensors over a patch of ground. One of these robots was a tracked EOD vehicle equipped with a motor controlled three degree of freedom Cartesian frame [Colon 2002]. The frame could sweep sensors across a patch of ground to the side of the vehicle. Control of the tracked vehicle was position based with an operator specifying where and when the vehicle should move. A remote camera was used to track a coloured ball mounted on the vehicle and calculate the position of the robot.

Idaho National Lab has developed a robotic countermine system in conjunction with CMU and the Space and Naval Warfare Systems Center [Walton 2006]. This small vehicle was equipped with a standard issue army metal detector and a dye spraying system. As the robot moved it marked a lane using green dye. Each time a probable landmine was detected a red dye spot was sprayed. A combination of differential GPS, wheel encoders and a compass were used for localisation. A scanning laser range finder was the primary means of obstacle detection. It was reported that the system was capable of sweeping a specified region of interest for mines.

Humboldt State University have created a semi-autonomous mine flail from a normally teleoperated system [Owens 2004]. The standard vehicle was equipped with a combined differential GPS receiver with inertial navigation system for localisation. Coverage of a rectangular area could be achieved. The vehicle was first driven under teleoperation to the corners of the mined area to set the size of the area to flail. Autonomous operation would then follow a pattern similar to a Zamboni repairing the surface of an ice rink. In this way the mined area was covered by a sequence of overlapping loops.

3.1.9 State of the Art

From this general review of UGV programs it is possible to determine state of the art. The minimal role of the operator is to specify a route as a sequence of map waypoints to be followed autonomously. Waypoint location and separation are critical factors in how reliant the UGV must be on perception sensors to follow this route. If necessary the described vehicles can always be teleoperated to find a way around obstacles that the vehicle cannot negotiate alone.

GPS is now the predominant outdoor localisation technology. Relative localisation systems are almost always fused with the GPS data. This acts both as a back up for when GPS data is unavailable and in some cases to improve the update rate or accuracy of the GPS solution.

LADAR is presently the most popular perception technology. It works day and night in any lighting conditions. This gives a considerable advantage over most camera based obstacle detection systems. Detection range seems to be a limiting factor with current LADAR. Other sensing technologies are often used in conjunction with LADAR for long range warning of upcoming obstacles.

3.2 UGV LOCALISATION

Off the shelf solutions are now available for UGV localisation. Typical devices combine a GPS receiver with relative localisation technologies such as dead reckoning and inertial measurement. Stable accurate absolute position data is output from these systems with low latency. Unfortunately the cost of these devices is very high so research is presently being conducted into low cost alternatives.

Stand alone GPS receivers are available at low cost. The same is true for relative localisation equipment such as wheel encoders and electronic compasses. So the hardware for a combined

absolute and relative UGV localisation system can be acquired for a fraction of the cost of a ready made off the shelf solution. The combination of GPS and relative localisation can offer accurate absolute position estimates with a much higher refresh rate than GPS alone. As such the relative localisation system fills the gaps between successive GPS updates. Localisation error from the relative localisation equipment is bounded by the accuracy of the GPS.

The difficult part of the localisation task is fusing the data from the GPS receiver and relative localisation equipment. Raw output from the localisation sensors is normally too noisy to be combined directly. It is common to use probabilistic techniques to fuse data from various localisation sensors. Bayes filters are a family of probabilistic techniques used for state estimation that are historically the most popular approach for position tracking problems [Thurn 2005].

3.2.1 Bayes Filters

A Bayes filter provides a recursive method of estimating the state of a system over time [Arulampalam 2002]. The filter maintains a distribution of the probability that the system lies in each possible state. At each time step a model is used to make a prediction about the states to which the system could next migrate given a control input. A set of observations of the system are then used to update the probability distribution for the current system state.

Bayes Theorem is used to incorporate new observations into the filter. So the probability distribution maintained by the filter is the probability that the system lies in each state given the sequence of prior observations. Both the control input and system observations are expected to contain an element of noise with known probability distributions that must be considered.

For the purposes of UGV localisation the system state that is being estimated can be thought of as the vehicle position. The output of the relative localisation system would form the control

input for the predictive model. Observations from the absolute localisation system are used to update the probability distribution for the system state. Examples of Bayes filters include the Kalman filter [Welch 2006] and the particle filter [Thurn 2005] although there are many others.

3.2.2 Kalman Filter

A Kalman filter calculates the exact probability density function for the state of the system at each time step. For a Kalman filter the assumption is made that this density function will follow a normal distribution. Similarly noise models for the control input and system observations are assumed to follow normal distributions. It is also specifeded that the predictive and update models must be linear with respect to the current system state. If these assumptions are satisfied the Kalman filter will produce an optimal estimate of system state [Arulampalam 2002].

Several variants of the Kalman filter exist that relax the assumption that predictive and update models are linear. These filters cope with nonlinear models by approximating them with linear functions. An example is the extended Kalman filter [Welch 2006]. The first two terms of a Taylor expansion are used to approximate both models within the extended Kalman filter. Periodically this linear approximation must be recalculated for the current state estimate.

The standard Kalman filter works on the fact that any linear transformation of a normal distribution results in another normal distribution. When a normal distribution is transformed by a non-linear function the resulting distribution will not be normal and could even be multimodal. So the extended Kalman filter makes a normal approximation of the real distribution. Hence if the estimated system state is too far from the actual system state the linear model approximations will be poor and the filter could diverge [Dudek 2000].

Some variant of the Kalman filter is used to fuse absolute and relative position estimates in most current UGV projects. In the UGV projects described earlier a Kalman filter was implemented by NIST [Murphy 1996] for localisation of their robotic HMMWV. The HMMWV was equipped with a localisation payload including differential GPS, an inertial navigation system and rear axle odometry. Humboldt State University also used a Kalman filter to fuse data from GPS and an inertial navigation system on their semi-autonomous mine flail.

An extended Kalman filter has been developed to localise an autonomous mowing machine [Batavia 2002]. This vehicle was equipped with a differential GPS receiver for absolute localisation. A dead reckoning system based on wheel encoders and a gyroscope provided relative localisation. This localisation system allowed the machine to autonomously mow convex areas with a series of parallel stripes.

3.2.3 Particle Filter

In contrast a particle filter only produces an approximation of the probability density function for the system state at each time step. This approximation is based on a large number of samples from the density function called particles [Thurn 2005]. As a result the particle filter can model systems where the probability density function for system state is not a normal distribution. Also no assumptions are made about the linearity of either the predictive or update models with respect to the state of the system. An estimate of the state of a system is typically made using a weighted average of the particles.

A popular basic particle filter implementation is the Sequential Importance Sampling (SIS) Algorithm [Arulampalam 2002]. At each time step the current set of particles are translated. Individual particle translation happens according to some control input with the addition of an appropriate randomly generated noise component. This has the effect of translating and

spreading the particles. Next the importance factor of each particle is calculated. This is based on the probability that the most recent observation could happen if the particle represented the true state of the system. The noise model for observations is used to derive this probability. Finally a state estimate is made from the weighted average of the particles. Normalised importance factors are used as the weights.

This basic particle filter implementation will suffer from degeneracy. After a number of iterations of the filter many particles may have a negligible importance factor [Thurn 2005]. This implies that the particles represent an unlikely system state. To combat this problem any particle with low importance could be removed and resampled.

Resampling can take place each time step or be triggered by a measure of the effective number of particles in the filter. One such measure is the inverse sum of squares of the importance factors [Arulampalam 2002]. The resampling process generates a new particle population from the existing population. Typically the importance factor of each particle is used as the chance that a particle will be included in the new population. The process selects particles with replacement so the new population can include multiple copies of a particle.

The use of particle filters for UGV localisation is not well documented. Variants of the Kalman filter are used in nearly all cases. However particle filters have been applied to the localisation problem for mobile robots operating indoors. The only research encountered where a particle filter has been proposed to fuse GPS data with other localisation equipment is for a project developing a navigation system for the blind.

For the localisation of indoor mobile robots map matching techniques are often employed. A well known approach is called Monte Carlo Localisation. This is a form of Markov Localisation that was introduced in Chapter 2. Monte Carlo Localisation uses a particle filter

to localise the robot. The robot MINERVA successfully demonstrated this form of localisation while operating as a museum tour guide [Dellaert 1999].

Initially MINERVA was given the belief that it could be anywhere within a mapped area of a museum. This was represented by a random and uniform distribution of particles across the map. Once MINERVA began to move the particles would rapidly converge to the true location of the robot on the map.

Wheel odometry provided the control input to translate the particles. Images from an onboard camera pointing upwards were also compared with a brightness map of the museum ceiling. This step was done to hone the importance factors of the particles. Resampling of the particles was performed at each time step with probability of selection proportional to importance factor.

A particle filter was proposed for the problem of fusing GPS and relative position estimates [Ceranka 2003]. This research considered part of a navigation system for blind people moving around an urban environment. Position estimates from a dead reckoning system based on a pedometer and compass were fused with GPS via the particle filter. The output of the particle filter was spatially constrained to only consider positions on a network of pavements. Each time particles were translated they were cross checked against a database of pavement regions. Particles that fell outside the pavement network were resampled. No experimental results were reported from this research.

3.3 3D RANGE IMAGERY

Autonomous navigation through natural terrain requires a 3D model of the terrain to be constructed. From this model obstacles can be detected. Chapter 2 introduced a variety of sensors used for UGV perception. Of those sensors some can directly perceive 3D images of the ground and others need to be swept along the ground to generate a 3D model.

This section reviews some of the research into 3D imaging that is being applied to the UGV domain. The review is restricted to literature concerning laser range finders and similar time of flight devices. These were the only sensors considered during the research program. The primary reason for this was that there is significant experience within the Robotics Research Group of using this mode of perception [Ye 2006]. These devices also offer high accuracy and resolution.

Low cost techniques to generate 3D range data are discussed here. First strategies to generate 3D range data using an off-the-shelf single axis scanning laser range finder are presented. Two axis scanning laser range finders are available off-the-shelf but not low cost. Next the use of a 3D time of flight range camera is considered. These new devices generate 3D range data directly and are available as advanced prototypes from several manufacturers. For each of the range data generation techniques examples of obstacle detection are given.

3.3.1 3D Range Data from a 2D Sensor

There are two common ways to generate 3D range data using a single axis scanning range finder. The first and most simple method is to angle the scanning plane to sweep along the ground as the vehicle moves. This allows a 3D model to be built as a sequence of 2D line scans. An alternative technique is to mount a single axis scanning range finder on a device to continuously alter the pitch or roll of the range finder. In this way the plane scanned by the range finder is rotating and will effectively scan a portion of a spherical volume.

The so called push broom style of laser scanning was employed for obstacle detection during a robotic search for meteorites in the Antarctic. Researchers at CMU developed a vehicle called Nomad that was equipped with a spectrometer for meteorite analysis [Apostolopoulos 2000]. Nomad primarily used a single axis laser range finder sweeping the ground in front of the

vehicle for obstacle detection. Nomad also had stereo vision but this was found to perform poorly in the icy environment due to a lack of texture.

On Nomad the laser ranger finder was fixed to scan a line on the ground 5m in front of the vehicle. Range data was processed by fitting a line through the points from a single scan using least squares regression [Moorehead 1999]. The fitted line was considered to be the ground. A grid based goodness map was used to represent the sensed area around Nomad as the vehicle moved. The goodness value for each grid cell was inversely proportional to the average residual length for range data that fell in the cell. If a large change in the ground level was detected between adjacent scans this indicated a step and all associated grid cells were given a low goodness value.

For the 2005 DARPA Grand Challenge the Stanford Team used five single axis laser range finders on their vehicle Stanley [Thurn 2006]. These sensors measured cross sections of the terrain at various distances up to 25m in front of the vehicle. Multiple sensors were used for redundancy. Range data from each sensor was projected into a global coordinate frame according to the estimated pose of the vehicle and combined into a 3D point cloud.

Obstacle detection for Stanley used the 3D point cloud. If two nearby points differed in height by more than a threshold value the points represented an obstacle. For greater efficiency this test was performed over a grid rather than point wise. Only the maximum and minimum values for each grid cell were used. The obstacle labelling scheme considered the probability that the difference in maximum and minimum values between adjacent grid cells exceeded the threshold. This made the system robust to errors in vehicle pose estimation.

Researchers at the University of Hanover have experimented with mounting a single axis laser scanner on various rotating mechanisms [Wulf 2003]. In one experiment they trialled a

mechanism that rapidly adjusted the pitch of the sensor as a way to build up a 3D image of the environment. Another 3D imaging scheme was to mount the single axis scanner on a spinning shaft so that it scanned a hemispherical region in front of the sensor. Asynchronous time stamped data from the laser scanner and a rotating mechanism was periodically received. From this data linear interpolation was used to project each range point as a consistent point in 3D space.

In later work from the same group a vehicle was constructed to autonomously navigate in unknown but structured outdoor environments [Brenneke 2003]. The vehicle used a continuously rotating 2D laser range finder to model the environment. Vertical lines were scanned by the range finder that was fixed to a pan mechanism and rotated repeatedly left then right. A localisation system based on wheel encoders and a gyroscope was used to ensure points were accurately projected into world coordinates. From the 3D point cloud obstacles were segmented and projected as a 2D obstacle plan. The obstacle plan was then used as input to a simultaneous localisation and mapping algorithm to aid navigation of the vehicle.

3.3.2 3D Time of Flight Range Cameras

Limited research has been reported concerning the use of 3D time of flight range cameras as the technology is still at an early stage. All known experiments have been aimed at the use of this technology in structured environments. The perception system within NOVA that is described later in this thesis assumes no structure and is intended to analyse the surface of the ground in front of the UGV.

In a joint project between CSEM (Centre Suisse d'Electronique et de Microtechnique) and the Swiss Federal Institute of Technology a 3D time of flight camera has been evaluated on a mobile robot [Weingarten 2004]. The work employed a CSEM Swiss Ranger camera mounted

on an indoor mobile robot with the optical axis of the camera parallel to the floor. Control of the robot was achieved using a reactive obstacle avoidance system.

Range data from the image sensor within the Swiss Ranger was analysed by pixel column. In this way the 3D data was squeezed into a 2D obstacle plan with any objects detected above the robot ignored. A conclusion from this work was that the robot was proficient at avoiding collision with overhanging obstacles due to the available 3D data. However a complete transient obstacle map was required due to the limited field of view of the Swiss Ranger.

As part of the Industrial Autonomous Vehicle Project the National Institute of Standards and Technology (NIST) have also performed experiments with a CSEM Swiss Ranger [Bostelman 2005]. These experiments aimed to evaluate the use of 3D time of flight range cameras for industrial robot applications. To that end NIST mounted their Swiss Ranger on an experimental factory AGV again with the optical axis of the camera parallel to the floor.

NIST developed an algorithm to processes the data from the camera. The first step of their algorithm fitted a floor plane to the intensity image. Next the edges of the robot path were superimposed so only obstacles on the path are considered. The portion of the intensity image on the robot path was then segmented to find obstacles. Any pixels belonging to potential obstacles were crosschecked with the range image to test if they were above the fitted floor plane. Output from the algorithm was then used to build a 2D top down obstacle map.

A potential problem with 3D time of flight range cameras in the AGV application was identified by the NIST experiments. As the modulation of the light source is sinusoidal these cameras have a specific non-ambiguity measurement range of 7.5m. Objects detected beyond the extreme of this non-ambiguity range are recorded as being inside the 7.5m. For example an obstacle detected at 11.5m will be recorded at a range of 4m as the reflected light will have the

same phase shift. A possible solution is to threshold the distance image based on pixel intensity [Einramhof 2007]. Pixels that have a genuine short distance should also have a relatively high intensity.

In the Swiss MINORA project a time of flight range camera was developed for use in automotive applications such as adaptive cruise control. For this sensor an approach to obstacle detection in the low resolution image sequences was proposed by researchers at the University of Bern [Sobottka 1998]. Their approach was pixel based and used radial slope to classify obstacles in a simulation of a road scene ahead of a vehicle carrying the range camera.

Radial slope is the 2D slope between two vertically adjacent image pixels when transformed into 3D coordinates. Pixels with a radial slope above a threshold value were defined as obstacles. The resulting obstacle pixels were then connected together to segment the pixels into objects. This connected component analysis used the rule that any pixel in the neighbourhood of an obstacle pixel should be connected if their depth differed by at most 1m.

An additional step was later added to the analysis in which objects could also be merged if they were close in 3D space [Meier 1998]. This had the effect of combining regions that belonged to the same obstacle but had been split due to incomplete data. These methods were successfully demonstrated for segmenting vehicles and roadside obstacles that were subsequently tracked by the system.

3.4 TERRAIN TRAVERSABILITY

As 3D range data is acquired it can be used to determine the traversability of the terrain around the UGV. There are several important tasks in this process. First range data must be stored efficiently as it would be unwise to try to hold all of the generated range data. Obstacles must

be detected within the range data. Obstacles should then be marked so they can be avoided. Also as the UGV will be in motion these tasks are time constrained.

There is no set order in which the tasks must be completed. Obstacle detection can be performed on raw range data frame by frame. Alternatively range points can be first stored and obstacle detection then performed over an abstract representation of the terrain. Some systems have used a combination of both.

Storage of range data implies the creation of some type of map. For navigation in natural terrain a spatial representation of the world seems the most appropriate. Most common spatial representations are based on a uniform 2D grid of cells. Although for large areas this type of map could consume vast quantities of computer memory. Other spatial representations exist that use non uniform cells to reduce the storage requirement. Of course use of a limited size map also reduces the storage requirement.

Popular spatial representations include binary obstacle maps, elevation maps and certainty maps. Binary obstacle maps indicate only if individual grid cells are traversable. Elevation maps store information about the height of data points that fall in each cell. Either of these types of map can form the basis for a certainty map. Certainty maps simply associate a confidence value with the information stored about each cell.

It is typical in the realm of indoor robotic navigation for obstacles to be defined as surface points above the ground plane. Once the ground plane is identified obstacle segmentation is relatively straightforward. In natural terrain the concept of a ground plane is not easy to apply as the surface may not be planar. Undulations in the ground may themselves form untraversable obstacles for the robot so a different methodology is required.

Geometric analysis of range data is widely regarded as the most robust approach to obstacle detection for rough terrain navigation. Obstacles may be positive (protrusions from the ground), negative (depressions in the ground) or overhanging. A step in height between adjacent points of range data is a critical indicator that an obstacle is present. Useful information about the average slope and roughness of terrain can also be found by considering groups of range data. Analysis of these features is like applying a vehicle model to the range data.

If raw sensor data is of sufficiently high resolution obstacle detection can be performed directly on new sensor data. Any detected obstacles could then be noted straight onto a binary obstacle map which could in turn be used for path planning. This would not be possible if sensor data was sparse. In this case multiple frames of sensor data would be needed to accurately detect obstacles.

Thus low resolution range data would have to be collected in an elevation map. There would be no need to keep all range data that fell into each map cell. Appropriate summary statistics can convey as much information and use less storage space. As new range data is added to the elevation map the traversability of grid cells would be tested. If summary statistics indicated that a cell contained an obstacle it would be marked as untraversable. The output of this operation would be like a binary obstacle map and could be used for path planning.

It is difficult to discuss the stages of traversability analysis in separation. The various tasks are highly interleaved. In the following subsections a number of approaches are described. Certainty grids are discussed first as they are the most popular form of traversability analysis for robotic vehicles. After this strategies for traversability analysis from the well known UGV projects are presented.

3.4.1 Certainty Grids

The concept of a certainty grid [Moravec 1985] or occupancy grid [Elfes 1989] dominates the literature as a probabilistic representation of spatial information for mobile robotics. A certainty grid models the world as a uniform grid of 2D or 3D cells. Each grid cell represents the probability that an obstacle occupies the corresponding space in the environment.

In the original implementation range sensor readings were placed into the grid using sensor models. The models were probabilistic and described the certainty that an object existed in individual grid cells given the observed sensor response. Accurate placement of sensor readings in the grid required the position and orientation of the sensor when the reading was taken. Sets of readings from different sensors and taken at different positions were progressively added to the grid enhancing the sensed boundaries of obstacles. Bayes Theorem was used to combine sensor data and update the certainty values for each cell.

Borenstein and Koren recognised that building a certainty grid could be computationally intensive. As a result they developed their own variant of the certainty grid known as the histogram grid [Borenstein 1991]. The histogram grid differs from the certainty grid in the way it is built and updated. An example histogram grid is shown in Figure 3.7.

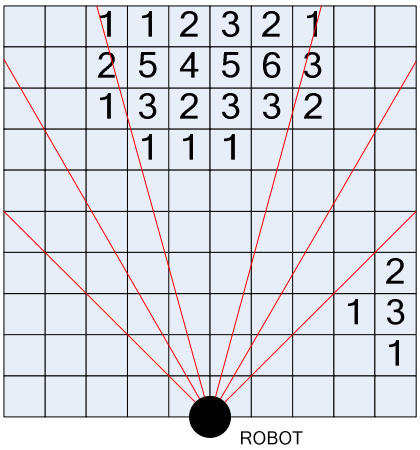

Figure 3.7 - Example Histogram Grid

The certainty grid projected the implications of each sensor reading onto the grid by way of probability profiles for the sensor that updated many cells. In contrast for each range reading inserted into the histogram grid only the count in a single grid cell was incremented. The count for all grid cells on the line between the observed obstacle and sensor was correspondingly decremented. As such the histogram grid created an observed probability distribution for obstacle locations from the continuous sampling of the sensors as the robot moved.

3.4.2 Autonomous Land Vehicle

Some of the earliest research into obstacle detection and mapping for a UGV came from the Autonomous Land Vehicle (ALV) program. Section 3.1.1 introduced this project. ALV used a two axis scanning laser to produce 3D range images of the terrain. The initial obstacle detection capabilities for the ALV allowed avoidance of sparse positive obstacles on road [Dunlay 1987]. Obstacles were detected using a basic method that derived the vertical height of points in each range image relative to the sensor.

The assumption was made that the ground in front of the ALV was a flat plane within the field of view of the sensor. From a few sample range points the tilt of the sensor was calculated with respect to the road surface. A vertical height was then derived for each range point via a trigonometric operation. Using a simple height threshold a binary obstacle image was produced. This allowed for obstacles to be segmented and recorded in a list of obstacle points called a Scene Model. It was acknowledged that this system was not adequate for rough undulating roads.

Subsequently Hughes Research Laboratories developed techniques for the ALV to map and detect obstacles in cross country terrain [Olin 1991]. This system detected obstacles within a Cartesian elevation map built from the laser range data. Traversability of the map was determined according to a number of constraints. These included permissible slopes, vehicle

ground clearance and an acceptable state for the ALV suspension. The constraints were tested using a simplified model of the ALV. The model was fitted to the elevation map at intervals along possible vehicle trajectories to evaluate the traversability of each trajectory.

Several different vehicle models were trialled [Daily 1987]. The first approximated the ALV as a plane specified by points that represented the wheels of the vehicle. This model was applied by first fitting the plane to the elevation map by least squares regression. The fitted plane was then shifted vertically to see if a position existed where all of the vehicle wheels could touch the ground within the allowed suspension range. From the fitted plane the slope and ground clearance constraints could be easily checked. For computational efficiency a second model that represented the ALV as a line was also tested. This model allowed only the slope and ground clearance constraints to be evaluated.

3.4.3 CMU NAVLAB

The Navigation Laboratory (NAVLAB) program at Carnegie Mellon University (CMU) was a long running UGV project. The aims of NAVLAB were to build complete systems capable of autonomous navigation on road and in cross country environments. A number of different vehicles were used as NAVLAB test platforms. NAVLAB II was specifically developed for operation in natural terrain and was based on a HMMWV chassis.

For the original NAVLAB vehicle Hebert and Kanade developed methods to analyse range images from a two axis laser scanner [Hebert 1988]. They introduced the concept of a bucket map which was a regular grid defined on a horizontal reference plane. Each grid cell or bucket contained a set of measured points from one or more consecutive range images. In processing the range images first a median filter was applied to help remove image artefacts such as mixed pixels and values beyond the non-ambiguity range of the sensor. Next points from the laser

scanner were converted to the coordinate frame of the UGV and allocated to map buckets. NAVLAB maintained a vehicle centred portion of the bucket map as the vehicle moved.

Obstacle detection with the map used a four step algorithm. First the bucket map would be searched for elevation discontinuities over a 2 x 2 bucket neighbourhood. Discontinuities signalled the edge of obstacles so whole obstacle regions could then be extracted. Next a surface normal would be computed for each map bucket. Buckets with a critically steep normal would also be marked as obstacles.

A behaviour based system for off-road navigation was developed for the NAVLAB II vehicle that used a similar method of grouping range points [Langer 1994]. As before range data from a 3D LADAR was converted to the coordinate frame of the UGV before being allocated to cells in a discrete traversability grid. For this research they introduced the concept of Early Traversability Evaluation. This meant the perception module used single range images to make traversability decisions. When a decision was made by the perception module it could not be revisited at a later stage by other parts of the system.

Summary statistics were calculated for each cell in the traversability grid. Minimum height, maximum height, variance of height and a weighted least squares slope were found for each cell. A cell was then classified as untraversable if either the height range or slope was outside acceptable bounds. This corresponds to testing each cell for vehicle undercarriage clearance and acceptable tilt.

The output of this perception module was a set of untraversable terrain regions made up of cells. These cells were passed to a local mapping module known as GANESHA. This module maintained a separate 2D grid based map of the environment around the UGV. As the UGV moved objects in the map were moved from cell to cell relative to the vehicle pose.

A local map was preferred to minimise the effects of accumulated localisation errors. For the initial implementation the grid extended 20m in front of the UGV and 10m to either side with individual cells 0.4m x 0.4m in size. The map resolution was larger than the typical error accumulated in localisation while traversing an area the size of the map.

The GANESHA algorithm had four steps. First the pose of the vehicle was read from the localisation system. Then cell positions on the grid were updated according to this new pose. Hence as the vehicle moved the map content was translated and rotated [Langer 1997]. Cells that moved outside the active area of the map were simply discarded. New untraversable objects detected by the perception module were then added to the grid. Finally the attributes of each cell in the grid were updated. Cell attributes included a history value and an indication if the cell was presently within the field of view of the sensor system. Cell history represented the number of times the current cell content had been observed.

GANESHA featured a map maintainer function. This ensured that cells wrongly classified as untraversable were removed by checking for their presence in subsequent input from the perception module. The history value of a cell would be decremented if an object in the field of view was not detected in the cell during a sensing cycle. The decay amplitude was such that objects always took the same number of cycles to disappear regardless of how many times they were observed before disappearance.

A novel method to overcome the limited field of view of the NAVLAB II sensor system was also provided by GANESHA [Langer 1994]. Artificial field of view obstacle points were generated on the local map along the extremes of the field of view. This prevented the UGV from turning blindly into unobserved regions to either side of the vehicle. Figure 3.8 illustrates the idea. These artificial obstacles were observed only when the vehicle came within 1m. At that point the field of view obstacle was treated like any other untraversable region of the map.

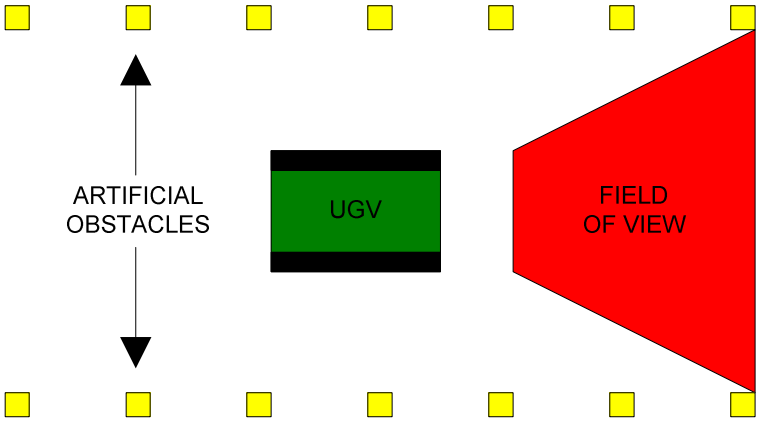

Figure 3.8 - Artificial Field of View Obstacles

3.4.4 DEMO II

The DEMO II program was introduced in section 3.1.3. Much of the software developed for the CMU NAVLAB II vehicle was adopted for the DEMO II UGV program. NASA JPL added obstacle detection algorithms for use with the stereo vision system used on the DEMO II vehicles. These algorithms were demonstrated in conjunction with the GANESHA local mapping module [Matthies 1995].

The obstacle detection algorithms developed by JPL operated on column oriented 1D range profiles from stereo or two axis LADAR range images. Each image column was analysed for sizable steps in the corresponding range profile. For positive obstacle detection it was assumed that each pixel in the image contained the base of a step obstacle of the maximum traversable height. Given the geometry of the sensor the corresponding pixel in the same image column could be determined in which the top of the assumed obstacle should appear. The range value for this upper pixel was then tested to see if the step was too large.

For negative obstacle detection the assumption was made that the ground was approximately flat. A predicted range value was determined for each pixel in the image based on the

observed range value from the pixel below in the same column. If the actual range was much larger this indicated the presence of a negative obstacle.

All traversability decisions were based on a single range image. Once an image was processed for obstacles a blob filter was applied to remove spurious isolated obstacle pixels. This removed the effects of noisy sensor data.

3.4.5 DEMO III

Section 3.1.4 introduced the DEMO III program. A combined obstacle detection and grid based mapping system was developed for the DEMO III XUV [Hong 2002]. Contributions came from a number of supporting institutions including the Intelligent Systems Division at NIST and NASA JPL. The system was initially implemented by NIST and then refined by JPL.

The NIST obstacle detection system first converted two axis LADAR range data to vehicle coordinates then searched for obstacles. Positive obstacle detection employed an algorithm similar to that used for DEMO II. Column oriented 1D profiles from each range image were analysed to find positive obstacles. In this work the assumption was made that there was at least one traversable ground point for each column of the image. This ground point was initialised as a location under the front axle of the UGV [Hong 2002].

The algorithm started from the newest ground point and would classify adjacent points in the image column. Another point would be declared as a ground point if it satisfied conditions in terms of relative elevation and slope from the last ground point. If the vertical separation of the points was more than a threshold value this represented an untraversable step in elevation. Likewise an untraversable gradient was detected if the slope between the points was steeper

than a threshold angle. Thresholds for step and slope traversability were set according to what the UGV could safely negotiate.

Negative obstacles would also be detected by the positive obstacle detection algorithm [Chang 1999]. However due to sensor location the method was not good enough to detect negative obstacles at sufficient distance for high speed autonomous driving. Instead a map based approach to negative obstacle detection was favoured [Hong 2002].

Ground points classified by the positive obstacle detection algorithm were accumulated in a map. Each new ground point found in a range image was projected onto the map for comparison. A check was made to see if any other ground points on the map lay within a threshold ditch crossing distance. If not the closest ground points were tested to see if they were higher by more than a step threshold. Failure of both of these tests indicated the new point represented a negative obstacle.

The earlier obstacle detection algorithms for DEMO II and DEMO III were prone to missing obstacles for which the surface normal pointed away from the sensor. This was due to the column oriented fashion in which range data was processed. JPL addressed this with an improvement to the algorithm that clustered together obstacle points [Talukder 2002]. The new algorithm used an obstacle definition similar to that proposed by NIST above. As such range points were classified as obstacles if they represented an untraversable step or gradient with respect to an adjacent point. This implied that any points that fell in a particular region of 3D space above another range point were obstacles. The definition described a region shaped like a truncated cone.

Figure 3.9 illustrates the obstacle detection process. In the figure blue points are obstacle points while red points represent traversable ground. H_T is the maximum step size the vehicle can climb and θ_T is the maximum allowed gradient.

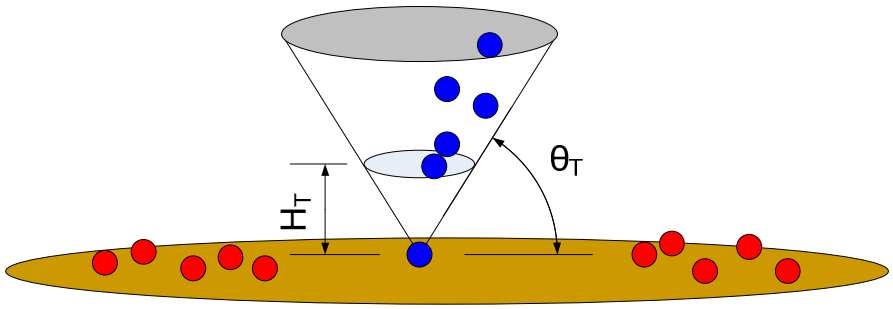

Figure 3.9 - Obstacle Detection using Point Clusters

For computational efficiency the vertical extent of the cone was limited to reduce the search space. Choice of the maximum height was critical for the detection of overhanging obstacles. Additionally the search process only considered single range images. This allowed the search for obstacles to happen in the image plane rather than over an obstacle map. As such only projected triangular regions of the image needed to be searched for obstacle points.

As part of the DEMO III UGV program NIST also developed a confidence based mapping system to facilitate vehicle navigation [Chang 1999]. The system used a 2D grid based obstacle map following the ideas of the CMU bucket map [Hebert 1988]. For each cell in the map there was a record of the terrain type, an obstacle list, measure of surface slope, average elevation and a confidence value associated with the elevation data.

The method of updating the confidence map for the DEMO III XUV was based on earlier research at NIST. NIST developed a spatial mapping system as part of an Autonomous Underwater Vehicle (AUV) project [Oskard 1990]. This system was based on a set of depth

maps of the terrain on the bottom of a lake. Each map had a different resolution and higher resolution maps depicted an area within the coverage of a lower resolution map.

The lowest resolution map contained known geographical data for mission planning. Higher resolution maps were used for individual vehicle path planning and were periodically updated according to new sensor readings. Each AUV also had a very high resolution local map that depth sensor readings and obstacle information were incorporated into directly. This local map was nominally centred on the AUV. As the vehicle moved a new local map was generated each time the AUV entered the final 25% of the map. New local maps were built by overlaying the known geographical data for the region with any recent sensor data.

This local mapping system could deal with dynamic environments. Each cell of the local map had an associated confidence value. When conflicting information was detected between different sensor readings or geographical data the confidence values for the cell were decreased. This continued until a threshold was reached at which point the newly sensed value was used on the map if consistent. Confidence values were increased linearly if new sensor readings were in agreement. A percentage based confidence value was used to allow the system to quickly react if the environment changed after remaining static for a long time.

3.4.6 PRIMUS

PRIMUS was introduced in section 3.1.5. For the initial phase of PRIMUS an evaluation vehicle was developed that was capable of autonomous operation in open terrain [Schaub 2004]. This vehicle was equipped with a two axis laser scanner that was constructed specifically for the project.

Lux & Schaefer implemented the first terrain modelling and obstacle detection scheme for the vehicle [Lux 1991]. They modelled the environment as a mesh of voxels (3D pixels) and

processed range data by voxel. Conceptually their work was very similar to the bucket map developed by CMU.

First range data from the two axis laser scanner was transformed into a 3D representation and associated with a voxel in the mesh. Next individual voxels were checked for untraversable steps in elevation and steep slopes. A count was made of all point heights inside a particular voxel that differed from the lowest by less than the threshold. These counted as votes in favour of the voxel being traversable. A gradient check was then made on each point in the voxel by comparison with the lowest point in adjacent voxels. Each time the maximum slope was exceeded gave a vote in favour of voxel containing an obstacle. If sufficient points existed in the voxel that pass or fail the tests then the voxel would be marked traversable or obstacle respectively. Obstacle votes overruled others.

For later phases of the PRIMUS program the obstacle avoidance system was substantially modified. Schwartz detailed the obstacle avoidance system used in Phase C of the project [Schwartz 2000]. Here 3D range images were converted to a 2D obstacle map which was evaluated for traversability by the tracked Wiesel 2 vehicle. Analysis took account of allowed slopes, vehicle ground clearance and trench crossing potential when calculating if terrain could be negotiated. A traversable direction was also calculated and marked in the obstacle map cells. Each cell could be crossed in up to 8 possible directions.

For local navigation PRIMUS used a 60m x 60m obstacle map split into 36cm x 36cm pieces. Cells in this map were coded as unknown area, obstacle, no obstacle or direction dependent traversable. Map cells also had an associated risk value which was used to adapt vehicle speed when traversing the cell. The risk value was associated to the proximity of detected obstacles.

3.4.7 MDARS-E

Section 3.1.6 introduced the ongoing work for MDARS-E. As part of the MDARS-E system a 2D obstacle grid map was used to fuse data from all of the obstacle detection sensors [Kurtz 1997]. This was implemented as a local map centred on the vehicle and the orientation of the vehicle on the map changed according to the vehicle heading. As the vehicle moved the map would scroll. Old map cells would roll off the edge of map behind the vehicle and new clear cells would appear in front.

Each cell of the obstacle map had an associated confidence that an obstacle was present in that location. Confidence values were updated as obstacles were detected with a weighting according to the false alarm rate for the sensor and the strength of the detection. If an obstacle was not detected for an area of the map within the current sensor field of view the confidence value was slowly decayed. This allowed the vehicle to continue moving if an obstacle blocking the path subsequently moved.

3.5 LOCAL NAVIGATION CONCEPTS

This section reviews some of the popular approaches to path planning given a local obstacle map. A local path planner can plan a path from the current UGV position to the edge of the obstacle map. Global path planning was not considered in this research. It was assumed that the UGV operator will have specified the best known global path as a set of map waypoints.

Local path planners perform the task of avoiding obstacles not known about ahead of the UGV mission. As such local path planning is a reactive operation. Ordinarily the local planner will guide the UGV along the ideal direct route from the present position to a goal. When an obstacle is encountered the planner will analyse the obstacle map to find the best path to circumnavigate the obstacle and still progress towards the goal.

Local path planners have a number of deficiencies. Most importantly their analysis of the terrain is myopic. Only the immediate area around the UGV is considered. At best this is the extent of the obstacle map but when exploring new terrain may be limited to just the field of view of a sensor. As such a UGV that relies solely on local path planning is susceptible to becoming trapped in cul-de-sac obstacle arrangements [Stentz 1995]. Figure 3.10 shows a UGV driving into a cul-de-sac as it lies on the ideal path to the goal and is deeper than the extent of the obstacle map.

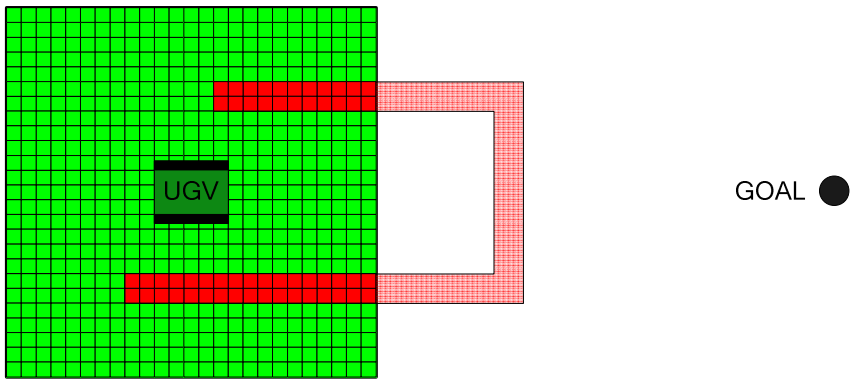

Figure 3.10 - Cul-de-sac not seen on Obstacle Map

A variety of local path planning methods have been developed to guide robotic vehicles. The most simplistic technique would be to search for obstacles on possible paths across the obstacle map. Other techniques that are commonly employed are based on artificial potential fields or polar obstacle density. In the following subsections these different types of path planner are discussed. Techniques used to escape from cul-de-sac scenarios are also reviewed.

3.5.1 Linear Path Search

The linear segment search is the simplest local path planning method. Each time the planner is run it searches for obstacles on straight line path segments across the obstacle map. Typically segments start at the UGV location and head towards the edges of the obstacle map. Segments need to be at least the same width as the UGV to ensure safe passage.

Normally a segment following the ideal heading will be searched first. If this segment is found to contain an obstacle it is discounted and other segments are systematically searched. The clear segment with smallest angular offset from the ideal path would be selected.

A similar local path planning strategy was adopted for the MDARS-E vehicle [Kurtz 1997]. The algorithm projects a line of set length on the obstacle map from the front of the vehicle along the current vehicle heading. A path segment of predefined width centred on the projected line is then searched for obstacles. If an obstacle is detected a new line is projected on the map with an incremental angular offset. The new path segment is then searched. This process continues testing segments to the left and right of the current vehicle heading until a clear path is found.

For the ALV program a reflexive path planner was developed [Daily 1988]. It periodically examined a small number of linear paths for the ALV to follow. Candidate paths radiated out from the front of vehicle. Each path was tested by applying a vehicle model to a Cartesian elevation map. The result was a safe distance to travel along each possible trajectory. Each of these linear path segments was called a Vehicle Model Trajectory (VMT).

The planner was based a group of behaviours that searched for the clearest path to the goal. If no VMT existed with a safe distance greater than a threshold then ALV would stop. The vehicle would then turn autonomously to look for alternative paths. Otherwise the planner selected the best VMT based on path length and the difference between VMT and goal headings.

3.5.2 Steering Arc Search

Other research has looked at evaluating possible steering arcs as a method of local path planning. This perhaps better acknowledges the way that nonholonomic vehicles actually

move. Typically a set of steering arcs are used that start at the front of the UGV and diverge to the edges of the obstacle map. These arcs will be repeatedly tested for the presence of obstacles.

This was the basis of the CMU NAVLAB II vehicle [Stentz 1995] and later used for the DEMO II fleet [Firschein 1997]. The navigation system was made up of behavioural modules for various tasks including obstacle avoidance using a local obstacle map. Each behavioural module provided output in terms of votes for candidate steering arcs. The set of arcs used ranged from the smallest radius left turn to smallest radius right turn [Langer 1994]. Votes from all behaviours were combined by the DAMN (Distributed Architecture for Mobile Navigation) arbiter as a weighted sum. The final steering choice made by DAMN was the candidate path with the highest average vote.

A set of rules determined the magnitude of the vote given to each steering arc by the obstacle avoidance behaviour. If an obstacle lay on the arc within a near threshold distance the arc scored a minimum value. Arcs that met obstacles between near and far threshold distances were scored according to the distance along the arc the vehicle could travel before meeting the obstacle. A maximum score could only be given to an arc that was clear all the way to the far threshold distance. As such the algorithm voted against choosing arcs that led to a direct collision.

Arcs that did not lead to direct collision were additionally moderated by the lateral distance to the closest obstacle. For near misses the arc was initially scored as if the closest obstacle was directly on the arc. Then the score was increased in proportion to the distance the obstacle lay from the arc. This moderation served to ensure the system preferred arcs that were wider than the vehicle.

For DEMO III researchers at NIST developed the Autonomous Mobility Path Planner [Coombs 2000]. This planner used a local obstacle map that stretched out 50m from the UGV and was 50m wide. To support the planner a large web of potential paths was calculated offline. These paths emanated 50m out from the front of the vehicle.

Each possible path was made up of two parts. The first 20m was represented by a clothoidal path segment. A clothoid is an arc with curvature that varies linearly along the arc. This models the steering angle of the UGV changing at a constant rate. The remainder of the path was made up from a network of unfeasible straight line segments. This was done for computational efficiency as the planner would refresh before the UGV had traversed the chosen clothoidal segment.

The Autonomous Mobility Path Planner started with a set of path segments for a number of different initial steering angles and vehicle speeds. These path segments were pruned each time the planner ran. Path segments that were blocked by obstacles were eliminated. To do this efficiently the planner used a look up table to find the obstacle map cells that each path segment passed through. Path segments had an associated width to account for the vehicle width and some clearance. Lateral path clearance was indexed by speed. Higher speed movement required a wider path and correspondingly speed was restricted for the UGV to pass through narrower gaps.

The final path choice of the Autonomous Mobility Path Planner was made according to a cost function [Hong 2002]. This cost function is based on a number of factors. Key factors included the maximum possible speed & length of path. Other considerations were the relative elevation of cells on the path and what proportion of the path had tree cover for stealthy movements.

3.5.3 Artificial Potential Fields

Artificial Potential Fields are based on idea of electrostatic forces. Electrically charged objects will exert a force on each other due to their charge. Objects with like charge repel while objects with opposite charge attract. Charles Coulomb showed that these forces were inversely proportional to the square of the distance between the objects [Breithaupt 1999].

For mobile robot path planning the robot is treated as a charged particle free to move around the environment. Obstacles exert repulsive forces on the robot. Goals attract the robot. As such collision between the robot and obstacles should not happen and the robot will be drawn towards a goal. Figure 3.11 illustrates the repulsive potential fields in an example environment with two obstacles. In this diagram the magnitude of the repulsive force is indicated by height.

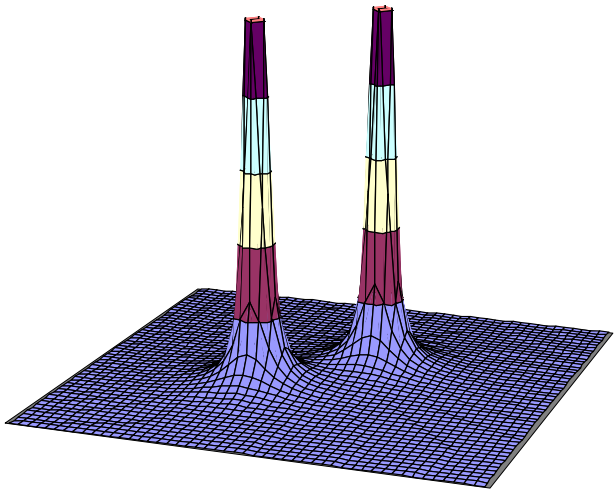

Figure 3.11 - Potential Field Obstacles

Khatib was one of the first people to suggest this approach to obstacle avoidance for mobile robots [Khatib 1986]. Potential fields have since been applied to both global and local path planning. For the global path planning problem force vectors could be calculated off-line for each cell of a global grid map. In contrast for local path planning it is normal to calculate just the momentary force vector for the robot. This corresponds to a separate force calculation for

each obstacle map cell. Individual force vectors are then summed to calculate the overall force vector for the robot.

There have been numerous implementations of potential field based local path planners. Well known developments include the Vector Force Field (VFF) from Borenstein and Koren [Borenstein 1991] and motor schema from Arkin [Arkin 1998]. They are also gaining popularity as a method to avoid obstacles when travelling at high speed in outdoor environments [Hamner 2006].

The VFF was based on a histogram grid obstacle mapping system [Borenstein 1991]. Cells from a vehicle centred local portion of the histogram grid were treated as a series of repulsive force vectors for the robot. A known target location was treated as a constant attractive force to guide the robot towards the goal.

The magnitude of individual repulsive forces was proportional to the count in a grid cell and inversely proportional to the square of the distance between the cell and the robot. Each force was taken to act along a line between the cell and the centre of the robot. Vector summation was used to derive the overall repulsive force vector.

In Chapter 2 motor schemas were discussed in the context of behaviour based architectures. A motor schema is a behaviour that generates an action vector defining the way a robot should move in response to the perceived environment [Arkin 1998]. A path planner based on motor schemas would combine action vectors from the different motor schema by vector summation.

Motor schemas have been developed to encode many robot behaviours. Move to goal and avoid static obstacles are schemas that satisfy the basic potential field concept. Other examples include a stay on path schema that drives a robot towards the centre of a path for following a

road or corridor. The noise schema generates a random force to allow a robot to escape traps. Finally a schema has been implemented that makes the last recorded robot position a repulsive field and ensures the robot moves away from previously explored space.

An explicit obstacle map is not necessarily needed by motor schemas. Perception schemas are associated with the motor schemas to generate appropriate sensory data. Typically the output from a perception schema is based on only the latest sensor readings and specific to the task of the motor schema. As such motor schema are good for path planning in dynamic environments.

Researchers at CMU have developed a potential field based local path planner for navigating at high speed in outdoor environments [Hamner 2006]. Their approach was different from the conventional potential field concept. Rather than creating a potential field over the location of the vehicle it was applied only to the vehicle heading.

Their work assumed a global path had been specified for the UGV as was the case for the DARPA Grand Challenge. A moving goal point attracted the vehicle heading. This goal point was set 10m in front of the vehicle along the specified path. The attraction force was proportional to the difference between the vehicle and goal headings.

Obstacles exerted repulsive forces on the UGV heading. Repulsion increased with decreasing angle and decreasing distance to each obstacle. The repulsion was also weighted in inverse proportion to the perpendicular distance the obstacle lay from the goal heading. This ensured the vehicle paid more attention to obstacles close to the global path than obstacles that were at some distance away but momentarily in front of the UGV.

A repulsive force was calculated for each obstacle point in a grid and overall repulsion was found by summing these forces. The attractive force was then added to generate a desired angular velocity for the UGV. Overall the vehicle dodged between obstacles while attempting to track the global path. Vehicle speed was set separately according to the distance and angle to the closest obstacle.

The potential field approach to local path planning has a number of well reported problems [Koren 1991]. Potential fields are particularly vulnerable to local minima. These are places where the attractive force of the goal is balanced by the repulsive force of an obstacle. A closely related problem is that a robot may not be able to pass through a narrow gap between multiple obstacles. This can happen due to the summation of fields from multiple obstacles.

Potential field based path planners can also exhibit cyclic or oscillating behaviour. This is caused by particular arrangements of obstacles in relation to the goal. If avoiding an obstacle causes the robot to stray away from the goal it may subsequently get attracted back towards the same obstacle. A cycle of avoiding and returning could ensue. Alternatively when traversing a narrow passage between parallel obstacles the robot may oscillate from side to side as it is repulsed by one obstacle then the next.

3.5.4 Vector Field Histogram

Borenstein and Koren also developed the Vector Field Histogram (VFH) local path planner for mobile robots [Borenstein 1991]. The workings of this planner are closely related to the artificial potential field concept. As with their Virtual Force Field approach a histogram grid was used to record range readings from the robot sensors. A vehicle centred local portion of the histogram grid was used to influence local path planning.

For the VFH path planner this local grid was transformed into a polar histogram. To perform the transformation the histogram grid was first split into a number of angular sectors around the robot. Next obstacle vectors were calculated for each cell that fell within a sector. Each obstacle vector had a value proportional to the square of the histogram count within the cell and also linearly dependent on the distance the cell was from the robot centre. The cumulative obstacle vector was then calculated for each sector and smoothed to remove the effect of mapping the coarse square grid onto angular sectors. The smoothed cumulative obstacle vector for each sector was known as a Polar Obstacle Density (POD) value.

From the resulting POD histogram the steering angle for the mobile robot was calculated. In the presence of multiple obstacles the POD histogram would feature a number of peaks and valleys. A threshold was applied and peaks in the POD histogram above the threshold were considered blocked. Valleys in the POD histogram below the threshold were considered navigable. The valley closest to the target direction would be selected by the path planner. See Figure 3.12 for an example POD histogram.

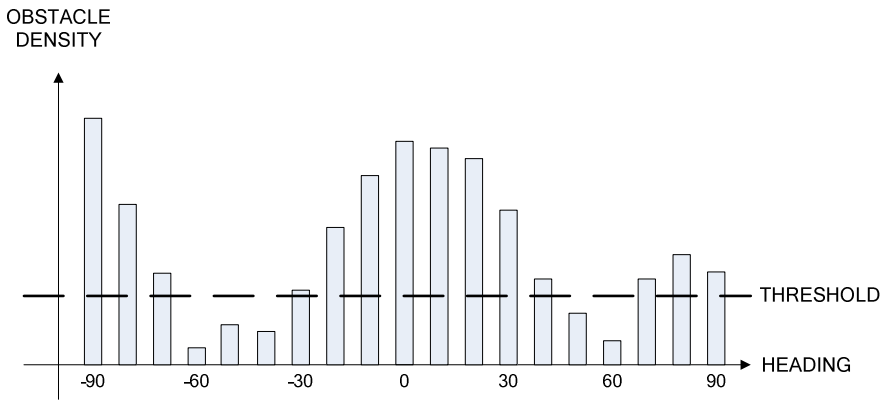

Figure 3.12 - Example Polar Obstacle Density Histogram

From within the selected valley a steering angle for the robot was chosen. If the direct path to the target lay within the valley then that would be selected. Else the width of the selected valley was determined. For a wide valley the robot would be driven at an acceptable angular

distance from the edge of the valley closest the target direction. Otherwise the robot would follow the central sector of the valley.

Various authors have suggested enhancements to the original VFH method. Most notably Ulrich and Borenstein introduced VFH+ to overcome a number of the identified shortcomings [Ulrich 1998]. The VFH approach does not take account of the width of the robot and as such can exhibit a tendency to cut corners. VFH+ deals with this weakness by enlarging obstacle cells by the robot radius plus a safe distance when the POD histogram is built. Once this is performed a robot can be safely treated as a point.

VFH+ also considers permissible trajectories for the robot where the original implicitly assumed the robot was free to instantly move in any direction at any point in time. This is implemented by assuming the robot will follow one of a small number of circular arcs when picking a steering direction and masking out those directions that appear clear but would lead to collision with an enlarged obstacle.

The Traversability Field Histogram algorithm is another variant of the VFH method developed to plan paths through rough terrain [Ye 2004]. For this method 3D terrain data was stored as a pair of discrete 2D grids [Ye 2003]. One grid held elevation data and the other held corresponding certainty values. The elevation data was filtered based on the certainty values to eliminate any errors. Holes in the sensor coverage of the elevation map were then filled by interpolation.

Similar to the original VFH method only a robot centred local portion of the elevation map was used to influence path planning for the Traversability Field Histogram algorithm. For each cell in this local map a traversability index was calculated which expressed the difficulty of traversing the cell. Traversability indices were derived by fitting a plane large enough to

envelop the robot to the neighbourhood around each map cell using least squares regression. The index given to a cell was a weighted function of the slope of the fitted plane and the variance of the residuals between data points and the plane.

This traversability analysis had the effect of growing the obstacles so that the robot could be treated as a point during navigation. The resulting traversability map was then used to generate a polar histogram and guide the robot. This was done using a near identical method to that used to create the VFH polar obstacle density histogram. The only difference was that the traversability indices replaced the histogram counts from the original VFH method.

3.5.5 Cul-de-Sac Escape

An interesting point to note about the vehicles described in the previous subsections is that they are not infallible. While much is written about the success of these projects the failings are often never reported. A common flaw is the possibility that the vehicle can drive into a cul-de-sac shaped arrangement of obstacles. Many of the UGV programs have no answer for this problem.

It was reported [NRC 2002] that the DEMO III XUV could become trapped in cul-de-sacs. This was attributed to the limited sensor range of the vehicle. The PRIMUS vehicle was known to suffer similar problems requiring intervention from a teleoperator [Schaub 2004]. A means to automatically recover from a cul-de-sac situation is something that has received little research attention.

The CMU NAVLAB II vehicle demonstrated a control architecture that incorporated global and local path planning modules [Stentz 1995]. This vehicle was also susceptible to becoming trapped in cul-de-sac obstacles due to limited sensor range. However once the vehicle had been manually backed up it would avoid driving straight back into the cul-de-sac.

The global path planner for NAVLAB II evaluated the best path to a goal based on the cost of traversing terrain. Obstacles were given a much higher traversability cost than benign terrain. Additionally large radius high cost buffers would be placed around untraversable areas. So when encountering a cul-de-sac for a second time the system would plan a path around the high cost area avoiding the cul-de-sac. It seems likely that this method could still be fooled by deep cul-de-sac obstacles.

Other reported cul-de-sac escape approaches seem more haphazard. They may lead the vehicle out of the trap but do not necessarily prevent the vehicle from driving back in at a later time. For instance Arkin developed an avoid past motor schema which made the last robot position like a repulsive potential field [Arkin 1998]. This had the effect of generally forcing the robot to move away from the end of a cul-de-sac. Others have suggested following the perimeter of obstacles as a method to escape a trap [Ye 2004].

The iRobot Research Group considered cul-de-sac escape within their Wayfarer PackBot project [Yamauchi 2005]. Local path planning for this robot was based on a variant of the VFH method. This path planner included a specific cul-de-sac escape behaviour that maintained a history of the recent robot path and treated this as an obstacle to be avoided. When a cul-de-sac was detected the robot would stop and wait for the path history to clear. The path planner would then guide the robot out of the cul-de-sac with the path history obstacles preventing the robot from immediately returning.

3.6 SOURCES OF INSPIRATION FOR RESEARCH

The Nottingham Off-road Vehicle Architecture (NOVA) has been developed for the application of autonomous route proving. To allow NOVA to be used in this role the architecture contains complete systems for UGV localisation, autonomous navigation and obstacle avoidance. This section identifies literature that guided the implementation decisions

made during this research. As such aspects of NOVA bear some resemblance to the research efforts highlighted here.

3.6.1 Autonomous Route Proving Behaviours

An objective for NOVA was to create a system that would record the path that the UGV had followed. This recorded path could then be used to back through the environment along the proven route. Alternatively parallel swaths could be generated to allow the proven route to be widened. Work by NIST as part of the DEMO I UGV program demonstrated that this was a realistic possibility [Murphy 1996]. Their robotic HMMWV recorded the path it followed and featured a retrotraverse behaviour to autonomously drive the vehicle back along the recorded path.

Swath coverage of an area was another objective for NOVA. Several projects confirmed that this was an achievable goal. Researchers at CMU have developed an autonomous combine harvester [Pilarski 1999] and in separate work an autonomous mowing machine [Batavia 2002]. Both of these systems create coverage plans for their work area off-line then autonomously follow the generated path.

3.6.2 UGV Localisation

The localisation system within NOVA required a means to fuse position data from a GPS receiver and relative localisation equipment. Jumps in position would not be acceptable so naïve methods that rely on relative localisation with periodic GPS updates were discounted. A probabilistic method was required. Initial ideas for the fusion algorithm used within NOVA were based on a proposal for a navigation system to be used by the blind [Ceranka 2003]. This work looked at the possibility of combining GPS location estimates with those from a dead reckoning system using a particle filter.

3.6.3 Obstacle Avoidance

At the beginning of the research it was recognised that for a UGV to navigate through natural terrain a 3D model of the ground would be required. Sensors that directly produce 3D images are normally high cost. To that end it was hoped that an existing 2D laser range finder could be used to generate a 3D model if it was swept across the terrain. A similar device had been used in a push broom configuration on the Nomad rover constructed by CMU [Moorhead 1999]. This vehicle was developed to search for meteorites in Antarctica and had successfully demonstrated obstacle avoidance using the laser range finder.

At a later stage in the research 3D time of flight range cameras were investigated as an alternative means to sense the terrain. Both the Swiss Federal Institute of Technology [Weingarten 2004] and NIST [Bostelman 2005] evaluated these devices on indoor mobile robots. Early reports from the use of these cameras suggested that they could be a viable low cost alternative to push broom scanning.

The means used to store the model of the terrain is crucial to efficient obstacle detection and path planning. As such two different obstacle mapping systems were devised for NOVA. First a 2D system was implemented to provide a simple yet robust mapping capability. This was devised as a foundation for later work. Next a more complex 3D mapping system was developed that could account for the UGV pose when mapping the terrain. These two systems are detailed in Chapter 7.

The 2D obstacle mapping system in NOVA is based on the concept of a vehicle centred local confidence map. It uses sensor models similar to those from the well known histogram grid [Borenstein 1991]. Range data from the sensors is incorporated into the map by updating obstacle confidence values. Obstacle confidence increases for cells in which range data falls. Confidence decreases for cells between the range data and sensor.

For 3D terrain mapping a voxel based local obstacle mapping system was chosen for NOVA. This allowed the terrain to be modelled as a sequence of grid cells each holding a summary of the points that have fallen in the cell. Inspiration for this came from the early work on the CMU Navlab vehicle [Hebert 1998] and the initial research by Dornier for PRIMUS [Lux 1991].

Global path planning is not performed by NOVA instead it is expected that the operator will provide a waypoint based route to guide the UGV to a goal. NOVA does contain a local path planner to plan paths to the edge of the local obstacle map. The implemented planner is similar to the iterative linear path method used by MDARS-E for obstacle avoidance [Kurtz 1997] but has been radically extended. NOVA combines this with a polar histogram to make path choices. This histogram is closely related to the polar obstacle density histogram used by the VFH local path planner [Borenstein 1991].

From the beginning of this research it was seen as essential for NOVA to bound the autonomous motion of the UGV. Upon seeing a wide obstacle a UGV may take evasive action that follows the obstacle away from the goal waypoint. To this end the concept of an allowed corridor of operation has been developed within NOVA. This was inspired by the waypoint corridor segments specified during the DARPA Grand Challenge [Thurn 2006].

NOVA generates phantom obstacles to realise corridors of operation and other bounds on where the UGV may travel. These obstacles are placed on the obstacle map and are treated by the path planner like any other obstacle. A similar idea was used on the CMU Navlab II vehicle. GANESHA the local map management module for Navlab II placed artificial obstacles at the extremes of the sensor field of view to prevent the vehicle from turning into unobserved space [Langer 1994].

A final objective for NOVA was to include a decision recording system that may allow a UGV to escape from obstacles that form a cul-de-sac. Literature from a number of other research projects showed that this is a common failing of present UGV systems. A simplistic method to overcome this problem by expanding obstacles on a cost map was demonstrated as part of the CMU Navlab II system [Stentz 1995]. NOVA includes a novel approach to this problem based on the polar histogram used by the local path planner.

4 ARCHITECTURAL DEVELOPMENT

During this research the Nottingham Off-road Vehicle Architecture (NOVA) was developed as a modular control architecture for a UGV. A fully functional UGV has also been constructed to evaluate the capabilities of NOVA. The intended role of this UGV is autonomous route proving in natural terrain.

This chapter begins by describing the direction that was followed during the research. An overview of the research is provided here with subsequent chapters adding more detail to key aspects of the implementation. The novelty of the research is also highlighted in this chapter and compared to the UGV research reviewed in the previous chapter.

The remainder of this chapter looks at the development of NOVA. First the overall operational concept for the route proving UGV is presented. This concept has guided the development of NOVA. Following this the framework of NOVA is described. The description includes how NOVA realises the operational concept for the UGV. Implementation details for NOVA are then discussed and the modules within NOVA are detailed.

4.1 RESEARCH OVERVIEW

The overall aim of this research was to construct a UGV with a set of autonomous behaviours that could be used for the task of route proving. Building on a foundation of existing literature an appropriate UGV control architecture has been devised. This architecture is known as the Nottingham Off-road Vehicle Architecture (NOVA).

NOVA includes modules for the essential tasks of UGV localisation, obstacle detection and navigation while avoiding obstacles. Special behaviours for the route proving task have also

been implemented. To facilitate those behaviours a novel route and decision recording process has been created.

An experimental UGV platform was developed to validate the new control architecture. This UGV is known as the Autonomous Route Proving (ARP) vehicle. It is based on a tracked vehicle platform with skid steering. Sensors for localisation and perception have been fitted to the ARP vehicle. In addition computing hardware has been added to interface with the sensors and execute NOVA.

The ARP vehicle features a GPS receiver for absolute localisation. A complementary relative localisation system is also used. This is based on wheel encoders and an electronic pose sensor. Fusion of the GPS and relative position data allows absolute UGV position estimates to be generated with smaller error than GPS can provide alone. The localisation system within NOVA uses a particle filter algorithm to perform the data fusion probabilistically.

Several perception sensors have been trialled on the ARP vehicle. A single axis scanning laser range finder was made available for the project. This has been tested in horizontal and push broom configurations. In the push broom configuration the scanner sweeps the ground in front of the UGV generating 3D cross sections of the terrain. A prototype time of flight range camera has also been tested as an alternative means to perceive the terrain. This camera can directly generate 3D range images.

The range data produced by the perception sensors is used by NOVA to detect and avoid obstacles. All new sensor data is first transformed into UGV centred coordinates. Next this data is added to a local obstacle map. Traversability analysis is then performed on the obstacle map to determine which regions of the map contain untraversable obstacles. Finally a

summary of the obstacle map is used by a local path planner to determine the best path for the UGV to follow.

The ARP vehicle has allowed the design of NOVA to be thoroughly evaluated. It has also acted as a platform for experiments to be conducted. The majority of the experiments have been aimed at demonstrating the novel parts of NOVA. Some experiments have compared alternate approaches for the operation of the NOVA modules.

4.2 RESEARCH NOVELTY

In Chapter 3 sources of inspiration for this research were identified. These represent ideas that have been successfully used by other researchers in the development of unmanned vehicles. It was described how this literature had been used as a basis for the development of NOVA and the ARP vehicle.

There are three major aspects of novelty to the work described in this thesis. The first is the route and decision recording system devised to support the autonomous route proving behaviours within NOVA. Next is the local path planner which features an adjustable lookahead. When obstacles are encountered the lookahead of the path planner can be reduced to ensure all alternate paths are detected. The final novelty comes from the demonstration of a particle filter to fuse localisation data from a GPS receiver and relative localisation equipment.

4.2.1 Route and Decision Recording System

When the mission plan for a UGV does not describe a benign route the UGV needs to be able to find a drivable path to the objective. The UGV must rely solely on sensors and obstacle mapping to assess terrain traversability. It will be possible to trap the UGV with obstacle arrangements that form a cul-de-sac deeper than the maximum range of the sensors on the UGV. As discussed in section 3.5.5 various authors have suggested methods to escape this

kind of obstacle. These methods are not always that effective because they do not actually consider the path options for the UGV.

A more intuitive way to view the route exploration task would be as a topological representation. Assume nodes in a graph represent the points where UGV path choices are made. Branches in the graph would represent the available route choices. Now suppose the UGV was approaching an obstacle. If there was clear space either side of the obstacle that would give the corresponding node two path choices that could be explored.

An objective for this research was to maintain a record of the route the UGV had traversed. For that purpose NOVA periodically records the UGV location during autonomous motion. This allows the UGV to retrace proven routes at a later time. The route recording system also provides the basis for a topological exploration graph.

Imagine each recorded location along the UGV route represents a node of an exploration graph. Alongside this the corresponding path choice made by the planner is stored for each node. Then if the UGV encounters a cul-de-sac it can retrace the route that has been proven using the nodes of the graph like close together map waypoints. The UGV would return to a node in the exploration graph with unexplored path choices. At this revisited node the UGV could then choose to follow an unexplored alternate path.

The concept of an exploration graph has been implemented within NOVA. It uses the output of a local path planner to describe the possible path choices available at each node. When the local path planner cannot find a path towards the next waypoint it will prompt NOVA to back up the UGV and explore alternative routes.

Route recording and retracing has been demonstrated in other UGV research. In the DEMO I UGV program a robotic HMMWV was developed that recorded the path it followed at 1m intervals [Murphy 1996]. On command this HMMWV could retrace the recorded route. However no literature has been encountered where route recording has been combined with decision recording.

A full description of the route and decision recording process within NOVA is given in Chapter 7. The exploration graph concept is discussed there in the context of the NOVA obstacle avoidance system. Tests of the ARP vehicle using exploration graphs to escape from a cul-de-sac are reported in Chapter 8.

4.2.2 Adjustable Path Planner Lookahead

A UGV local path planner should search for the best path segment from the current UGV location towards the next waypoint on the required route. The lookahead of the planner determines the distance in front of the UGV that is considered when deciding which way to go next. A longer lookahead implies obstacles are avoided earlier and so the UGV would stay further away from these obstacles. A shorter lookahead implies UGV can get closer to obstacles and even consider a twisty path between obstacles. Different situations require different lookahead distances.

In other UGV research the lookahead for path planners is normally fixed. For this work experiments have been performed to investigate the effect of different planner lookahead distances. A path planner with a novel adjustable lookahead feature has also been devised. This path planner adjusts lookahead based on the obstacle density around the UGV. In situations with sparse obstacles the lookahead should stay high. When operating in a confined space the lookahead will drop to allow shorter path segments to be considered and so increase the chances of finding a way through.

To control the rate at which the planner lookahead falls in the presence of obstacles the concept of claustrophobia is used. An operator configurable claustrophobia parameter sets a limit on the allowed proportion of possible UGV path segments that are blocked. If the actual proportion is too high then the lookahead is reduced until an appropriate proportion is seen. Additionally a method was devised to determine the lookahead that allows the planner to find the maximum number of alternate paths.

Chapter 7 provides a detailed description of the local path planner and the operation of the lookahead adjustment feature. The results of the investigation into the effects of lookahead distance are reported in Chapter 8. Also in Chapter 8 is a demonstration of path planner claustrophobia and use of the lookahead distance that finds the maximum number of alternate paths.

4.2.3 Particle Filter Localisation

The ARP vehicle has both relative localisation equipment and a GPS receiver to estimate the UGV position. Either can be used in isolation but have associated disadvantages. GPS position estimates are noisy and are only available if satellites are within line of sight. Errors in the position estimates from the relative localisation system accumulate as the UGV moves. On longer runs the relative localisation needs to be periodically reinitialised to place an upper bound on the growth of the position error.

A particle filter has been implemented as a means to fuse the GPS and relative localisation system. No literature has been encountered that suggests this has been tried before for UGV localisation. It is common practice to use a variant of the Kalman filter for this task. This research demonstrates that a particle filter is a viable alternative that is robust to the errors associated with the underlying localisation equipment.

A particle filter was considered ahead of the Kalman filter as it is a more intuitive idea. The particle filter represents the probable state of the system as a series of samples from the underlying probability distribution. It makes no assumptions about the predictive and update models of the filter being linear as is the case for the Kalman filter. In fact the control inputs for the filter used in this research are trigonometric functions so a linear approximation would have to be made.

The equipment used for the localisation of the ARP vehicle is introduced in Chapter 5 of this thesis. Details of the particle filter algorithm are given in Chapter 6 which discusses the UGV localisation system. A demonstration of the particle filter fusing position estimates from the GPS receiver and relative localisation system is presented in Chapter 8.

4.3 CONCEPT OF OPERATION

In this section an overall concept of operation for the route proving UGV is presented. This was devised at the outset of this research and has been maintained throughout. The concept has served as the basis for developing NOVA.

Route proving is a slow and hazardous task. Military and humanitarian forces often need to prove routes when they are working in areas affected by military conflicts. The task ensures paths that must be followed with essential equipment or medical supplies are clear, free of traps and traversable. Normally route proving is performed using vehicles that carry tools to detect or destroy traps such as landmines.

Automation of the route proving task is a desirable goal. It means that a driver would not be required inside the route proving vehicle. Instead the vehicle would be teleoperated. The remote operator would act as a supervisor for the UGV that would otherwise autonomously

prove the route. To support this idea the following features are seen as essential for a route proving UGV control architecture:

- Basic route specification by a human operator
- Autonomous obstacle detection and negotiation
- Autonomous backtracking along proven routes
- Autonomous search of geographical areas

Future route proving tasks will involve a human operator providing a sequence of waypoints for a UGV to visit in turn such that together they form a well defined route to a target location. On route to each waypoint the UGV would observe the environment and negotiate obstacles autonomously. A corridor of allowed deviation from the waypoint defined route could also be used. This would ensure that the UGV made use of the particular terrain features intended by the operator such as following a valley between two steep ridges.

The use of waypoints allows the operator to break a complex navigation task down into manageable segments and to specify a route that will guide the UGV away from any impassable obstacles that are known in advance. Figure 4.1 demonstrates the idea with a UGV that needs to cross a river to reach the target location. The UGV in the figure is likely to pick the most direct route to the target but may not be able to detect or safely cross the river. Using waypoints the UGV can be guided to a safe crossing point.

Employing a waypoint based control methodology is aimed at increasing the level of the UGV control interface. The use of positional information to describe a waypoint is the most intuitive way to develop the concept. Waypoints can then be specified from maps or reconnaissance information. Translation of positional data into motion commands would be performed by the

UGV control architecture. In this way the low level control of the UGV could be hidden from the operator.

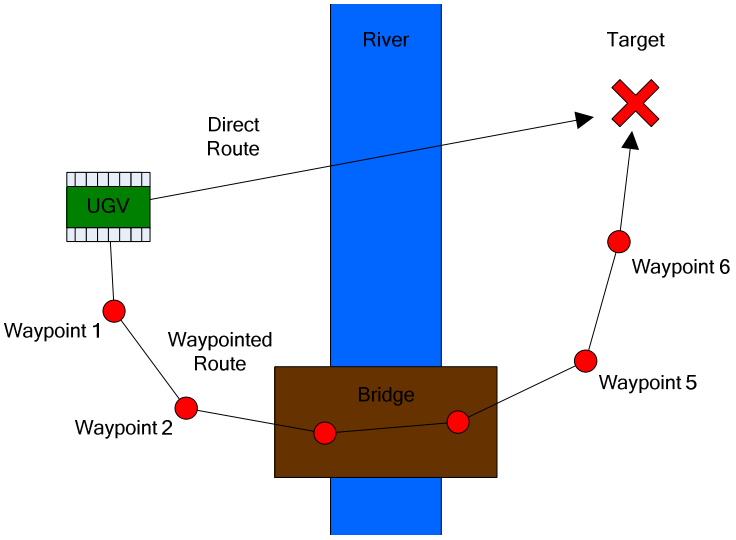

Figure 4.1 - Use of waypoints to specify a path

To provide a foundation for the more advanced functionality the UGV would maintain a record of the route that it follows during a task. Real-time positional data from the UGV localisation system is recorded. Figure 4.2 below depicts a UGV recording the path it has taken through the environment while autonomously navigating between waypoints.

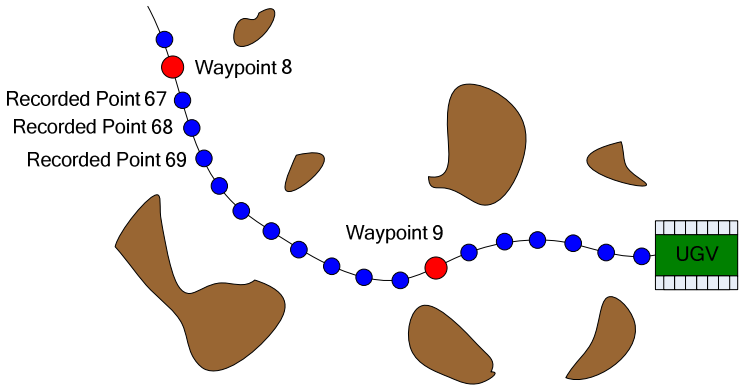

Figure 4.2 - UGV Path Recording

There are several reasons for maintaining a record of the route traversed. The most important is that it allows the route to be followed in reverse. This could be essential if the route is found to be obstructed. In applications such as landmine detection it is necessary to return along the proven route to ensure mines are not accidentally triggered by traversing unproven terrain.

The record could also be transmitted to following vehicles to describe the location of the proven route supplementing the waypoints they are using. Additional information about the route could then be added by the following vehicles if they were equipped with different sensors or new obstacles were detected as they traversed the route. Finally the record would provide accurate positional information about the proven route for battlefield mapping purposes. These peripheral facilities were not developed in this research.

Autonomous backtracking is a natural extension to this combined route proving and recording system. This would allow the UGV to return through an unknown environment along the proven route. The recorded position data would be used as a series of close together waypoints to make the UGV closely track the previously traversed path in reverse.

Together the route proving and backtracking behaviours would be used by the UGV to prove swaths through the environment. Once a swath has been proven it may be desirable to widen this safe route for use by larger vehicles or to search an entire area using distinct swaths. Hence the logical extension of this autonomous swathing behaviour is to perform multiple parallel swaths of a territory using the first pass as a guide.

Ultimately the UGV should be capable of searching an area specified by a human operator. A coverage plan of the area would be generated by the UGV. This plan would describe a route that led the UGV to prove the entire area. The behaviour would make the UGV suitable for

proving strategic areas of ground such as airfields. Use of overlapping swaths in this role may be advantageous to ensure all ground is covered several times.

NOVA embodies the concept of operation described above. During autonomous driving the UGV route is recorded as a list of locations. Manually driven routes may also be recorded if the operator desires. Autonomous backtracking is performed on demand using the recorded route as waypoints for the UGV to follow.

A parallel swath to the left or right of a proven route can also be generated by NOVA. The recorded route is used as a basis for the parallel swath. Each point in the recorded route is translated laterally to form the swath path. In this way the new swath effectively widens the proven route. The UGV can then be made to drive autonomously along the new swath. This is accomplished by making the UGV follow the points in the generated swath like waypoints.

If the operator specifies a set of waypoints that form a closed path a coverage plan of the interior area can be generated by NOVA. A coverage plan consists of a sequence of waypoints that will guide the UGV to search the entire area. The plan will cover the search area using parallel swaths. It is possible for the operator to specify the required overlap between adjacent swaths.

4.4 NOTTINGHAM OFF-ROAD VEHICLE ARCHITECTURE

This section introduces the framework of the Nottingham Off-road Vehicle Architecture (NOVA). NOVA is a complete UGV control architecture constructed from a set of task specific modules. Modules have been developed to perform all essential tasks for a UGV to operate autonomously in natural terrain. These tasks include localisation, obstacle detection, obstacle mapping and path planning. In addition NOVA includes the special behaviours for autonomous route proving that were introduced in the previous section.

In the following subsections the architectural concept of NOVA is first presented. An overview is then given of the framework for NOVA and how the modules interact. Finally the operation of each module within the NOVA framework is individually described.

4.4.1 Architectural Concept

To best explain the architectural concept of NOVA it should be presented in terms of the UGV control architectures described in Chapter 2. NOVA is a hybrid of the reactive and deliberative architectural paradigms. For maximum flexibility the framework of NOVA was implemented on top of a blackboard system. While inside NOVA the operation of the UGV has been broken down into five distinct modules that perform the key tasks.

The Teleoperator Module provides a GUI that allows an operator to specify waypoints for the UGV. Using the latest sensor data the Location Module works out the present UGV position and orientation. Obstacle mapping is performed by the Protection Module. The Route Planner Module finds the best route towards the next waypoint. Finally the Motion Module is responsible for ensuring the UGV drives in the correct direction. Figure 4.3 shows how these modules relate.

The hybrid nature of NOVA is best demonstrated by the way the UGV heading is controlled. Two of the NOVA modules are involved. One is a reactive controller and the other is deliberative. Both controllers run periodically with the reactive controller executing more frequently to ensure a rapid response to heading deviations.

Within NOVA the task of following a desired heading is a purely reactive function. As soon as the UGV is seen to have deviated from the desired heading the UGV will be turned. This role is performed by the Motion Module. UGV path planning is performed less frequently and is more deliberative in nature. An obstacle map is used by the path planner to represent the state

of the world. Using this obstacle map the planner evaluates the traversability of potential paths and selects the best. This task is handled within the Route Planner Module.

At the heart of NOVA is a shared storage area known as the UGV Monitor. Each module within NOVA has access to the contents of the UGV Monitor. The modules save their output in the UGV Monitor and can also use data within the UGV Monitor as input. As such the UGV Monitor is the basis of the blackboard system.

Figure 4.3 shows a simplified version of the NOVA blackboard system. No attempt is made to show all of the data stored in the actual UGV Monitor. The complete UGV Monitor includes the group of data shown in Figure 4.3 and many more implementation specific variables.

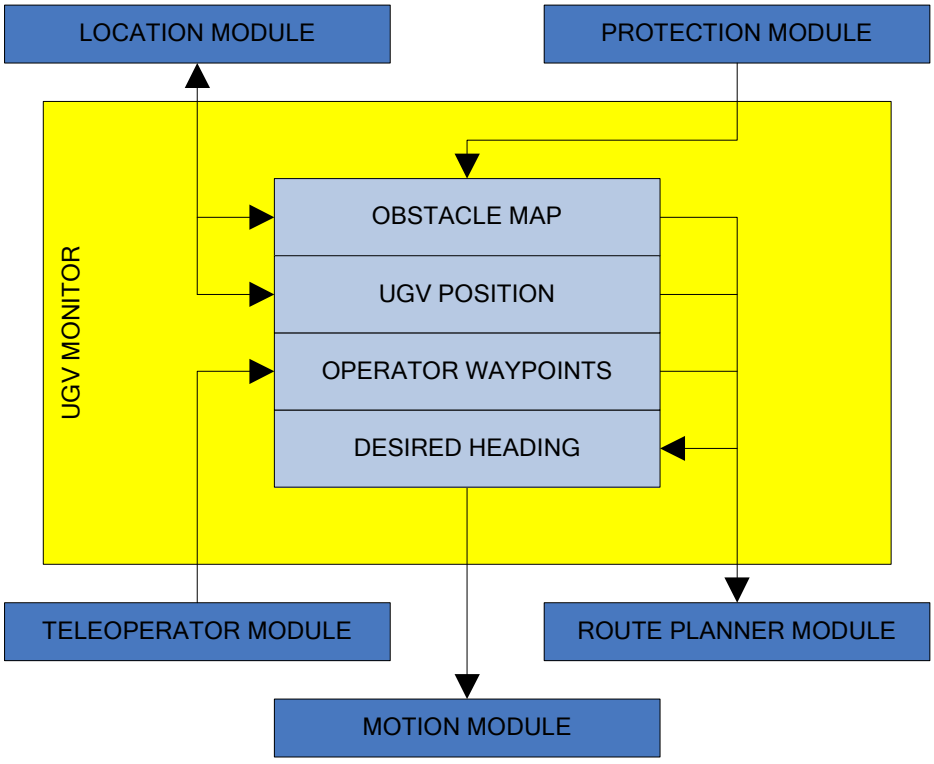

Figure 4.3 - Simplified NOVA Blackboard

Every NOVA module is responsible for processing particular data. This data may be operator commands, input from sensors or data from the UGV Monitor. Their output may be stored in the UGV Monitor and used subsequently as input to other modules. Alternatively a module output may be used to directly control some part of the UGV.

If an operator specified a set of waypoints for the UGV to follow they would be stored in the UGV Monitor. The Location Module also maintains a record of the current UGV position in the UGV Monitor. Using the stored operator waypoints and UGV position the Route Planner Module can calculate a desired heading for UGV motion. The Motion Module then ensures that the UGV follows the desired heading.

When the obstacle avoidance function of NOVA is enabled a local obstacle map is also maintained. This obstacle map is updated with perception sensor data from the Protection Module. The Route Planner Module will use the obstacle map when considering alternate headings for the UGV to ensure obstacles are avoided.

4.4.2 NOVA Framework

A different way to visualise NOVA is to consider how data flows between the task specific modules. The blackboard system provided by the UGV Monitor handles these data flows at a low level. So for the remainder of the discussion the UGV Monitor will be treated as a transparent mechanism for data exchange between modules.

Figure 4.4 illustrates the flow of data through the NOVA framework. NOVA can be considered as being essentially layered. Modules send information asynchronously to lower levels in the architecture. All modules in the NOVA framework can be thought of as executing continuously in parallel. In reality timers control when each module should execute. The modules perform their tasks periodically at intervals appropriate to the task.

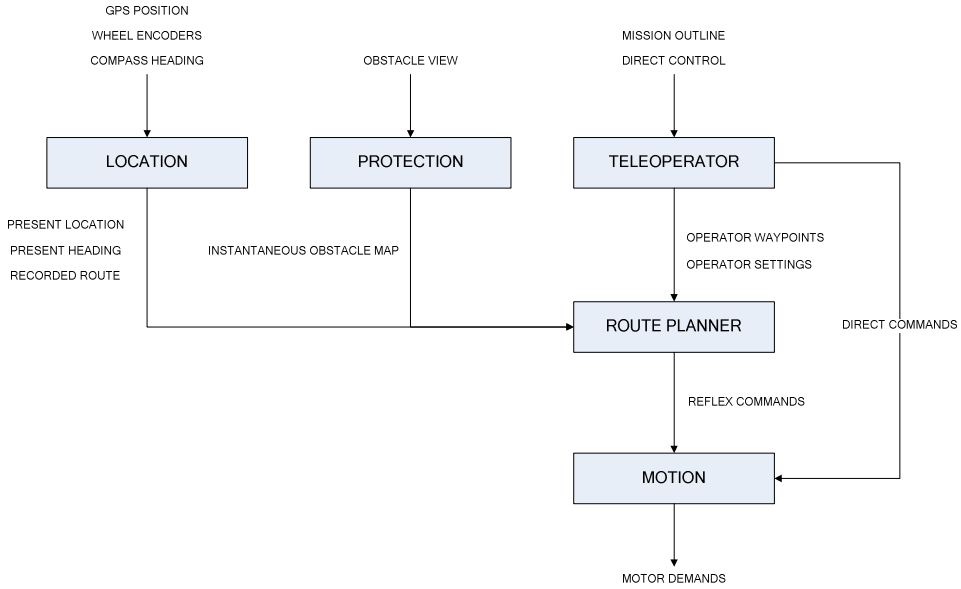

Figure 4.4 - Data flow in NOVA

The top level of the architecture processes input from the operator and sensors onboard the UGV. This level contains three modules the Teleoperator Module, Location Module and Protection Module. When the UGV is operating autonomously all these modules provide input for the Route Planner Module on the level below.

As a new UGV mission is outlined by the operator the Teleoperator Module will create a waypoint defined route for the Route Planner Module. All output from the Location Module and Protection Module is also passed to the Route Planner Module. The Location Module reports the current UGV location and heading along with a record of the route followed by the UGV. Instantaneous obstacle maps are produced by the Protection Module when new range data is received from the perception sensors.

If the UGV is driven manually the flow of data is different. The Teleoperator Module bypasses the Route Planner Module. Commands from the Teleoperator Module are passed directly to the Motion Module on the lowest level of the architecture. Any output from the Location Module, Protection Module or Route Planner Module is ignored.

The intermediate level of the architecture contains only the Route Planner Module. This module calculates the heading and distance from the current UGV position to the next operator waypoint. If the obstacle avoidance system is enabled it also checks if this ideal route is obstacle free. In the event that the route is obstructed a new path towards the waypoint is planned that avoids the detected obstacles. Heading and distance to a sub goal along this alternative path are then calculated. The output from the Route Planner Module describes the required motion of the UGV in order to reach the next waypoint or sub goal.

At the lowest level is the Motion Module that takes as input motion commands from the higher levels in the architecture. Motion commands may be either direct movement commands from the Teleoperator Module or reflex movement commands from the Route Planner Module. These correspond to demands for manually controlled or autonomous UGV motion respectively. In response the Motion Module generates control signals to drive the UGV.

4.4.3 Implementation Details

NOVA has been implemented in C++ using the Microsoft Visual Studio .NET 2003 development environment. C++ is the industry preferred object oriented programming language. Each of the five task specific modules within NOVA has been realised as an object class and the UGV Monitor has been implemented as a simple data storage class.

It was also necessary to develop a hardware interface for the UGV. This has been contained within a single communication management object known as the Serial Port Master. The object buffers all incoming data from GPS receiver, pose sensor & perception sensors. Buffered data is processed asynchronously by the Location Module and Protection Module.

As well as interfacing with the sensors the Serial Port Master also handles communication with the low level control electronics onboard the UGV. The UGV electronics are housed within a

Vehicle Control Unit (VCU). This VCU includes motor drives that control the UGV tracks. Demands for the motor drives are generated by the Motion Module and relayed by the Serial Port Master.

Additional classes have also been implemented to represent the essential structures within NOVA. These include UGV position nodes and route lists which provide the backbone for all point to point navigation performed by NOVA. An obstacle map and obstacle map voxels are other structures used by NOVA for obstacle avoidance.

A position node is a basic record of the UGV position or waypoint for the UGV to reach. The duality of the definition allows records of the UGV position to be used as waypoints. This is employed when the UGV is commanded to retrace a route that has been driven. Route lists are lists of position nodes. NOVA uses route lists to store operator specified waypoints, record the route the UGV has followed and to generate swath patterns for the route proving behaviours.

The obstacle map is a square map of the environment that is local to the UGV. In the 3D mapping system within NOVA it is made up from a 2D array of voxel objects. A voxel can be visualised as a 3D pixel. Each voxel represents a square patch of ground covered by the obstacle map. A summary of range data from the UGV perception sensors is stored in each voxel.

NOVA has been developed to be event driven. GUI events trigger the function of the Teleoperator Module. These events correspond to operator input. The data displayed on the GUI is updated periodically in line with a timer event that triggers a few times each second.

The function of the other NOVA modules is also driven by timers. Timers run for intervals appropriate for the modules to process new sensor data and stay in control of the UGV.

Executing most frequently is the Protection Module processing around 20 sensor frames each second. The Motion Module checks the UGV heading approximately 5 times a second. Other modules execute less frequently but at least twice a second.

4.4.4 Teleoperator Module

The role of the Teleoperator Module is to maintain the operator interface for NOVA. This interface is known as the NOVA GUI and is described in detail in Chapter 5. The two functions of the GUI are to display feedback from the UGV and to gather any operator input.

Raw feedback is displayed on the GUI from all of the sensors on the UGV. This includes the latest position estimate from the GPS Receiver. The current UGV pose from the electronic pose sensor is shown as well as the cumulative step counts from the UGV wheel encoders. A representation of the most recent data from the perception sensor is also provided.

In addition to the raw sensor data some processed sensor data is also made available. The present UGV position calculated by the Location Module is shown on the GUI. A view of the UGV obstacle map is also provided. This indicates regions around the UGV that are known to be traversable or contain obstacles.

Operator input for the Teleoperator Module can be split into three categories; mission outlines, direct control and system options. Mission outlines allow the operator to specify a heading or a more complex waypoint based route for the UGV to follow. Once a mission is outlined autonomous UGV behaviour can be initiated. Direct control accepts demands for control of the UGV via an onscreen joystick. Finally system options allow the operator to enable parts of the obstacle avoidance system or modify the swathing behaviours.

To ensure that all parts of NOVA know if the UGV is under manual or autonomous control the Teleoperator Module maintains a record of the current control mode. The Teleoperator Module sets the UGV control mode as MANUAL when the joystick control is used to move the UGV. When the UGV is moving autonomously the control mode will be AUTO if the movement is forward or BACKTRACK if the UGV is retracing a route. If the UGV should not be moving the control mode is set to SAFE.

The Teleoperator Module makes extensive use of the blackboard system within NOVA. All new operator input and the present UGV control mode are stored in the UGV Monitor. Additionally raw and processed sensor data is gathered from the UGV Monitor for display on the NOVA GUI. This sensor data represents the output of other modules in the NOVA framework.

4.4.5 Location Module

The UGV localisation system is contained within the Location Module. This is one of the novel aspects of this research. Only a brief introduction to the Location Module is given here as Chapter 6 discusses the operation of the UGV localisation system in detail.

The Location Module is responsible for gathering inputs from all of the localisation equipment on the UGV. That includes the GPS receiver, electronic pose sensor and wheel encoders. From these inputs the Location Module will derive the present position and heading of the UGV.

Two estimates of the UGV position are provided by the localisation equipment onboard the UGV. The GPS receiver directly provides absolute position estimates. Data from the wheel encoders and electronic pose sensor is combined by the Location Module to produce relative estimates of the UGV position.

Fusion of the GPS and relative position estimates is performed using a probabilistic method. The Location Module contains a particle filter algorithm that has been developed to derive a combined estimate of the UGV position. Relative position estimates are used as the control input for the predictive step of the filter. Position estimates from the GPS receiver are used as observations for the update step of the filter.

In addition to UGV localisation the Location Module has another role. It is responsible for keeping a record of the route followed by the UGV. The UGV route is recorded whenever the UGV moves autonomously. Manually driven routes can also be recorded if specified by the operator.

The UGV route is recorded as a list of position nodes. Each position node represents the location the UGV was at when the record was made. As the UGV drives along new position nodes are periodically added to the list. Position nodes are stored in the same way as the waypoint routes specified by the operator to outline a UGV mission. This allows the Route Planner Module to use the position nodes as waypoints to guide the UGV if a recorded route needs to be retraced.

When the UGV route is recorded the latest path planning decision from the UGV obstacle avoidance system can also be recorded with the position node. All path planning decisions are made by the Local Path Planner component within the Route Planner Module. The list of combined position and decision nodes is known as an exploration graph.

Exploration graphs are a novel aspect to this research. The concept allows the UGV to escape from cul-de-sacs encountered while autonomously following an operator specified route. They are described in detail in Chapter 7 in the context of the obstacle avoidance system.

If an obstacle free path cannot be found towards the next operator waypoint NOVA will backtrack the UGV to an earlier node in the exploration graph. Decision records allow NOVA to decide if there are unexplored paths at any of the earlier nodes in the graph. These unexplored paths may lead towards the next waypoint. When a node with an unexplored path is found NOVA will stop backtracking the UGV and explore this alternative path.

4.4.6 Protection Module

The Protection Module provides a key element of the obstacle avoidance system within NOVA. This module produces an instantaneous obstacle map for each new frame of sensor data received from the perception sensors. Only an overview of the task performed by the Protection Module is given here. A full description of the obstacle avoidance system is given in Chapter 7.

All range data from the perception sensors is processed by the Protection Module. Within the Protection Module sensor specific functions are used to process the range data. Each function uses a model of the corresponding sensor to map range data into 3D coordinates with respect to the sensor coordinate frame.

The origin of the sensor coordinate frame is at the focal point of the sensor. From here the positive x-axis points forward from the front of the sensor. Hence this axis passes through the centre of the sensor image. The positive y-axis points to the right of the sensor and the positive z-axis points up from the top of the sensor.

Range points are then transformed from the sensor coordinate frame to the UGV coordinate frame. Figure 4.5 illustrates the relationship between the two coordinate frames. Perception sensors are mounted centrally at the front of the UGV. The sensors are not necessarily mounted horizontally in fact the image axis may be pitched down towards the ground.

For the UGV the coordinate frame is fixed to the base of the vehicle. The origin is under the centre of rotation for the UGV. As with the sensor coordinate frame the positive x-axis for the UGV points forward, the positive y-axis points to the right and the positive z-axis points up. With the UGV the x-axis runs along the centre line of the UGV. The y-axis also runs along the base of the vehicle and the z-axis is normal to the UGV pose.

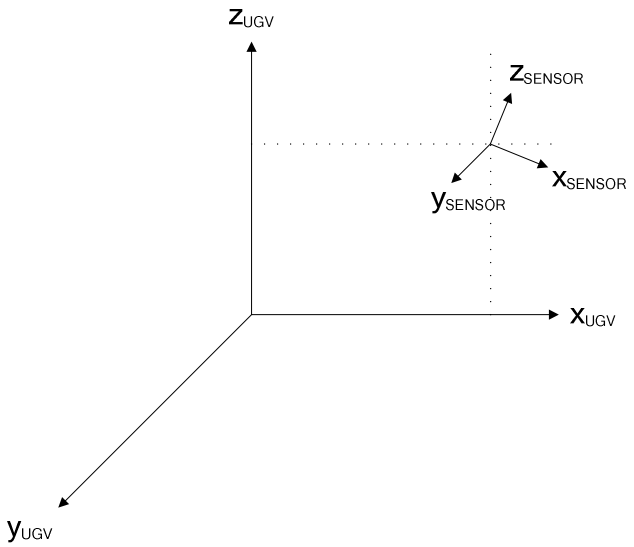

Figure 4.5 - Coordinate Frames for UGV and Perception Sensor

To transform range points from sensor coordinates to UGV coordinates the Protection Module uses a series of homogenous transformations [Kelly 1994]. First each point is rotated around the y-axis of the sensor to account for the sensor pitch. The points are then translated along the z-axis to account for the mounting height of the sensor. Finally each point is translated along the x-axis to account for the distance between the UGV centre of rotation and the sensor mount.

Next the new range points are turned into an instantaneous obstacle map. This involves another set of transformations from the UGV coordinate frame to the map coordinate frame. Map coordinates are tied to the UGV obstacle map.

The obstacle map is a grid based representation of the environment around the UGV. Each cell of the obstacle map holds information about range points generated by the perception sensors. Traversability analysis is performed on the map by the Route Planner Module when trying to plan a path for the UGV. The traversability of a cell is decided based on the range points that have fallen into that cell. A full description of obstacle mapping is given in Chapter 7.

Only a small area local to the UGV is covered by the obstacle map and this area is roughly centred on the UGV. When the UGV moves the obstacle map is scrolled but never rotated. The Protection Module is responsible for scrolling the obstacle map to follow the UGV.

The obstacle map is north oriented. At the centre of the map is the origin for the map coordinate system. For the map coordinate frame the positive x-axis points due north and the positive y-axis points due east. Figure 4.6 shows how the map and UGV coordinate frames relate.

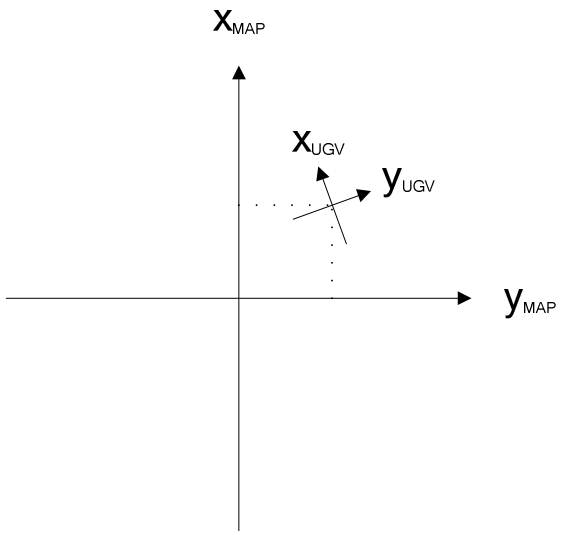

Figure 4.6 - Coordinate Frames for Obstacle Map and UGV

To transform range points between the UGV coordinate system and the map coordinate system the Protection Module must again rotate and translate all of the range points. First range points

are rotated around the x-axis and y-axis to compensate for the UGV pose with respect to the ground. A further rotation is then required to account for the UGV heading and is made round the z-axis of the UGV. The translation is required as the UGV may not be at the exact centre of the obstacle map. As before the Protection Module uses homogenous transformations to convert the range points to the map coordinate frame.

Each item of range data received from a single sensor frame is thus processed by the Protection Module. This processing transforms the raw range values into 3D points in map coordinates. Together these range points form an instantaneous obstacle map of the UGV environment. The final task performed by the Protection Module is adding the contents of this instantaneous obstacle map to the complete obstacle map.

4.4.7 Route Planner Module

When the UGV is operating autonomously the Route Planner Module is responsible for deciding which direction the UGV should drive. The obstacle avoidance function of NOVA is largely contained within the Route Planner Module. An overview of the tasks performed by the Route Planner Module is given below. Chapter 7 provides a full description of the UGV obstacle avoidance system.

The function of the Route Planner Module is split into four parts. These are known as the Offline Route Planner, Online Route Adaptor, Cartographer and the Local Path Planner. Figure 4.7 shows how the different parts interact.

All offline route planning tasks are performed by the Offline Route Planner. This includes generating parallel swaths of a recorded route and coverage plans to search an operator specified area. Another task of the Offline Route Planner is to generate route corridors. These place a limit on how far the UGV can stray from an operator specified route. A route corridor

is created by flanking the specified route with artificial obstacles. These appear on the UGV obstacle map and are treated like any other obstacle by the obstacle avoidance system.

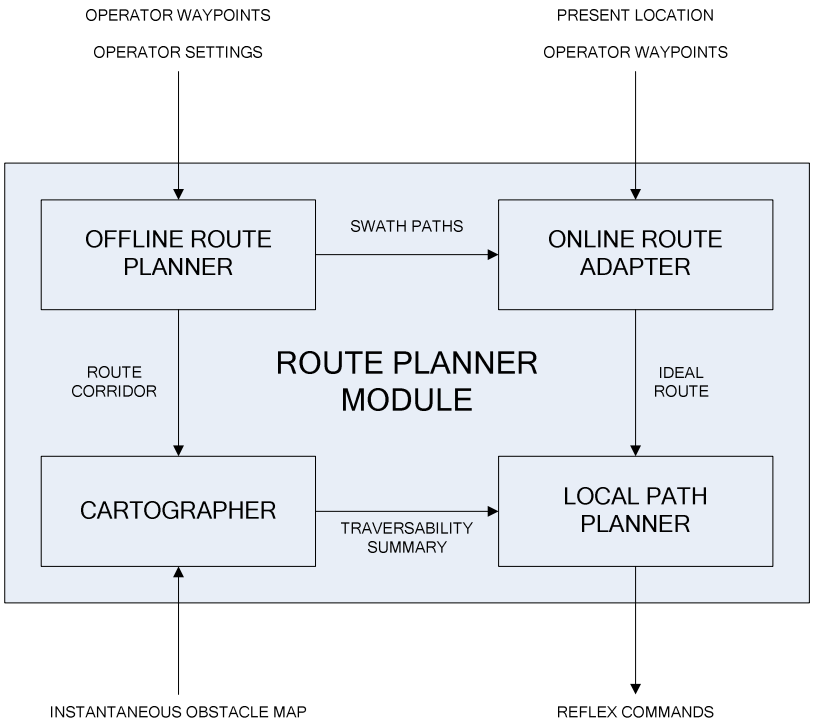

Figure 4.7 - Expansion of the Route Planner Module

Parallel swath routes are generated from the UGV route recorded by the Location Module. There are two steps in the swath generation algorithm. First the heading of the chord between the initial two points in the recorded route is calculated. Then each recorded point is translated in a direction perpendicular to this heading. The generation process is illustrated in Figure 4.8.

The size of the translation is determined from the operator options set by the Teleoperator Module. Relevant operator options are the swath width and swath overlap parameters that can be specified by the operator on the NOVA GUI. The swath width parameter indicates the width of the swath made by the UGV. It is used to account for the width of any equipment attached to the UGV such as a metal detector for landmine detection. The swath overlap

parameter specifies how much adjacent swaths should overlap. This is used to ensure an area of ground is searched thoroughly.

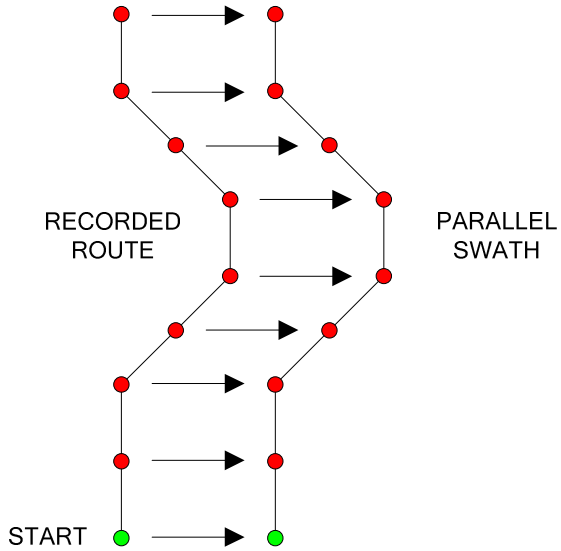

Figure 4.8 - Parallel Swath Generation

Coverage plans can be generated for sets of operator waypoints that form an enclosed route. As with parallel swaths a coverage plan is represented as a list of waypoints the UGV should follow. The idea of a coverage plan is that the UGV will be led around the perimeter of the area then cover the interior area with a series of swaths.

To perform the coverage plan the UGV will first drive the specified enclosed route. This ensures a proven circuit around the area to be covered in which the UGV can use to reposition at the end of each swath. The UGV will then be guided up and down the series of swaths turning 180° at the end of each swath to reposition for the next. Figure 4.9 shows the idea for a five waypoint enclosed route where waypoints are illustrated by red circles and the start point is a green circle.

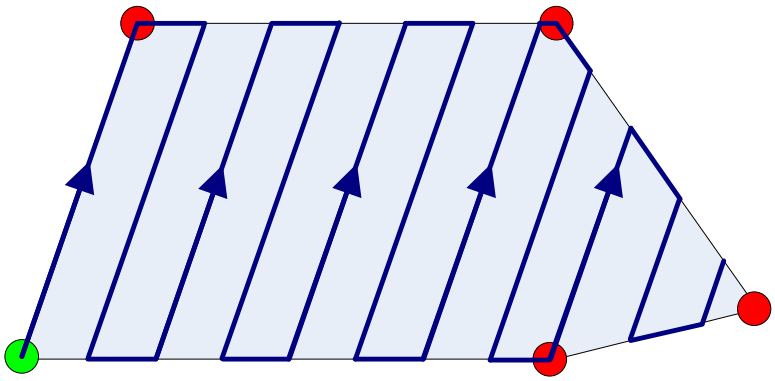

Figure 4.9 - Area Coverage Idea

Coverage plans can be generated for convex or concave enclosed areas. For concave areas if a swath is split into multiple pieces the UGV will use the perimeter circuit to travel between the separate parts of the swath. In the present implementation the coverage plan will finish all separate parts of a swath before moving to the next swath.

The process to generate a coverage plan from the enclosed operator route is quite complex. First the operator specified enclosed route is rotated about the map origin so the first waypoint leg faces due north. Next the easting of each rotated waypoint is tested to find the leftmost and rightmost extent of the route. Then starting from the leftmost waypoint swath lines running north to south are laid across the enclosed area at appropriate separation.

For each swath line a list of intersections with the chords between waypoints in the operator specified route are found. Two intersections indicate the swath line passing through the top and bottom of the enclosed area. If more intersections are found then the enclosed area must be concave and the swath will have two or more distinct parts. For split swaths the list of intersections is expanded by inserting the required waypoints the UGV must pass through to get between the end of one part of the swath and the beginning of the next. Figure 4.10 shows an example of a coverage plan passing through a waypoint in a split swath.

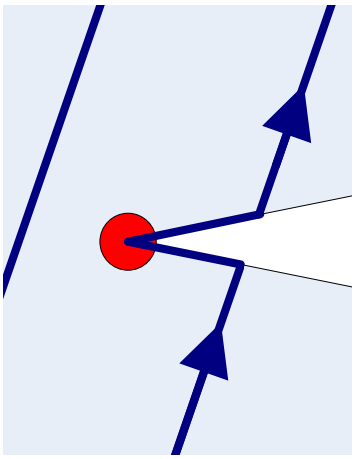

Figure 4.10 - Example of a Split Swath

Each list of intersections is then appended to the coverage plan. At this stage the direction of adjacent swaths is toggled so that the UGV will head up one swath then down the next. Once the entire enclosed area has been checked for swath line intersections all of the waypoints in the coverage plan are rotated so they are back in line with the original operator specified route.

In a coverage plan the separation of the swaths is controlled through operator options set by the Teleoperator Module. The same options affect the coverage plan and the generation of parallel swaths. So the swath separation is based on the swath width and overlap parameters that the operator specifies via the NOVA GUI.

During autonomous operation the Online Route Adaptor will calculate the ideal route for the UGV to follow. The calculated route is simply the heading and distance to the waypoint that the UGV is currently trying to reach. This calculation is performed by comparing the present location of the UGV to the location of the waypoint. The Online Route Adaptor has no knowledge of obstacles around the UGV.

Operator specified routes generated by the Teleoperator Module are stored as lists of waypoints. The Online Route Adaptor keeps track of the current waypoint that the UGV is

headed towards. Each time the UGV gets close to a waypoint the Online Route Adaptor switches attention to the next waypoint. In this way the Online Route Adaptor guides the UGV to autonomously follow the waypoint route.

The Online Route Adaptor also coordinates the UGV backtracking and swathing behaviours. When the operator wishes to backtrack along a recorded route the Online Route Adaptor will follow the recorded points instead of the operator waypoints. Following the recorded points will lead the UGV back towards the UGV start point. If a parallel swath or coverage plan is generated the Online Route Adaptor follows the points in the swath path.

The main function of the Cartographer is to perform traversability analysis on the UGV obstacle map. This produces a traversability summary of the obstacle map detailing which regions are traversable. Whenever new instantaneous obstacle maps are generated by the Protection Module the latest traversability analysis is invalidated. So the Cartographer must repeat the traversability analysis before each path planning decision is made.

An obstacle free path is selected for the UGV by the Local Path Planner. This is based on the traversability summary generated by the Cartographer. The planner will select the ideal route from the Online Route Adaptor if it is clear as far as the next waypoint or if closer the edge of the obstacle map. If the ideal route crosses an untraversable region the planner will select another path to make progress to the next waypoint.

The output from the Route Planner Module is a reflex movement command. This command describes the location of a point the UGV should reach by driving autonomously. The location is specified in terms of the heading and distance the point is from the present UGV position. This command will be used by the Motion Module to drive the UGV.

4.4.8 Motion Module

The role of the Motion Module is to control the UGV motor drives. To do this the Motion Module produces demands for the motor drives based on the latest inputs that are received from other modules in the NOVA framework. Input for the Motion Module comes in the form of commands for UGV motion. These commands can be either direct or reflex movement commands.

Direct commands come from the Teleoperator Module. They represent commands for manually controlled motion of the UGV. Under manual control the motion of the UGV is unrestricted. Both track drives may be driven either forward or backwards at any speed. Reflex commands come from the Route Planner Module. These are commands for autonomous UGV motion. During autonomous motion the Motion Module will only move the UGV in a straight line or rotate the UGV on the spot.

This restriction was made as speed control of the motor drives is difficult due to limitations of the underlying hardware. Analogue output from each wheel encoder is fed back into the corresponding motor drive. The resolution of the Analogue to Digital Converter (ADC) on the motor drives is only 8-bit. This is too low to calculate more than a coarse estimate of track speed from the wheel encoders. Thus the UGV cannot be reliably steered while moving forward or back.

To ensure the UGV does drive in straight lines during autonomous motion the track drives are trimmed. Trim controls have been provided on the NOVA GUI. There are separate trim controls for forward and reverse motion. Each control has a sliding scale. Positioning a trim control towards the left end of the scale will make the UGV turn more to the left and vice versa. Trimming is achieved by reducing the demand to one motor drive inline with the

position of the trim control. In reality the UGV does not drive in a straight line without trimming the drives.

The function of the Motion Module is illustrated in Figure 4.11. It is split into three parts that operate in series. These parts are known as the Control Arbiter, Command Interpreter and the Demand Dispatcher.

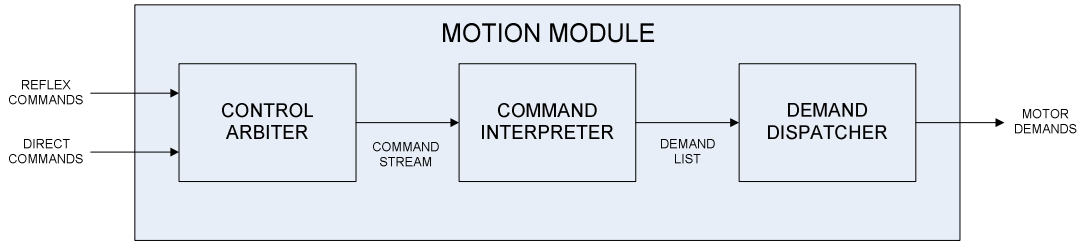

Figure 4.11 - Expansion of Motion Module

The Control Arbiter determines whether the UGV should respond to direct movement commands, reflex movement commands or do nothing. This choice is made based on the current UGV control mode as maintained by the Teleoperator Module. If the control mode is SAFE the Motion Module sets null demands for the UGV motor drives. Other control modes require use of the Command Interpreter.

Once movement commands have passed through the arbiter they are translated into a basic set of motor demands by the Command Interpreter. This has been implemented as a set of functions that interpret the different kind of motion commands.

For direct movement commands the Command Interpreter uses the position of a joystick control on the NOVA GUI to determine the direction and power of the demand for the UGV motors. If the joystick is pushed forwards both UGV tracks will move forwards at a power

proportional to the joystick displacement. Moving the joystick to one side will make the UGV turn. The further the joystick is moved the greater the rate of turn.

For example as the joystick is moved from the forward position to the right the right hand UGV track will slow down in proportion to the joystick position. When the joystick is at 45° to the forward position the right hand UGV track should stop. Further movement towards the 90° position will reverse the right hand UGV track. At 90° the UGV will be performing a neutral turn to the right. Figure 4.12 illustrates the complete joystick map.

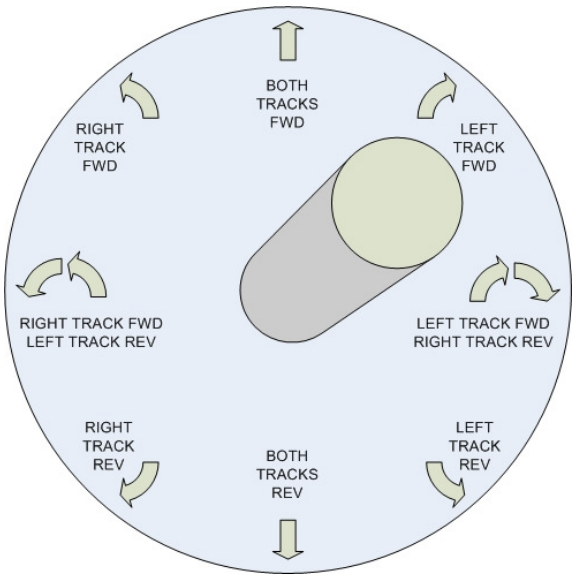

Figure 4.12 - Mapping Joystick Position to UGV Motion

For autonomous control the Route Planner Module provides motion commands in the form of a heading and distance to a point that the UGV should reach. The Command Interpreter compares the present UGV heading with desired heading. If the difference between the present and desired headings is greater than an allowed deviation the Motion Module will rotate the UGV. Otherwise the UGV will be driven in a straight line.

The direction of autonomous straight line movements can be forward or back and is determined by the UGV control mode. In AUTO control mode the UGV will move in the forward direction. The UGV will reverse if the control mode is BACKTRACK.

Autonomous UGV rotation has several steps. First all UGV motion is stopped before the motor drives are sent demands to turn the UGV to the desired heading. The Motion Module will then wait until the motor drives report completion of the turn. When the drives report back a check is made to see if the UGV heading is appropriate. If the difference between the desired heading and UGV heading is still too large the rotation process is repeated.

While the UGV is fitted with an electronic pose sensor it cannot be used during a turn to check the UGV heading. The device has a short latency during rotations that makes instantaneous readings inaccurate. When the sensor was used alone to measure turns the UGV was observed hunting for the desired heading. The nature of the turn process allows the pose sensor to be used to check the UGV heading after the turn. A short wait is required at the end of each turn to give time for the pose sensor reading to stabilise.

Rotation of the UGV can be thought of as the centre of the tracks travelling round the perimeter of a circle. As such the turn demand sent to the motor drives is based on the encoder count that should be observed by each motor drive during the turn. This encoder count is calculated using the well known formula for the length of a circular arc $\lambda = \theta r$. Where θ is the angle that the UGV needs to turn and r is the distance from the centre of either track to the centre of rotation for the UGV. The resultant arc length λ is multiplied by the number of encoder steps per metre to yield the estimated encoder count for the turn. Figure 4.13 illustrates the idea.

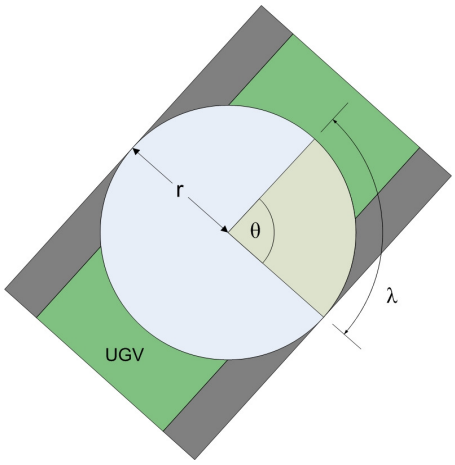

Figure 4.13 - Calculation of UGV Turn Demand

During UGV rotation the motor drives are crudely speed controlled. The encoder step counts are used to measure track speed. Meanwhile the power of the motor drive is manipulated to keep the UGV turning at a reasonable rate. A turn rate of 30 degrees per second is used as a maximum threshold. This was chosen as it is the maximum turn rate that can be reliably controlled. As mentioned above the ADC resolution of the motor drives is too low to allow proper speed control using the encoder feedback.

Bespoke software has been developed for the UGV motor drives to aid UGV rotation. The motor drives keep track of the wheel encoder count during each rotation. This count is compared against the initial turn demand to prevent overshooting. Also to ensure the reaction time of the drive system is taken into account a predictive value is added to the encoder count. This value is based on the latest encoder feedback and predicts what the final encoder count would be if the UGV stopped rotating immediately. If the prediction overshoots the initial turn demand UGV rotation is slowed to a stop.

The final stage of the Motion Module is the Demand Dispatcher. This updates the motor demands stored in the UGV Monitor with the latest values. These demands are then sent to the UGV motor drives.

5 RESEARCH PLATFORM

This chapter introduces the Autonomous Route Proving (ARP) vehicle that was developed as a platform for this research. The overall hardware design for the ARP vehicle is described first. Individual components are then identified and discussed. The remainder of this chapter looks at the control interfaces that have been developed for the ARP vehicle. Particular attention is paid to how the operational concept for this research was realised through one of the control interfaces.

5.1 DESCRIPTION OF UGV PLATFORM

5.1.1 ARP vehicle

A UGV known as ARP has been constructed as part of this research. This vehicle is based on the chassis of a bomb disposal robot. New equipment was integrated to allow monitoring of UGV location, pose and the proximity of obstacles. In addition the basic control unit was augmented with sufficient computing power to undertake autonomous navigation and obstacle avoidance. Figure 5.1 is a recent photograph of ARP.

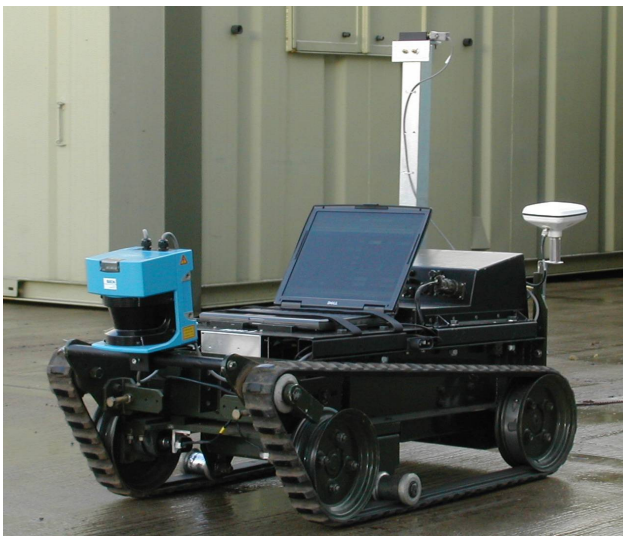

Figure 5.1 - ARP (Autonomous Route Proving) Vehicle

The ARP vehicle control system comprises two computing entities communicating through a wireless link. The NOVA Computer is fitted onboard ARP and is responsible for the control and monitoring of all other equipment on the UGV. A second remote computer is used to teleoperate the ARP vehicle. The operator interface from the NOVA Computer is replicated on the screen of the remote computer allowing full manual or autonomous control of the UGV.

Figure 5.2 shows a schematic of the hardware onboard ARP at present. The UGV platform is controlled by a microcontroller based Vehicle Control Unit (VCU). Within the VCU are electronic modules to drive the tracks of the UGV and distribute power to other onboard equipment. Management of the VCU is performed by the NOVA Computer sending demands via a serial interface.

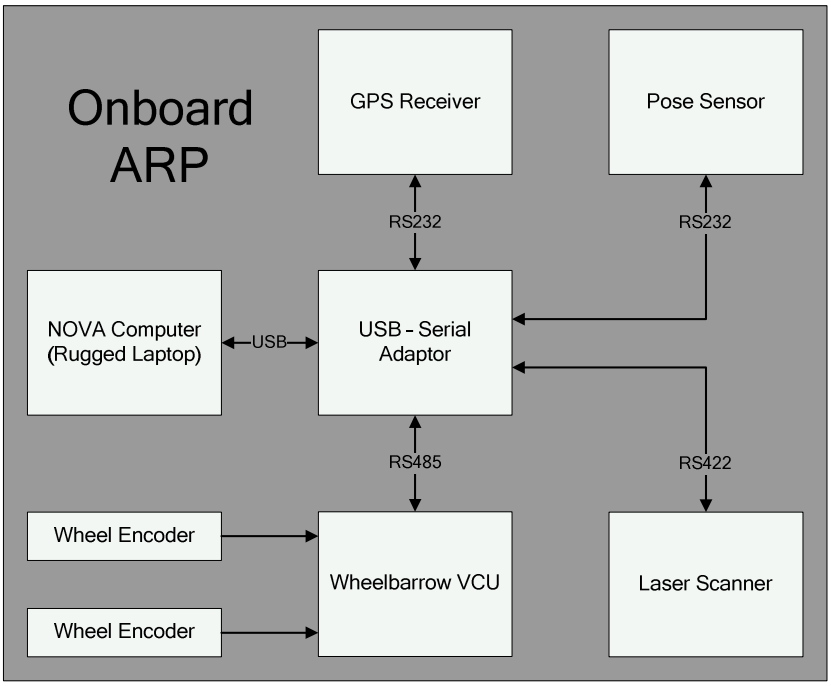

Figure 5.2 - System Schematic showing data interfaces

Absolute global localisation for the ARP vehicle is provided by a GPS receiver with sub-metre accuracy. A three axis pose sensor and wheel encoders provide an auxiliary relative means of

localisation. Obstacle detection is performed using a laser scanner with a 180° planar field of view. The data output from each of these devices is monitored by the NOVA Computer.

Serial communication between the NOVA Computer and other devices on the ARP vehicle employs a USB to serial port adaptor. This versatile adaptor has ports that can be configured to communicate using any of the RS232, RS422 or RS485 electrical interfaces. Through this adaptor the NOVA Computer communicates with the RS485 based VCU. Data from the GPS receiver and pose sensor are received from separate RS232 channels. The laser scanner streams data over an RS422 bus.

Wheel encoder data for the UGV is gathered by the NOVA Computer via the VCU. This was done as the chosen encoder devices provide an analogue feedback signal to indicate the position of the encoder. The VCU has the capacity to monitor analogue feedback from these components.

As part of the UGV construction bespoke control software has been developed for the NOVA Computer and the VCU. Two distinct items of software have been developed for the NOVA Computer the ARP Interface and NOVA GUI as discussed in section 5.4. Software for each of the electronic modules within the VCU was also developed. This was necessary to establish the communication protocol with the NOVA Computer, provide sensible low level behaviours to control the UGV tracks and process feedback from the wheel encoders.

5.1.2 Chassis

ARP is based on the robust chassis of a Mk7 Wheelbarrow Remote Control EOD (Explosive Ordinance Disposal) Vehicle provided by Remotec UK. The chassis came complete with drive train and power pack.

The Mk7 Wheelbarrow was used by the British MoD until around 1984 and has now been superseded by several new generations of the Wheelbarrow [Ryder 2006]. Design requirements for the Mk7 Wheelbarrow included the capability for driving in loose or muddy terrain and stair climbing. Hence the vehicle is equipped with compliant rubber tracks and a low centre of gravity making it suitable for use on a wide variety of terrain.

Each side of the chassis has a driven rear wheel and a free turning front wheel. A rubber track is fitted around both of these wheels and held in tension by a roller at the front of the chassis. Power is transferred to each rear wheel from a 24V DC motor via an oil filled reduction gearbox. Thus movement of the rear wheels propels the tracks. Figure 5.3 shows the idea.

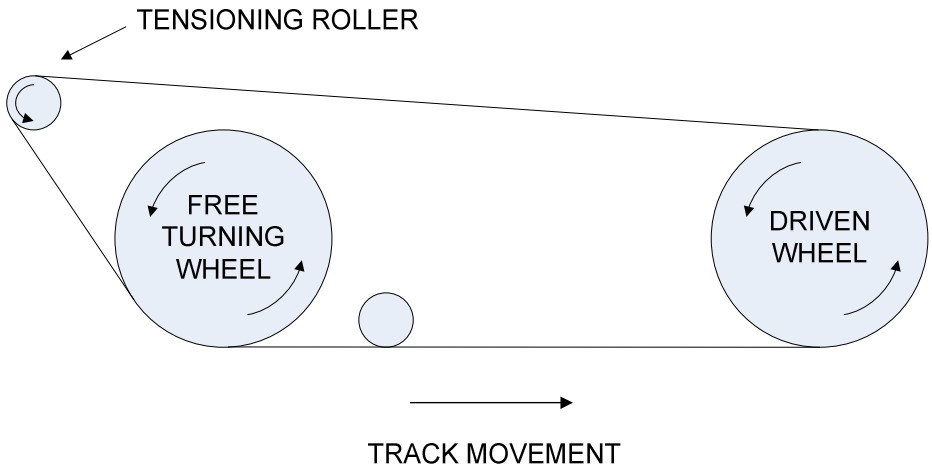

Figure 5.3 - Mk7 Wheelbarrow Track Configuration

All power for the ARP vehicle comes from a 24V 55Ah gel electrolyte battery pack fitted inside the chassis. A runtime of over 3 hours for the UGV is possible if the battery is fully charged. The battery is connected to the VCU which regulates and distributes the battery input for all other equipment. A VCU regulated 5V DC supply powers the wheel encoders and pose sensor.

Power for the UGV motors and other devices is not regulated by the VCU. A separate protected 24V DC-DC converter has been added to the UGV to power the GPS receiver and laser scanner. This is to ensure they are not affected by noise that may get onto the 24V battery supply from the VCU during operation of the drive motors.

The NOVA Computer has a built-in battery pack and does not take power from the UGV. This battery offers an uninterruptible power source for the backup of all system data if the UGV battery is discharged below the operational threshold.

5.1.3 New Control System

Originally the Mk7 Wheelbarrow featured a relay based control system. The interface between the operator control unit and vehicle used a series of analogue electrical signals. Such a system would be difficult to interface with modern digital computing equipment. In order to generate appropriate analogue control signals special hardware would be required.

A better solution was to directly upgrade the control system onboard the Wheelbarrow. Remotec had preformed initial development of a microcontroller based control system for the Remotec Super M a modern variant of the Mk7 Wheelbarrow. This new control system included a replacement Vehicle Control Unit (VCU) and Hand Controller (HC) as shown in Figure 5.4. The VCU was retro fitted to the Mk7 Wheelbarrow chassis with only minor mechanical modifications.

The VCU contains a number of electronic modules. Originally a coordination module received all data from the HC and combined it with proprioceptive information to form the final drive and actuation demands for the vehicle. Finalised demands were broadcast on an internal serial bus for the other modules to receive, act upon and respond if necessary. Control of the Wheelbarrow was thus distributed with all modules performing different tasks simultaneously.

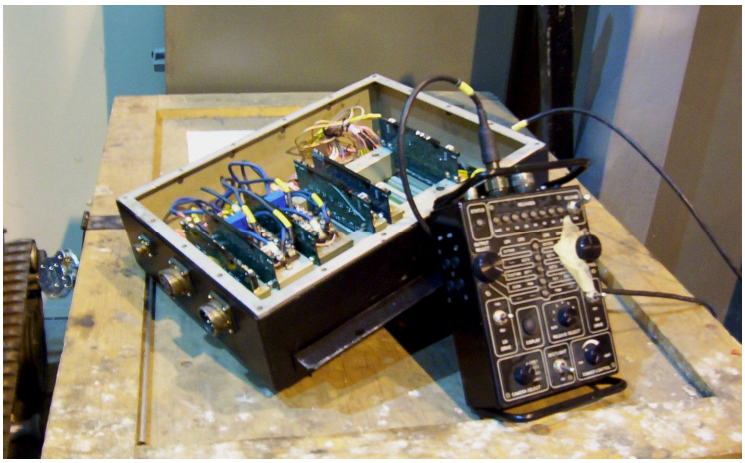

Figure 5.4 - Prototype VCU and HC for the Remotec Super M

Initially within the prototype VCU a module with an 8-bit microcontroller was responsible for coordinating the behaviour of the vehicle. It was clear that this microcontroller had neither the speed of execution nor memory capacity to perform the tasks required for this project. The obvious upgrade path was to replace this control module with a much more powerful computer.

5.1.4 NOVA Computer

The NOVA Computer runs the main control application for the ARP vehicle and as such provides a local means of control for the ARP vehicle. Tasks performed by this computer include gathering data from the GPS receiver, pose sensor & laser scanner as well as issuing commands to modules within the VCU. Figure 5.5 shows the present NOVA Computer mounted on ARP.

Use of a laptop computer rather than a headless embedded computer offered a number of advantages. A laptop gave a convenient means to view performance information onboard the vehicle as it has an LCD screen. As a fully functioning computer the laptop was also used as the platform on which to develop and test the control software instead of a separate development computer.

Figure 5.5 - NOVA Computer

It is the trend with computers that a high specification machine today will be rapidly superseded. Hence in the event that greater computing power is required for a future expansion the system has been designed in such a way that only the NOVA Computer would need to be replaced. At present the full version of NOVA increases the CPU load of the NOVA Computer by around 15% and uses around 35MB of RAM.

The machine used as the NOVA Computer is a semi-rugged laptop. It has the following salient features:

- Intel T2400 Core Duo CPU running at 1.83 GHz
- 1GB RAM
- 60GB gel cushioned hard disk drive

Any equipment attached to the Wheelbarrow Chassis will be subject to shock and vibration during basic manoeuvres and this is amplified if the terrain is not smooth and flat. As such a

key feature of the NOVA Computer is the gel cushioned hard disk drive that prevents errors while the UGV is in use. An alternative would be to use a large capacity solid state storage device.

A mount for the NOVA Computer has been created at the front of the vehicle above the UGV battery. The computer sits flat on an aluminium deck flanked by a series of small wedges that prevent lateral movement. Straps are used to restrain the NOVA Computer from vertical movement. These straps are adjustable so the NOVA Computer can be secured with either the screen open or closed.

5.1.5 USB to Serial Adaptor

Figure 5.6 shows the Sealevel SeaPORT+4 USB to serial adaptor that is used onboard the ARP vehicle. This device provides the NOVA Computer with sufficient serial ports to asynchronously communicate with the GPS receiver, pose sensor, laser scanner and VCU. These four extra serial ports are controlled as if they were native to the NOVA Computer.

Figure 5.6 - Sealevel SeaPORT+4

The SeaPORT+4 has a series of switches to select the characteristics required for each of the four serial ports separately. Each port can be configured to send and receive asynchronous data

supporting the RS232, RS422 or RS485 electrical interface with data transfer rates up to 921.6 kbaud.

500 kbaud is the highest data rate that needs to be supported for real-time RS422 data from the laser scanner. Compare this with the 12Mbps maximum transfer rate of a typical USB port supporting the USB1.1 standard. Hence USB has a much greater capacity than is required by each of the serial ports. The serial port adaptor works reliably and has very low latency.

A separate power supply was not required for the SeaPORT+4 as it takes power from the NOVA Computer via the USB port. The USB to serial adaptor is housed within the VCU alongside the GPS receiver to make it easy to separate the chassis and control system for the purposes of bench testing.

5.1.6 VCU

With the introduction of the NOVA Computer it was clear that the operation of the VCU must change. All of the modules within the VCU are connected by an internal serial bus. This is a RS485 communication network. The NOVA Computer has been made the master of this bus network with all other modules as slaves. Figure 5.7 illustrates the communication bus from the NOVA Computer and within the VCU.

Across the internal serial bus the NOVA Computer sends demand messages and requests for feedback to the modules within the VCU. The serial communication protocol used was developed especially for this project. Messages from the NOVA Computer are always composed of four words sent with a gap of approximately 250μs between adjacent words.

In every message the first word is an address word with the second word being an exact duplicate. The address words allow the VCU modules to pick out the start of a new message

and indicate which modules need to act upon the message. These are followed by a sub-address word and then a data word. The sub-address word indicates what type of message is being sent, either drive demand or feedback request. Finally the data word encodes the actual demand or feedback request. Each word in a message comprises of 10 bits. The start bit is a '0', this is followed by an 8-bit information segment then a single stop bit with value '1'.

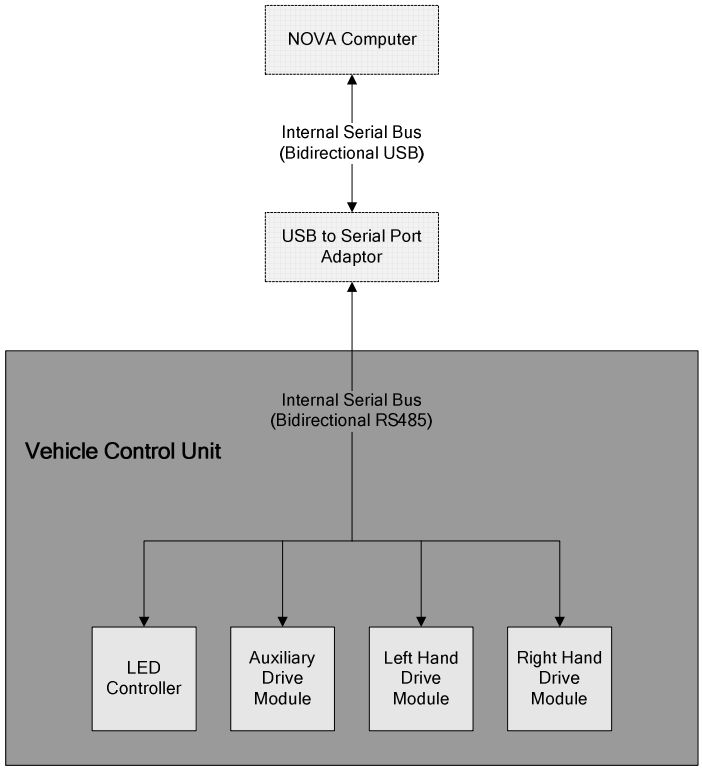

Figure 5.7 - Communication Bus within VCU

Within the VCU is a LED Controller, one Auxiliary Drive Module and a pair of Drive Modules. The primary role of the LED Controller is to control the battery status LED and communication status LED on the outside of the VCU. These lamps indicate the state of the UGV battery and if valid control data is being received from the NOVA Computer. Two additional LED drives are also used as diagnostic indicators. At present these indicators report the quality of the GPS receiver lock and whether the vehicle is avoiding an obstacle.

A 24V bus to power the obstacle detection sensor and GPS receiver is provided by the Auxiliary Drive Module. The original role for this module would have been to drive the various motors and actuators onboard the front deck of a Wheelbarrow.

Two Drive Modules are also fitted within the VCU and these are used to drive the tracks of the UGV. Their function has been modified to ensure feedback from the wheel encoders can be reported to the NOVA Computer. They receive motion demands from the NOVA Computer. An onboard microcontroller then provides inputs for a single high power H-bridge bi-directional motor driver. Speed control of the motor drive is achieved by pulse width modulation.

As a safety feature all modules within the VCU are taken to a safe state if no messages are received from the NOVA Computer for 0.5s. In this situation all drives are ramped down to null power and the LED Controller flashes all lamps red to indicate the loss of communication.

5.1.7 Base Station

A remote computer has been employed as a base station for teleoperation of the ARP vehicle. Via the remote computer the operator can use the ARP Interface or NOVA GUI to control the UGV. At present a rugged laptop is being used as the remote computer since it is portable and can be used outdoors. Specifications of the remote computer are not as critical to the operation of the UGV as for the NOVA Computer. All real-time UGV control is performed onboard the ARP vehicle. The remote computer only displays the control interface.

Communication between the NOVA Computer and remote base station computer uses an IEEE 802.11b/g Wireless Ethernet connection. This technology is now ubiquitous for wireless networking worldwide and operates in the licence free 2.4GHz radio band. A secure ad-hoc wireless network has been set up between the two computers. Data rates of up to 54Mbps are

theoretically possible with this network. A range of up to 500m at a reduced rate of at a rate of 1Mbps is also claimed. In practice the link has sufficient data rate and range for the vehicle test environment.

Dedicated software is not used to teleoperate the UGV. Rather a remote desktop connection is established to the NOVA Computer. This is a feature of the Windows XP operating system. It allows a client computer to run an application on a networked host machine displaying the user interface and responding to input as if the application was running directly on the client machine. Additional threads of execution & communication for this feature are managed by the operating system.

5.2 DESCRIPTION OF UGV PERCEPTION EQUIPMENT

During this research two sensors have been trialled for obstacle detection onboard the ARP vehicle. Both of these sensors are used to observe the ground around the UGV and the generated range data is transformed into 3D coordinates with respect to the vehicle. From this data a 3D point cloud model of the terrain can be constructed. Different techniques are used to generate the point cloud model using the two trialled sensors. For one sensor the motion of the UGV is used to build up the model as the sensor is swept along the ground. The other sensor is capable of generating 3D range data directly. In the following subsections the two sensors are introduced.

5.2.1 Laser Scanner

The ARP vehicle has been equipped with a Sick LMS 200 scanning laser range finder. This sensor is in common use by the robotics community and a study has been performed to characterise the performance of the device [Ye 2002]. On both of the previous robots constructed by the Robotics Research Group a Sick LMS 200 was used for obstacle detection. Figure 5.8 shows the Sick LMS 200 fitted to the front of the ARP vehicle.

Figure 5.8 - SICK LMS 200

A significant feature of this sensor is that it provides 180° coverage in front of the robot via a single axis laser scan. The beam of the laser is pulsed and distance is measured by time of flight. Within the laser scanner a rotating mirror deflects the laser pulses to systematically scan the whole region surrounding the device. Figure 5.9 shows an example scan.

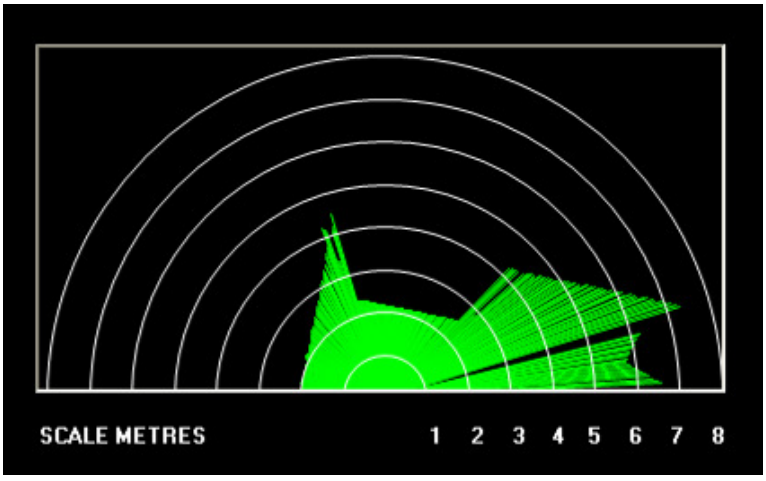

Figure 5.9 - Example Scan from SICK LMS 200

Output from the scanner is a sequence of range values representing the distance and angle to the reflecting obstacle. Angular resolution is presently set at 0.5° which allows new scans at a

rate of 37.5Hz. While the maximum measurement range of the device is adjustable in steps from around 8m to over 80m. Currently the scanner is configured for data communication using the RS422 electrical standard. This allows the scanner to stream data at the maximum rate of 500kbaud.

On the earlier Robotics Research Group robots the scanner was mounted at the front of the chassis so the laser scanned horizontally. In this configuration the Sick LMS 200 is ideal for use indoors or outdoors on flat ground. By mounting the sensor at the front of the robot it ensures that most positive obstacles the robot encounters whilst travelling forward or turning will be detected. Negative obstacles will be overlooked as demonstrated in Figure 5.10.

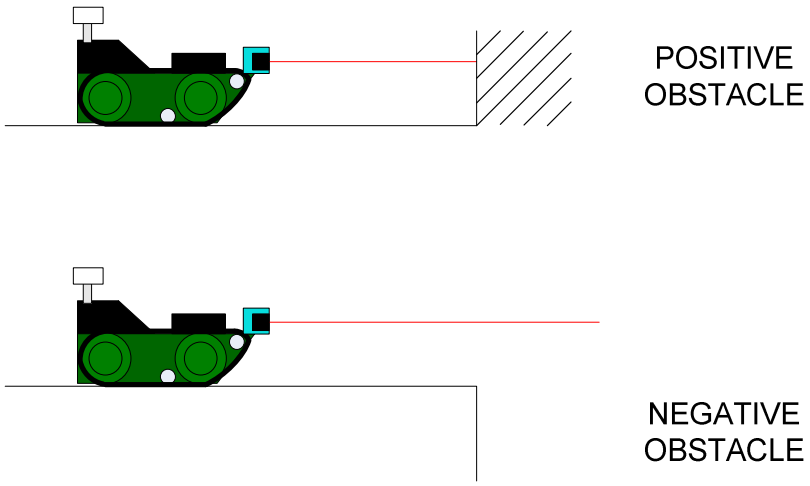

Figure 5.10 - Obstacles with Horizontal Scanner Configuration

For this research effort another mounting configuration was required to allow the ground surface to be analysed before the UGV attempts to drive across. The scanner was mounted at the front of the ARP vehicle with an adjustable bracket that allows the scanner pitch to be easily altered. In this way the scanner can be set to sweep across the ground in front of the UGV to detect positive and negative obstacles. Figure 5.11 illustrates the idea.

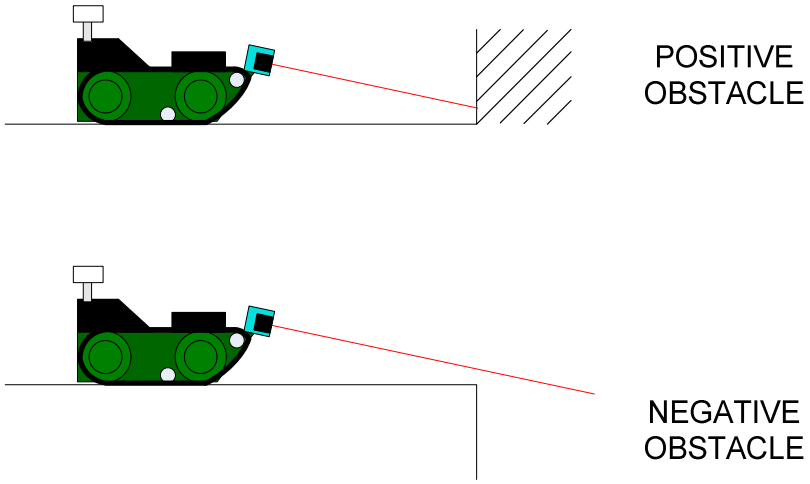

Figure 5.11 - Obstacles with Angled Scanner Configuration

By orienting the laser scanner like this a straight line is scanned across the ground in front of the UGV. The distance of the line in front of the UGV is dictated by the scanner pitch. As the UGV moves forward this divides the space into three categories, unobserved, currently observed and previously observed. The currently observed space lies within the field of view of the scanner. Lying behind the scanner field of view is the previously observed space that has been scanned as the UGV moved forward. All other space is unobserved.

Range points produced by the laser scanner are converted into 3D coordinates using homogenous transformations [Kelly 1994]. This method can be used to build a high resolution 3D picture of the terrain but does have some disadvantages. The most important is that the UGV needs to move forward to analyse the ground. Also overhanging obstacles may not seen be until just before a collision.

Several solutions exist for these problems. More sensors could be added to the ARP vehicle with different fields of view. The present scanner could be made into a 3D sensor. Alternatively a 3D sensor could be added to the ARP vehicle in place of the laser scanner.

As mentioned in Chapter 3 a number of low cost 3D imaging schemes have been trialled by researchers at the University of Hanover [Wulf 2003]. One method used a single axis laser scanner mounted on a tilting mechanism that rapidly adjusted the pitch of the sensor. Another scheme was to mount the single axis scanner on a spinning shaft. Either of these methods could be replicated relatively cheaply using off the shelf components.

While the construction of a 3D laser scanner from the existing 2D scanner was an attractive project it was not pursued in this research. Instead a 3D range sensor was purchased on the premise that a comparison with the angled laser scanner could be made.

5.2.2 3D Time of Flight Range Camera

Affordable 3D time of flight sensors are available from PMD Technologies GmbH [Möller 2005] and CSEM (Centre Suisse d'Electronique et de Microtechnique SA) [Oggier 2004]. Both organisations produce a range of prototype cameras that provide greyscale images and associated 3D data.

After discussion with both manufacturers the decision was taken to purchase the PMD [vision]® 1k-S over a CSEM Swiss Ranger. This was because only PMD Technologies GmbH claimed their product was suitable for use outdoors because of built in technology for suppression of background illumination. Successful outdoor experiments have been performed in fully overcast conditions using the CSEM Swiss Ranger but CSEM report that this camera is only reliable with indoor lighting conditions [Bostelman 2005].

The PMD [vision]® 1k-S is shown in Figure 5.12 fitted to the ARP vehicle in place of the laser scanner. This camera produces 64x16-pixel greyscale video images with associated depth values for each pixel. The standard field of view is 12° in the vertical direction and 34° horizontal but this can be altered by swapping the lens. Frame rates of up to 50Hz are possible

through a FireWire (IEEE 1394) interface. The ambiguity interval for this device is 7.5m and is determined by the illumination modulation frequency.

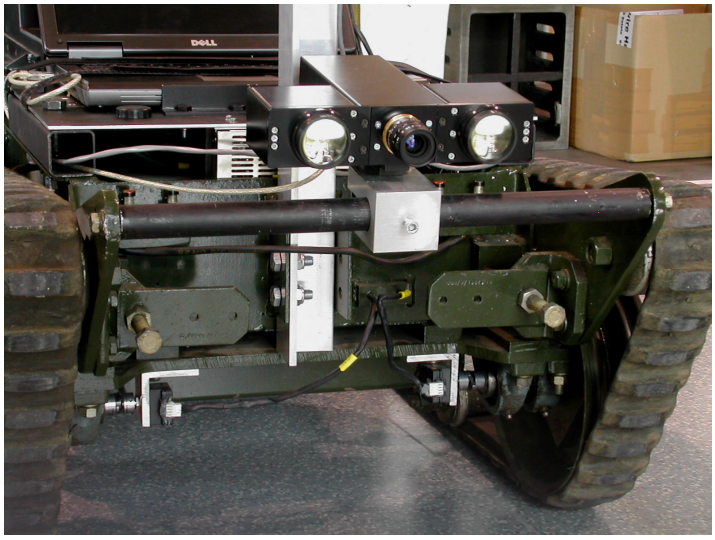

Figure 5.12 - PMD [vision]® 1k-S

Within the PMD [vision]® 1k-S the custom image sensor is sensitive to near infrared light (~870nm wavelength). Two banks of LED illuminators integrated into the camera produce light of this wavelength. This active illumination is modulated at a frequency of 20MHz. The emitted and captured illumination signals are correlated on an individual pixel basis. So the distance to the scene can be calculated from the phase difference of the reflected light. The resulting greyscale and distance image frames are made available as 1024 entry arrays.

Figure 5.13 shows a distance image frame captured from the PMD [vision]® 1k-S. To aid 3D visualisation colour indicates depth with red pixels close to the camera and blue pixels farthest away. A digital photo of the same warehouse scene is shown in Figure 5.14. Elements of the scene are clearly recognisable in the distance image which shows good detail and has very little noise.

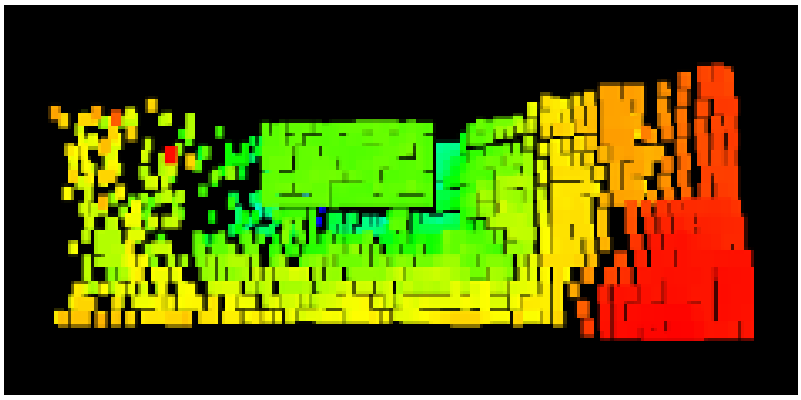

Figure 5.13 - Frame from PMD [vision]® 1k-S

Figure 5.14 - Corresponding Warehouse Scene

Communication with the camera from the NOVA Computer is simplified by the use of the PMD MiniSDK. This software development kit includes functions to setup a connection with the camera so explicit configuration of the FireWire port is not necessary. Additional functions are provided to grab a frame from the camera and then extract distance or greyscale values. All of the underlying communication details are hidden from the software developer.

For initial testing the PMD [vision]® 1k-S was mounted at the front of the ARP vehicle in the centre. It was mounted using a clamp onto a rail allowing the camera pitch to be easily

adjusted. The advantage of this location was that no part of the UGV fell in the field of view of the camera.

The camera was mounted on the ARP vehicle so the lens was pitched 6° below horizontal. This meant that the top of the field of view was approximately horizontal. At this angle the chances of receiving distance data from beyond the non-ambiguity range of the camera are much reduced as the majority of pixels will be receiving reflected infrared light from the ground. Figure 5.15 illustrates the field of view of the camera when fitted to the ARP vehicle.

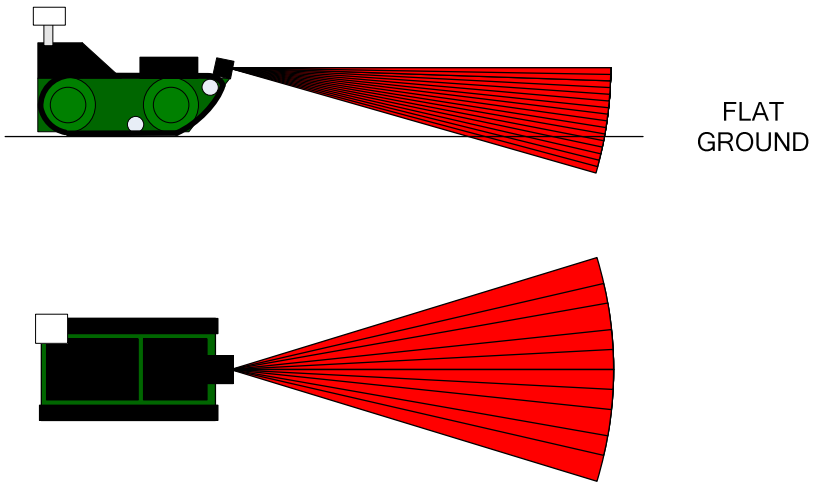

Figure 5.15 - Field of View for PMD [vision]® 1k-S

5.3 DESCRIPTION OF UGV LOCALISATION EQUIPMENT

The ARP vehicle is fitted with a GPS receiver for absolute localisation. A combination of encoders and a pose sensor are used for relative localisation. The relative localisation system is used alone when GPS signals are not available and provides a useful enhancement to GPS accuracy at all other times.

With position accuracy of around a metre for EGNOS corrected GPS it is not suitable for high precision navigation. For short straight line movements wheel encoder accuracy is much

better. So the GPS data can be used to bound any position errors accumulated by dead reckoning.

Additionally GPS receivers can only give dynamic heading derived by tracking recent position translations. Thus when the UGV turns on the spot auxiliary equipment is required to measure rotation. The pose sensor provides an independent method to detect the UGV orientation.

5.3.1 GPS Receiver

A CSI Wireless Mini MAX GPS Receiver with a MGL-3 Antenna has been fitted to the ARP vehicle [CSI 2005]. This system can provide positional updates at a rate of up to 5Hz. The receiver is pictured in Figure 5.16 removed from the vehicle. Normally it is housed within the VCU alongside the USB to Serial Port Adaptor. The antenna has been mounted on a mast at the rear of the ARP vehicle.

Figure 5.16 - CSI Wireless mini MAX GPS Receiver

The CSI Wireless Mini Max is a 12 channel GPS receiver that supports civilian GPS signals and also analyses the carrier wave to increase the accuracy of position estimates. Two of the available channels can be used to receive EGNOS differential correction data as provided by the European Space Agency. Use of EGNOS allows this receiver to provide horizontal

position estimates with sub-metre accuracy 95% of the time compared with 5m accuracy using GPS signals alone. These accuracy values assume ideal conditions.

The receiver provides the NOVA Computer with real time position information for the location of the GPS antenna. This is communicated as latitude and longitude values measured in degrees with decimal minutes. An altitude value measured in metres is also given. Additional derived data such as heading and velocity are available as well as a measure of the error associated with the calculated position.

For the present the GPS receiver has been configured to stream the NMEA 0183 standard GGA and GST messages at 5Hz using a data rate of 19200 baud. The GGA message contains detailed GPS position information including latitude, longitude and altitude. Current GPS error information is contained in each GST message.

5.3.2 Wheel Encoders

As part of the chassis upgrade a wheel encoder was coupled to the axle of the non-driven wheel on each side of the UGV. These devices have been wired to take power and return an analogue signal to the corresponding drive module within the VCU. Each drive module reports the cumulative movement of the encoder to the NOVA Computer on request. Figure 5.17 shows the right hand encoder mounted on the ARP vehicle.

The wheel encoders are Vishay Spectrol Full 360° Smart Position Sensors [Vishay 2003] donated by Remotec UK. These potentiometer like devices produce an analogue output that varies as the centre shaft is rotated with respect to the body of the sensor. The output is a saw tooth signal that climbs or drops in proportion to the rotation of the sensor shaft. A transition between 0V and 5V feedback happens at the sensor origin. Figure 5.18 illustrates the output signal.

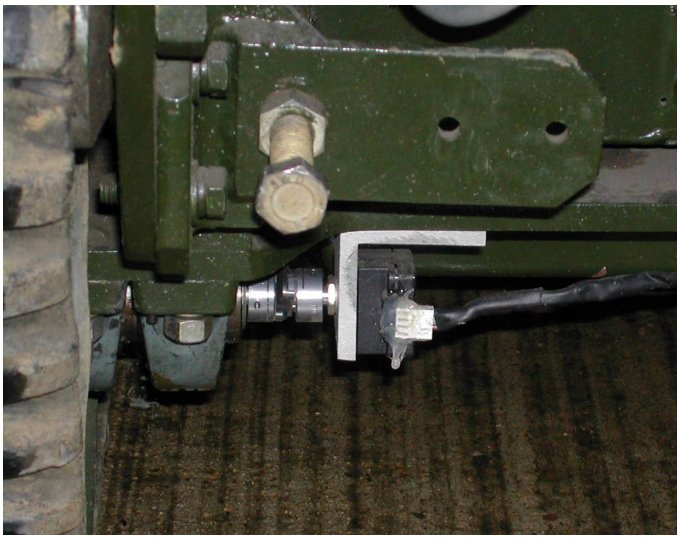

Figure 5.17 - Right Hand Encoder fitted to ARP Vehicle

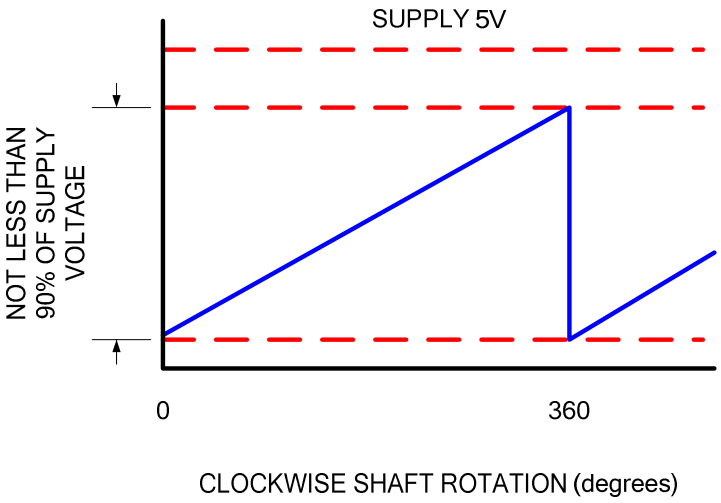

Figure 5.18 - Output from Wheel Encoder with Shaft Rotation

An analogue to digital converter (ADC) on each drive module periodically converts the encoder feedback to an 8-bit count. The drive module software compares new feedback with the last encoder position and calculates the encoder movement. A cumulative encoder count is maintained by the drive module. On request from the NOVA computer the drive module will report the cumulative value. At this time the cumulative encoder count held by the drive module is reset to zero.

From the cumulative encoder counts the NOVA Computer can calculate how far each UGV track has turned since it last requested encoder feedback. One metre of track movement is approximately equal to an encoder count of 363. It has been observed that the drive train on either side of the UGV is fairly symmetric. As such the left and right hand encoder counts do not particularly diverge if the UGV is driven in a straight line on a smooth surface. The NOVA Computer uses the average encoder count to measure distance travelled.

5.3.3 Pose Sensor

To provide pose information for the UGV a Honeywell HMR3000 Electronic Compass Module has been fitted to the vehicle. This sensor can provide pose updates at rates of up to 20Hz. The manufacturer reports that heading values have a resolution of 0.1° and standard deviation better than 1.5° [Honeywell 1999]. Figure 5.19 shows the compass module.

Figure 5.19 - Honeywell HMR3000 Pose Sensor

Best performance is achieved from this sensor if it is mounted away from the effects of strong electromagnetic fields such as those produced by the UGV drive motors. The original plan was to mount this sensor inside the VCU. When the compass was first fitted to the ARP vehicle it

was observed that driving the vehicle would deflect the compass. The direction and power of drive were both factors. At full power the compass deflection was around 30°.

Alternative positions were investigated for the pose sensor away from the drive motors. Using the Honeywell sensor calibration utility it was found that most potential mounting locations were unsuitable. As the UGV chassis is made of steel this in itself was affecting the compass heading. It is possible to calibrate the sensor to ignore the presence of nearby ferrous metal but this was never entirely successful due to saturation from the chassis.

The final mounting solution was to fix the pose sensor to a non-ferrous mast. This mast holds the pose sensor over 0.5m above the chassis and drive motors. At this height the sensor is unaffected by any magnetic interference from the UGV. The rear of the ARP vehicle is pictured in Figure 5.20 showing both the pose sensor mast and GPS antenna mount.

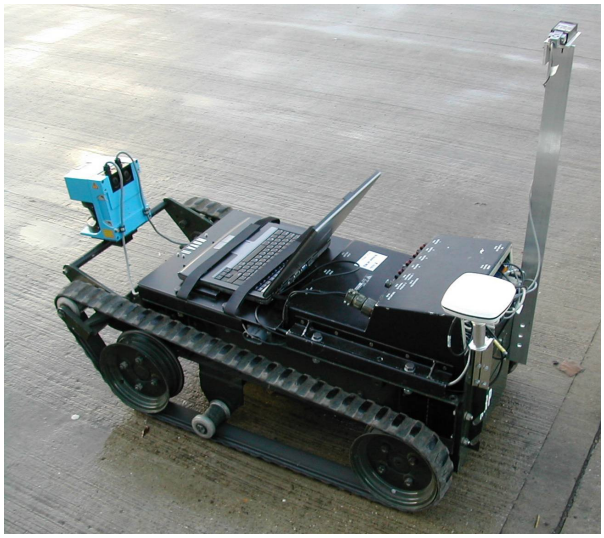

Figure 5.20 - Rear of ARP Vehicle

This pose sensor contains a three axis magnetometer for determining the UGV heading. An integrated two axis tilt sensor is also used for measuring the UGV pitch and roll. A large

variety of pose information is available from the sensor including raw data and processed heading, pitch & roll values.

At present the sensor has been configured to periodically output NEMA 0183 standard XDR messages 825 times a minute using the fastest available data rate of 19200 baud. XDR messages encode the raw transducer measurements from the tilt sensor and magnetometer. From these measurements the NOVA Computer repeatedly calculates the UGV heading at a rate of 5Hz. Heading values are then smoothed to reduce noise spikes.

5.4 DESCRIPTION OF UGV CONTROL INTERFACES

Two operator interfaces were implemented for control of the ARP vehicle. The ARP Interface was developed first and provides a basic interface for purely manual vehicle operation. Following this the NOVA GUI was developed to provide a full suite of autonomous behaviours. The concept of operation proposed in Chapter 4 has been realised within the NOVA control interface.

5.4.1 ARP Interface

The ARP Interface was initially developed to allow the hardware interfaces of the ARP vehicle to be tested in isolation. This interface has been maintained throughout the research. The latest version of the ARP Interface is actually a cut down version of NOVA providing manual control of the UGV platform and displaying basic output from the peripheral equipment onboard the ARP vehicle. Figure 5.21 illustrates the ARP Interface GUI.

Manual control of the ARP vehicle is possible using both the ARP Interface & NOVA. An onscreen interactive joystick graphic is used to move the UGV around. It can be used by mouse control or touch screen interface. Figure 5.22 illustrates the joystick control in use. In NOVA the joystick control is on the joystick tab on the left hand side of the GUI.

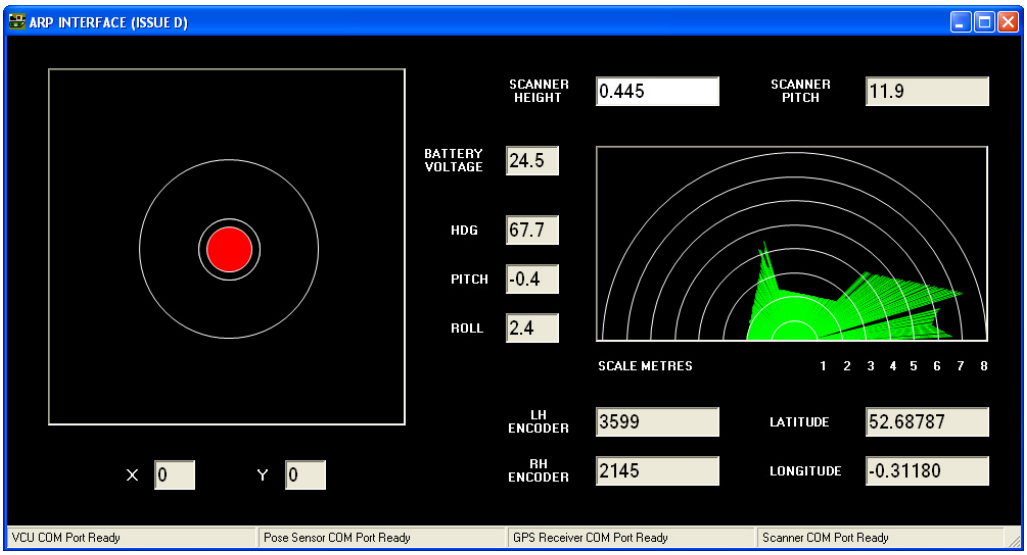

Figure 5.21 - ARP Interface GUI

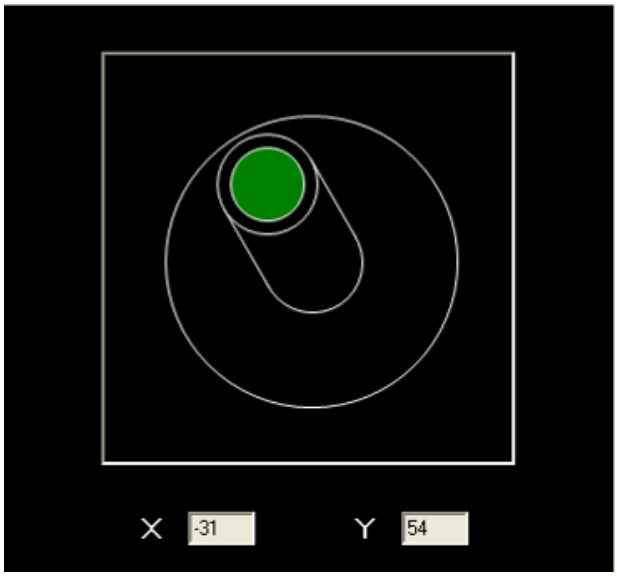

Figure 5.22 - Joystick control used to drive UGV

For safety the joystick features a button that acts as a dead man's handle. When the dead man's handle is held the joystick movement controls the UGV. Both tracks of the UGV are driven simultaneously at speeds relating to the joystick position. Moving the handle allows the full range of UGV movement from forward at full speed to reverse at full speed and all

possible turn rates. When the dead man's handle is released the joystick returns to the centre of the gate and the UGV stops.

On the right side of the ARP Interface GUI feedback from the other equipment on the ARP vehicle is displayed. A graphic shows the raw range values from a 180° laser scan of the environment. This scan has been produced by the Sick LMS 200 laser scanner. All other feedback is displayed in text boxes.

A BATTERY VOLTAGE text box displays the present UGV battery voltage. HDG, PITCH and ROLL indicate the orientation of the UGV as reported by the pose sensor. Cumulative measures of left hand and right hand track movement are provided by LH ENCODER and RH ENCODER. Finally LATITUDE and LONGITUDE provide the position for the GPS patch antenna fitted to the ARP vehicle in degrees, decimal degrees and minutes.

Unique to the ARP Interface is a basic method to calibrate the pitch of the laser scanner. The user specifies the height of the laser origin above the ground. The default value is 0.445m as measured on the ARP vehicle. From this the interface looks at the central range value of each 180° scan and calculates the scanner pitch. For accurate results calibration should be performed on flat ground that is obstacle free.

5.4.2 NOVA GUI

Figure 5.23 illustrates the NOVA GUI. On the left hand side are control tabs that provide the means to drive the UGV manually or describe a path to be followed autonomously. Additionally within these tabs are controls to specify parameters and options for autonomous movements such as enabling the local path planner.

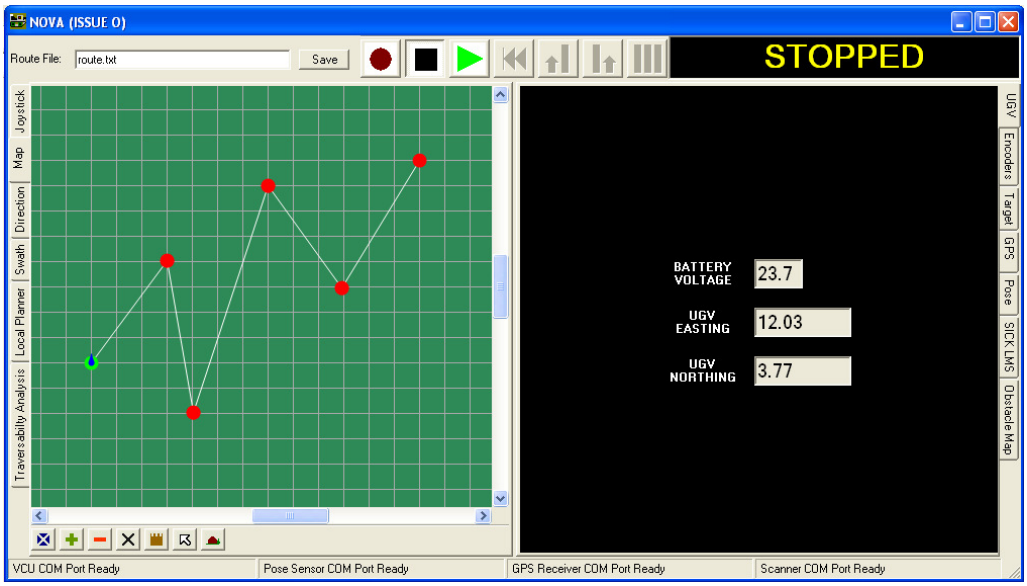

Figure 5.23 - NOVA GUI

The ARP Interface software displays the raw output of the peripheral equipment onboard the ARP vehicle such as the GPS receiver, pose sensor and obstacle detection hardware. On the NOVA GUI this same information is available via the diagnostic tabs on the right hand side of the GUI. In addition NOVA also has an obstacle map tab which shows the obstacle environment around the UGV.

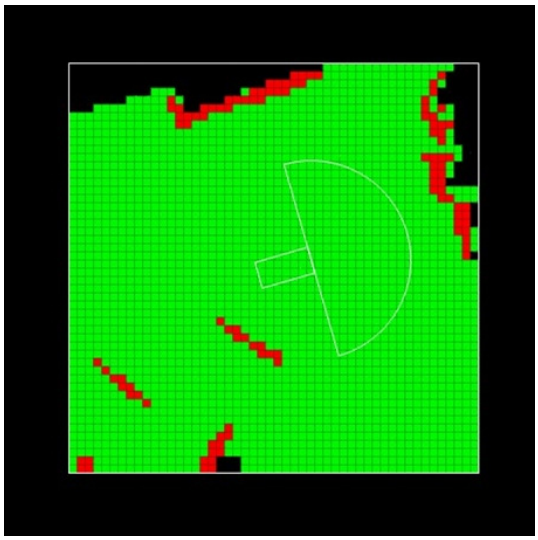

Figure 5.24 - Obstacle Map Tab

Figure 5.24 shows the obstacle map tab after the UGV had observed a number of obstacles. Red cells on the graphic indicate obstacles while the green cells are traversable. The present location and pose of the UGV with respect to the obstacle map is represented by the wire frame vehicle overlay.

5.4.3 Autonomous Control

Autonomous control of the ARP vehicle is only possible using the NOVA GUI. A toolbar at the top of the GUI is used to initiate autonomous behaviour. This toolbar is similar in style to the controls for a CD player with record, stop, play, rewind & track skip buttons. Figure 5.25 illustrates the toolbar.

Figure 5.25 - Autonomous behaviour controls

The record button begins the process of manual path recording. If selected then as the UGV is driven around using the onscreen joystick the path followed is recorded as a sequence of position nodes. These nodes may be retraced at anytime by selecting the rewind button then replayed by selecting the play button. Alternatively a parallel swath to the left or right of the recorded path can be played and retraced by selecting either the left or right swath button. Recorded paths and the corresponding swath paths can be viewed on the map tab of the GUI.

Selecting the stop button at any time will stop all autonomous UGV movement. Autonomous movement can be restarted by selecting either the play button or rewind button. Movement in the forward direction is initiated by the play button. This button is used to start (or restart) navigation of a user defined route or following of a recorded path or swath. Similarly

autonomous movement in the reverse direction is initiated by the rewind button. Selecting this button will cause the UGV to backtrack through a user defined route, manually recorded path or swath.

To the right of the rewind button on the toolbar are the left swath and right swath buttons. These buttons are used to make the UGV follow a swath parallel to a manually recorded route or previously explored set of user defined waypoints. When a swath button is selected NOVA generates an appropriate swath path to the left or right of the original route. Separation of the swath path from the original route is determined by the swath width and swath overlap parameters defined on the swath tab of the NOVA GUI.

At the far right of the toolbar is the area coverage button. This button is only enabled when an enclosed route is drawn on the map tab. Selecting the area coverage button generates a coverage plan for the enclosed area using the first waypoint leg to dictate swath direction. Swaths within the coverage plan are all parallel and separation is again determined by the values of the operator defined swath width and swath overlap parameters.

5.4.4 Autonomous Guidance

Two methods for guiding the autonomous motion of the UGV have been implemented within the NOVA GUI. The first is map based and requires the operator to specify a sequence of positional waypoints to be visited by the UGV. Alternatively the operator can use the direction control to set a heading for the UGV to autonomously follow.

Figure 5.26 shows an example of an operator specified route drawn on the map tab of the NOVA GUI. The start point is shown as a green circle with subsequent waypoints shown as red circles connected by chords. Present UGV position and orientation on the map are shown by a blue teardrop shape.

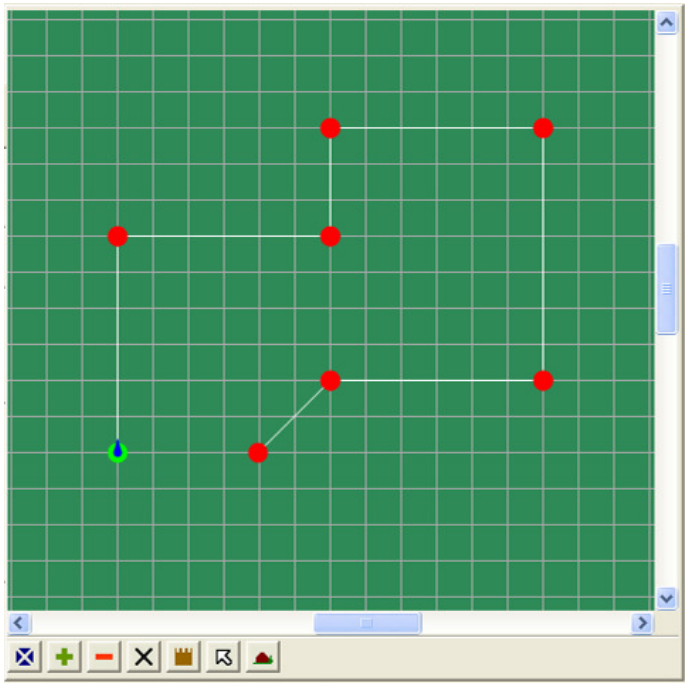

Figure 5.26 - Map Tab with Operator Specified Route

Along the base of the map are controls to add, remove or clear waypoints. To add a waypoint the operator selects the add waypoint button then clicks the desired location on the map to place the waypoint. An actual map can be used as the backdrop for the map tab.

Whenever a sequence of waypoints is drawn the play button will be enabled to allow autonomous behaviour. Selecting the play button indicates that autonomous movement is required and the UGV should begin to move towards the first waypoint.

UGV movement follows a simple turn to face the next waypoint and drive straight forward philosophy. If the UGV heading strays on route to the waypoint then NOVA will bring the UGV to a halt. The vehicle will then rotate until an acceptable heading is achieved before returning to straight line movement. When the UGV approaches the next waypoint in the route NOVA switches attention to the following waypoint.

Autonomous movement continues until the UGV reaches the vicinity of the final waypoint at which time the UGV shall revert to a safe state. UGV movement can also be paused at any time by selecting the stop button. Selecting the play button again will restart autonomous movement.

During autonomous forward movement NOVA will periodically record the location of the UGV as a position node. The recorded route is identical to those that can be recorded driving the UGV manually. Recording of the UGV path in this way allows for NOVA to backtrack the UGV along the path it has followed or indeed follow a parallel swath. Selecting the snail trail button on the map toolbar displays the recorded path on the map. An example recorded path is shown in Figure 5.27. Small yellow circles indicate the recorded points.

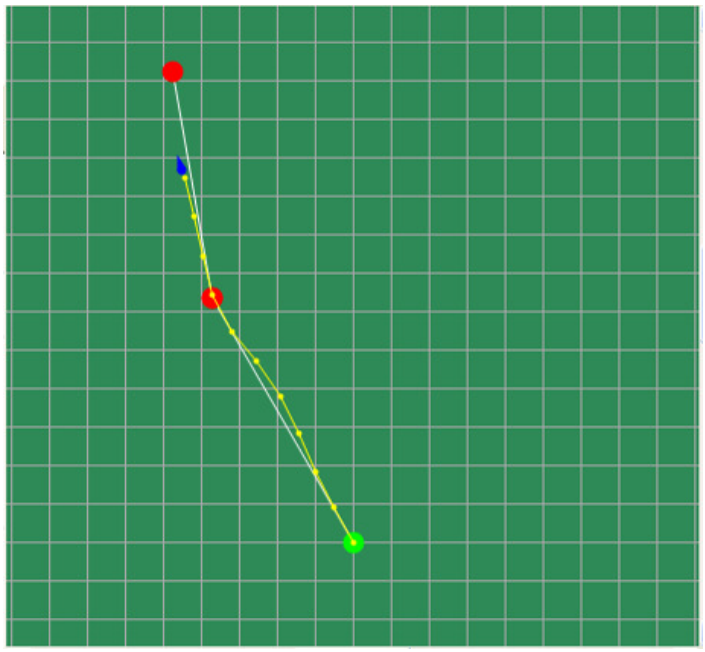

Figure 5.27 - Recorded UGV Path

Backtracking can be initiated at any point after the UGV has begun forward autonomous movement. Selecting the rewind button will switch NOVA into backtrack control mode. In this control mode the UGV shall move in the reverse direction using the recorded path as a

sequence of waypoints. Backtracking continues until the UGV has returned to the vicinity of the original start location where the UGV shall revert to a safe state. As with forward autonomous movement selecting the stop button at any time will pause the UGV.

When the UGV has backtracked towards the start location selecting the play button again will return NOVA to auto control mode. The UGV will autonomously follow the recorded path in the forward direction until reaching the last recorded point. At the last recorded point NOVA will again turn attention to reaching the next unexplored operator waypoint.

An alternate means to control the UGV has been developed in the form of a direction control as illustrated in Figure 5.28. During testing it was found that this control method offered a convenient way to navigate longer distances. Additionally it does not require the operator to have prior route knowledge unlike specifying waypoints on the map.

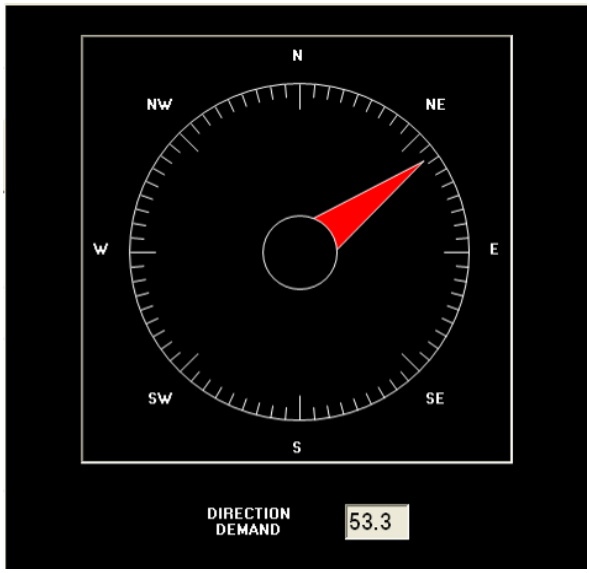

Figure 5.28 - Direction Control

To use the control the operator simply grabs the needle via the mouse pointer or touch screen and drags it to a new heading. In response to the control being moved the autonomously

moving UGV will turn to match the new heading. If local path planning is enabled then when NOVA encounters an obstacle the path planner will steer the UGV around the obstacle before reverting to the operator set heading.

As with waypoint control autonomous motion using the direction control is initiated by selecting the play button on the centre toolbar. The UGV path is also recorded as a sequence of position nodes. Selecting the rewind button allows the route to be followed in reverse and a parallel swath can be generated and driven by selecting either the left or right swath buttons.

5.4.5 Driving Swaths

The UGV can follow swaths parallel to routes that have been driven manually or autonomously. Selecting either the left swath or right swath button will cause NOVA to generate a swath path. The generated path will be parallel to the current recorded path but offset to the left or right as appropriate. Figure 5.29 shows an example right swath path generated from the recorded path pictured in Figure 5.27.

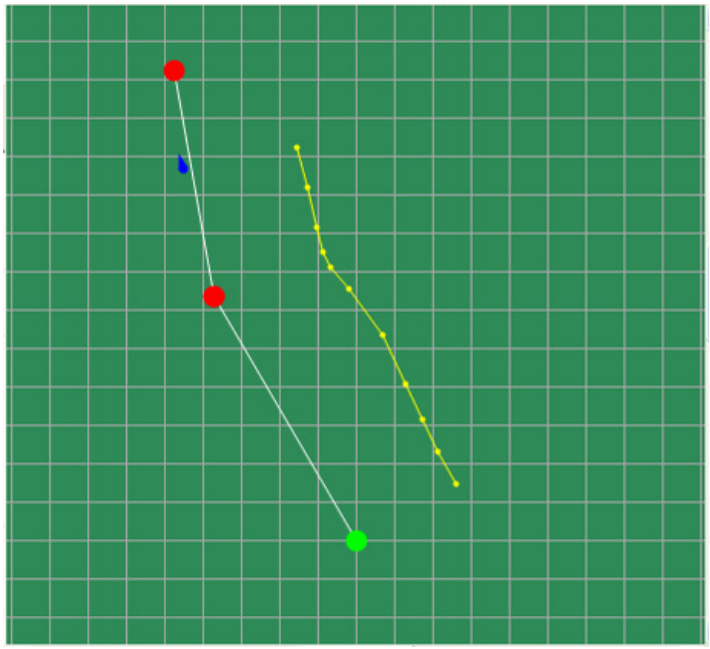

Figure 5.29 - Right Swath of UGV Path

Separation of the swath path from the recorded path is dependent on the swath width and swath overlap parameters. These values can be set on the swath tab of the NOVA GUI. Figure 5.30 below shows the swath tab.

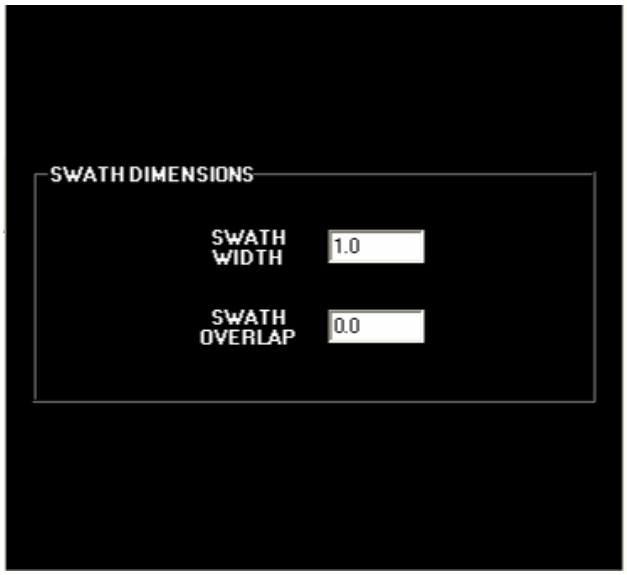

Figure 5.30 - Swath Tab

Swath width is a measure of the width of the swath covered by the vehicle and is intended to account for peripheral equipment such as metal detector or a crop spraying gantry. Swath overlap is a measure of the width of the overlap between adjacent swaths. In a mine clearance application the ground may need to be swept more than once to ensure a high probability of detection so the swath overlap would be set to at least half the swath width.

As a safety feature selecting a swath button takes the UGV to a safe state. Selection of either the play or rewind button is required to continue autonomous movement in the forward or reverse direction respectively. All subsequent UGV movement will be to navigate between points on the swath path until the swath button is unselected. Again as a safety feature when a swath button is unselected the UGV shall revert to a safe state.

5.4.6 Area Coverage

When a set of operator waypoints drawn on the map forms an enclosed area it is possible to generate a sequence of swaths that cover the entire area. The method works regardless of whether the enclosed area is convex or concave. Area coverage is not however possible for routes that contain crossed or overlapping waypoint chords but such routes could be easily redrawn without these features.

Selecting the area coverage button creates a coverage plan that begins by proving the perimeter of the enclosed area. This provides a safe circuit of the area that the UGV can use for repositioning between swaths. Next the UGV will be guided across the enclosed area in a sequence of swaths turning 180° at the end of each swath to reposition for the next. The orientation of the swaths will match the chord between the start point and first waypoint in the original operator waypoints. Figure 5.31 shows an example coverage plan for an enclosed area.

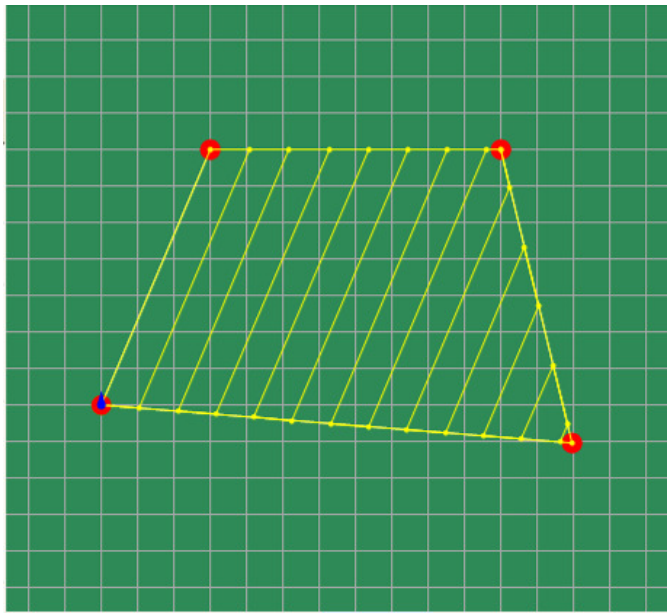

Figure 5.31 - Coverage Plan for Enclosed Area

Coverage plans are made up of a number of waypoints that describe the path the UGV should take. As with parallel swaths the autonomous behaviour controls are used to make the UGV drive a coverage plan. Selecting the play button will make the UGV autonomously visit each waypoint in the coverage plan in order. If the rewind button is selected at any time the UGV will stop and backtrack through the coverage plan waypoints.

6 LOCALISATION

Many types of system have been created to give an indication of the present location for a UGV. These can be divided into the two groups of absolute and relative location sensing. Absolute location sensing typically uses a network of reference beacons in known locations to compute the UGV position. Relative location sensing provides a notion of how far the UGV has travelled relative to some earlier position or start location.

All localisation systems are subject to error and so deriving the true location of a UGV is a challenging task. A popular solution to this is to combine different types of localisation system. The aim is to use complementary localisation systems that fall foul of errors in different ways. Then errors from one localisation system can be detected and ignored with reference to another system.

In this chapter the two localisation systems onboard the ARP vehicle are discussed. The ARP vehicle is fitted with an absolute localisation system based on a GPS receiver and a relative localisation system based on wheel encoders and a digital pose sensor. For each of these localisation systems the sources of error are identified.

A particle filter algorithm is then introduced as a means to fuse the location estimates made by the two localisation systems. This filter is probabilistic in nature and models the range of possible UGV locations with a series of weighted particles. Particles in the filter are translated to match UGV movement and their weights are regularly updated in line with the probability that the particle indicates the true UGV location.

6.1 ABSOLUTE LOCALISATION

Absolute location information for the ARP vehicle is provided by the onboard GPS receiver. This stand alone device continuously streams location and error information to the NOVA Computer. In order to utilise the location information the computer must perform a map projection and then account for the position of the GPS antenna relative to the rotation centre of the UGV.

The following subsection describes the map projection applied to the GPS location data. Characteristics of the GPS location data are then discussed. Next the issues of using GPS alone to guide the UGV are highlighted. Finally the problem of error estimation for GPS location data is tackled.

6.1.1 Map Projection

Raw GPS location data is based on the WGS84 (World Geodetic System 1984) coordinate system. The latitude, longitude & altitude information associated with this coordinate system specifies a geodetic position relative to an ellipsoidal model of the Earth. This reference ellipsoid is known as GRS80 (Geodetic Reference System 1980).

In normal operation the UGV will be requested to follow sequences of operator waypoints. It is most intuitive to allow these waypoints to be set using map coordinates specified by northing and easting. Thus knowing the UGV position in terms of latitude, longitude & altitude is not useful. However geodetic position can be readily converted to northing & easting using a map projection.

Ordnance Survey is the national mapping agency for Great Britain. The map projection used by Ordnance Survey is known as the National Grid. This projection is an example of a Transverse Mercator projection [OS 2007]. A projection of this kind is like resting the centre

of a sheet of card on a line of longitude. Along the centre of the card all distances are maintained. Either side of the centre the projection is distorted due to the curvature of the reference ellipsoid. Figure 6.1 illustrates the point.

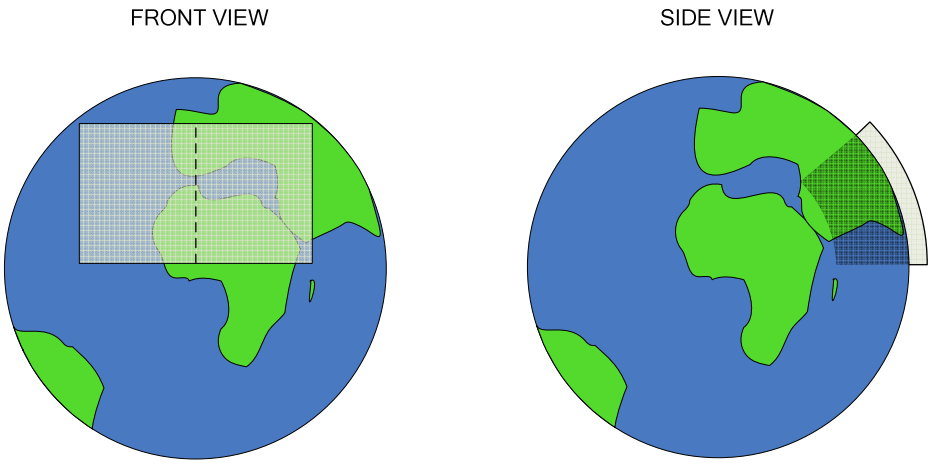

Figure 6.1 - Idea of Transverse Mercator Projection

Ordnance Survey has developed OSTN02 the National Grid Transformation. This allows projected positions to be transformed accurately from the GPS coordinate system to the OSGB36 (Ordnance Survey Great Britain 1936) system used in Ordnance Survey maps. OSGB36 is based on the Airy 1830 ellipsoidal model rather than GRS80. This reference ellipsoid is a better approximation to the shape of the Earth around Britain.

The error introduced by the OSTN02 transformation has a standard deviation of around 0.1m [OS 2005]. A software library that includes the transformation formulae is freely available from Ordnance Survey. Map projection functions for geodetic positions are also provided. This library was incorporated into the software developed during this research to transform GPS coordinates to OSGB36 northing and easting.

6.1.2 Characteristics of GPS

It is widely accepted that the horizontal coordinates of a GPS solution follow an approximate bivariate normal distribution with the true location of the GPS antenna as a mean. To that end GPS location estimates are often provided with a description of an error ellipse which describes the shape of the underlying distribution. Thus GPS location data streamed from a receiver should be expected to fluctuate continuously and at random.

Figure 6.2 illustrates the fluctuation of GPS location data well. Two plots are shown, the first is for basic GPS and the second GPS with EGNOS differential corrections. To generate each plot the ARP vehicle was left standing in an open space with a fairly clear view of the sky for a period of approximately 1 hour. The GPS receiver was set to report new position estimates at a rate of 1Hz.

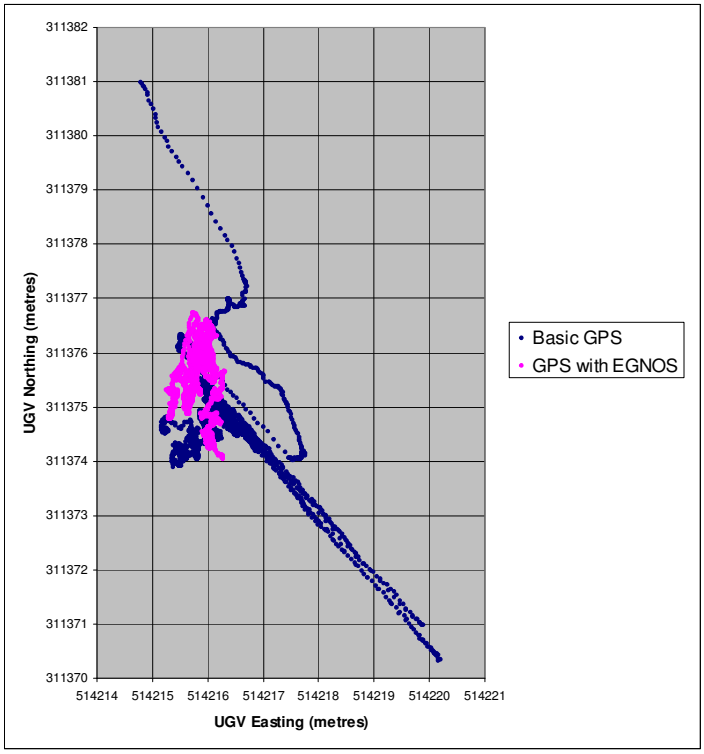

Figure 6.2 - Plot of GPS Location Estimates over 1hr

It can be seen from the two plots that the basic GPS location estimates are spread much more widely than the location estimates from GPS with EGNOS corrections. The basic GPS location estimates have a range of around 10.5m northing and 5.5m easting. With EGNOS corrections the range was around 2.75m northing and 1.5m easting.

An important observation from Figure 6.2 is the fact that the basic GPS location estimates can wander several metres in the space of a few seconds. On this basis the basic GPS location estimates could not be used to reliably localise the ARP vehicle. So for all remaining work in this project the more accurate EGNOS corrected GPS was used.

To check the performance of the GPS receiver Figure 6.3 shows the standard deviation of the northing and easting of points from the plots in Figure 6.2. The claimed accuracy of the receiver under ideal conditions using EGNOS is sub-metre 95% of the time [CSI 2005]. In statistical terms this means that the standard deviations should be less than 0.5m. This is not the case here for the northing error but the value is quite close. The test was performed in an area containing a number of buildings so the conditions were not ideal. Some amount of satellite occlusion and signal multipathing should be expected which would increase error.

	Northing Standard Deviation	Easting Standard Deviation
GPS	1.057875364	0.803332589
GPS with EGNOS	0.650200522	0.19743485

Figure 6.3 - Standard Deviation of GPS Location Estimates over 1hr

This leads us to another important characteristic of GPS. When in motion if the UGV comes close to trees or buildings the error associated with the GPS location estimate will rise. If satellites are occluded the GPS receiver can lose EGNOS corrections and in severe cases may

not be able to compute a location estimate at all. This is the case if the UGV drives inside a building.

6.1.3 Using GPS to Guide a UGV

Given the characteristics of GPS it is clear that it could only be used to guide a UGV when the antenna has a clear view of the sky. The random fluctuation of the GPS location estimates would cause the UGV to frequently adjust heading when on route to a waypoint. Additional errors caused by satellite occlusion and signal multipathing would further compound this problem.

Figure 6.4 illustrates the major issue. To create this screenshot from the NOVA GUI the ARP vehicle was requested to follow a waypoint route that described a 10m by 10m square. The figure shows the path that NOVA believes the UGV followed as a yellow line. For this test the ARP vehicle used the onboard pose sensor to determine which direction it was facing. The GPS receiver alone was used to localise the vehicle.

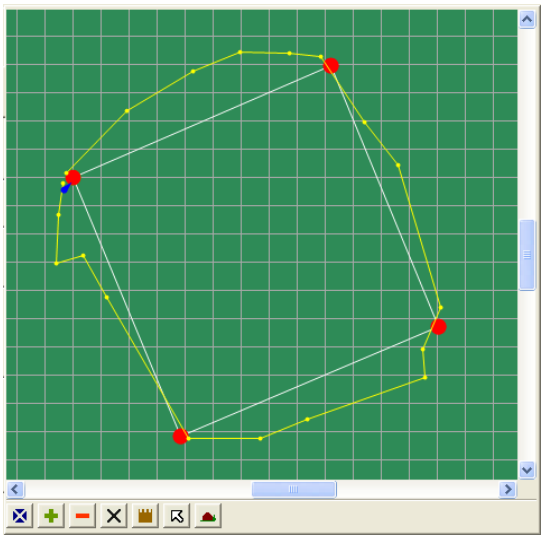

Figure 6.4 - Record of GPS Guided Route

It can be seen that the recorded route deviates from the requested route in places by several metres. This was caused purely by the fluctuations in the GPS location estimates. The result was that the ARP vehicle had to stop and turn every few metres as it seemed to jump around relative to the waypoints. If the distance between waypoints was much larger the effect of the fluctuations would be less pronounced.

The conclusion to be drawn here is that GPS alone can not be used for precision UGV navigation. Something else is required to overcome the random fluctuations in the location estimates and other sources of error.

6.1.4 Error Estimation

The intent of this part of the research is to improve on the location estimation capabilities of GPS alone. Knowing the magnitude of error associated with GPS location estimates is important. When combining GPS estimates with those from other localisation equipment the amount of uncertainty in each estimate must be taken into account if a plausible combined estimate is to be produced.

The GPS receiver onboard the ARP vehicle can provide estimates of the error associated with the latest location estimate. Estimates of the error in terms of latitude and longitude are available. Figure 6.5 shows graphs of the estimated standard deviation for latitude and longitude. These graphs correspond to the plot of location estimates from the GPS receiver using EGNOS in Figure 6.2.

A good way to visualise the quality of a GPS location estimate is to look at an error ellipse for the estimate. An error ellipse defines a line of constant probability for the bivariate normal distribution on which the location estimate is based [Johnson 2007]. So an error ellipse for a

GPS location estimate could be described using just the standard deviation values for latitude and longitude error.

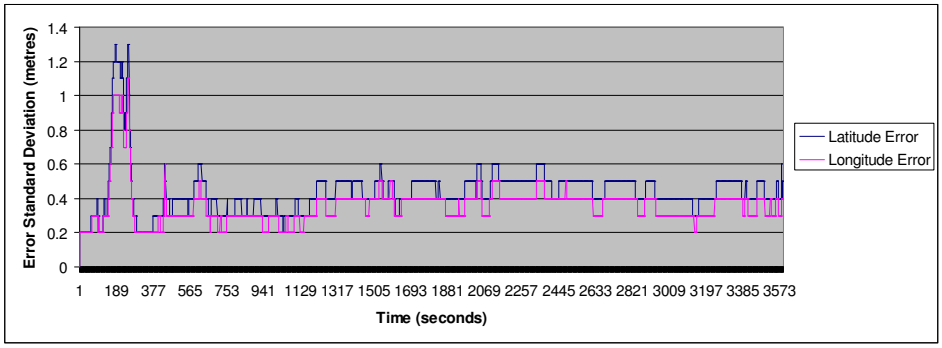

Figure 6.5 - Latitude and Longitude Error over 1hr

However this is not the full story. Latitude and longitude errors for GPS location estimates are correlated. This correlation is determined by the position of the satellites used to compute the location estimate. As a consequence the bivariate normal distribution that describes a GPS location estimate will not always be north oriented.

The GPS receiver onboard the ARP vehicle can output estimates of the correct error ellipse parameters. The length and orientation of the semi-major axis is available along with the length of the semi-minor axis. These parameters represent the standard deviation of location error in the direction of the error ellipse axes. Together they describe an ellipse that should contain the true antenna location 39.4% of the time. Multiplying the axis lengths by a factor of 2.447 gives an error ellipse that should contain the true location 95% of the time.

The following two figures show how the error ellipse parameters changed during the 1hr plot of location estimates using GPS with EGNOS differential corrections. Figure 6.6 shows the standard deviation of the location error along axes of the error ellipse. The orientation of the

error ellipse is depicted in Figure 6.7. It can be seen that the ellipse orientation changes slowly as the GPS satellites orbit above.

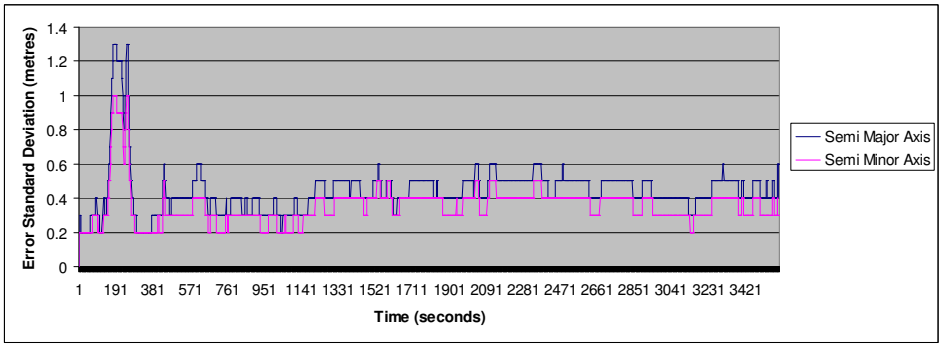

Figure 6.6 – Semi-Major and Semi-Minor Axis Length over 1hr

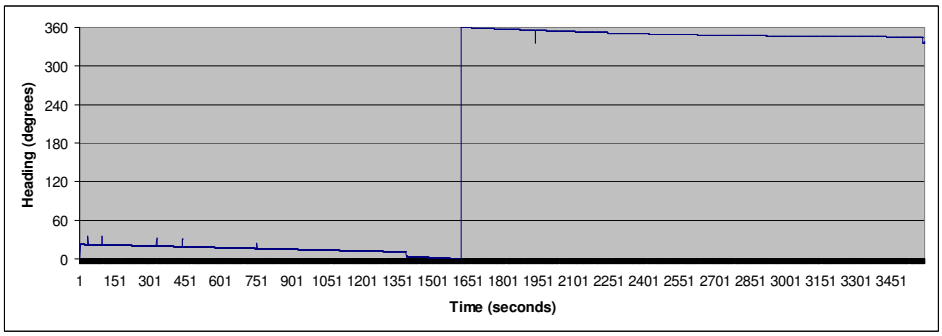

Figure 6.7 - Error Ellipse Orientation over 1hr

The GPS receiver derives the latitude and longitude error estimates introduced earlier from the shape of the GPS error ellipse. Consider Figure 6.8 which shows the error ellipse plotted on axes representing latitude error and longitude error. The radius of the error ellipse equates to the standard deviation of error at each possible orientation. So the standard deviation of latitude error ($\sigma_{Latitude}$) is given by the error ellipse radius that coincides with the latitude error axis. Likewise the error ellipse radius that coincides with the longitude error axis gives the standard deviation of longitude error ($\sigma_{Longitude}$).

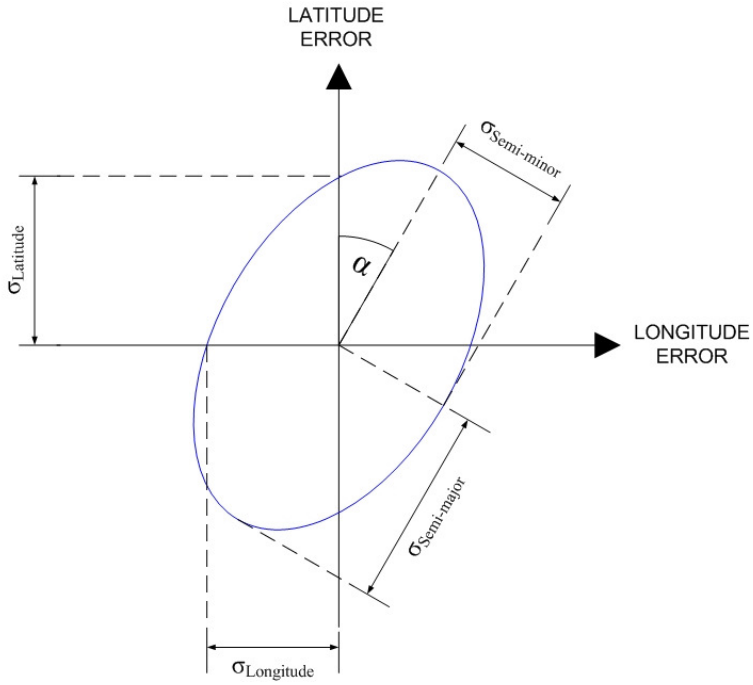

Figure 6.8 - Latitude & Longitude Error Standard Deviation Calculation

To calculate $\sigma_{Latitude}$ and $\sigma_{Longitude}$ the intersections between the error ellipse and the two graph axes must be found. One method to do this is to use a second set of axes aligned with the major and minor axes of the error ellipse. Imagine this as Figure 6.8 being rotated anti-clockwise by an angle α which is the orientation of the error ellipse.

Assume that the major axis runs along the y-axis and the minor axis runs along the x-axis. The equation of the error ellipse relative to the y-axis and x-axis is:

$$\frac{y^2}{\sigma^2_{Semi-major}} + \frac{x^2}{\sigma^2_{Semi-minor}} = 1$$

Where $\sigma_{Semi\text{-}major}$ is the length of the semi-major axis of the error ellipse and $\sigma_{Semi\text{-}minor}$ is the length of the semi-minor axis.

The equation of the latitude error axis relative to the y-axis and x-axis is:

$$y = -x \ / \ \tan(\alpha)$$

By substitution the ellipse equation becomes:

$$\frac{x^2}{\sigma_{Semi-major}^2 \ \tan(\alpha)^2} + \frac{x^2}{\sigma_{Semi-minor}^2} = 1$$

This can be quickly rearranged to give:

$$x^2 = \frac{\sigma_{Semi-major}^2 \ \sigma_{Semi-minor}^2 \ \tan(\alpha)^2}{\sigma_{Semi-minor}^2 + \sigma_{Semi-major}^2 \ \tan(\alpha)^2}$$

Substituting this result back into the ellipse equation gives:

$$\frac{y^2}{\sigma_{Semi-major}^2} + \frac{\sigma_{Semi-major}^2 \ \tan(\alpha)^2}{\sigma_{Semi-minor}^2 + \sigma_{Semi-major}^2 \ \tan(\alpha)^2} = 1$$

Which can be rearranged as:

$$y^2 = \sigma_{Semi-major}^2 \left(1 - \frac{\sigma_{Semi-major}^2 \ \tan(\alpha)^2}{\sigma_{Semi-minor}^2 + \sigma_{Semi-major}^2 \ \tan(\alpha)^2}\right)$$

These results yield the coordinates of intersection between the error ellipse and latitude error axis so $\sigma_{Latitude}$ can be calculated by Pythagoras Theorem:

$$\sigma_{Latitude} = \sqrt{x^2 + y^2}$$

By repeating this calculation using the orientation of the semi-minor axis of the error ellipse in place of the orientation of the semi-major axis $\sigma_{Longitude}$ can also be found.

Unfortunately the GPS receiver does not provide an estimate of the correlation or covariance between the errors in latitude and longitude. However this can be derived from the parameters of the error ellipse that are available. To start the covariance matrix C_{GPS} of the bivariate normal distribution underlying GPS location estimates must be defined. Within the covariance matrix σ_{Cov} is the covariance between the errors in latitude and longitude.

$$C_{GPS} = \begin{bmatrix} \sigma^2_{Latitude} & \sigma_{Cov} \\ \sigma_{Cov} & \sigma^2_{Longitude} \end{bmatrix}$$

For a bivariate normal distribution the length of the semi-major axis and semi-minor axis of the error ellipse correspond to the square root of the eigenvalues of the covariance matrix. Likewise the eigenvectors of the covariance matrix encode the orientation of the error ellipse axes [Johnson 2007].

Normally to find eigenvalues of the GPS covariance matrix it would be necessary to find the roots of the following characteristic equation:

$$det\left|C_{GPS} - \lambda I\right| = 0$$

In the characteristic equation λ is a dummy scalar used to find the eigenvalues and I is the 2x2 identity matrix. Thus the characteristic equation can be expanded to give:

$$det\begin{vmatrix} \sigma^2_{Latitude} - \lambda & \sigma_{Cov} \\ \sigma_{Cov} & \sigma^2_{Longitude} - \lambda \end{vmatrix} = 0$$

Solving the determinant yields:

$$(\sigma^2_{Latitude} - \lambda)\ (\sigma^2_{Longitude} - \lambda) - \sigma^2_{Cov} = 0$$

This can be rearranged as a quadratic equation in terms of λ:

$$\lambda^2 - \lambda \left(\sigma^2_{Latitude} + \sigma^2_{Longitude}\right) + \sigma^2_{Latitude}\,\sigma^2_{Longitude} - \sigma^2_{Cov} = 0$$

Roots of this quadratic equation are then given by:

$$\lambda = \frac{\left(\sigma^2_{Latitude} + \sigma^2_{Longitude}\right)}{2} \pm \sqrt{\left(\frac{\sigma^2_{Latitude} - \sigma^2_{Longitude}}{2}\right)^2 + \sigma^2_{Cov}}$$

These roots are the eigenvalues for the GPS covariance matrix. However the GPS receiver can provide the length of the semi-major axis and the semi-minor axis of the error ellipse. Thus these eigenvalues can be calculated by simply squaring each of the semi axis lengths.

Let us call the larger of the eigenvalues λ_1. This eigenvalue corresponds to the semi-major axis of the error ellipse. So the magnitude of λ_1 is given by the following equation:

$$\lambda_1 = \frac{\left(\sigma^2_{Latitude} + \sigma^2_{Longitude}\right)}{2} + \sqrt{\left(\frac{\sigma^2_{Latitude} - \sigma^2_{Longitude}}{2}\right)^2 + \sigma^2_{Cov}}$$

This can be rearranged to give:

$$\sigma^2_{Cov} = \left(\lambda_1 - \frac{\left(\sigma^2_{Latitude} + \sigma^2_{Longitude}\right)}{2}\right)^2 - \left(\frac{\sigma^2_{Latitude} - \sigma^2_{Longitude}}{2}\right)^2$$

Thus the covariance of GPS location estimates can be expressed as a function of the largest eigenvalue, the variance of the latitude error and the variance of the longitude error. If required ρ the correlation of latitude and longitude errors can also be derived using the following equation [Rice 1995]:

$$\rho = \frac{\sigma_{Cov}}{\sigma_{Latitude}\ \sigma_{Longitude}}$$

A comparison of the true GPS error ellipse and the north oriented error ellipse based purely on the estimates of latitude and longitude error is shown in Figure 6.9. This figure was drawn using real error data. It is based on the location estimate after 4 minutes and 6 seconds of the 1hr long plot of location estimates made by the GPS receiver with EGNOS corrections.

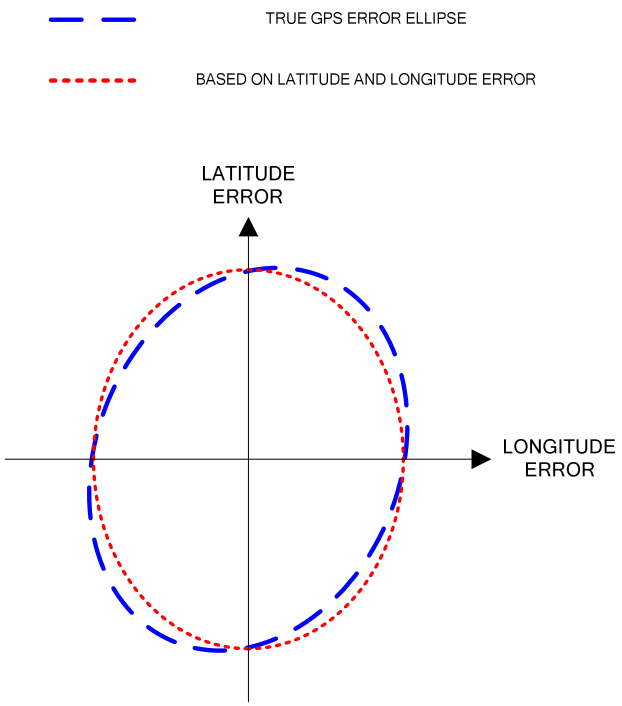

Figure 6.9 - Comparison of Error Ellipses

From Figure 6.9 it could be concluded that the north oriented error ellipse based purely on the estimates of latitude and longitude error is a fairly good approximation of the true GPS error ellipse. In fact this should be the case provided that the correlation between latitude and longitude error is small.

6.2 RELATIVE LOCALISATION

Output from the pose sensor and wheel encoders are combined to calculate the relative location of the ARP vehicle. Pose sensor data is streamed continuously to the NOVA Computer. This data includes estimates of the current vehicle heading, pitch and roll. The NOVA Computer also frequently requests the latest encoder changes from the onboard Vehicle Control Unit.

The recursive dead reckoning method used to calculate the relative position of the ARP vehicle is described in the next subsection. Following that the characteristics of the relative localisation system are discussed. Finally the magnitude of error associated with the system is considered.

6.2.1 Position Calculation

The relative localisation system maintains a cumulative estimate of the vehicle movement. This estimate is relative to the vehicle start position and is stored in terms of northing and easting. Updates of the relative vehicle position are calculated by the NOVA Computer at a rate of 5Hz.

At each update the current vehicle heading is grabbed. The distance the vehicle has travelled is then calculated from the average of the left and right hand encoder changes since the last update. An encoder count of 363 equates to 1m of forward motion. Finally the new vehicle position is estimated by adding the new distance travelled along the current heading to the last position estimate.

In mathematical terms this simple kinematic model can be described by the following two equations:

$$n(t) = n(t-1) + d(t)\ \cos(\theta(t))$$

$$e(t) = e(t-1) + d(t) \sin(\theta(t))$$

In these equations n(t) and e(t) represent the UGV northing and easting at time t respectively. The distance traveled in the time interval (t-1, t] is given by d(t) while θ(t) is the UGV heading at time t.

This method of dead reckoning assumes that the ARP vehicle will only travel in straight lines. To ensure that the ARP vehicle does not arc to the left or right when it is supposed to be travelling straight the track drives have been trimmed. Controls have been included on the NOVA GUI to allow the track drives to be trimmed when necessary. Trimming involves slowing down the speed of one track drive by reducing the duty cycle sent to the motor. If the vehicle is veering to one side trimming may be used to slow the faster track drive.

Steering is achieved by stopping the vehicle and performing a neutral turn. As such any change observed by the encoders during a turn can be ignored as the left and right encoder values should be equal and opposite. Any disparity will be the result of track slippage. NOVA maintains a variable describing the turning state of the UGV. While this variable indicates the UGV is turning encoder changes are not used for localisation.

6.2.2 Characteristics of Relative Localisation System

The characteristics of the relative localisation system are most unlike those of the location estimates produced by the GPS receiver. Most fundamental is the fact that the relative localisation system can only estimate where the UGV is in relation to the start position. This of course makes it difficult to relate location estimates to any kind of map. So in order to use the relative localisation system to navigate a complex route the intended path needs to be measured out beforehand.

However location estimates from the relative localisation system are not subject to continuous random noise like those from the GPS receiver. The error mode for the relative localisation system is completely different. Error tends to accumulate in the relative localisation system as the UGV moves.

The wheel encoders on the ARP vehicle are intended to measure the distance the vehicle has travelled horizontally forwards and backwards. If the terrain is not smooth and flat the encoders will be measuring the distance the tracks have had to travel to overcome any bumps and slopes. Figure 6.10 illustrates the magnitude of error when the encoders measure the same distance on flat ground, a slope and over bumps.

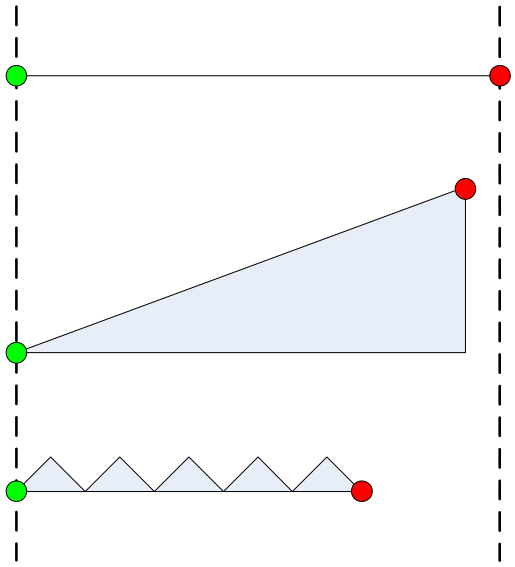

Figure 6.10 - Encoders on Flat, Sloped & Bumpy Ground

Another source of encoder error is track slippage. Since the UGV uses skid steering track slippage is expected during turns. As a result encoder changes are deliberately ignored whenever the UGV is performing a neutral turn. Track slippage has not been observed during straight line motion.

The assumption is made that the UGV always turns on the spot. However track slippage can cause the UGV to turn off centre. Two different sources of track slippage have been observed during turns. The first is when the outside of the track slips on the ground. The drive wheel has also been seen to slip on the inside of the track.

If one track slips while the other gets traction the UGV will tend to rotate around the centre point of the slipping track. Typically this will only cause a few centimetres of error in subsequent location estimates made by the relative localisation system. At worst this error can be equal to half the width of the UGV.

The heading of the ARP vehicle as measured by the pose sensor is also subject to noise. Normal errors associated with the magnetoresistive sensor measurement of the Earth's magnetic field accounts for much of this noise. Preliminary tests have showed that the UGV heading could sway by several degrees while the UGV was driving straight along smooth flat ground.

Changes in pitch and roll of the UGV will additionally affect the precision of the pose sensor heading value. This is because the Earth's magnetic field has a vertical component in the UK as it dips towards the planetary pole. If the pose sensor is not horizontal some of this vertical field component will be measured by the horizontal magnetoresistive sensors.

Experiments were performed to see if pitch and roll measurements from the pose sensor can be used to correct these errors. Preliminary experiments showed that the pitch and roll sensor is affected by vibration as the UGV is driving. Different filters can be applied to the pitch and roll measurements to see if the effects of the UGV vibration can be removed. This is discussed further in Chapter 8.

Other sources of magnetic field were also seen to affect the pose sensor heading value during preliminary testing. If the UGV moves so that the pose sensor is close to any ferrous metal the heading value is dragged to point at the metal. It has also been observed that electromagnetic fields emitted by machinery can deflect the pose sensor heading.

6.2.3 Using Relative Localisation to guide the UGV

The characteristics of the relative localisation system suggest that over shorter distances it should be quite accurate. As the UGV travels further away from the start location the precision of location estimates will decrease. If the terrain is sloped or rough the precision of location estimates can be expected to deteriorate at a faster rate.

Figure 6.11 shows the result of the ARP vehicle being requested to follow a 10m by 10m square waypoint route over rough undulating grass. The square was also marked out on the ground. This allowed the deviation of the UGV from each corner to be measured as the corresponding waypoint was reached.

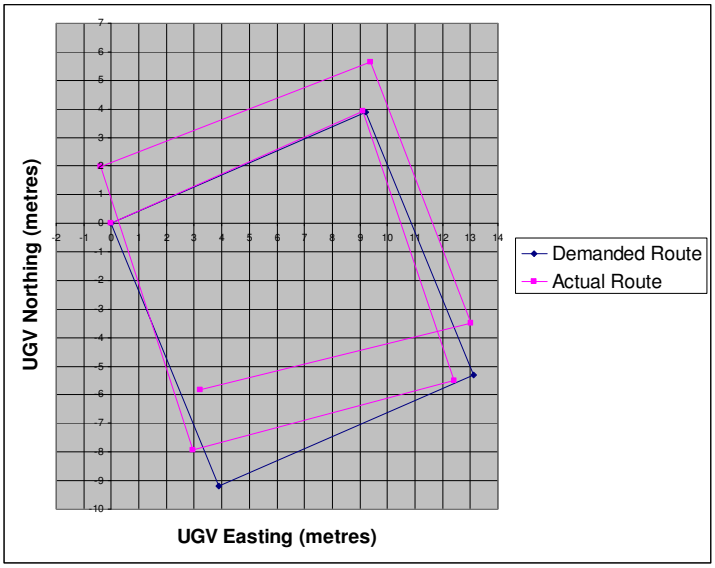

Figure 6.11 - Accumulation of Error with Dead Reckoning

It is clear from Figure 6.11 that with each lap of the square the UGV strayed further away from the markings on the ground. The first leg of the route was near perfect. Errors then accumulated in the dead reckoning calculation due to undulations in the ground near the south eastern corner of the square.

This result agrees well with the expected behaviour of the relative localisation system. Over short distances the accuracy is good. When the terrain is not smooth and flat errors accumulate in the location estimate. On flat ground the relative localisation system fares better.

An attractive feature of the relative localisation system is highlighted when the UGV is backtracking. The UGV will accurately follow the path it has recorded when driving forwards. This implies that the UGV backs up over the same ground and so is subject to the same localisation errors. By the time the UGV has returned to the start point the localisation error has mostly disappeared.

Contrast this with the performance of the GPS. If the UGV was asked to backtrack using GPS localisation it may stray several metres from the original path due to random fluctuation in the location estimates. So when the UGV needs to retrace a route the relative localisation system should perform better.

6.2.4 Error Estimation

It is important to know the magnitude of error associated with individual location estimates from the relative localisation system. These errors must be taken into account when fusing location estimates from the relative localisation system and GPS receiver. Two error sources in the relative localisation system have already been identified. The first is error in the heading data from the pose sensor. Error in the encoder count is the second. These two sources of error are uncorrelated.

It was possible to derive separate standard deviation values for the encoder and heading changes on the ARP vehicle. This was done by driving the vehicle autonomously around the test area while recording sensor data. New data from the encoders and pose sensor was recorded each time the relative localisation system made a new location estimate. The recorded values represent several hundred metres of straight line driving.

It should be noted that during autonomous operation the ARP vehicle travels at a constant speed. This is helpful as the encoder values are thus not affected by the vehicle speeding up or slowing down.

The recorded data had a mean encoder count of 33.9957346 between location estimates. Given that the relative localisation system runs at a rate of 5Hz and an encoder count of around 363 equates to 1m of travel the average speed of the vehicle was $0.47ms^{-1}$. From the data the standard deviation for encoder changes was found to be 5.902924. This equates to a standard deviation for distance travelled $\sigma_{Distance}$ of 0.01626m

Since noise can equally well increase or decrease the pose sensor heading it should be expected that the average heading change is close to 0°. A mean heading change of -0.002193844° was found from the recorded data. The associated standard deviation for heading change $\sigma_{Heading}$ was found to be 0.376463°.

It is fair to say that these error statistics would be different for another test area with different ground conditions. The standard deviation values should be larger for an area with rougher terrain. To account for this issue more conservative standard deviation values were used when fusing location estimates with those from the GPS receiver.

6.3 SENSOR FUSION

At the outset of this research it seemed a reasonable strategy to use GPS to localise the UGV whenever the receiver could provide a location estimate. The relative localisation system would then only be needed if the UGV was indoors or surrounded by tall objects that occluded the satellites. In this case location updates from the relative localisation system would be integrated with the last GPS location estimate until the GPS receiver again had a clear view of the sky.

Once the characteristics of the GPS location estimates had been fully examined this naïve strategy was abandoned. It was clear that GPS alone was not accurate enough to be used for precision navigation of the ARP vehicle. A better strategy was required that incorporated data from the otherwise redundant relative localisation system.

The error in the relative location estimates tends to accumulate as the UGV moves. Yet over short distances this system is typically more accurate than GPS. In contrast absolute location estimates produced by the GPS receiver fluctuate around the true antenna location. This implies that small UGV movements are not readily detected. However the error for absolute location estimates does not accumulate as the UGV moves.

Thus the relative and absolute localisation systems onboard the UGV are complementary in nature. What is required is a fusion strategy that periodically corrects the location estimate from the relative localisation system with a GPS location estimate. Such a system should outperform the GPS alone.

In this research a probabilistic technique known as a particle filter has been used to fuse location estimates from the two localisation systems. An overview of the particle filter is given

in the next subsection. The remainder of the chapter then looks at the various steps of the particle filter algorithm.

6.3.1 Overview of Particle Filter

Particle filters can be used to recursively estimate the state of a system when the actual state is unknown. The filter makes use of a large number of weighted particles to track the true state of the system. A weighted average of the particles in the filter is then used to produce probabilistic state estimates.

Consider a particle filter as describing a probability distribution over the possible states of the system. At the end of each time step the particle filter represents what is known as the posterior probability distribution. This is the probability that the system is in a particular state given the set of preceding control inputs and state measurements.

The posterior probability distribution is updated using a two stage process at every time step. The first stage is known as the predictive stage and integrates a new control input. A corrective stage then follows that accounts for a new state measurement. Bayes Theorem is used to add new control inputs and state measurements into the particle filter [Thurn 2005].

The control input can be thought of as a measurement of a change in the system state. A state measurement is then an observation of the system state after the change described by the control input. Both the control input and state measurements are assumed to be subject to measurement error. So the particle filter must capture this error when updating the posterior probability distribution.

For this project the state of the system being estimated equates to the location of the UGV. Estimates of the control inputs for the filter are provided by the relative localisation system

onboard the UGV. Measurements of the system state are available from the GPS receiver. Error models have been derived for both control inputs and state measurements.

The particle filter algorithm has five steps that are executed repeatedly. These steps can be summarised as:

i) Translate Particles

ii) Update Particle Weights

iii) Normalise Particle Weights

iv) Resample Particles

v) Estimate System State

At each time step the particles are first translated according to the latest control input for the particle filter. A measurement of the control input is given by the relative localisation equipment onboard the UGV. At the time of translation random noise is added to each particle location to account for the expected measurement error in the data from the sensors.

An example particle translation is shown in Figure 6.12 below. This data was generated in an initial test of the particle filter algorithm. In the graph the blue points represent the particles at one time step and the pink points are the same particles after they have been translated. The control input for this translation had a heading of 90.87° and distance travelled of 0.26m.

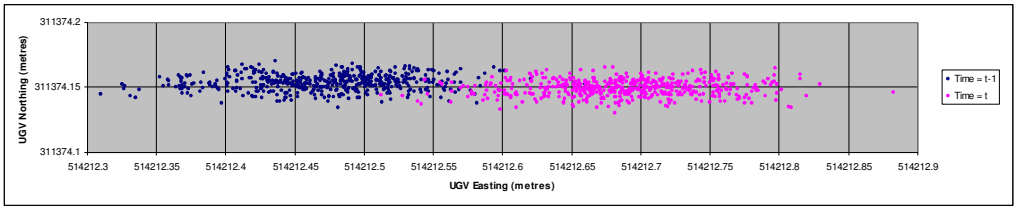

Figure 6.12 - Particle Translation

The translated particles are compared with the latest state measurement in the second step of the algorithm. This state measurement is provided by the GPS receiver. Particle weights are updated in line with probability that the state measurement is true given the particle location. This is known as the measurement probability.

Following the update of the individual particle weights the total weight of the particle set is not guaranteed to be unity. As a result the particle weights must be normalised. This is the third step of the particle filter algorithm.

The fourth step changes the members of the particle population. Particles are resampled from the existing population with the chance of reselection proportional to their weight. This tends to focus the filter on to the particles with higher measurement probability.

Figure 6.13 demonstrates the particle resampling process. Before resampling all of the particles on the graph are within the particle set. After resampling only the pink particles have survived and the blue particles are discarded. On average the blue particles will have lower weight than the pink particles.

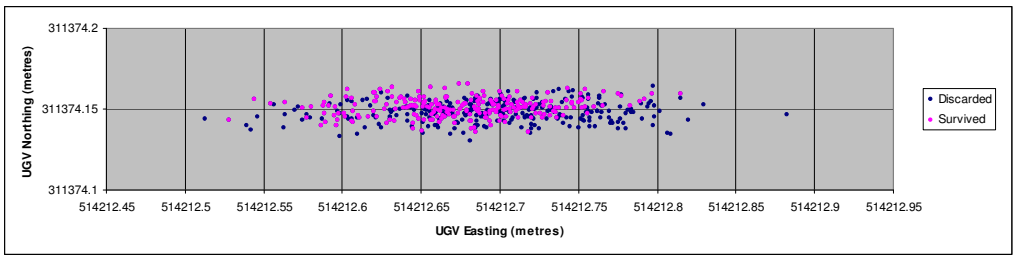

Figure 6.13 - Particle Resampling

The final step of the algorithm is to estimate the system state. This corresponds to estimating the UGV location. The mean of the posterior probability distribution is used as this estimate. It is calculated as a weighted average of the particle locations.

6.3.2 Definition of Particles

The posterior probability distribution for the system state is represented by the set of particles in the filter. So individual particles are like samples from this posterior.

Let the set of particles be denoted by P(t) where t signifies the time step of the particle filter. The particle set contains M particles. As an initial value M was set as 500. Individual particles shall be denoted $p^i(t)$ where i is the index used to identify each particle. This can be expressed as:

$$p^i(t) \in P(t) \qquad (1 \leqslant i \leqslant M)$$

Particles store their location in terms of northing and easting. Let $p^i_n(t)$ represent the northing value and $p^i_e(t)$ represent the easting value. The particle position can then be given in the following vector form:

$$p^i(t) = \begin{bmatrix} p^i_n(t) \\ p^i_e(t) \end{bmatrix}$$

Each particle also has an associated weight. This weight is based on the state measurements made by the GPS receiver. Particle weights estimate the probability that recent state measurements have been true given the location of the particle. The following statement defines the weight of each particle:

$$weight\ of\ p^i(t) = w^i(t)$$

When the particle filter is initialised the UGV has no idea where it is in terms of absolute position. In relative localisation terms it knows that it is at the start location. Hence until valid GPS location and error data is received the particles can only track the relative location of the UGV. All particle locations are initialised to reflect this using the vector:

$$p^i(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Likewise before the GPS receiver has provided a measurement of the system state the weight of the particles cannot be updated. So when the particle filter is initialised the particles are all given equal weight. This is represented by the following expression:

$$w^i(0) = \frac{1}{M}$$

6.3.3 Definition of Control Input

The control input for the particle filter provides a prediction of movement in the system state at each time step. This prediction is subsequently corrected by state measurement. It is not possible to observe the control input directly but it can be measured subject to measurement errors. The relative localisation system provides these measurements.

Control input measurements shall be denoted u(t) where t signifies the time step of the particle filter. Measurements consist of a distance travelled value and a heading value as was introduced in section 6.2.1. Let d(t) represent the distance travelled and θ(t) represent the heading value. The control input measurement can thus be expressed in the following vector form:

$$u(t) = \begin{bmatrix} d(t) \\ \theta(t) \end{bmatrix}$$

Control input measurements need to be converted into an estimate of UGV movement before they can be used by the particle filter. To achieve this task the relative localisation calculation from section 6.2.1 is used. This calculation is encapsulated in a new vector f(u(t)). Within the

new vector $f_n(u(t))$ represents the northing value and $f_e(u(t))$ represents the easting value. Hence the following is an expression of UGV movement since the last time step:

$$f(u(t)) = \begin{bmatrix} f_n(u(t)) \\ f_e(u(t)) \end{bmatrix} = \begin{bmatrix} d(t) \cos(\theta(t)) \\ d(t) \sin(\theta(t)) \end{bmatrix}$$

The control input measurements have a corresponding error model. This model is based on the measurement errors associated with the relative localisation system that were discussed in section 6.2.4. Relative location estimates have an error that can be modelled as a bivariate normal distribution. If the error model is denoted as $e_u(t)$ then it has the following bivariate normal distribution:

$$e_u(t) \sim N(0, C_{Relative})$$

The covariance matrix for the error model is given by $C_{Relative}$. This matrix contains the variance values derived for distance and heading measurement error. As distance and heading measurement errors are uncorrelated the matrix has a null covariance term.

$$C_{Relative} = \begin{bmatrix} \sigma^2_{Distance} & 0 \\ 0 & \sigma^2_{Heading} \end{bmatrix}$$

6.3.4 Definition of State Measurements

State measurements are observations of the system state. They are intended to correct predictions made by the particle filter control input. These measurements correspond to estimates of the UGV location made by the GPS receiver.

Let state measurements be denoted by z(t) where t signifies the time step of the particle filter. Then the northing and easting components of the state measurement can be represented by $z_n(t)$

and $z_e(t)$ respectively. Thus the GPS location estimate may be written in the following vector form:

$$z(t) = \begin{bmatrix} z_n(t) \\ z_e(t) \end{bmatrix}$$

State measurements have a corresponding error model. Since state measurements correspond to GPS location estimates the error model for GPS can be used. This model is based on the GPS error ellipse that was discussed in section 6.1.4. The GPS error ellipse describes the shape of the distribution and corresponds to the GPS covariance matrix. If the error model is denoted as $e_z(t)$ then it has the following bivariate normal distribution:

$$e_z(t) \sim N(0, C_{GPS})$$

In this case we assume the GPS covariance matrix to be:

$$C_{GPS} = \begin{bmatrix} \sigma^2_{Northing} & \sigma_{Cov} \\ \sigma_{Cov} & \sigma^2_{Easting} \end{bmatrix}$$

This is possible as the location estimate error information from the GPS receiver is already given in metres. So it is reasonable to say:

$$\sigma_{Latitude} \approx \sigma_{Northing}$$

$$\sigma_{Longitude} \approx \sigma_{Easting}$$

6.3.5 Fusion of Initial GPS Estimate

There is normally a short delay after the GPS receiver is powered up before it begins to make location estimates. Hence state measurements may not be available from the GPS receiver when the particle filter is first initialised. Until the GPS receiver is ready the particle filter

maintains the relative location of the particles. During this period particle weights are not updated and no particle resampling takes place.

When the first GPS location estimate with error information is available the entire particle set is translated. The first GPS location estimate is added to each particle location. This ensures that all particles are now absolute location estimates and that any UGV movement already captured by the relative localisation is maintained.

To account for the uncertainty associated with the GPS location estimate noise is also added based on the initial GPS error information. The translation from relative to absolute location can thus be represented by the following expression:

$$p^i(t) = p^i(t) + z(t) - e_z(t)$$

To simplify the error generation process it is assumed that the error covariance for GPS is negligible. This seems reasonable with reference to Figure 6.9. Two random values n_1 and n_2 can then be generated from a standard normal distribution. If these random values are then scaled by the standard deviation of GPS error in northing and easting respectively they generate the coordinates of a point from the error distribution. So the fusion expression becomes:

$$p^i(t) = \begin{bmatrix} p^i_n(t) \\ p^i_e(t) \end{bmatrix} + \begin{bmatrix} z_n(t) \\ z_e(t) \end{bmatrix} - \begin{bmatrix} n_1 \sigma_{Northing} \\ n_2 \sigma_{Easting} \end{bmatrix}$$

6.3.6 Translating Particles

At each time step the particle set is translated. Particle translations are made in line with the filter control input. This is known as the predictive stage of the particle filter.

A measurement of the unknown control input is given by the relative localisation equipment onboard the UGV. This is used to replicate the dead reckoning calculation introduced in section 6.2.1. The general particle translation function is given by the following equation:

$$p^i(t) = p^i(t-1) + f\left(u(t) - e_u(t)\right)$$

For each particle the distance travelled and UGV heading values from the control input are coloured with noise. In this way the particle filter accounts for the likely measurement errors. The noise added is based on the error model for the control input.

In practice adding the noise requires generation of a pair of values from a standard normal distribution. So if n_1 and n_2 are new standard normal values generated for this stage of the particle filter then the particle translation equation can be written as:

$$p^i(t) = \begin{bmatrix} p_n^i(t-1) \\ p_e^i(t-1) \end{bmatrix} + \begin{bmatrix} (d(t) - n_1\sigma_{Distance}) \cos(\theta(t) - n_2\sigma_{Heading}) \\ (d(t) - n_1\sigma_{Distance}) \sin(\theta(t) - n_2\sigma_{Heading}) \end{bmatrix}$$

6.3.7 Updating Particle Weights

Updating the particle weights is the corrective stage of the filter. In this way new state measurements from the GPS receiver are incorporated into the particle filter. Each particle weight represents the probability of the state measurement being true given the location of the corresponding particle. This probability is known as the measurement probability.

At every time step the weight of individual particles is scaled by the measurement probability. Consecutive state measurements are incorporated by multiplication. Hence the equation for updating particle weights is given by:

$$w^i(t) = w^i(t-1)\ Prob\left(z(t) \mid p^i(t)\right)$$

There is an alternative way to say the probability that the state measurement is true given the location of the particle. It is the same as the probability that the error in the state measurement is equal to the difference between the GPS location estimate and particle location. This can be stated as:

$$Prob\left(z(t) \;/\; p^i(t)\right) = Prob\left(e_z(t) = z(t) - p^i(t)\right)$$

GPS location estimates have an error that can be modelled as a bivariate normal distribution. This distribution is centred at zero error and has a spread described by the GPS covariance matrix. Hence knowing details of the GPS error ellipse that was discussed in section 6.1.4 of this chapter is essential to calculating this probability of error. The GPS error ellipse describes the shape of the distribution and corresponds to the GPS covariance matrix.

In this case we can assume the covariance matrix to be:

$$C_{GPS} = \begin{bmatrix} \sigma^2_{Northing} & \sigma_{Cov} \\ \sigma_{Cov} & \sigma^2_{Easting} \end{bmatrix}$$

Hence the inverse of the GPS covariance matrix is then given by:

$$C^{-1}_{GPS} = \frac{1}{\sigma^2_{Northing}\sigma^2_{Easting} - \sigma^2_{Cov}} \begin{bmatrix} \sigma^2_{Easting} & -\sigma_{Cov} \\ -\sigma_{Cov} & \sigma^2_{Northing} \end{bmatrix}$$

So the error probability can be expressed by the following equation where η is a constant that is corrected for in a later normalisation process:

$$Prob\left(e_z(t) = z(t) - p^i(t)\right) = \eta \;\exp\left(-\frac{1}{2}\;(z(t) - p^i(t))^T\; C^{-1}_{GPS}\;(z(t) - p^i(t))\right)$$

The key part of this expression is the matrix multiplication within the exponent. This can expanded to give:

$$(z(t) - p^i(t))^T \; C_{GPS}^{-1} \; (z(t) - p^i(t)) =$$

$$\frac{1}{\sigma^2_{Northing}\sigma^2_{Easting} - \sigma^2_{Cov}} \begin{bmatrix} z_n(t) - p^i_n(t) \\ z_e(t) - p^i_e(t) \end{bmatrix}^T \begin{bmatrix} \sigma^2_{Easting} & -\sigma_{Cov} \\ -\sigma_{Cov} & \sigma^2_{Northing} \end{bmatrix} \begin{bmatrix} z_n(t) - p^i_n(t) \\ z_e(t) - p^i_e(t) \end{bmatrix}$$

Performing the matrix multiplication then yields:

$$\frac{\left(z_n(t) - p^i_n(t)\right)^2 \sigma^2_{Easting} - 2\left(z_n(t) - p^i_n(t)\right)\left(z_e(t) - p^i_e(t)\right)\sigma_{Cov} + \left(z_e(t) - p^i_e(t)\right)^2 \sigma^2_{Northing}}{\sigma^2_{Northing}\sigma^2_{Easting} - \sigma^2_{Cov}}$$

This is the unwieldy formula that is used within the particle filter. However if the covariance of the GPS error distribution is assumed to be negligible the last equation statement simplifies to:

$$\left(z_n(t) - p^i_n(t)\right)^2 \sigma^2_{Northing} + \left(z_e(t) - p^i_e(t)\right)^2 \sigma^2_{Easting}$$

The consequences of making this simplifying assumption were tested by experiment. The outcome of this experiment is detailed in Chapter 8.

6.3.8 Normalisation of Particle Weights

Particle weights indicate the probability that latest state measurement was true given the location of the particle. By the law of total probability these weights must sum to 1. Typically after the particle weights have been updated this would not be true.

To account for this each of the particle weights must be normalised. First the total weight of the particle set is found by summation:

$$total\ weight = \sum_{i=1}^{M} w^i(t)$$

Individual particle weights are then normalised by dividing by the total weight of the particle set:

$$normalised\ weight = unnormalised\ weight \ / \ total\ weight$$

6.3.9 Resampling Particles

This stage of the particle filter is not performed at every time step. Resampling is triggered by a fall in the proportion of effective particles in the particle filter [Arulampalam 2002]. A small proportion of effective particles implies that the particle filter has degenerated meaning many particles have very low weight. In this case the particles are representing regions of the state space that have low probability.

The number of effective particles is calculated by finding reciprocal of the sum of squared particle weights.

$$effective\ particles = 1 \ / \ \sum_{i=1}^{M} (w^i(t))^2$$

Initially the minimum allowed proportion of effective particles in the filter has been chosen to be 2/3. This value was proposed in other particle filter work [Ceranka 2003]. A higher proportion would imply that resampling would take place more often and so the particle filter location estimate would tend to follow the GPS more closely. By contrast a lower value would allow particles to spread more and be less affected by changes in the GPS state measurements. Thus particles would be resampled less frequently and get less pull toward the GPS location estimates.

The idea behind the resampling process is survival of the fittest. Particles with low weight are likely to be discarded. As such resampling refocuses the particle filter to states with higher posterior probabilities.

Resampling is performed with replacement on the particle set. First a uniformly distributed random decimal is generated from the range [0, 1]. Let u denote this random number.

$$u \sim U[0,1]$$

Next particle weights are summed in order until the cumulative weight of particles is greater than or equal to the random number. This condition is represented by the following statement:

$$\min_{j} \sum_{i=1}^{j} w^{i}(t) \geqslant u \qquad (1 \leqslant j \leqslant M)$$

The particle that makes the cumulative weight at least the value of the random number is selected and added to the resampled particle set.

This resampling process is repeated until a completely new particle set has been generated from the old particle set. In theory the new particle set should contain clones of the fitter particles which will account for their original higher weighting. Thus the final stage of the resampling process is to assign equal weight to each of resampled particles.

$$w^{i}(t) = \frac{1}{M}$$

6.3.10 Estimating UGV position

The final step of the particle filter algorithm is to estimate the state of the system. This of course corresponds to estimating the UGV location. The basis for the estimate is the average

particle location which estimates the mean of the posterior probability distribution for the particle filter.

The following vector denotes the average particle location at time t and shows that it consists of components of northing and easting:

$$\bar{p}(t) = \begin{bmatrix} \bar{p}_n(t) \\ \bar{p}_e(t) \end{bmatrix}$$

Average particle location is calculated using a simple sum of particle weight multiplied by particle position for each particle. The two equations below show the calculation for the northing and easting components of the average particle location:

$$\bar{p}_n(t) = \sum_{i=1}^{M} w^i(t)\, p_n^i(t) \qquad \bar{p}_e(t) = \sum_{i=1}^{M} w^i(t)\, p_e^i(t)$$

6.3.11 Generating Random Normally Distributed Numbers

As can be seen in the preceding subsections the particle filter algorithm requires normally distributed random numbers to be generated at each time step. Computers are good at generating uniformly distributed pseudo random numbers. However special algorithms are typically required to generate random numbers from other distributions.

Here a rejection sampling method is introduced to generate random normal values. This method is known as the Box-Muller Polar Method. The generated values come from a normal distribution with mean 0 and variance 1 [Knuth 1998].

First two random values are generated from a uniform distribution as a basis for the required normal random values. These random values are scaled to be decimals in the [-1,1] range. Call these values u_1 and u_2 then:

$$u_1 \sim U[-1,1] \quad u_2 \sim U[-1,1]$$

Consider these numbers as specifying the coordinates of a point in 2D. Then the distance the point lies from the origin can be found using Pythagoras Theorem. In polar coordinate terms this is the radius of the point. Hence the radius can be expressed by the following equation:

$$radius = \sqrt{u_1^2 + u_2^2}$$

Next the radius is tested to check if it is non zero and also less than one. This is equivalent to testing if the point lies within the unit circle but is not at the origin. If either test is failed the point is rejected and two new values must be generated from the uniform distribution.

Once a point within the unit circle is generated the Polar Box-Muller Transform is calculated. This transform is a function of the radius and is specified by the expression below:

$$Polar\ Box\ Muller\ Transform = \sqrt{\frac{-2\ln(radius^2)}{radius^2}}$$

The transform is then applied to each of the coordinates of the point. This generates two new values n_1 and n_2:

$$n_1 = u_1 \times Polar\ Box\ Muller\ Transform$$

$$n_2 = u_2 \times Polar\ Box\ Muller\ Transform$$

These new values should then be normally distributed with mean 0 and variance 1:

$$n_1 \sim N(0,1) \qquad n_2 \sim N(0,1)$$

7 OBSTACLE AVOIDANCE

This chapter introduces the obstacle avoidance strategy that was devised for this research and is implemented within NOVA. The process of obstacle avoidance has three steps, processing range sensor data, obstacle mapping and local path planning. For each step the idea was to achieve a basic capability then try different ways to expand this capability.

In the following sections the steps of the obstacle avoidance process are looked at in turn. First methods for processing range sensor data are described. This builds three sensor models to reflect the different obstacle sensing methods that were available for use on the ARP vehicle. A simple 2D obstacle detection system is provided by the first of these models. The other models allow obstacle detection in 3D.

Next the obstacle mapping system is introduced. A basic 2D obstacle map design is presented first. This map is for use with the 2D obstacle detection system. A more advanced 3D version of obstacle map designed for use with the 3D obstacle detection models is then discussed. Both systems provide a way to store the locations of obstacles detected around the UGV.

The local path planning strategy is then detailed. This path planner has been devised to work equally well with either the 2D or 3D obstacle mapping system. The role of the planner is to guide the UGV towards an operator waypoint while avoiding any detected obstacles.

Finally the concept of an exploration graph is explained. This is an extension to the basic local path planner. Exploration graphs are used to store the decisions made by the local path planner. They allow the UGV to effectively explore and escape from maze like arrangements of obstacles.

7.1 RANGE DATA PROCESSING

Within NOVA all processing of range sensor data is performed by the Protection Module. This architectural module has two key roles. The first is to generate range points in 3D coordinates from raw sensor data. It must also account for the pose of the sensor on the UGV.

A sensor model is required to generate range points from sensor data. For this research three separate sensor models have been devised. Two of these models are for use with a Sick LMS 200 laser scanner. The third is for use with a PMD [vision]® 1k-S range camera.

The first sensor model created for the Sick LMS 200 allows for 2D obstacle detection. This provides a basic capability that gave a foundation for the complete obstacle avoidance system. The second model for the Sick LMS 200 allows the sensor to be used for 3D obstacle mapping when mounted appropriately. 3D obstacle mapping is also possible using the sensor model for the PMD [vision]® 1k-S range camera.

Regardless of which sensor is used the range points generated by the Protection Module portray the closest objects within the current field of view for the sensor. For the 2D sensor model all generated range points represent obstacles that must be avoided. In contrast for the 3D sensor model the range points only represent the surface in front of the UGV which may or may not be traversable.

In order to use range points for obstacle detection and mapping they must first be transformed to account for the pose of the sensor on the UGV. When range points are generated they are initially specified in the coordinate frame of the sensor. So a transformation is required within each sensor model to take the range points from the sensor coordinate frame into a UGV centred coordinate frame.

To simplify the coordinate transformation it is split into stages and each is performed using a homogenous transform [Kelly 1994]. The first stage compensates for the pitch of the sensor relative to the UGV. This transforms the range points from the sensor coordinate frame to the sensor mount coordinate frame. In the second stage the range points are transformed to the UGV coordinate frame. This translates range point coordinates to account for the location of the sensor mount relative to the turning centre of the UGV.

The pose of the UGV also needs to be considered before new range points can be added to an obstacle map. For the 3D sensor models this means each range point must be transformed to compensate for the measured pitch and roll of the UGV. This transform takes range points from the UGV coordinate frame to the so called ground coordinate frame.

All sensor models then feature a final transform to account for the UGV heading and ensure the coordinate frame is north oriented. Range points are taken to the map coordinate frame by this final transform. The map coordinate frame is used by the obstacle mapping system described later in this chapter.

7.1.1 Horizontal LMS 200 Model

This subsection introduces the basic sensor model for use with a Sick LMS 200 laser scanner. A description of the characteristics of the LMS 200 can be found in section 5.2.1. For the purposes of this model it suffices to know that the sensor has a semicircular 180° planar field of view.

The basic model assumes that the sensor is mounted on the UGV so that the scanned plane is horizontal. As a consequence all generated range points can be assumed to have similar height values. Thus obstacle detection is only possible in 2D using this model.

The 2D nature of obstacle detection means that when using this model the UGV should only be operated on relatively flat ground. If the UGV drives onto a slope the sensor may see the ground and detect it as an obstacle. Additionally the sensor will not be able to see negative obstacles or positive obstacles that are below the sensor plane.

Range data is continuously streamed from the laser scanner to the NOVA Computer onboard the ARP vehicle. This data encodes each 180° scan as 361 range values taken at 0.5° angular increments moving anticlockwise from the right hand extreme. The sensor model converts this raw data to a sequence of 2D coordinates.

For the LMS 200 the origin of the coordinate frame is taken as the rotating mirror within the sensor. The positive x-axis is assumed to go forward from the sensor along the centre line and the positive y-axis goes out from the right of the sensor. This means that the generated range points are then within ±90° of the positive x-axis. Figure 7.1 illustrates the sensor coordinate frame.

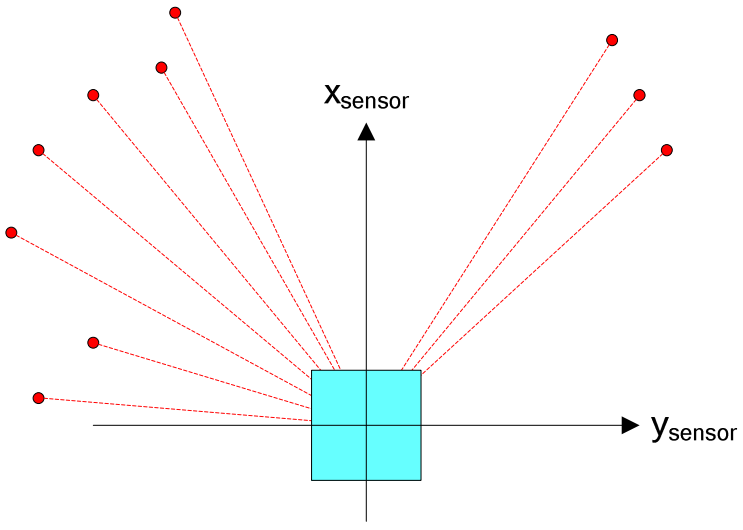

Figure 7.1 - Basic Sensor Model of LMS 200

The sensor model uses the following trigonometric equations to convert each of the range values r_i into 2D coordinates in the sensor coordinate frame:

$$\begin{bmatrix} x_{sensor} \\ y_{sensor} \end{bmatrix} = r_i \begin{bmatrix} \sin(0.5i) \\ \cos(0.5i) \end{bmatrix}$$

$$\forall i \; : \; 0 \leqslant i \leqslant 360$$

With this sensor model only two steps are required to transform the range points from the sensor coordinate frame to the map coordinate frame. This is because the sensor is mounted horizontally so the sensor mount coordinate frame is identical to the sensor coordinate frame. So for all 361 range points:

$$\begin{bmatrix} x_{sensor\ mount} \\ y_{sensor\ mount} \end{bmatrix} = \begin{bmatrix} x_{sensor} \\ y_{sensor} \end{bmatrix}$$

The transform to UGV centred coordinates requires an offset to be added to the x coordinate of each range point. This offset is equal to the distance the sensor mount lies in front of the UGV centre of rotation. For the ARP vehicle this value is 0.66m. No offset is required for the y coordinate as the sensor is mounted on the UGV centre line. So each range point can be transformed by:

$$\begin{bmatrix} x_{ugv} \\ y_{ugv} \end{bmatrix} = \begin{bmatrix} x_{sensor\ mount} + offset \\ y_{sensor\ mount} \end{bmatrix}$$

Finally the range points are rotated around the centre of the UGV to account for the UGV heading θ. This transform takes the range points to the map coordinate frame which has axes pointing north and east. The following equations are used to transform each range point:

$$\begin{bmatrix} n_{map} \\ e_{map} \end{bmatrix} = \begin{bmatrix} x_{ugv} \cos(\theta) - y_{ugv} \sin(\theta) \\ x_{ugv} \sin(\theta) + y_{ugv} \cos(\theta) \end{bmatrix}$$

7.1.2 Push Broom LMS 200 Model

This subsection introduces the second sensor model devised for use with the Sick LMS 200 laser scanner. For this model it is assumed that the scanner is mounted on the UGV in a push broom configuration. This means that the scanning plane is pitched down towards the ground as illustrated in Figure 7.2.

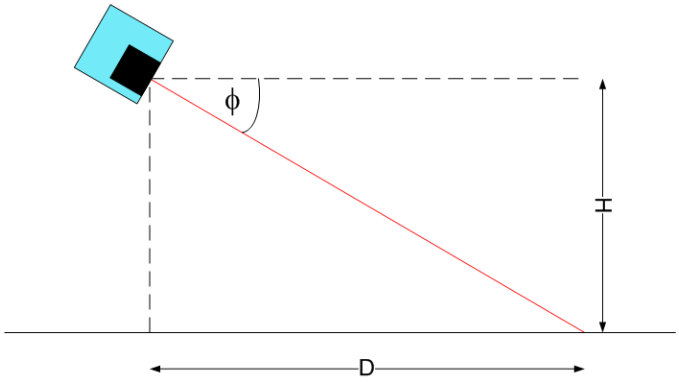

Figure 7.2 - Calculating Ground Intersection

With the laser scanner mounted in this fashion as the UGV drives forward it scans cross sections of the ground. Successive scans can be used to build a 3D model of the ground the UGV is approaching. To do this the sensor model needs to generate range points with 3D coordinates. A 3D obstacle detection process is then possible.

Using this sensor model a UGV would not have to be restricted to relatively flat ground. It would be able to detect both positive and negative obstacles that the UGV approached. The only limitation is that UGV would have to move towards an obstacle in order to detect it.

An important question here is how does the pitch of the scanner affect how far the obstacle detection system can see ahead? Assuming the UGV is on flat ground then this can be easily calculated using trigonometry. The scene is shown in Figure 7.2. Let ϕ denote the pitch of the

scanner below the horizontal. The height of the scanner above the ground is H which is 0.435m on the ARP vehicle. Thus the intersection of scanner plane with ground is given by:

$$D = \frac{H}{\arctan(\phi)}$$

Figure 7.3 shows the calculated ground intersection for the scanner plane at pitch angles up to 45°. It is fair to say that the shallower the pitch the further the sensor is looking ahead. However this longer look ahead comes at the cost of reduced obstacle detection sensitivity. Figure 7.4 shows how far the UGV must travel forward in order to detect a discrete obstacle 100mm in height. It should be noted that Figure 7.4 is actually a scaled version of Figure 7.3.

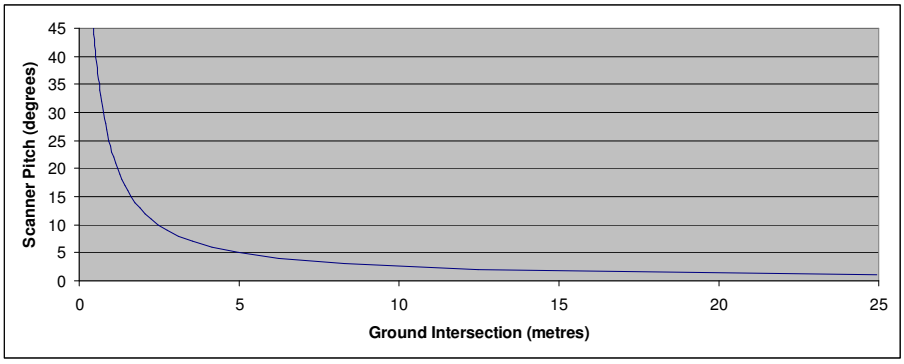

Figure 7.3 - Effect of Pitch on Ground Intersection of Scanning Plane

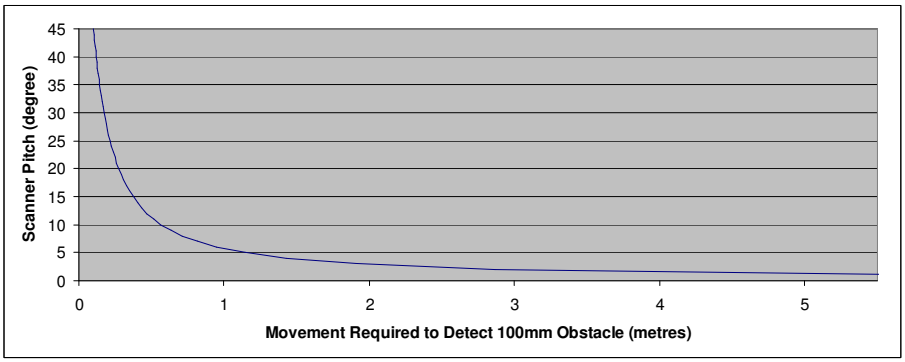

Figure 7.4 - Obstacle Detection Sensitivity

So it is important to strike a balance between the distance the sensor is looking ahead and the distance the UGV must travel in order to detect obstacles. An initial scanner pitch value of 11° was used for experimentation and this is reflected in the sensor model. This implies the UGV has to travel around 0.5m forwards in order to detect an obstacle 100mm in height.

At this stage it will become clear that this sensor model is simply an extension of the basic LMS 200 sensor model. To derive the initial 3D coordinates for each of the range points the sensor model uses the same method as in the 2D case. The z coordinate for each range point is set to zero. Thus the equations to transform each of the range values r_i into xyz-triplets are:

$$\begin{bmatrix} x_{sensor} \\ y_{sensor} \\ z_{sensor} \end{bmatrix} = \begin{bmatrix} r_i \ \sin(0.5\mathrm{i}) \\ r_i \ \cos(0.5\mathrm{i}) \\ 0 \end{bmatrix}$$

$$\forall i \ : \ 0 \leqslant i \leqslant 360$$

Next the range points must be transformed from the sensor coordinate frame to the map coordinate frame. Unlike the basic 2D model the 3D case requires a four step coordinate transform. The first step is a rotation about the y-axis to account for the pitch of the scanner $\phi = 11°$. The homogenous transform to take each range point to the sensor mount coordinate frame is thus given by:

$$\begin{bmatrix} x_{sensor\,mount} \\ y_{sensor\,mount} \\ z_{sensor\,mount} \end{bmatrix} = \begin{bmatrix} x_{sensor} \ \cos(\phi) \\ y_{sensor} \\ -x_{sensor} \ \sin(\phi) \end{bmatrix}$$

A translation is then required to account for the location of the sensor mount relative to the UGV centre of rotation. As in the basic sensor model the x-coordinate of the 361 range points must be offset by 0.66m to account for the fact that the sensor is mounted at the front of the

UGV. The z-coordinate of each range point is also moved vertically by 0.435m the height of the sensor mount above the ground. Thus the transformation is given by:

$$\begin{bmatrix} x_{ugv} \\ y_{ugv} \\ z_{ugv} \end{bmatrix} = \begin{bmatrix} x_{sensor\ mount} + offset \\ y_{sensor\ mount} \\ z_{sensor\ mount} + height \end{bmatrix}$$

Once range points have been transformed to the UGV coordinate frame it is necessary to compensate for the pose of the UGV on the ground. Range points must be rotated around y axis to correct for UGV pitch then rotated around the resulting x axis to correct for UGV roll.

Estimates of UGV pitch and roll relative to the horizontal are provided by the pose sensor on the ARP vehicle. Let α denote the pitch angle of the UGV and β denote the roll angle as measured by the pose sensor. Then a pair of homogenous transformations can be used to transform from the UGV coordinate frame to the ground coordinate frame. These two transformations can be combined into a single step given by following equation:

$$\begin{bmatrix} x_{ground} \\ y_{ground} \\ z_{ground} \end{bmatrix} = \begin{bmatrix} x_{ugv}\cos(\alpha) + z_{ugv}\sin(\alpha) \\ y_{ugv}\cos(\beta) - (-x_{ugv}\sin(\alpha) + z_{ugv}\cos(\alpha))\sin(\beta) \\ y_{ugv}\sin(\beta) + (-x_{ugv}\sin(\alpha) + z_{ugv}\cos(\alpha))\cos(\beta) \end{bmatrix}$$

The final step in the coordinate transform is also identical to the basic sensor model. Range points are rotated around the z-axis to account for the heading of the UGV. This leaves the z-coordinates of the range points unchanged yielding the 3D map coordinate frame which has axes pointing north, east and up. The final transform is given by the following equations:

$$\begin{bmatrix} n_{map} \\ e_{map} \\ z_{map} \end{bmatrix} = \begin{bmatrix} x_{ground}\cos(\theta) - y_{ground}\sin(\theta) \\ x_{ground}\sin(\theta) + y_{ground}\cos(\theta) \\ z_{ground} \end{bmatrix}$$

7.1.3 PMD [vision]® 1k-S Camera Model

A sensor model for the PMD [vision]® 1k-S camera is introduced in this subsection. This alternative sensor produces range images with a resolution of 16 x 64 pixels. These images have a 12° vertical field of view and 34° horizontal field of view. For a full description of the range camera see section 5.2.2.

This sensor model generates 3D range points from each frame of camera range data. In this way a camera frame becomes a 3D picture of the ground the UGV is approaching. Thus using this sensor model 3D obstacle detection is possible.

Using this model the UGV is not restricted to operation on flat ground. As with the last sensor model both positive and negative obstacles should be detectable. The major advantage over the push broom Sick LMS 200 model is that UGV does not have to move in order to detect obstacles.

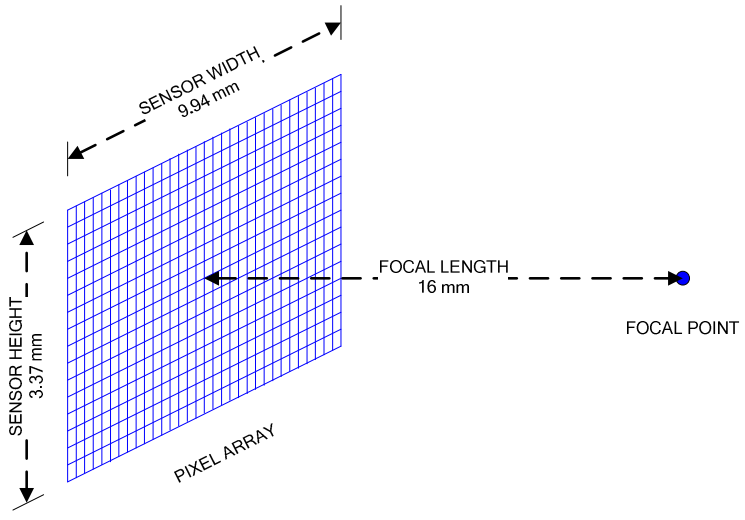

Figure 7.5 - Visualisation of Pixel Array and Focal Point

Dimensions of the pixel array and camera focal length for the PMD [vision]® 1k-S are known [PMD 2005]. The array has a height of 3.37mm split equally into columns of 16 pixels. The

width of the pixel array is 9.94mm and is split equally into rows of 64 pixels. Finally the focal length of the camera is given as 16mm. These key dimensions are depicted in Figure 7.5.

This means that the coordinates for the centre of each of the 1024 camera pixels can be calculated in 3D. For this it is assumed that the focal point of the camera is the origin of the coordinate system. Due to the large number of pixels these calculations are performed off line.

Next the coordinates for each pixel are normalised so the length of the vector from the focal point to each pixel centre is 1m. This is the same as the 3D coordinates of a range point at a distance of 1m in front of the focal point of the camera. So the coordinates of points at arbitrary distances from the camera can be calculated by simply multiplying each coordinate by the required distance.

These normalised 3D pixel coordinates are stored in the software as a look up table. The look up table is used to transform pixel range values from each camera frame into 3D range points. In this way range points are generated in the sensor coordinate frame with very little computational overhead.

Next the range points must be transformed to account for the pose of the sensor on the UGV and also the pose of the UGV on the ground. The transform from the sensor coordinate frame to the map coordinate frame follows the same method used in the push broom Sick LMS 200 sensor model. In this case the sensor pitch value ϕ will be 6° for initial experimentation. This ensures that the top of the camera field of view is horizontal so the camera looks mainly at the ground in front of the UGV.

7.2 OBSTACLE MAPPING

The last section showed how range sensor data can be transformed into range points specified in map coordinates. As the UGV moves around it will generate an ever expanding cloud of these range points. So methods to rationalise this point cloud and detect obstacles within it are necessary.

Individual frames of range sensor data cannot be used as a basis for obstacle detection and avoidance. None of the sensors used in this research produce complete high resolution images of the environment around the UGV. Sensor imagery is restricted to either a cross section of the environment from the Sick LMS 200 or a narrow low resolution 3D image from the PMD camera.

What is required is a method to store and relate multiple frames of range sensor data. Caution must also be taken in how the data is stored as the range sensors rapidly generate range points. The Sick LMS 200 laser scanner will produce in excess of 10000 range points each second and the PMD [vision]® 1k-S camera has the potential to produce over 50000 points a second.

In this research two types of obstacle map have been devised to store range points. The first is a 2D obstacle map and is for use with the horizontally mounted 2D Sick LMS 200 and the corresponding basic sensor model. The second is a 3D obstacle map and is designed for use with a 3D sensor model and the corresponding sensor.

The obstacle map represents the area around the UGV in the real world and divides this space up into small square cells. As new range points are added to the obstacle map traversability analysis is performed on each cell. This results in the cell being classified as untraversable if it contains an obstacle. Figure 7.6 illustrates the idea with green cells being traversable ground and red cells containing obstacles.

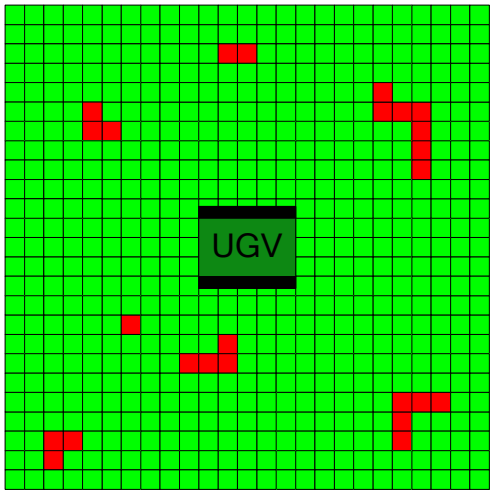

Figure 7.6 - Obstacle Map Idea

The 2D obstacle mapping system is based on the popular idea of confidence mapping. If an obstacle is detected within a cell on the obstacle map the obstacle confidence value for that cell is increased. If the space represented by the cell on the map is observed but no obstacles are seen the obstacle confidence value is decreased. There is some similarity between the method introduced here and the concept of a histogram grid [Borenstein 1991].

This 2D system was devised to provide a basic obstacle mapping capability. It is based on the implicit assumption that the ground around the UGV is relatively flat due to the planar nature of the associated range sensor. The attractive features of the system are that it is fast updating, robust to sensor noise and has a capability to handle moving obstacles.

In contrast the 3D obstacle mapping system is based on the idea of voxel mapping. Each obstacle map cell contains a record of the recent range points that fell into the cell. Recorded range point coordinates are then used to calculate traversability statistics for the corresponding cell and classify the cell as traversable or untraversable. In other research the bucket map implemented on the CMU NAVLAB vehicle is the closest similar system [Hebert 1998].

Effectively the 3D mapping system maintains a sparse model of the ground around the UGV. The cell traversability statistics indicate if a step or a hole is detected in the surface described by the range points. This system should allow the UGV to operate in more rugged terrain than the 2D mapping system.

The remainder of this section is split into three parts. First the common implementation details of the two mapping systems are discussed. Then the 2D obstacle mapping strategy is introduced. Finally the design of the 3D obstacle mapping system is presented.

7.2.1 Common Implementation Details

Within NOVA obstacle mapping for the UGV has been realised using a local obstacle map. The map is north oriented and scrolls as the UGV moves through the environment. Scrolling allows the map to be kept roughly centred on the location of the UGV. Thus the obstacle map acts as a short term memory of range sensor data for the moving UGV.

This subsection discusses how the general obstacle mapping concept is implemented. The features introduced here are common to both the 2D and 3D obstacle mapping systems. This is because the 3D mapping system was developed to be a direct replacement for the 2D system when the UGV is equipped with a means to generate 3D range images.

Map size and granularity are key considerations when designing an obstacle map. For a given granularity the larger the map the more storage space it will require and the longer it will take to update. Likewise increasing the map granularity will also increase required storage space and update time. Obstacle mapping and avoidance have to run in real time. This is because avoidance decisions must be made by the UGV before colliding with any obstacles that have been sensed.

The Sick LMS 200 has a range of 8m while the PMD [vision]® 1k-S has a non-ambiguity range of 7.5m. When moving into unobserved territory there is little point in having a map that extends further than the sensor range in front of the UGV. Also range points observed at the extreme sensor range are often the result of noise. Hence it was thought best to ensure the map did not show the extremes of the sensor range.

A local 10m x 10m obstacle map centred on UGV has been chosen for initial experiments. This ensures that a distance of at least 5m in front of the UGV is shown by the obstacle map regardless of the UGV heading. Given the speed of the UGV this should be more than adequate.

The UGV localisation calculation is performed at a rate of 5Hz and the top speed of the UGV is around 0.5 ms^{-1}. Hence the UGV can move at most 0.1m between localisation updates. This represents the worst case of error when new range points are integrated with the obstacle map. So using map cells any narrower than 0.1m could lead to some amount of obstacle map smearing.

Map cells with a 20cm x 20cm base have been trialled. Larger values seemed inappropriate since the ARP vehicle is only 0.66m wide and the vehicle needed to be capable of operation in tight spaces. This means the 10m x 10m obstacle map has 2500 cells. Preliminary experiments showed that the NOVA computer has sufficient speed to maintain an obstacle map of this size.

Due to the small size of the obstacle map the contents of the map must be translated as the UGV moves. With this translation it is important to maintain the observed spatial relationships between the UGV and all recorded range sensor data. Scrolling of the obstacle map thus

happens at a rate to ensure the map remains roughly centred on the latest position reported by the UGV localisation system.

Map scrolling is performed by shifting all of the cells in the obstacle map together. Each shift is done in steps the same size as an obstacle map cell. This makes it easy to preserve spatial relationships between the data recorded in the obstacle map. Cells will be shifted either north, south, east or west approximately in the opposite direction to the UGV heading.

Once a row or column of map cells is scrolled outside the defined map area all information for those cells is discarded. As these cells are discarded new cells are added to the leading edge of the map. New cells contain no obstacle information and require range sensor data to be gathered from the corresponding space. Figure 7.7 illustrates the scrolling process. In the diagram green cells are traversable while red cells contain obstacles. The black cells are new and unobserved.

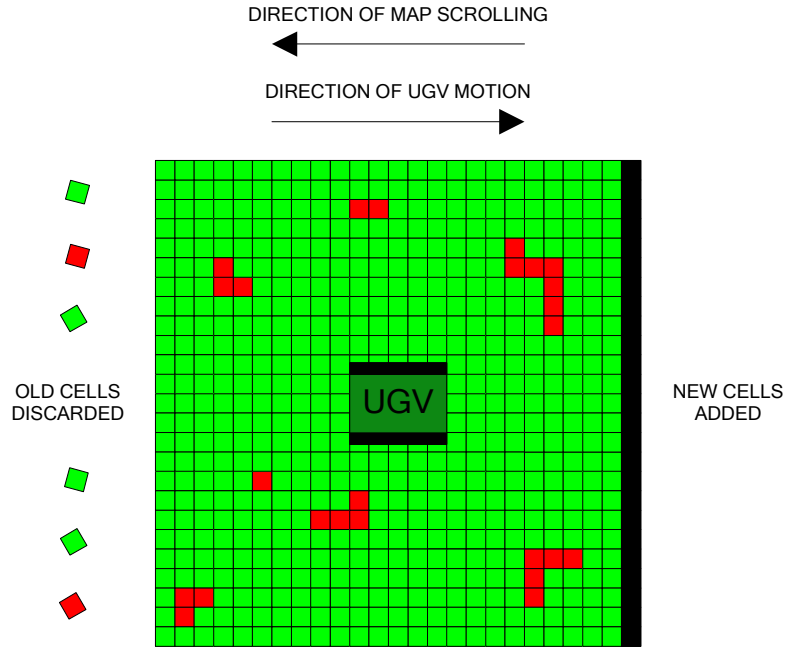

Figure 7.7 - Example of Map Scrolling

Obstacle map scrolling is performed by the NOVA Protection Module. Each time new range sensor data arrives the map is scrolled if required. Since the map is scrolled in discrete steps it is necessary to let the UGV roam around the centre of the obstacle map. Map scrolling is then triggered when the location of the UGV moves more than one cell from the centre of the obstacle map.

An aspiration with the obstacle mapping system is to try mapping an entire UGV mission on a continuous map. The UGV test area is around 40m wide so maps up to this width could be tried. Such a map would have 40000 cells. It was expected that the mapping system would have a much lengthier update process and will require much more memory using this size of map. In Chapter 8 an experiment is described that used a much larger obstacle map to see if mission mapping was plausible using the system devised in this research.

7.2.2 2D Confidence Mapping

The 2D obstacle map is based on three separate mapping arrays. Each array represents the same area of ground but stores different attributes of the map cells. Together the arrays summarise all that is known about the real world space that corresponds to each cell and they form a coherent obstacle map.

An example of the three mapping arrays is shown in Figure 7.8. On the left is the confidence map. This array holds obstacle confidence values associated with each map cell. The array in the centre is known as the observation mask and indicates which map cells have been observed by the UGV. Finally the traversability summary is shown on the right. The traversability summary shows the latest traversability classification of the map cells.

As new frames of range sensor data arrive they are processed using the appropriate sensor model. All generated range points are then incorporated immediately into the confidence map

by updating the obstacle confidence values associated with each cell. Obstacle confidence values are measure of how likely the space represented by the cell is to contain an obstacle.

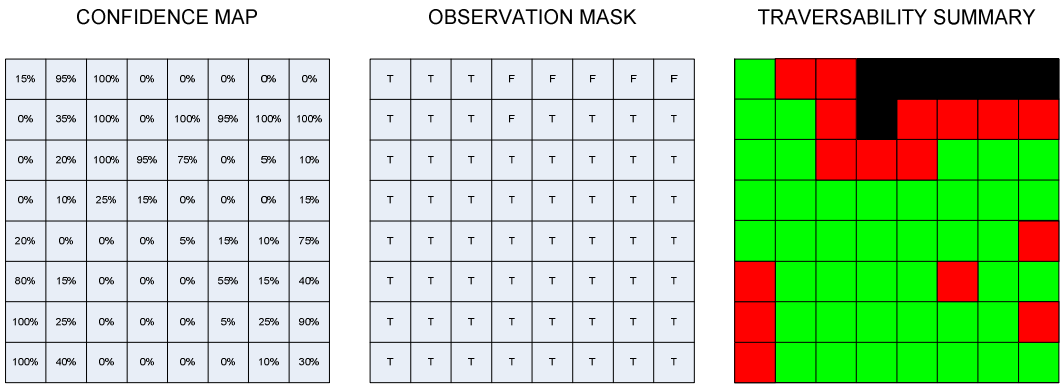

Figure 7.8 - Example of Confidence Mapping System

The confidence map stores the obstacle confidence value for each obstacle map cell as a percentage. As such individual confidence values have a minimum value of 0% and a maximum value of 100%. The confidence map is thus a very lean way of representing a large amount of range sensor data.

A mapping array known as the momentary map is used to update confidence map values. The momentary map is like an instantaneous obstacle map representing only the latest sensor frame. Each cell of the momentary map can represent an obstacle, clear space or unobserved space. The content of the momentary map cells is determined based on the coordinates of the range points from the latest sensor frame.

To populate the momentary map each range point from the latest sensor frame is considered in turn. The coordinates of range point are tested and the corresponding cell of the momentary map is found. If a range point falls in a cell the confidence value for the cell should increase. Ray casting is then used to indicate the cells between the range point and sensor are obstacle

free. If a ray passes through a cell the confidence value for the cell should decrease provided it does not already contain a range point.

Populated momentary map cells can contain one of three possible values. If the cell contained one or more range points it is given a value of +5%. If a ray between the sensor and one or more range points passed through the cell it is given a value of -5%. Finally the cell will have a value of 0% if the cell was not observed directly or by ray casting. These values have been determined by preliminary tests to ensure when obstacles first come into sensor range they are detected within a second.

The confidence map update process is fast. Momentary map values are simply added with the corresponding cell in the confidence map. The update process limits confidence values to ensure they remain in the 0-100% range. An example of a momentary map updating a confidence map is shown in Figure 7.9.

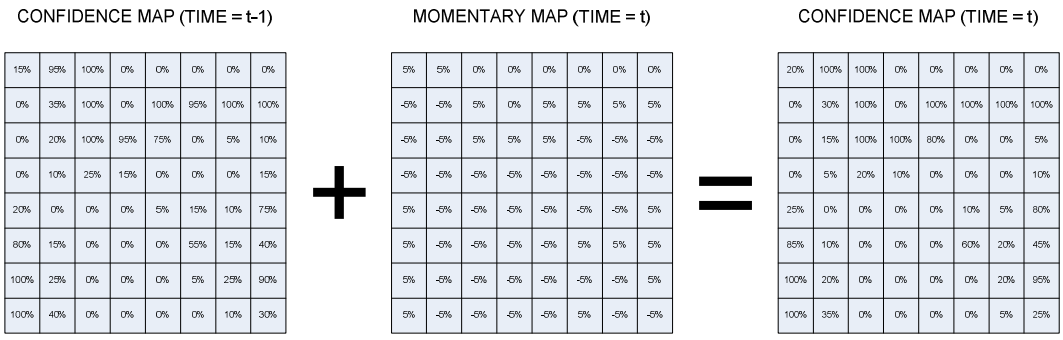

Figure 7.9 - Updating Confidence Values

For reasons of safety the UGV cannot be allowed to drive blindly into unobserved space. Thus it is important to know what portions of the obstacle map have been observed by the range sensor. The observation mask mapping array is used to provide an indication of which of the

obstacle map cells have been observed. Each cell of the mask holds a simple true or false value.

The content of the momentary map is again used to update the observation mask after the arrival of new range sensor data. If the momentary map indicates a map cell contained a range point the corresponding cell in the observation mask is set to true. Likewise if a cell was indirectly observed by ray cast the corresponding cell in the observation mask is also set to true. No action is taken if the momentary map indicates that a cell was unobserved since it may have been observed in an earlier sensor frame.

Traversability analysis is performed periodically on the cells within the obstacle map. This process examines what is known about each obstacle map cell and classifies them as either traversable, unobserved or containing an obstacle. The output from this analysis is the traversability summary mapping array.

Information is taken from both the observation mask and confidence map when generating the latest traversability summary. A cell will be marked as unobserved in the traversability summary if it is marked as unobserved within the observation mask. If a cell on the confidence map has an obstacle confidence value above a threshold it is marked as untraversable in the traversability summary. For initial experimentation this obstacle threshold has been set to a value of 50%. This makes the system tolerant to sensor noise but still fast updating if the environment changes.

In terms of NOVA the three arrays that make up the 2D obstacle map are stored as part of the underlying blackboard system. The confidence map and observation mask are updated by the Protection Module each time new range sensor data arrives. The traversability summary is

revised at 0.5 second intervals by the Cartographer within the Route Planner Module. On the NOVA GUI the traversability summary is displayed to allow the obstacle map to be visualised.

7.2.3 3D Voxel Mapping

The 3D obstacle mapping system is also based on three mapping arrays that together represent the obstacle map much like the 2D system. Again each of these arrays represents the same space in the real world and each stores different attributes about the obstacle map cell. For the 3D mapping system the mapping arrays are a voxel map, an observation mask and a traversability summary.

The voxel map replaces the confidence map in the 3D mapping system. This map maintains a record of recent range points that have fallen into each obstacle map cell. The role of the observation mask remains as an indicator of which portions of the obstacle map have been observed. Also the traversability summary continues to show if map cells are unobserved, traversable or contain some kind of obstacle.

As new data arrives from one of the 3D range sensor configurations it is processed by the appropriate sensor model. This generates corresponding range points with 3D coordinates. To incorporate this new sensor data into the obstacle map these range points are added directly to the voxel map.

You can think of the voxel map as being made up of lots of 3D pixels. These 3D pixels are known as voxels as they describe a small unit of volume. In this research voxels have square bases and their height is dependent on the range points that they contain. Figure 7.10 illustrates what a single voxel would look like.

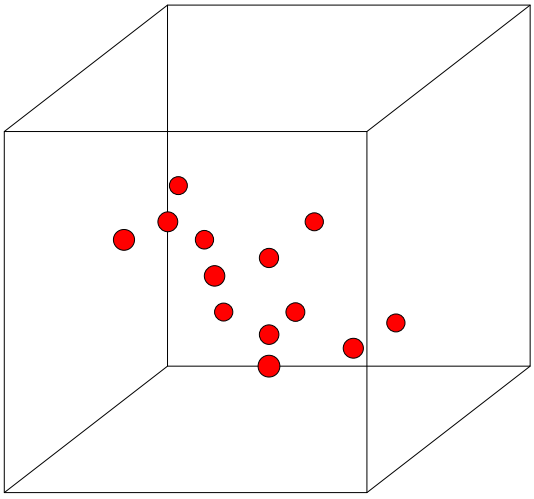

Figure 7.10 - Voxel Concept

The voxel map records the coordinates of recent range points that were generated by the range sensors. Each voxel records range points that have coordinates in the space represented by the voxel. These records are held in a first in first out style list. For initial experiments a maximum of 20 range points were recorded by each voxel. This value was derived from the expected number of range points that might fall into a voxel during one second when using the PMD [vision]® 1k-S. If a voxel contains the maximum number of range points the oldest point would usually be discarded to make room for a new arrival.

Untraversable obstacles for the UGV are typically seen as discontinuities in the ground. So in order to detect obstacles it suffices to look for steps or holes in the voxel map. To address this all voxels have an associated traversability statistic which gives a basic measure of height change within the voxel. Each time a new range point is added to a voxel the traversability statistic for the corresponding voxel is updated.

Two subtly different traversability statistics have been devised. The first statistic is the difference between the maximum and minimum height of range points currently in a voxel. The alternative is to use the difference between the maximum and minimum height of all range

points ever recorded within that voxel. Chapter 8 details experiments that were performed to compare the two statistics.

Using the first statistic requires that the UGV does not discard range points with either maximum or minimum height from a voxel when stationary. If this is not done obstacle filled voxels may be reclassified as traversable when the older points are discarded. This would happen as the range sensor would be continuously seeing the same patch of ground until the UGV began to move again.

In contrast when the UGV is moving the second traversability statistic will be more susceptible to the effects of sensor noise. Consider the case where the range point that has maximum or minimum height in a voxel has been generated out of place due to range sensor or pose sensor noise. After approximately a second this range point will be discarded by the voxel. Using the first traversability statistic the effects of the noise affected range point will disappear. The second statistic will continue to use this poor quality data until a range point with more extreme characteristics is found.

When mapping in 3D it is especially important to know which parts of the obstacle map are unobserved. This is because large negative obstacles may cause some map voxels to be unobserved as is illustrated in Figure 7.11. Hence the observation mask is also updated as new range points are generated.

Whichever voxel a range point falls into on the voxel map the corresponding cell of the observation mask is marked as true. In the 3D obstacle mapping system ray casting is not used to update the observation mask due to the risk of missing negative obstacles. As before no action is taken if a cell is not observed in any given sensor frame as it may have been observed earlier.

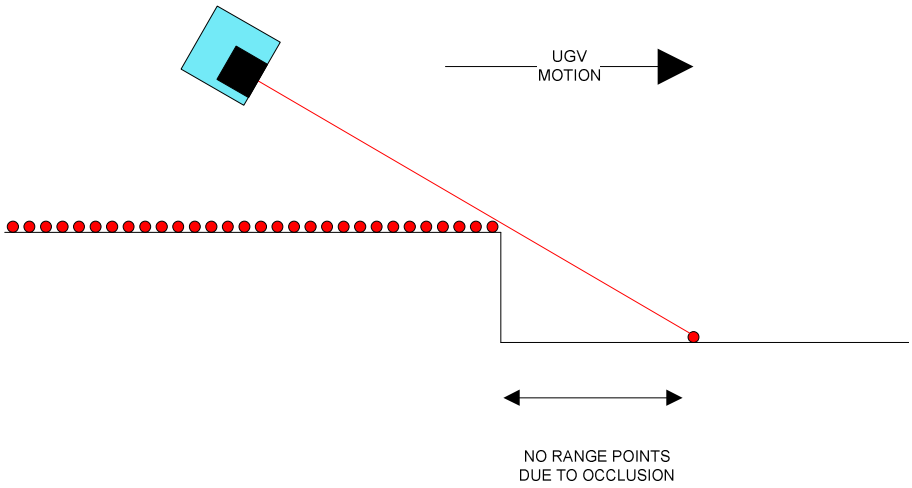

Figure 7.11 - Unobserved ground due to the negative obstacle

Periodically traversability analysis is performed on the obstacle map. This updates the traversability summary based on the latest content of the voxel map and observation mask. As with the 2D system cells in the traversability summary can be classified as unobserved, traversable or containing an obstacle.

Traversability analysis has two steps. First all cells marked as unobserved on the observation mask are given the same classification on the traversability summary. Next the traversability statistics from the voxel map are checked. Any cells where the traversability statistic is greater than a threshold are deemed untraversable and marked as such in the traversability summary.

The threshold for the traversability statistic has been given the initial value of 100mm. This represents the maximum size of obstacle the UGV can drive over without getting stuck if the obstacle lies between the tracks. It also imposes an implicit restriction on the steepness of slopes the UGV can climb of 26.5°.

Within NOVA the 3D obstacle mapping system is a direct replacement for the 2D system. The 3D obstacle map is kept within the blackboard storage area by NOVA in place of the 2D map.

Updates to the voxel map and observation mask are performed by the Protection Module whenever new range sensor data arrives. At intervals of around 0.5 seconds the Route Planner Module runs the Cartographer to revise the traversability summary. As before the traversability summary is the view of the obstacle map that is displayed on the NOVA GUI.

7.3 LOCAL PATH PLANNING

This section introduces the local path planning strategy that is used within NOVA. The local path planner is responsible for guiding the UGV around obstacles. It works equally well with either the 2D or 3D obstacle mapping systems. The planner uses the traversability summary from the obstacle map to decide where the UGV can be safely driven.

The coupling of the local path planner and the waypoint navigation system is seamless. UGV routes are specified using either map waypoints or the direction control on the NOVA GUI. The basic waypoint navigation system then guides the UGV to each waypoint in turn. The Online Route Adaptor in the NOVA Route Planner Module is in control of waypoint navigation. This routine calculates the ideal straight line path to the next operator waypoint ignoring the possibility of obstacles.

When local path planner is enabled the UGV follows the ideal route towards the next waypoint until it encounters an obstacle on the desired path. At that point the local path planner will take over and guide the UGV around the obstacle. The UGV will revert back to the ideal path generated by the Online Route Adaptor when it is seen to be obstacle free.

The following subsections discuss how the basic path planner works then introduce extensions that should improve performance. First the radial swath search process is presented. This is the method used by the local path planner to find a clear path across the obstacle map. A basic

path planning algorithm is then introduced. This uses the output of the swath search to guide the UGV.

Next the concepts of field of view protection and corridor obstacles are described. Field of view protection is a simple behaviour that prevents the UGV from being driven into unobserved space. Corridor obstacles provide a means to stop the UGV from straying far from the operator specified waypoint route.

Following these topics the important subject of lookahead for the path planner will be discussed. The planner lookahead determines how close an obstacle must be before the planner takes avoiding action. This leads onto the idea of adjusting the planner lookahead according to the environment. The novel concept of UGV claustrophobia is then introduced. In this case claustrophobia is used to control the lookahead adjustment according to the density of nearby obstacles.

7.3.1 Radial Swath Search

The radial swath search process is at the heart of the local path planner. It is so called because it searches swaths across the obstacle map. Each search swath radiates out from under the UGV towards the edges of the obstacle map. Thus these swaths should be seen as possible paths that the UGV could take across the obstacle map. The swath search process finds the distance from the UGV to the closest obstacle for each swath. These distance values are then ultimately used to guide the UGV.

Every cycle of the path planner a set of swaths around the ideal heading are searched. Since it is desirable for the UGV to always make progress towards the next waypoint the searched swaths cover the range ±90° around this ideal heading. Search swaths in this range have an

angular separation of 5°. This is the smallest separation the UGV can use without needing to stop and turn every few seconds due to heading noise.

With a 5° angular separation 37 swaths across the obstacle map must be searched each planner cycle. The direction of these search swaths relative to the UGV is illustrated in Figure 7.12. Since local path planner cycles happen at a rate of around 2Hz an efficient search process is thus required to ensure that the swath search can be conducted in real time.

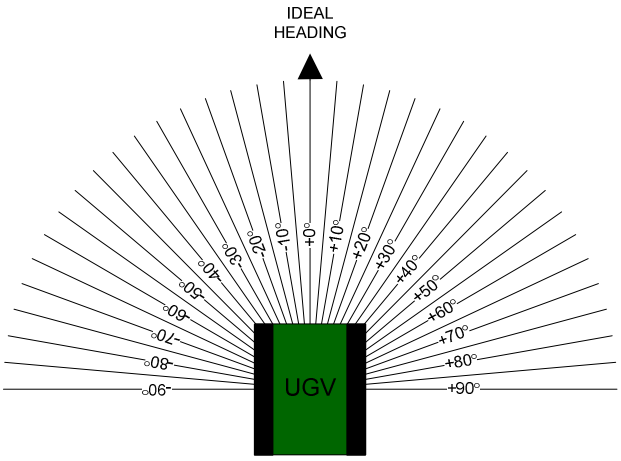

Figure 7.12 - Direction of Radial Swaths

A fast point sampling method is used to search each swath. This method makes two assumptions. First it is assumed that the UGV has the same heading as the swath being searched. Second it is assumed that the UGV is surrounded by a 5cm safety margin. The aim of making the UGV 10cm longer and wider in this way is to minimise the chance of collisions.

The sampling method starts by calculating the location of the back corners of the expanded UGV to determine a start line for the search. Next it generates a sampling line parallel to the heading of the search swath. This sampling line goes from the back left corner of the expanded UGV forward and off the edge of the obstacle map. Moving away from the start line the

obstacle map traversability summary is then sampled at regular intervals along the sampling line.

A sampling interval of 5cm has been chosen. This ensures every traversability summary cell under the sampling line has a high chance of being sampled. The sampling process ends when the edge of the obstacle map is passed or an untraversable cell is sampled. If an untraversable cell is found the distance of the sample from the start line is recorded.

Once the first parallel line has been sampled the process is repeated using other sampling lines parallel to the swath heading. In each case the sampling line begins at the start line and goes off the edge of the obstacle map. A lateral separation of 5cm has been chosen between adjacent parallel sampling lines.

Sampling continues until the sequence of parallel lines that have been sampled covers the whole width of the start line. Figure 7.13 shows the sampling lines for the swath that follows the ideal heading. This screenshot was generated during initial development of the search algorithm.

Each time an obstacle cell is found during the sampling process the distance from the start line along the sampling line is checked. A record of the minimum of these distance values is maintained throughout the whole swath search. This minimum value corresponds to the minimum straight line distance from the back of the UGV to an obstacle along the searched swath. If no obstacles are found in a swath search the distance to the closest obstacle is recorded as 10m. This value implies that the swath is clear at least as far as the edge of the obstacle map.

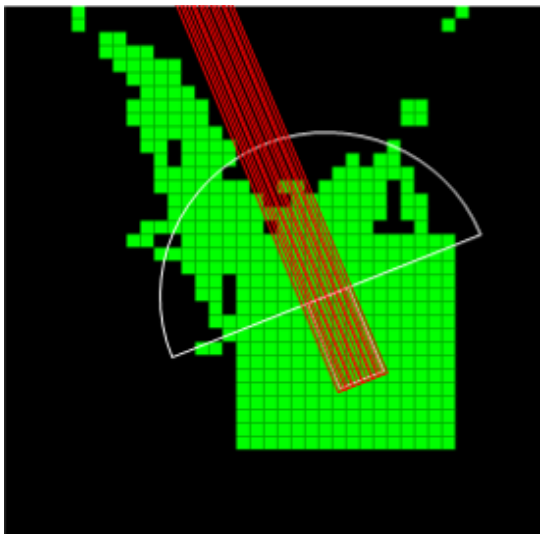

Figure 7.13 - Parallel sampling lines used in swath search process

If the searched swath is clear the search process will make no more than 2500 samples on a 10m square obstacle map. In the presence of obstacles this number could be much less. Figure 7.14 shows which cells of the obstacle map would be sampled when using this process to search the swath on the ideal heading. Again this screenshot was generated during initial development of the search algorithm.

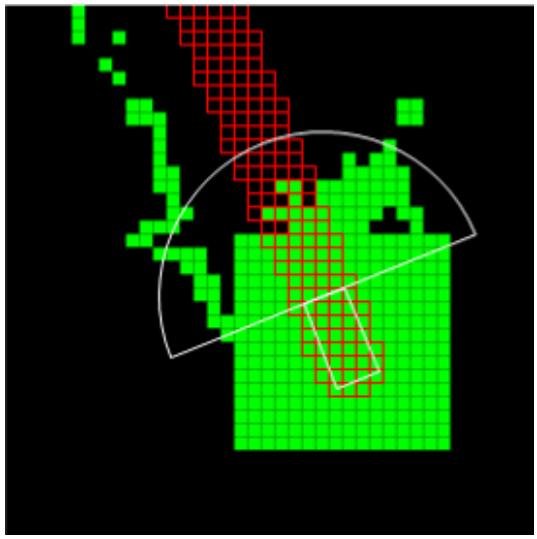

Figure 7.14 - Cells sampled by the swath search process

Each swath search yields the minimum distance to an obstacle if that swath was followed across the obstacle map. Together the results from the 37 searched swaths create a histogram. This histogram has been dubbed the Polar Obstacle Distance Histogram.

Figure 7.15 shows what the histogram would look like if the UGV was following the centre of a 2m wide corridor. It should be noted that the central bars of the histogram are all 10m. This is because the swath search in these swaths reached the edge of the obstacle map before an obstacle was encountered.

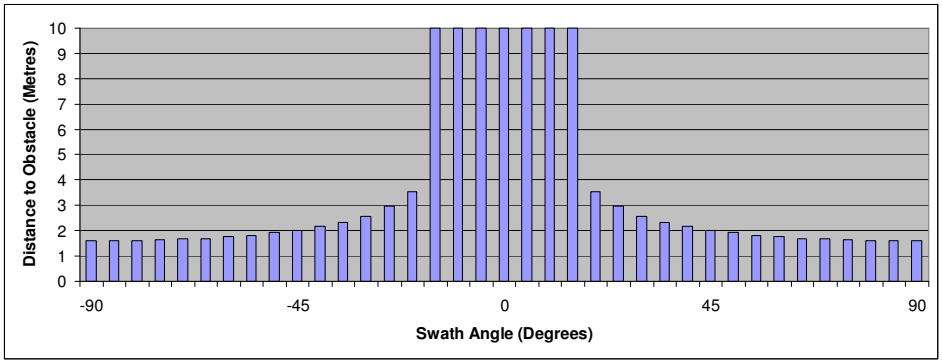

Figure 7.15 - Example Polar Obstacle Distance Histogram

Polar histograms have been used in other local path planners to represent the closeness of obstacles. The most well known is the Vector Field Histogram (VFH) method [Borenstein 1991]. With VFH the polar histogram represented obstacle density and each histogram bar showed the density for a narrow angular sector of the obstacle map around the robot. Obstacle density was calculated as the sum of obstacle confidence multiplied by distance from the UGV for each cell that fell within the angular sector.

The method introduced in this research is much more intuitive. The radial swath search is only concerned with the distance to the closest obstacle marked on the traversability summary. The search does not consider cells with low obstacle confidence as they are not marked as

untraversable on the traversability summary. Also when searching a swath the more distant untraversable cells are not considered. This makes sense since the UGV will not be able to drive any further along the swath than the closest obstacle.

7.3.2 Basic Path Planning Algorithm

The Polar Obstacle Distance Histogram is used to decide if the UGV needs to avoid an obstacle. It is also used to find the best path to avoid any detected obstacles. The path planning algorithm starts by checking the length of the swath on the ideal heading. If the swath length is greater than the distance to the next UGV waypoint or it is clear to the edge of the obstacle map then this swath will be selected.

In the event that the swath on the ideal heading is blocked then the planner will search for a path that best avoids the obstacle. Swaths to the left then right of the ideal heading are checked in order of increasing angular deviation until a clear swath is found. The order of checking is thus -5°, +5°, -10°, +10°, etc. Figure 7.16 illustrates the path planning process finding that the +20° swath avoids an obstacle with the smallest angular deviation.

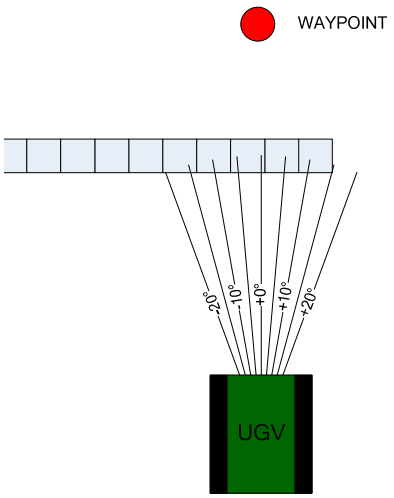

Figure 7.16 - Path Planning Process

If the path planner finds itself surrounded by obstacles it may not be able to find a clear path to the edge of the obstacle map. In this case the planner will select the longest swath in the Polar Obstacle Distance Histogram. The aim of this behaviour is to approach the most distant obstacle in hope that a clear path will appear before the obstacle is reached. If a clear swath does appear the path planner will opportunistically select it and lead the UGV into the clear space.

When the planner is following the longest swath, if a clear swath does not appear the UGV is automatically stopped. It will be brought to a standstill just short of the obstacle at the end of the swath. In this situation the UGV would then await further instructions from the operator. This would occur if the UGV drove into a cul-de-sac guided by this basic path planner. The final section of this chapter introduces an extension to the path planner that can be used to escape cul-de-sac obstacles.

When the path planner selects a swath to follow it generates a sub-goal. Sub-goals encode the heading of the selected swath and also the distance which the UGV should travel along the swath. When the swath on the ideal heading is clear the sub-goal will contain the distance to the next waypoint. In all other cases the sub-goal distance is set as the length of the swath indicated on the Polar Obstacle Distance Histogram less the stopping distance of the UGV. The latest sub-goal is always passed to the NOVA Motion Module at the end of each planner cycle to control the UGV.

The simple path planning algorithm introduced here ensures that the UGV will begin to avoid obstacles as soon as they are detected blocking the ideal heading. When avoiding an obstacle the UGV should take the minimum deviation path and skirt around the perimeter of the obstacle. The UGV would then select the ideal heading again once it is clear. Inspiration for

this basic path planning algorithm came from a system used on the MDARS-E UGV [Kurtz 1997].

7.3.3 Field of View Protection

Field of view protection is an extension to the basic path planning algorithm. It is only required when using the 3D sensor models and mapping system. The aim of this feature is to stop the UGV from driving blindly into unobserved areas of the obstacle map. In the real world doing this could have disastrous consequences like driving the UGV over a cliff edge.

Enabling field of view protection makes the radial swath search process treat any unobserved cells on the obstacle map as obstacles. Thus the path planning algorithm will then avoid any unobserved areas of the obstacle map as if they were untraversable. It was expected that this feature would work well with the wide field of view of the Sick LMS 200 sensor. However field of view protection could restrict the UGV motion when using the PMD [vision]® 1k-S camera.

Consider the case where a range sensor with a narrow field of view is used on the UGV. An example would be the PMD [vision]® 1k-S camera with the standard lens that gives a 34° horizontal field of view. It is possible to contrive a situation where the UGV drives up to an obstacle that is wider than the sensor field of view. To the path planner this is like being trapped in a cul-de-sac if field of view protection is being used. Figure 7.17 shows what the obstacle map may look like in this case.

To counter this problem a method to increase the observed space around the UGV has also been devised. The radial swath search has been expanded to record if the closest obstacle in any swath is unobserved. If this is the case and the path planner cannot find a clear swath the UGV will stop and scan the surrounding area. The path planner will turn the UGV to the left

then to the right in steps smaller than the horizontal sensor field of view. After each step the UGV will pause to gather range sensor images.

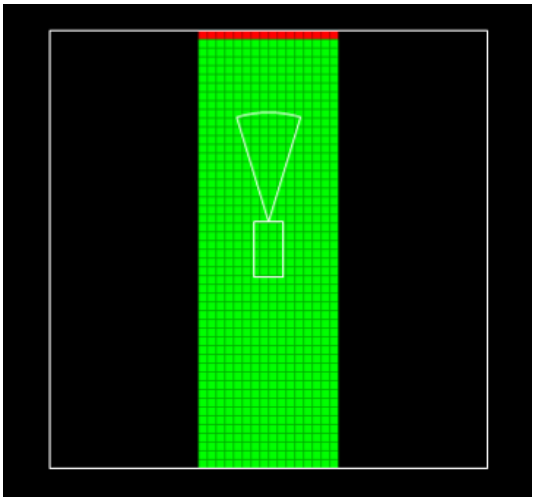

Figure 7.17 - Contrived example of sensor with narrow field of view

This observation process should provide the UGV with a more complete map of the surrounding environment. When the area has been completely scanned the path planner will again attempt to find a path around the obstacle which was originally blocking movement. If a path cannot be found the UGV would have to revert to a safe state and await operator instructions.

7.3.4 Corridor Obstacles

Another extension to the basic path planning algorithm is the concept of corridor obstacles. Corridor obstacles are phantom obstacles that can be placed on the obstacle map traversability summary. The purpose of these obstacles is to stop the UGV straying outside a defined corridor of operation during autonomous behaviour.

These phantom obstacles are treated exactly like untraversable cells on the obstacle map by the radial swath search process. Hence the path planning algorithm avoids corridor obstacles as if

they were real obstacles. The effect of this behaviour is to force the UGV to find a path to the next operator waypoint that stays within the route corridor.

Corridor obstacles can be enabled and disabled on the NOVA GUI. When enabled corridor obstacles are generated in lines parallel to the route segments between any operator waypoints that have been specified. The route segments always lie at the centre of the corridor that the obstacles delimit. A GUI control allows the operator to specify the required corridor width.

Figure 7.18 shows a corridor of width 5m generated on the obstacle map for the route drawn on the map tab of the NOVA GUI. Adjacent corridor obstacles have a separation of 0.5m. This is narrower than the ARP vehicle so there is no way that the UGV can find a way though. In the screenshot it looks as if the corridor edges are not straight however this is just the aliasing effect of colouring a whole map cell that contains a single point obstacle.

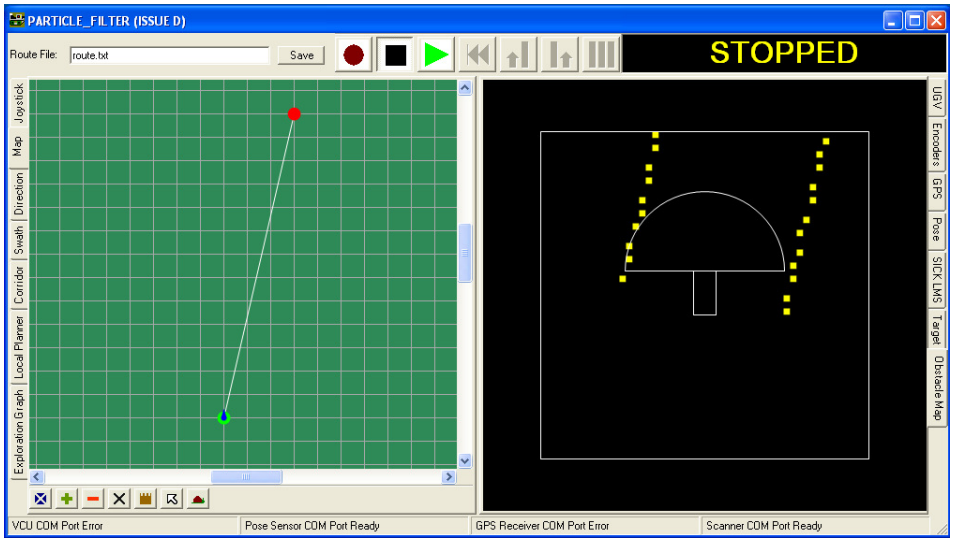

Figure 7.18 - Generating Corridor Obstacles

As the UGV moves the corridor obstacles drawn on the obstacle map are updated. When the obstacle map scrolls an algorithm cycles through the set of corridor obstacles for the current route segment and draws those that lie within the area of the map. When the UGV reaches a

waypoint the existing corridor obstacles are replaced with obstacles corresponding to the next segment of the route. The Cartographer within the NOVA Route Planner Module is responsible for drawing corridor obstacles on the map.

7.3.5 Planner Lookahead

The lookahead of a path planner can be defined as the distance in the front of the UGV that is considered when making obstacle avoidance decisions. For the basic path planner introduced in this chapter the lookahead is determined by the size of the obstacle map. The radial swath search process searches for untraversable cells from the UGV at the centre of the obstacle map out to the edges of the map. Thus due to the square shape of the obstacle map the basic path planner within NOVA has a lookahead of at least 5m.

In theory the path planner lookahead is a critical factor affecting how the UGV behaves when avoiding obstacles. Smaller values would allow the UGV to get much closer to obstacles before the path planner recognises obstacles are present and begins to avoid them. In this case avoiding actions will need to be more extreme. Larger values make the UGV respond to the presence of an obstacle earlier. Thus a more acute avoiding angle can be followed around the obstacle.

Figure 7.19 illustrates this point. The diagram shows a series of different lookahead values for the UGV path planner. The range of these lookahead values goes from 2m to 5m in steps of 0.5 m. If the planner has a lookahead of at least 3.5m then the best swath is at -25° from the ideal heading and is shown as a dashed rectangle. If the planner lookahead is shorter the UGV will move closer to the wall before obstacle avoidance is used.

On this basis it seems reasonable to assume that a longer path planner lookahead should be preferred. In fact there is no reason why this strategy would not work for avoiding sparse

obstacles such as that depicted in Figure 7.19. However when the obstacle density is much higher a path planner with a large lookahead may get in to trouble.

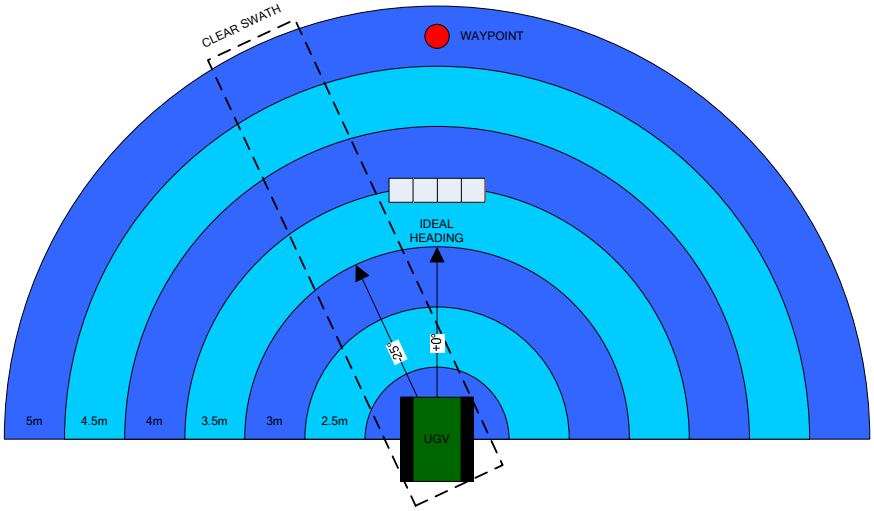

Figure 7.19 - Effect of planner lookahead value

Consider the scenario illustrated in Figure 7.20 where the UGV needs to drive through a narrow gap. Again the diagram shows a number of hypothetical lookahead zones. If the planner lookahead is less than 3.5m the UGV will continue on the ideal heading towards the waypoint. For longer lookahead values the path planner will not find an obstacle free swath and so the longest swath would be selected. In the diagram the longest swath is at +30° from the ideal heading and is shown as a dashed rectangle.

While selection of the longest swath is useful for escaping tight situations it should not be relied on. Clearly in this situation the path planner would fair better with a shorter a lookahead. In path planning research it is normal for the planner lookahead to be fixed. Typically the planner lookahead is tuned for the specific test environment. This was certainly the case for VFH path planner [Borenstein 1991] and the planner used on the MDARS-E UGV [Kurtz 1997].

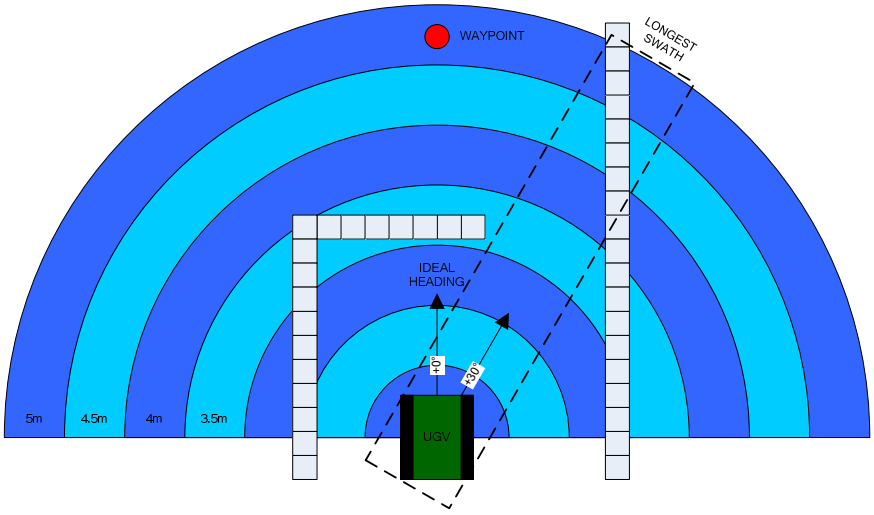

Figure 7.20 - Issue with long lookahead in tight space

For the path planner developed in this research it is easy to adjust the lookahead. The planner uses the Polar Obstacle Distance Histogram to decide which of the searched radial swaths are clear of obstacles. With the basic version of the path planner a swath is considered clear if it has a length of exactly 10m. In the extended version of the path planner this threshold length is a configurable parameter.

Experiments are discussed in Chapter 8 that test basic obstacle avoidance with the path planner lookahead set at a series of different values. These tests confirm the expectation above. That is the shorter the lookahead the closer the UGV will approach obstacles and so the more extreme the required avoiding action.

In conclusion it can be said that in some situations a long planner lookahead is desirable while others require a short lookahead. So what is required is a novel method to adjust the lookahead according to the environment. This would give the path planner best of both worlds. The density of obstacles around the UGV could be the trigger for this kind of the lookahead adjustment. This is the topic of the next subsection.

7.3.6 Adjustable Lookahead & Claustrophobia

Path planner lookahead needs to be quite short when the UGV is driving through areas with high obstacle density. In contrast when the UGV is in open space with sparse obstacles the planner lookahead can be much longer. So a novel method to measure the obstacle density in the vicinity of the UGV has been devised. As the obstacle density increases the path planner can then react by reducing the lookahead. This should allow the UGV to get closer to obstacles and drive through smaller gaps if necessary.

The path planner can measure obstacle density by calculating the proportion of clear swaths in the Polar Obstacle Distance Histogram. If the histogram indicates the distance to the closest obstacle for a swath is no smaller than the current planner lookahead then that swath is considered clear. In general for a greater lookahead there should be a smaller proportion of clear swaths. Figure 7.21 shows the relationship between the proportion of clear swaths and the planner lookahead for the example histogram in Figure 7.15.

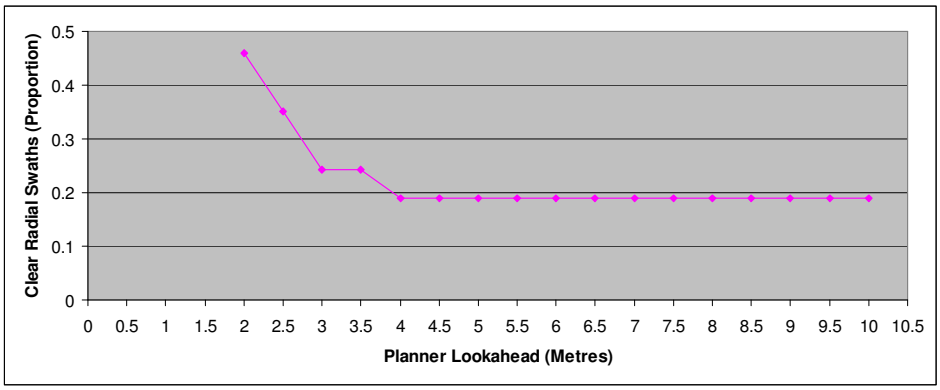

Figure 7.21 - Example relationship between lookahead and clear swaths

Specifying a minimum desirable proportion of clear swaths gives a trigger to adjust the path planner lookahead. At all times the planner should have the greatest practical lookahead. This ensures that the UGV begins avoiding obstacles as soon as possible. However if the proportion

of clear swaths falls below the desirable level the planner lookahead should be reduced. In this way the lookahead is adjusted in response to the measure of obstacle density.

Lookahead adjustment is performed each cycle of the path planner. The lookahead can be adjusted between preset maximum and minimum values. A maximum lookahead of 10m is used which ensures the path planner considers the full extent of the obstacle map. This would be equivalent to the basic version of the path planner. The minimum lookahead is 2m. This includes the length of the UGV since the radial swath search starts from the back of the UGV. Hence the minimum lookahead equates to considering obstacles up to about 0.5m in front of the UGV.

At the start of the adjustment process the planner lookahead is reset to the maximum value. The planner lookahead is then reduced in 0.5m steps. After each step the proportion of clear swaths in the Polar Obstacle Distance Histogram is checked. The adjustment process continues until the desired proportion of clear swaths is seen or the minimum lookahead is reached.

The complement of the minimum desirable proportion of clear swaths is the maximum desirable proportion of blocked swaths. This proportion is displayed as an operator configurable value on the NOVA GUI. Due to the effect of this value on the path planner behaviour it has been dubbed the claustrophobia parameter. From this claustrophobia setting the desirable proportion of clear swaths is calculated and passed to the lookahead adjustment process.

As a default the claustrophobia parameter is set to 0.75. This implies that the path planner should only reduce the lookahead if less than 25% of swaths are clear. Preliminary tests have shown that this value will allow the UGV to avoid sparse obstacles without the lookahead

dropping from the maximum value. If the UGV encounters a higher density of obstacles the lookahead will begin to fall.

Generally setting the claustrophobia parameter to a high value should have several effects. The path planner will be more reluctant to reduce the lookahead value. This will make the UGV avoid obstacles earlier. By avoiding obstacles at a greater distance the UGV is more likely to stay in areas of open space and avoid areas of high obstacle density. Thus the UGV will demonstrate the characteristics of claustrophobia.

Contrast this with the likely behaviour of the UGV when the claustrophobia parameter is set to a low value. The path planner should quickly reduce lookahead whenever the UGV is in the vicinity of obstacles. This will allow the UGV to get much closer to obstacles or even pass through narrow gaps between obstacles.

Figure 7.22 depicts the likely effect of the claustrophobia parameter on the UGV path. In the diagram the UGV needs to negotiate an obstacle field to reach the next waypoint. If the claustrophobia parameter has a high value the UGV will skirt around the perimeter of the obstacle field. With a low claustrophobia setting the UGV will attempt to drive between the obstacles.

Experiments have been conducted to test the adjustable planner lookahead concept. These experiments are detailed in Chapter 8 and confirm expectations about the effects of the claustrophobia parameter. A tight and twisty obstacle course was set up and the UGV requested to negotiate it. Experiments were repeated with the claustrophobia parameter given both high and low values.

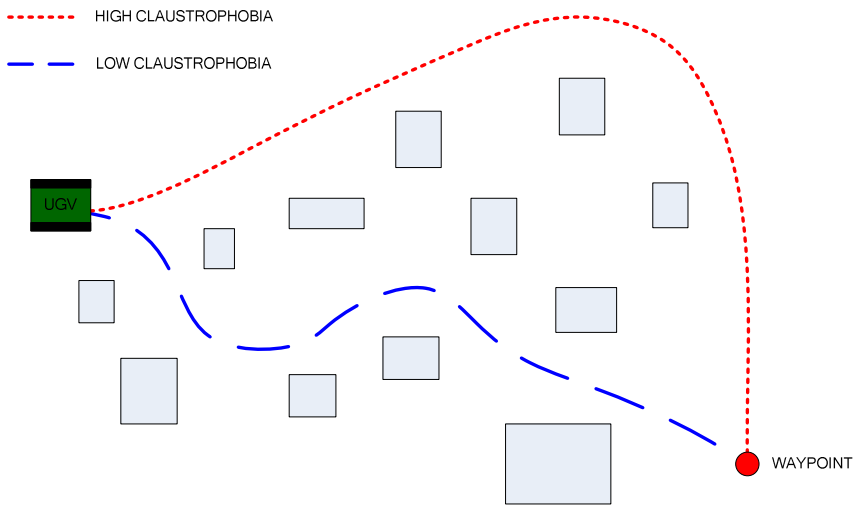

Figure 7.22 - Effect of Claustrophobia on UGV Path

7.4 EXPLORATION GRAPHS

This section introduces the novel concept of an exploration graph. Exploration graphs can be used to autonomously recover a UGV when it has encountered a dead end. The concept was devised in this research as an extension for the local path planning algorithm.

Local UGV path planning systems are known to fall foul of arrangements of obstacles that form cul-de-sacs. The problem occurs when the UGV drives into the mouth of a cul-de-sac but cannot sense the closed end due to limited sensor range. Surprisingly these systems often do not have a good method to detect and escape from this type of situation. The exploration graph offers a potential solution.

Before a meaningful description of an exploration graph can be given a few auxiliary ideas must be introduced. First the notion of peaks and valleys on the Polar Obstacle Distance Histogram is presented. Next a definition of an unexplored path for the UGV is given. Finally the concept of a decision node will be explained.

The local path planner makes decisions using the Polar Obstacle Distance Histogram. Radial swaths in the histogram are considered clear if they are at least as long as the current planner lookahead. By superimposing the lookahead onto the polar histogram the histogram will be split into peaks and valleys. Peaks represent groups of one or more adjacent radial swaths that are considered clear. Conversely valleys are groups of adjacent swaths that are shorter than the planner lookahead and so considered blocked.

Figure 7.23 illustrates this notion of peaks and valleys. Let us assume the current planner lookahead is 4m as shown by the dashed red line on the histogram. Then the example histogram has two peaks separated by a valley. The left hand peak includes all radial swaths in the range -90° to -30°. Swaths in the range +30° to +90° then make up the right hand peak. In reality this histogram would represent the UGV facing a slim obstacle.

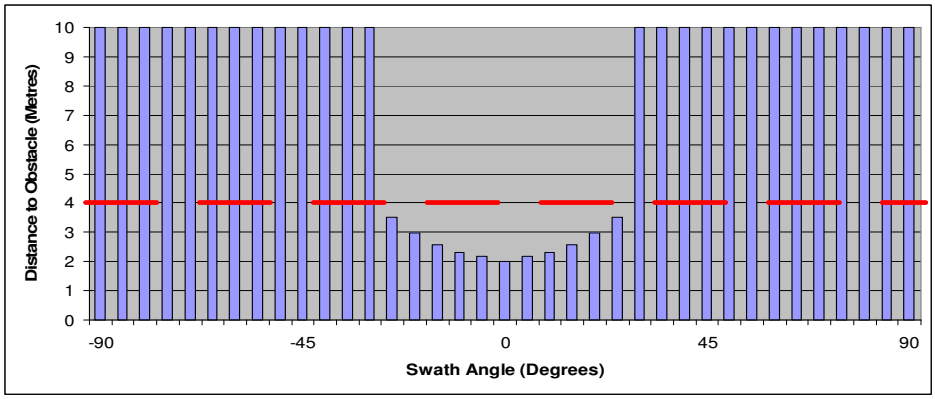

Figure 7.23 - Lookahead shown on Polar Obstacle Distance Histogram

In the example above the path planner can choose to pass the obstacle on the left hand side or the right. Each of these options represents an unexplored path for the UGV. The basic path planner will always select the best swath to reach the next operator waypoint. That is a swath that is considered clear and avoids any obstacles with the smallest angular deviation. In terms

of path choice all of the swaths in each peak can be considered equivalent as they will each guide the UGV round the same side of the obstacle.

In every cycle of the local path planner a decision is made about which direction the UGV should travel. When using exploration graphs the details of each path planning decision are recorded as individual decision nodes. These nodes store four key details about the decision that the path planner made. The first detail is the UGV location at the time of the decision. Next is the number of peaks that were counted in the Polar Obstacle Distance Histogram. The remaining values represent the type and rank of decision made by the path planner.

The decision type parameter indicates if the path planner selected a clear swath from the polar histogram when the decision was made. This parameter will also show if the planner was forced to select the longest swath or was completely blocked due to obstacles. The decision rank parameter records the number of times the decision node has been visited by the UGV. When a decision node is first created this parameter always has a value of 1.

Essentially each decision node can be thought of as providing two important facts. Firstly the node is a record of the number of paths that the UGV could explore from the corresponding location. Secondly it reports which of the possible paths the path planner chose to follow. This is because the path planner will always pick the best swath from the Polar Obstacle Distance Histogram.

Now that all of the essential ideas have been introduced the exploration graph concept can be properly defined. The best description of an exploration graph is as a connected network of decision nodes. It is implemented as an ordered list or stack structure. Thus each time the path planner makes a decision the newly generated decision node is added to the end of the list.

For a UGV mission the exploration graph gives both a record of where the UGV has been and the decisions the path planner made at each step. If a cul-de-sac was encountered the UGV could backtrack through the decision nodes to retrace the original path. The decision nodes could also be used to recognise where alternative paths exist and get the UGV to explore those instead.

In the following subsections the exploration graph concept is discussed in more detail. First a method is described for using exploration graphs to escape from cul-de-sac situations or explore mazes. After this the concept of obstacle flooding is introduced. The role of obstacle flooding is to prevent the path planner from choosing paths that have already been found to be blocked.

The remainder of the chapter then looks at enhancements for the exploration graph concept. A method to rationalise the generation of decision nodes is proposed first. Use of this method ensures exploration graphs are kept as simple as possible. Next an optimum lookahead distance for the path planner is introduced. This is known as the minimax lookahead and helps the planner to maximise the number of unexplored paths detected in any location.

7.4.1 Using Exploration Graphs

Consider the hypothetical situation depicted in Figure 7.24. The UGV has driven autonomously into a cul-de-sac that lay on the ideal path to the next waypoint. The set of decision nodes that make up the exploration graph are shown in the diagram as yellow circles. Underneath each decision node is a number that indicates how many paths the path planner detected when the node was created.

The decision nodes imply that when the UGV was at the left hand side of the diagram the path planner could see only one possible path option. That option would be to follow the corridor

straight from left to right. For the first half dozen decision nodes the Polar Obstacle Distance Histogram could be expected to look like the example shown in Figure 7.15.

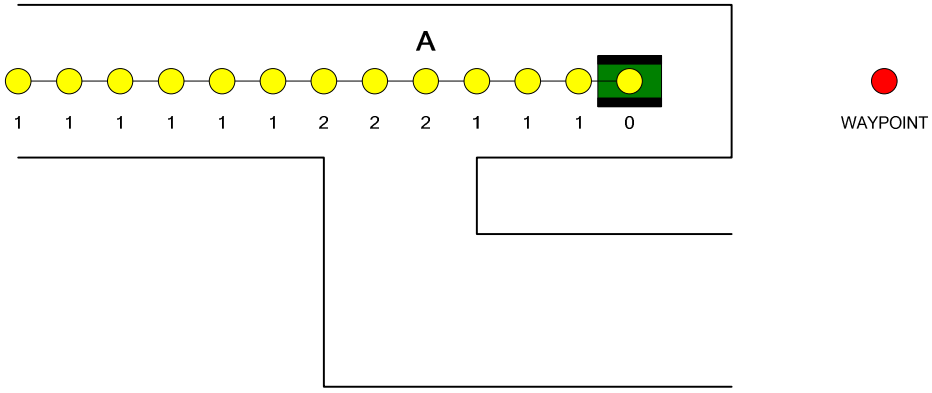

Figure 7.24 - Exploration Graph records UGV driving into cul-de-sac

When the UGV reaches the branch in the corridor the number of possible paths increases to two. The UGV can continue straight on or choose to follow the branch. For the set of decision nodes around the branch the Polar Obstacle Distance Histogram would be similar to that shown in Figure 7.25.

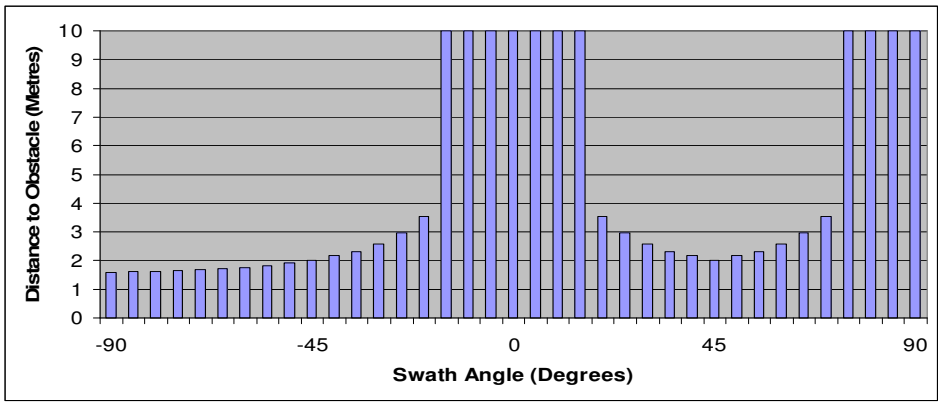

Figure 7.25 - Polar Obstacle Distance Histogram at Decision Node A

Once the UGV has passed the branch the number of potential paths temporarily falls back to one. This continues until the UGV senses the cul-de-sac and it is clear that the path planner

has no unexplored path options. The UGV would thus come to a standstill before reaching the end of the cul-de-sac.

The aim of the exploration graph concept is to allow the UGV to escape from these kind of situations autonomously. So clearly the UGV needs to back up to where it saw the branch in the corridor and try the unexplored path option. In the diagram this would equate to reversing the UGV until the decision node labelled A was reached.

In general when the path planner encounters a cul-de-sac it will begin by backtracking to the penultimate decision node in the exploration graph. Upon reaching this node the path planner will perform a radial swath search and build a new Polar Obstacle Distance Histogram. The planner will then use the histogram to check if there are any unexplored paths.

This is achieved by comparing the number of peaks in the histogram with the decision rank stored in the node. Recall that the decision rank shows the number of times the node has been visited and so the number of decisions made at the node. Also remember that the peaks in the polar histogram each represent potential paths. Thus if the decision rank is less than the number of peaks there will be a potentially unexplored path.

In the event that no unexplored paths are found at this decision node the UGV will continue to backtrack. The path planner will guide the UGV back through the exploration graph to each decision node in turn. The process continues until a decision node with an unexplored path is found.

If it is determined that there is an unexplored path from the decision node the path planning algorithm is rerun to find the best swath to follow. However this time the system is made

aware of the decision rank of the node. As a result the planning algorithm ignores all swaths in peaks of the polar histogram that correspond to previously explored paths.

The path planning algorithm will thus select the next best swath. This swath will belong to one of the unexplored paths and so the UGV will be instructed to head off down this alternative path. At this stage the exploration graph must be tidied up. The decision nodes at the tail of the exploration graph that lead into the cul-de-sac will be deleted. Finally the decision rank of the final node will be increased to reflect the fact the UGV returned to this node and selected an alternate path.

This use of an exploration graph can be likened to the concept of a depth first search [Luger 1999]. The depth first search is a well known method of searching state spaces using a decision tree. It is normal for the search algorithm to select the best option at each node. The algorithm then continues through the branches of the tree until a leaf node is reached. If the leaf node does not have the required characteristics the algorithm will backtrack. When backtracking the algorithm returns to the last node with unexplored branches and then repeats the search process through the unexplored nodes.

7.4.2 Obstacle Flooding

There is a problem with the algorithm defined in last subsection. When the UGV encounters a cul-de-sac it should autonomously backtrack to the last node in the exploration graph with an unexplored branch. Then it should correctly select the next best branch and continue. However at the next cycle of the path planner what is to stop the UGV driving straight back into the cul-de-sac?

The cul-de-sac is still likely to be the best path the next time the path planner is run. So it is necessary to make the swaths that correspond to the cul-de-sac look unattractive. To that end a

novel method has been devised to flood the radial swaths that correspond to explored paths with phantom obstacles. Subsequent cycles of the path planning algorithm then avoid the flood obstacles in the same way as normal untraversable cells on the obstacle map.

Figure 7.26 illustrates how obstacle flooding would be used with the example presented in Figure 7.24. Once the cul-de-sac was detected the UGV would backtrack to the node labelled A. Upon returning to node A the exploration graph algorithm would ensure that the UGV chose to follow the branch in the corridor. At this point the swaths corresponding to the cul-de-sac would be flooded. The path planner would then avoid the flooding and guide the UGV along the branch towards the next waypoint.

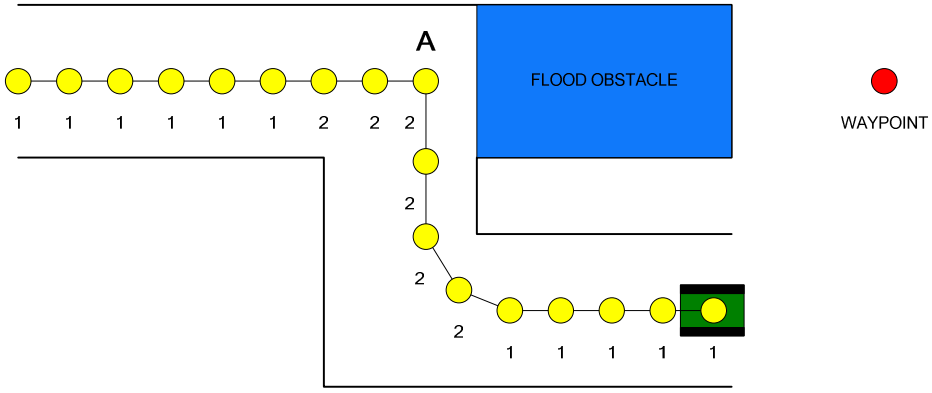

Figure 7.26 - Concept of Obstacle Flooding

Flood obstacles appear on the traversability summary within the UGV obstacle map. New flood obstacles are always integrated with the obstacle map via a flooding mask. The flooding mask is another mapping array of the same size as the obstacle map. Cells within the mask store a Boolean value that indicates if the corresponding cell of the traversability summary should be flooded. Use of the mask ensures old flood obstacles are not unintentionally removed when new flooding is added.

The actual obstacle flooding algorithm is a variation of the radial swath search process that builds the Polar Obstacle Distance Histogram. The algorithm is used to completely flood all of the swaths in explored paths. However obstacle flooding cannot be allowed to obstruct any other clear swaths. So a second stage of the algorithm is required to ensure all of the other swaths are free from flooding.

Before flooding begins each radial swath must be assigned a rank according to whether it belongs to the best path, second best, third best, etc. As an example Figure 7.27 shows how the swaths in Figure 7.25 would be ranked. The flooding algorithm then cycles through each radial swath checking if it belongs to an explored path. This is done by comparing the decision rank of the current decision node with the rank of each radial swath.

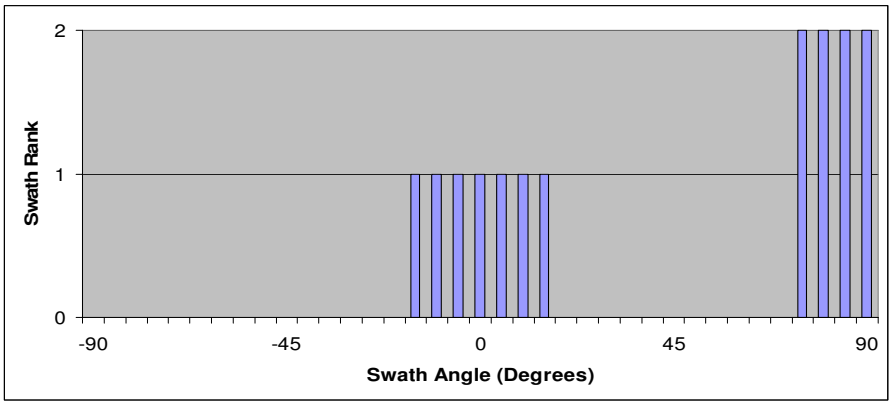

Figure 7.27 - Rank of Polar Obstacle Distance Histogram at Node A

For each radial swath belonging to an explored path all of the cells that fall inside the swath are marked on the flooding mask. Flooding in a swath starts from the under the UGV and goes up to the closest obstacle or the edge of the obstacle map if the swath is clear. Flood obstacles are added using the same point sampling process used in the radial swath search.

Next the algorithm cycles through each radial swath again checking if it does not belong to an explored path. For these swaths it is ensured that all of the flooding mask cells that fall inside the swath are not marked. The flood clearing process again follows the swath from under the UGV up to the closest obstacle if any. As before the point sampling process used in the radial swath search is used here to clear the flooding

Experiments have been performed to test the operation of the exploration graph and obstacle flooding concepts. For these experiments an obstacle course was constructed that deliberately led the UGV into a cul-de-sac. These experiments are detailed in Chapter 8. It was expected that an exploration graph would be demonstrated as a useful tool to use when escaping from a dead end. Furthermore it was presumed that the use of obstacle flooding in conjunction with an exploration graph would be shown to be essential.

7.4.3 Rationalising Decision Nodes

Over the course of an hour long UGV mission the local path planner would generate in the region of 7200 decision nodes. Not all of these nodes are required for the UGV to escape from cul-de-sac situations. In fact the path planner needs only store a small proportion of the nodes that are generated. So rationalising the decision nodes could lead to a much simplified exploration graph.

The critical decision nodes are those that represent the last opportunity to select a particular path before the path planner commits to an alternative. Consider the example from Figure 7.24 the critical node here is labelled A. This is the last opportunity the planner had to select the corridor branch before driving into the cul-de-sac.

An important question is how can these critical nodes be discriminated from the other decision nodes? The key is that they are the last decision node before the number of possible paths or

peaks in the Polar Obstacle Distance Histogram falls. Figure 7.28 illustrates a rationalised version of the exploration graph from Figure 7.24. This shows what the exploration graph would become if only these critical nodes were recorded. It is clear from the diagram that the rationalised exploration graph still contains all of the vital information. So it should be expected that the UGV will still be able to escape the cul-de-sac using the rationalised graph.

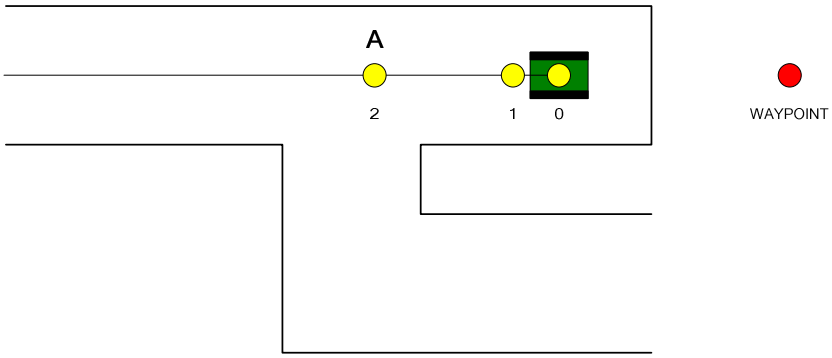

Figure 7.28 - Rationalised Exploration Graph

The exploration graph can be rationalised as it is constructed. Each time a new decision node is created by the path planner the peak count parameter is compared with that of the previous decision node. If the peak count of the new node is at least as big then the older node will be discarded. This process is very efficient.

Experiments have been performed to check that the exploration graph concept correctly functions when decision node rationalisation is used. It was expected that the UGV motion should also be smoother when escaping a cul-de-sac using a rationalised exploration graph. This is due to the fact that the UGV will stop less frequently to check for alternate paths as there are many fewer decision nodes. Results from these experiments are presented in Chapter 8.

7.4.4 Minimax Lookahead

The final enhancement proposed for the local path planner is a method to find the optimum lookahead when using exploration graphs. As was explained in section 7.3.5 it is good for the path planner to look as far ahead as possible. However this can lead to the planner overlooking viable path options when the obstacle density is high.

It would also be possible for the adjustable path planner lookahead to drop too low. This might happen if the claustrophobia parameter had a low setting causing the lookahead to fall quickly in the presence of obstacles. In this case the lookahead may make the path planner so short sighted that it does not notice alternative path choices.

Either of these situations would cause a major problem with the exploration graph. If all of the path choices are not recognised and recorded then the exploration graph is incomplete. Ultimately it could prevent the UGV from finding a path around a set of obstacles as the correct path may be overlooked.

An optimum lookahead would see the maximum number of path choices in a given location. Consider the situation depicted in Figure 7.29 this illustrates that for different lookahead distances the path planner could see a different number of paths. Figure 7.30 shows the likely relationship between the planner lookahead and the number of paths for this example.

The maximum number of path choices are seen when the path planner lookahead is greater than 4m. Thus 4.5m is the minimum lookahead that shows the maximum number of possible paths. This has been called the minimax lookahead for short and is coloured yellow in Figure 7.29. The minimum value has been purposely selected so that it can be used with the adjustable path planner lookahead feature. When enabled the minimax lookahead should act as a base value that the planner lookahead distance will never be allowed to fall below.

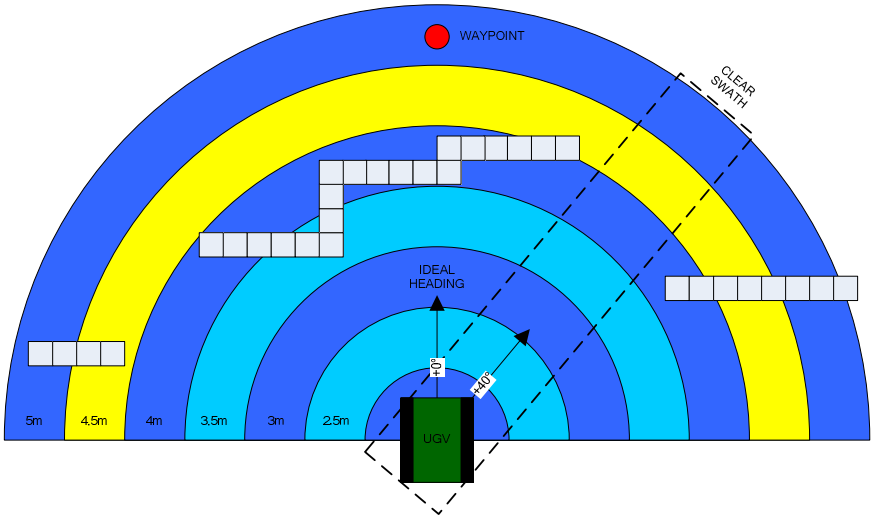

Figure 7.29 - Example of Minimax Lookahead

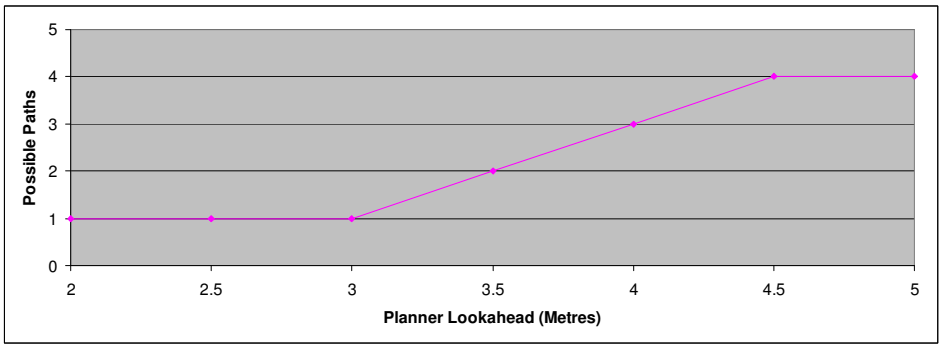

Figure 7.30 - Relationship between Planner Lookahead and Possible Paths

An efficient algorithm has been devised to find the minimax lookahead. It runs in each path planner cycle after the radial swath search process has built the Polar Obstacle Distance Histogram. The algorithm starts by assuming the planner lookahead is 10m and cycles through the polar histogram counting the number of peaks. Then the lookahead is lowered by 0.5m and the count repeated. This process continues until the peak count is seen to fall.

Generally as the planner lookahead drops the peak count should first increase up to a maximum. After this maximum the peak count will then fall as the lookahead continues drop

to the minimum distance of 2m. Hence when the peak count first falls the minimax lookahead has just been passed.

Experiments have been conducted to show the benefit of using the minimax lookahead. These have used the same obstacle course as experiments with the exploration graphs and obstacle flooding. The results are detailed in Chapter 8. It was expected that the minimax lookahead would protect against the situation where the claustrophobia setting was low and the planner lookahead falls too low.

8 EXPERIMENTS

In this chapter the experimental phase of the research is detailed. Experiments are described that used the ARP vehicle to test the different aspects of NOVA. These experiments demonstrate that NOVA is a fully functioning UGV architecture. In the accompanying discussion problems that were encountered during the implementation and ideas for further improvements to NOVA are also outlined.

The chapter begins by examining the scope of the UGV experiments. First the motivation behind the experiments is presented. This looks at the factors that governed the design of the experiments. Next the test location and facilities used during the experiments are described.

In the following sections the actual UGV experiments are then presented. The discussion is split according to the different tasks that NOVA performs. First experiments with the UGV localisation system are described. These demonstrate the operation of the particle filter to fuse the absolute and relative localisation systems on the ARP vehicle. A comparison of the particle filter performance against the two localisation systems is also given.

Experiments with the two obstacle mapping systems developed for NOVA are then detailed. It is shown that both the 2D and 3D mapping systems are viable. Problems associated with noisy sensor data are also discussed. Finally an example of mission mapping is presented.

Next attention is turned to the local path planning system within NOVA. First the performance of the basic system is shown. Unforeseen problems with this system are highlighted and solutions presented. The importance of the planner lookahead is then examined. This is followed by a demonstration of the adjustable lookahead feature and the effect of the UGV claustrophobia setting.

The experiments with exploration graphs are described in the penultimate section of this chapter. It is shown that obstacle flooding is necessary when using an exploration graph to escape from a cul-de-sac. Next the use of rationalised exploration graphs is trialled. The effect of the UGV claustrophobia setting on the cul-de-sac escape process is also investigated.

The chapter concludes with an evaluation of the results that have been presented. This section summarises the capabilities of each part of the UGV architecture. Ideas for further research are also presented here.

8.1 SCOPE OF EXPERIMENTS

This section describes the scope of the experiments performed in this research. Overall aims are stated for what the experimental phase of the research was meant to achieve. The motivation behind individual experiments is then discussed. Each key part of NOVA had to be demonstrated by experiment.

The facilities made available for the experiments are also detailed in this section. A description of the test site is given along with the type of ground surfaces. In addition the objects available for use as obstacles are briefly detailed. This gives an indication of how realistic the experiments were and how far away the UGV is from autonomous operation in the real world.

8.1.1 Motivation

There were three overall aims for the experimental phase of this research. Proving that NOVA is a fully functional UGV architecture was the first aim. Showing that the ARP vehicle is a capable UGV was the second aim. The final aim was to give an evaluation of the novel ideas that were devised during this research.

To achieve these aims a series of experiments had to be devised. These experiments set out to test four important features of NOVA. The four features are localisation, obstacle mapping, local path planning & the use of exploration graphs. Experiments were performed primarily to test that these features worked. The secondary goal of the experiments was to investigate and overcome any problems that were found. The experiments also provided an opportunity to tune key parts of the software.

From a localisation perspective three experimental targets were set. It was important to show that the particle filter was capable of fusing data from the absolute and relative localisation equipment onboard the ARP vehicle. It was essential to demonstrate that the particle filter provides plausible location estimates for the UGV. Finally it was necessary to show that particle filter location estimates do not fall foul of the errors associated with the two underlying localisation systems.

In terms of obstacle mapping these experiments must show that NOVA can produce good quality maps of the environment. Without good maps the advanced autonomous behaviours within NOVA would not be able to function correctly. The idea was to perform the obstacle mapping experiments in three stages. First the 2D obstacle mapping system would be demonstrated. Next the 3D obstacle mapping system would be tested on flat ground. Finally the performance of the 3D obstacle mapping system would be trialled in more natural terrain.

The initial aim of the local path planning experiments was to evaluate the basic local path planner design. It was important to establish if it had any flaws and find solutions. Next the effects of changing the path planner lookahead needed to be investigated. It was essential to check what happened to the UGV behaviour as the lookahead was reduced. Following this it was necessary to demonstrate the benefits of an adjustable lookahead. Finally the novel

concept of UGV claustrophobia and the effects of changing the underlying parameter had to be evaluated.

Several different obstacle courses needed to be devised for the path planning experiments. To test the basic path planner and planner lookahead distance ad hoc obstacle courses could be used with isolated obstacles. A more sophisticated obstacle course was needed to demonstrate the concepts of adjustable lookahead and claustrophobia. The course would need to be a field of obstacles that the UGV could find a way through or circumnavigate.

Exploration graphs and obstacle flooding are novel concepts for this research. So experiments were required to show that these two concepts worked and could be used to help the UGV escape from cul-de-sacs. To this end an obstacle course with a cul-de-sac was devised. A series of experiments were then used to evaluate the cul-de-sac escape behaviour and make it as effective as possible.

To facilitate the experiments NOVA was extended to record many of the key variables in the software. Hence NOVA acts like a black box recorder and after each UGV run variables can be plotted to show exactly what happened in the software. This recording facility is what has been used to create the numerous graphs in the remainder of this chapter.

8.1.2 Test Facilities

Experiments were performed at the Remotec UK production facility based at Market Deeping in Lincolnshire. GPS coordinates for the site are approximately 52.68792°N and 0.31194°W. This equates to 311360m north and 514196m east in the OSGB36 coordinate system. Figure 8.1 shows the smallest scale Ordnance Survey map available for the area around the test location [Digimap 2008]. The map has a scale of 1:10000 and squares are 1km wide. A blue circle on the map highlights the test location.

The test site has a large outdoor area that is used for testing robotic vehicles. This outdoor area includes a patch of grass with dimensions no less than 25m x 40m. Permission was granted to use this as the main test area for this research. The surface of the grassy area is relative flat with a few ruts and undulations. A number of access roads and pavements with smooth flat surfaces were also used during UGV experiments.

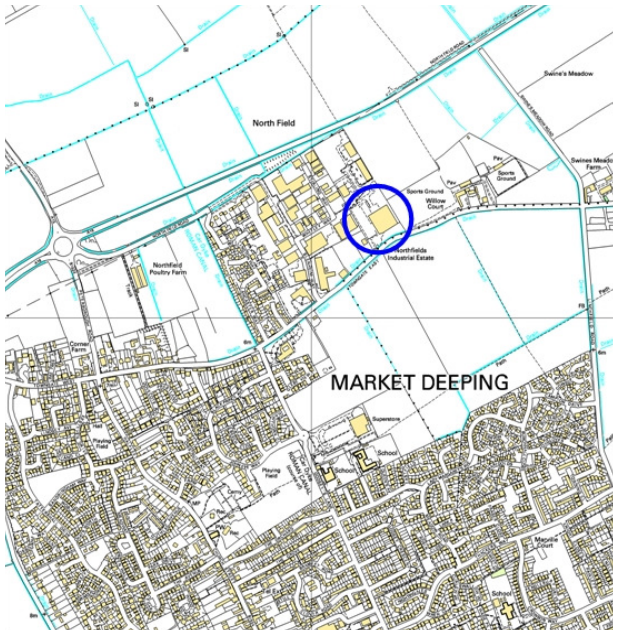

Figure 8.1 - Map Showing Test Facility 1:10000 Scale

To test the features of the obstacle avoidance system within NOVA it was necessary to construct obstacle courses for the UGV to autonomously navigate. The test site had a healthy supply of wooden pallets that could be erected as isolated obstacles. Flexible fencing made from tarpaulin and metal spikes was also procured to allow more complicated arrangements of obstacles to be built. To ensure experiments could be accurately repeated chalk paint was used to mark the positions of obstacles on the ground.

An important consideration is how the experiments detailed in this chapter compare to what the UGV might experience when operating in the real world. Firstly the experimental environment

is largely static. Secondly the majority of obstacles used were deliberately selected to be detected by both the 2D and 3D obstacle detection systems. Clearly these both represent simplifications of the world. It should be said that elements of NOVA have been designed to allow operation in dynamic environments. Additionally the 3D obstacle mapping system has the potential to detect a much larger variety of obstacles than the 2D system.

8.2 LOCALISATION EXPERIMENTS

In this section experiments with the UGV localisation system are detailed. The section begins by discussing an issue with the basic particle filter idea and showing the solution. Next the performance of the particle filter is evaluated. Comparisons are drawn with location estimates from the GPS receiver and dead reckoning system.

Further experiments are then described that were used to further tune the performance of the filter and simplify its operation. First an experiment was performed to investigate the need for the resampling step in the filter to prevent degeneration. Then experiments were performed to investigate how the model of the GPS error ellipse affects the filter.

8.2.1 Evaluation of Particle Filter

Early experiments with the particle filter showed a problem that was not considered in the initial filter design. Immediately after start up when the UGV was stationary the location estimate from the particle filter would drift. This drift could take the particle filter location estimate up to several metres from the initial location. Figure 8.2 illustrates the problem.

In the diagram the first particle filter location estimate has the coordinates 311367.59m north and 514239.34m east. Likewise the initial GPS location estimate has the coordinates 311367.53m north and 514239.31m east. Since the dead reckoning system provides only relative location estimates these are shown translated by the first GPS location estimate. The

figure illustrates that the location estimate from the particle filter drifted first to the south then east. This drift happened in a period of around 1 min.

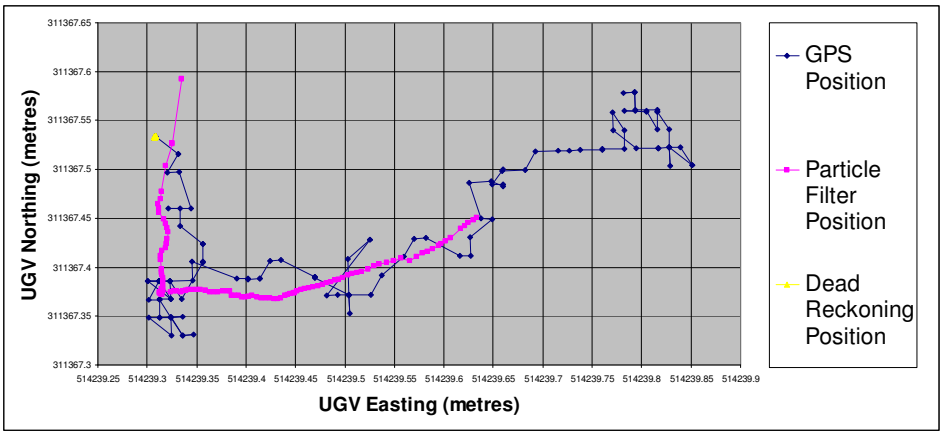

Figure 8.2 - Effect of GPS Drift when UGV Stationary

The encoders on the UGV were correctly measuring no distance travelled. Thus the control input for the particle filter was null. This implied that the particles in the filter were not being translated. However resampling of the particle set was taking place whenever the number of effective particles in the population fell below the resampling threshold.

Without particle translation the resampling process was gradually discarding the particles with lower weight. In their place clones of the fitter particles were being generated. Of course the weight of each particle is directly related to the distance between the particle location and the recent state observations provided by the GPS receiver.

It can be seen from Figure 8.2 that the particle filter was slowly following the drift of the GPS location estimates. Each time the particles were resampled the particle filter location estimate was dragged towards the latest GPS location estimate. This would continue until the resampling process had reduced the particle set to clones of only a handful of close together particles.

To combat this undesirable effect the original particle filter algorithm as described in Chapter 6 was modified. Two steps in the algorithm are now deliberately blocked whenever the UGV is stationary. These are the steps that update the particle weights and resample the particle set. By blocking these steps the new state observations from the GPS receiver are effectively ignored. Once the UGV begins to move the full particle filter algorithm is run as normal.

Next the performance of the particle filter was evaluated. To do this a 10m x 10m square was marked out on the ground. On the south side of the square there were some undulations in the ground but the surface was otherwise flat. The ARP vehicle was given a route that mimicked the markings on the ground and driven repeatedly round the route in the clockwise direction. A plot of the points where the UGV turned at the corners of the square route is shown in Figure 8.3.

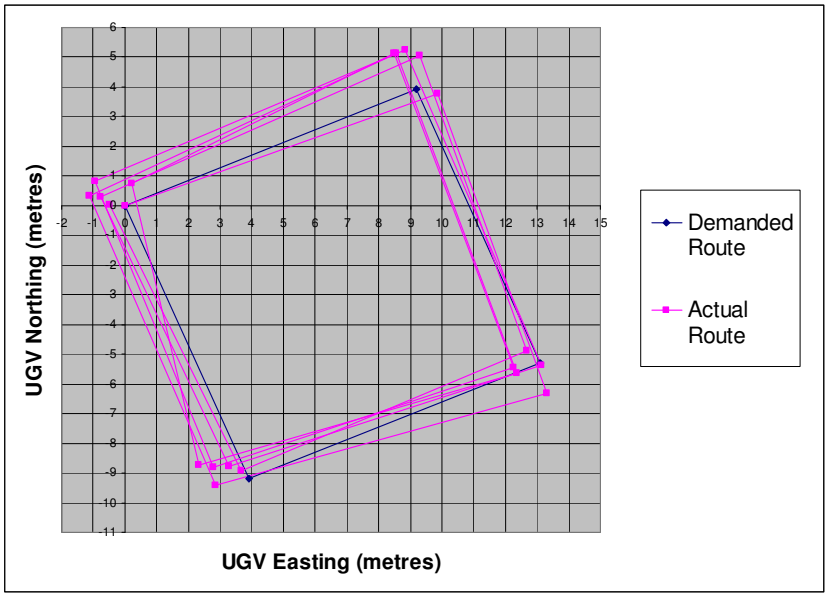

Figure 8.3 - Particle Filter Error following 10m x 10m Square

NOVA considers route waypoints as circles with a diameter of 0.6m. So the UGV may begin the process of stopping then turning at any point within that circle. As such some amount of spread has to be expected in the location of the turning points. The diameter for waypoints was

determined by the stopping distance of the ARP vehicle and is not related to the accuracy of the localisation system.

Figure 8.3 shows the UGV driving 5 laps of the square route giving a combined distance travelled of 200m. The points on this plot have an average error of 0.94m and the standard deviation for the error is 0.39m. There is no obvious trend showing that the UGV may be diverging from the square marked on the ground. In fact one of the later turning points was almost exactly at a corner of the marked square. On repeated runs of this experiment the UGV never strayed significantly further from the marked square.

The values given above for the average and standard deviation of the localisation error must be considered as upper bounds for the true localisation error. Clearly if it were possible to use much smaller route waypoints for the UGV a tighter bound could be placed on this localisation error. So the experiment reported here equates to an upper bound for localisation error of 0.78m 95% of the time. This is better than the corresponding value for the GPS receiver in ideal conditions.

8.2.2 Comparison with GPS and Dead Reckoning

Does the particle filter perform better than the GPS receiver or dead reckoning system? The intention of the particle filter was to fuse location estimates from the two complementary localisation systems and overcome the sources of error associated with each system. So an experiment was performed to compare the performance of the particle filter with the location estimates from the relative localisation equipment and the GPS receiver.

For this experiment the square painted on the undulating grass was again utilised. As before the ARP vehicle was given waypoints that mimicked the square route and the UGV autonomously drove round the square route 5 times. This gave a total traverse of 200m.

During the experiment location estimates from the particle filter, GPS receiver and dead reckoning system were recorded. The dead reckoning system was again initialised with the first location estimate from the GPS receiver. Figure 8.4 shows the results from the experiment. It was observed that the UGV followed the square route with the same kind of accuracy as in the last experiment.

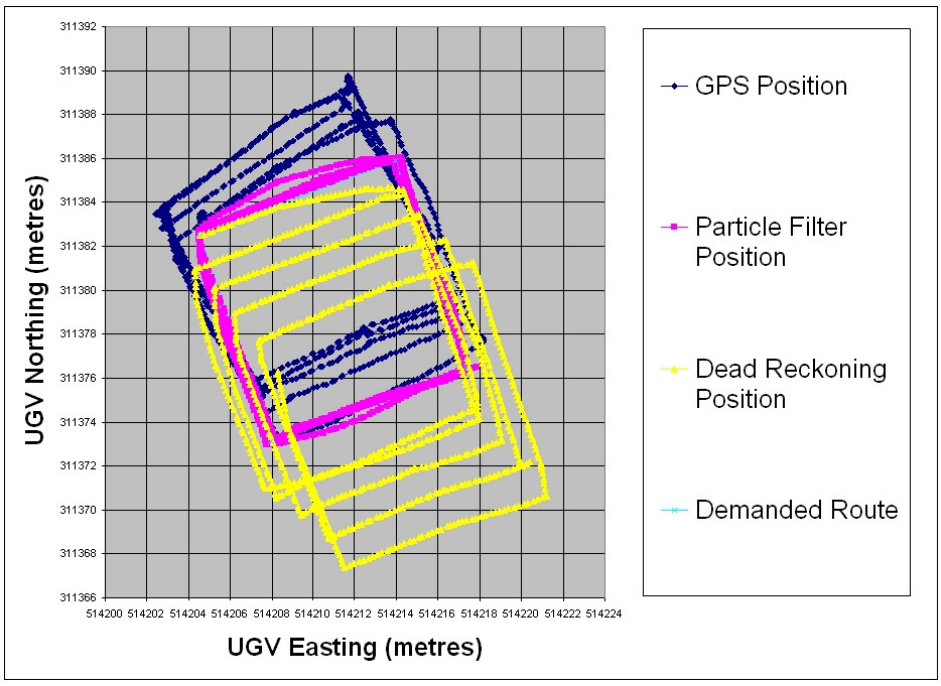

Figure 8.4 - Comparison of Localisation Systems

It is clear from Figure 8.4 that the location estimates from the dead reckoning system rapidly diverge from the true UGV location. The yellow line that describes these location estimates spirals away from the particle filter location estimates. After 5 laps of the square route there was a 7.24m difference between the location estimates.

As indicated in Chapter 6 the error in the dead reckoning system is mostly due to the fact that the ground has undulations. These induce error in the encoder measurement and UGV heading measurements. The encoders are measuring the distance the tracks move to overcome

undulations rather than horizontal movement of the UGV. Also when the pose sensor is not horizontal some of the vertical component of the Earth's magnetic field will be measured and so marginally deflect the heading measurement.

Location estimates from the GPS receiver are also affected by error. Unlike the dead reckoning system they do not diverge but they do drift from time to time. In the run shown in Figure 8.4 there were two drifts in the GPS location estimates. First the estimates drift around 1m north at start up. Then during the second lap of the square the estimates seem to drift north again by around 1m. After this the GPS location estimates seemed to stabilise around 2m north of the corresponding estimates from the particle filter.

This experiment was repeated a number of times. Similar results were seen at each attempt. Location estimates from the dead reckoning system would always spiral away from the true UGV location. From time to time GPS location estimates would drift a short distance typically to the north or south. Additionally the north side of the square route traced by the GPS always looked to be at a slight angle. The error statistics reported by the GPS usually seemed to be a little optimistic given the observed drifts and deformed shape of the square route.

During the runs of this experiment the variance of latitude error reported by the GPS receiver was generally much less than 0.5m with a couple of spikes up to 1m. Meanwhile the longitude error variance was even smaller typically under 0.1m. There was no obvious correlation between the spikes in the latitude error variance and the drift of the GPS location estimates. It is likely that the GPS drift is caused by the presence of the factory building on the south side of the test area. The building has the potential to cause satellite occlusion and signal multipathing which could both result in drift.

This experiment clearly shows that the particle filter can successfully fuse the two underlying localisation systems. Importantly the particle filter location estimates do not show any sign of the errors associated with either the dead reckoning system or GPS. So it can be said that the particle filter provides greater absolute localisation accuracy than the GPS receiver alone. Also for longer runs the particle filter is capable of tracking the UGV motion much more accurately than the dead reckoning system.

8.2.3 Particle Filter Degeneration

Once it was proved that the particle filter algorithm worked subsequent experiments concentrated on the individual steps of the algorithm. In Chapter 6 it was stated that the resampling step was a necessary part of the particle filter algorithm. The reasoning behind this was that without resampling the particle set would degenerate to a state where only a few particles had significant weight. An experiment was performed to observe the degeneration effect.

For this experiment the UGV was driven autonomously a distance of 10m due east over gently undulating ground. During the run the location of each particle in the population used by the particle filter was tracked. Figure 8.5(a) shows the location of the particle set at the beginning of the 10 run while the state of the particle set at the end of the run is shown in Figure 8.5(b). Critically the centre of the final particle set is approximately 10m due east of the initial particle set.

At the beginning of the run the spread of the particles is determined by the uncertainty associated with the initial state observation from the GPS receiver. As the UGV drives the particle set will be weighted according to subsequent state observations from the GPS receiver and the particle set will be resampled. As particles with lower weight are discarded by the resampling process the spread of the particle set tends to reduce. Thus the particle filter homes

in on the true location of the UGV. After a few seconds of movement the particle spread is determined by the error model for the dead reckoning system.

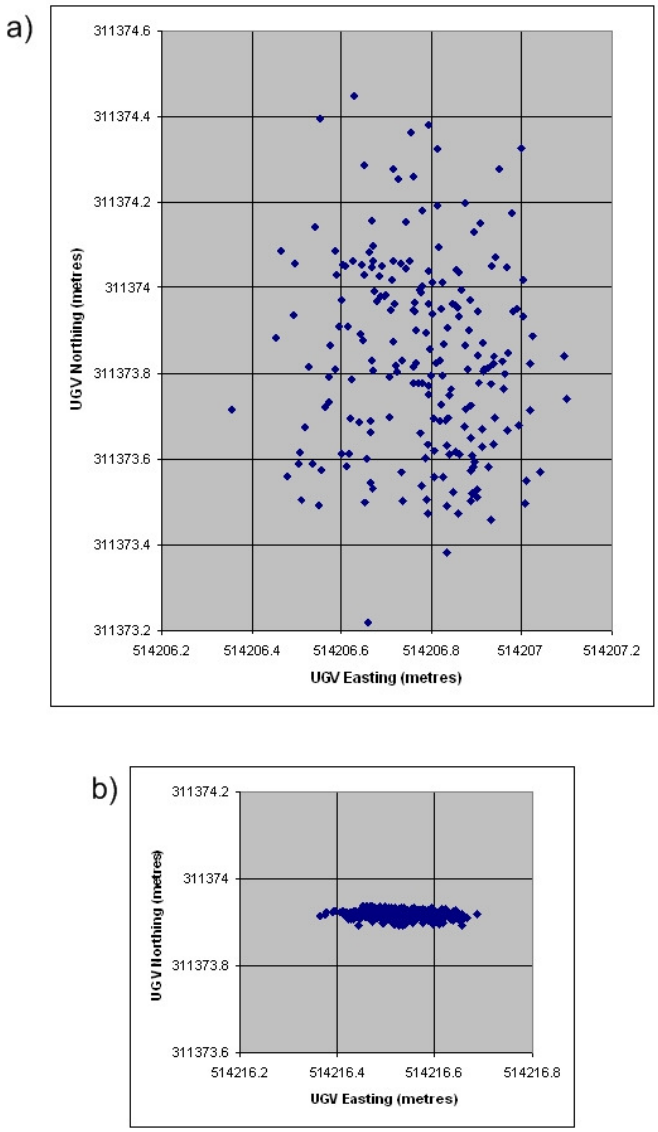

Figure 8.5 - Particle Spread during 10m Run with Resampling

Figure 8.5(a) shows at start up the particle set had a spread of around 1.3m north to south. In the east to west direction the particle spread was just under 0.8m. By the time the UGV had driven 10m the particle set had crowded together. The final north to south spread of the particles in Figure 8.5(b) was less than 0.1m. While in the east to west direction the spread

was just over 0.3m. This reflects the fact that the UGV was moving due east and so is dominated by the measurement error from the wheel encoders and compass.

Next the experiment was repeated with the resampling step of the particle filter deliberately blocked. Figure 8.6(a) shows the location of the particles at the beginning of the 10m run without resampling. The location of the particles at the end of the run is shown in Figure 8.6(b). As before the centre of the final particle set is approximately 10m to the east of the initial particle set.

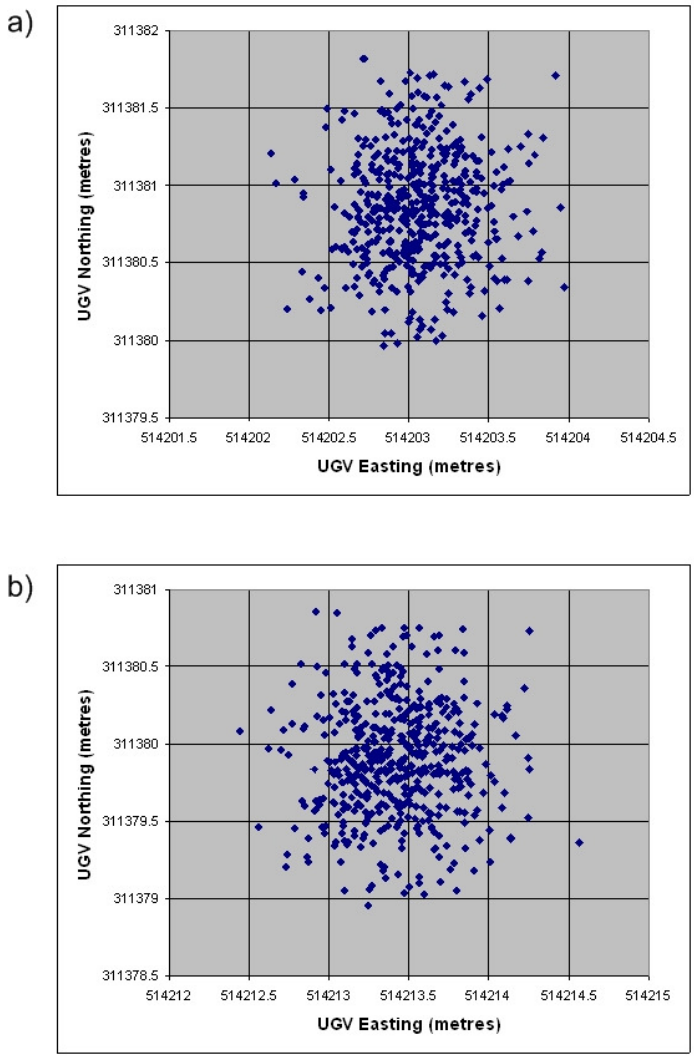

Figure 8.6 - Particle Spread during 10m Run without Resampling

This time there is a major difference. The spread of the final particle set is almost identical to that of the initial particle set. Both have a north to south spread of around 2m and an east to west spread of around 2m. Most importantly the particle set does not home in on the location of the UGV.

Without the resampling process the spread of the particles cannot change a great deal in a short run. As the UGV moves the particles are translated according to the control input given by the dead reckoning system. The measurement errors for the dead reckoning system are individually much smaller than the initial particle spread. Hence the spread of particles remains roughly the same as was determined by the uncertainty of the initial state observation from the GPS receiver.

It is important to know if the quality of the particle set was any worse at the end of the second run. One way to check this is to look at the number of effective particles in the particle set. In Chapter 6 the number of effective particles was defined as the reciprocal of the sum of squared particle weights. In the first run the resampling process ensures that at least two thirds of the particle population are effective. Figure 8.7 illustrates how the number of effective particles changed during the second run.

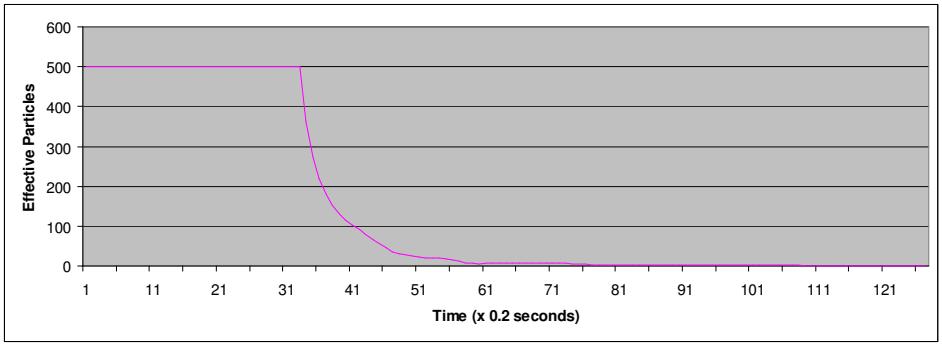

Figure 8.7 - Effective Particles during 10m run without Resampling

Until the UGV starts to move all of the particles have equal weight and so all particles are counted as effective. Once the UGV is in motion the number of effective particles seems to decay at an almost exponential rate. The run took around 25 seconds to complete but after 17 seconds there was only one effective particle. One particle had a weight of 0.99847 while all the rest had negligible weight. So from this point the particle filter location estimate was based almost entirely on the weight of a single particle.

This experiment demonstrates that the resampling step of the particle filter algorithm is essential. Without resampling the particle set cannot converge on the true UGV location. As a consequence nearly all of the particles will be representing unlikely locations for the UGV. The particle filter location estimate will be based on the location of the particle that was closest to recent state observations from the GPS receiver. Based on only one particle the accuracy of the particle filter location estimate will be poor.

8.2.4 Error Ellipse Model

The final set of localisation experiments concentrate on the GPS error ellipse model used within the particle filter. This model is used when updating the weight of each particle in line with the latest state observation from the GPS receiver. As such the model dictates how much the particle filter responds to new GPS location estimates.

In a sense the stricter the model of the error ellipse the more influence it should have on the particle set. This will be reflected by how the particle set is weighted. Stricter models will place higher weight on the particles near the latest state observation from the GPS receiver. Consequently this affects how often the particle set needs to be resampled. So a stricter model of the error ellipse would imply more frequent resampling.

In Chapter 6 two error ellipse models were proposed for the GPS receiver. The first was the true model that uses all of the error information available from the GPS receiver. This specifies an ellipse that has an orientation dictated by the momentary location of the GPS satellites. The second was a simplified model that ignored the covariance between the latitude and longitude errors. This specifies a simple north oriented error ellipse.

An experiment was performed to see what difference could be observed from using the simplified error model. For consistency the 10m by 10m square route marked on the ground was again used. As before the UGV was requested to drive 5 clockwise laps of the route giving a total distance travelled of 200m. During the first run the true error ellipse model was used. In a second run the simplified error ellipse model was tried.

For both runs the UGV closely followed the square marked on the ground as in the experiments described in section 8.2.1 and section 8.2.2. The only noticeable difference between the results from the two runs with different error ellipse models was in the degeneration of the particle filter. Figure 8.8 is a plot of the effective particles from the run using the true error ellipse. Contrast this with Figure 8.9 which is a similar plot for the run using the simplified north oriented error ellipse.

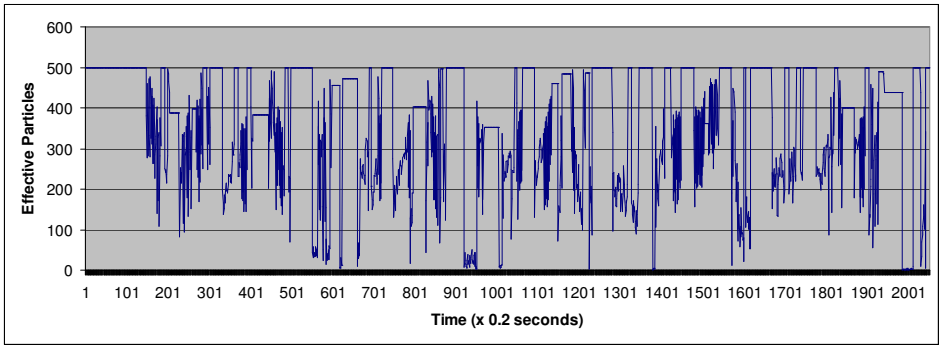

Figure 8.8 - Effective Particles when using True Error Ellipse

Two key observations can be drawn from the plots. First using the true error ellipse model the particle filter seemed to degenerate faster. This is supported by the fact that the number of effective particles often dropped to very low values. Second the particle filter resampled more often when using the true error ellipse model. Both of these observations imply that the true error ellipse model is stricter. Hence the particle set was being influenced more by location estimates from the GPS receiver when the true error ellipse model was used.

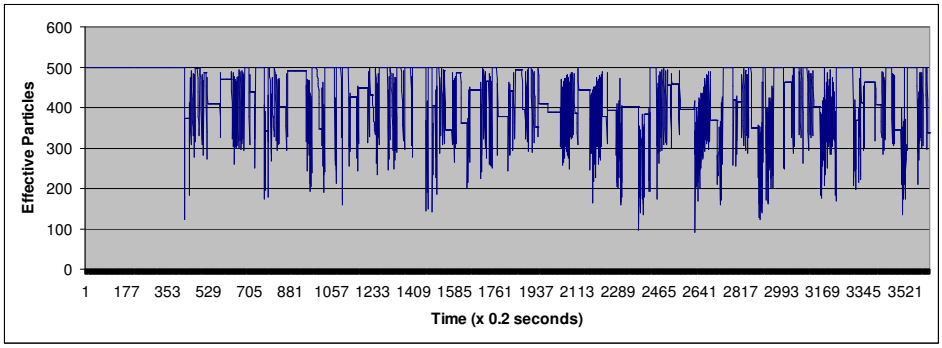

Figure 8.9 - Effective Particles when using North Oriented Error Ellipse

In section 8.2.2 it was stated that error data from the GPS receiver seemed a little optimistic. In reality the true error associated with the GPS location estimates had to be larger to explain the observed drifts in the estimates. So perhaps using the stricter error ellipse model is not a good thing. With the stricter model the particle weighting and resampling process is more likely to drag the particle set to follow GPS drifts.

To that end the experiment was run again using the simplified error ellipse model with a slight modification. This time the standard deviation values for latitude and longitude error from the GPS receiver were processed before use. All standard deviation values under 0.5m were deliberately inflated to 0.5m. Given the observed GPS drifts the idea of this modification is that the low error values reported by the GPS receiver should not be trusted.

As before the UGV made 5 laps of the route and again closely followed the square marked on the ground. Figure 8.10 shows the number of the effective particles during this third run. By comparison with the similar plots from the earlier runs it can be seen that degeneration of the particle set is lower still. The number of effective particles never drops to the lows seen with the previous error models.

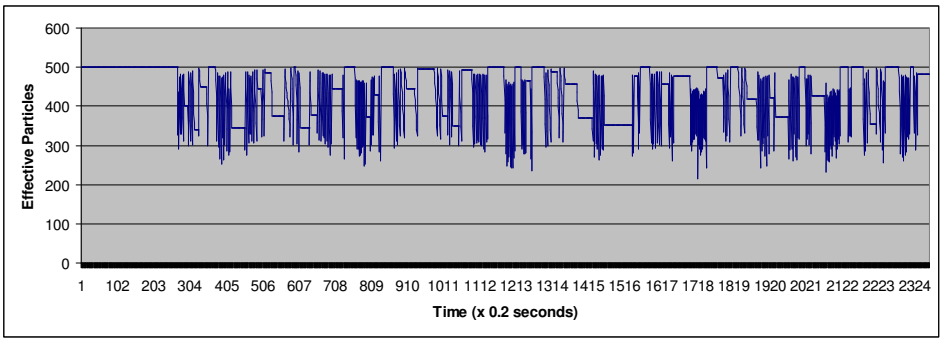

Figure 8.10 - Effective Particles when using Inflated Error Ellipse

Each of the three error ellipse models trialled allowed the UGV to repeatedly follow the 200m route without diverging. It is clear that the true error ellipse is the strictest model and results in a quicker degeneration of the particle filter. This in turn means more frequent resampling of the particle set is required. The simplified north oriented error ellipse models do reduce the influence of the GPS location estimates on the particle set. When the particle filter was used for other experiments in this research the inflated error ellipse model was preferred.

8.3 2D Obstacle Mapping Experiments

Experiments with the 2D obstacle mapping system are described in this section. First an unforeseen problem is highlighted that resulted in early obstacle maps smudging as the UGV moved around. A solution to this is demonstrated. An example is then given that shows the 2D mapping system works as designed. Subsequent experiments identify sources of error for the 2D obstacle mapping system. Finally the idea of mission mapping is demonstrated.

8.3.1 Smoothing Heading Measurements

Early experiments with obstacle mapping highlighted a problem that affected both the 2D and 3D obstacle mapping systems. Obstacle maps built while the UGV was stationary looked correct. However when the UGV was in motion even at slow speeds the obstacle map would seem to smudge. Figure 8.11 illustrates the problem.

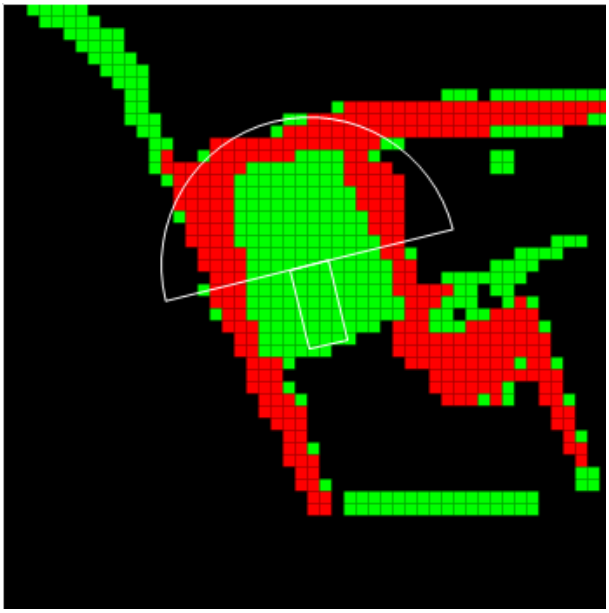

Figure 8.11 - Smudged Obstacle Map from Early Prototype

Figure 8.11 is a screenshot of an obstacle map taken from an early prototype of the obstacle mapping system. This obstacle map was generated as the UGV was driven manually into a cul-de-sac formed by several freight containers. In reality the sides of the cul-de-sac were straight and orthogonal to the end of the cul-de-sac. The obstacle map incorrectly showed the left hand side as encroaching into the cul-de-sac. Also the freight container obstacles were described by thick bands of untraversable pixels. The perimeter of straight edged solid obstacles should only be one or two pixels wide.

After some testing it was discovered that the smudging effect was being caused by the pose sensor on the ARP vehicle. When the UGV was in motion the heading measurement given by the pose sensor had a tendency to oscillate. This sensor noise then affected the mapping system when new range points were transformed into coordinates for the obstacle map. Since the range points are several metres in front of the UGV a small error in the heading measurement caused a noticeable lateral error in the placement of the range points on the map.

Figure 8.12 shows a trace of the raw UGV heading measurement during a short straight run on a relatively flat grass surface. As soon as the UGV started to move the heading measurement began to oscillate. The maximum heading error in this run was 6°. For range points more than 2m in front of the UGV a heading error of this magnitude would cause a lateral placement error wider than an obstacle map cell.

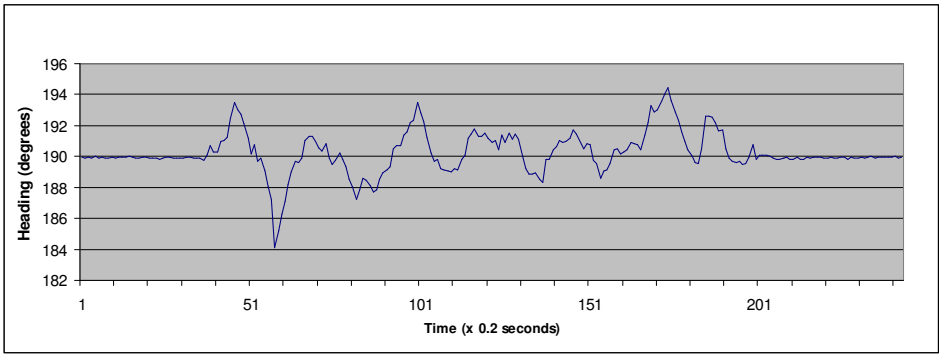

Figure 8.12 - UGV Heading Noise

What was needed was a method to smooth the raw heading measurements from the pose sensor on the ARP vehicle. A large variety of smoothing filters were tried to find a filter that gave acceptable performance. At first median filters and averaging filters were tested. Figure 8.13 shows the result of applying an averaging filter to the data in Figure 8.12.

To generate the average heading values plotted in Figure 8.13 a window of 3 seconds of heading measurements was used. This equated to averaging the last 15 measurements at each

time step. After smoothing the heading data from the initial run using this filter the maximum heading error was still in excess of 2°.

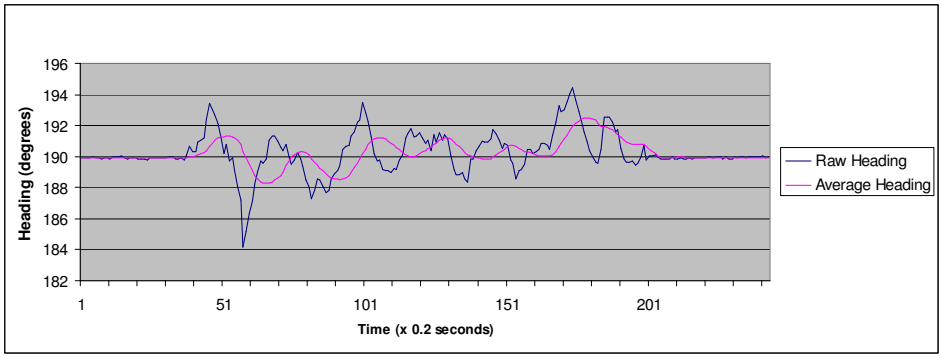

Figure 8.13 - Use of an Average Filter to Smooth Heading Data

For acceptable obstacle mapping the heading error must be small enough that range points placed at the extreme edges of the obstacle map have a lateral error less than half the width of a map cell. This implies that the largest tolerable heading error would be approximately 1.1°. To guarantee this performance using an averaging filter many seconds of heading measurements would have to be stored and processed.

At this point more advanced types of smoothing filter were investigated. Honeywell recommended the use of a non-linear filter to get the best performance from the pose sensor [Withanawasam 1998]. The recommended filter is similar to a traditional single pole recursive filter [Smith 1997]. Only the latest heading measurement is considered by the filter. At each time step the difference between the latest measurement and the current filter value is calculated. A proportion of this difference is then incorporated into the new filter value.

Let the smoothed heading value output by the filter at time t be denoted as s(t). Then if the raw heading measurement is denoted by r(t) the difference value d(t) would be given by the following equation:

$$d(t) = r(t) - s(t-1)$$

A recursive equation is used with the difference value to incorporate the latest raw measurement into the smoothing filter. The proportion of the difference value incorporated into the filter at each time step is given by a gain parameter G. Thus the recursive equation for the single pole filter is given by:

$$s(t) = s(t-1) + d(t)\,G$$

Unlike a traditional single pole recursive filter the proportion of the difference value that is incorporated into the filter at each time step is not a constant for the recommended filter. The filter gain is a variable g(t) that is related to the difference between the latest measurement and the current smoothed filter value. So the recursive equation for the Honeywell recommended filter can be written as:

$$s(t) = s(t-1) + d(t)\,g(t)$$

The gain calculation for the smoothing filter uses two constants S and L to determine the response of the filter to changes in the raw measurements. S is a value in the range 0 to 1 and specifies the amount of smoothing that should be applied to the raw measurement data. Smaller values imply greater smoothing. L is an integer greater than 0 and acts as a noise threshold for the filter. The gain variable for the Honeywell filter is given by:

$$g(t) = S\left(1 + \frac{d(t)}{L}\right)^2$$

Appropriate values for the constants in the gain calculation were found by trial and error. Best performance was seen when S was 0.01 and L had a value of 2. Figure 8.14 shows the result

of smoothing the heading data in Figure 8.12 using the non-linear filter with these hand tuned constants.

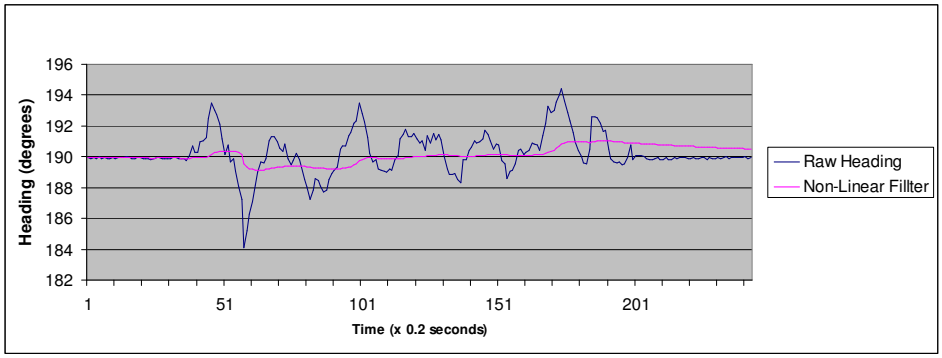

Figure 8.14 - Use of a Non-Linear Filter to Smooth Heading Data

The non-linear filter worked as intended and kept the heading error within the critical ±1.1° range for the example run. Subsequently it was adopted as the method to smooth heading measurements in real time within NOVA. Once this smoothing filter was in use the obstacle map smudging problem completely disappeared.

8.3.2 Basic 2D Confidence Mapping

Once the initial problems with heading noise were overcome the 2D confidence mapping system was quickly perfected. The parameters of the mapping system were then tuned to give an approximate one second response time. So the obstacle map confirms the presence of an obstacle within a second of it entering the field of view of the UGV. Likewise if the obstacle later moves the position of the obstacle is updated within the same time frame. This works exceptionally well and allows operation of the UGV at full speed.

Figure 8.15 shows an example of the ARP vehicle building a 2D obstacle map of the environment. Parts (a) to (e) of the figure are a time ordered sequence of snapshots from the

obstacle map. These snapshots were generated by recording the graphical output of the NOVA Computer to a video file. Between each snapshot approximately 5 seconds elapsed.

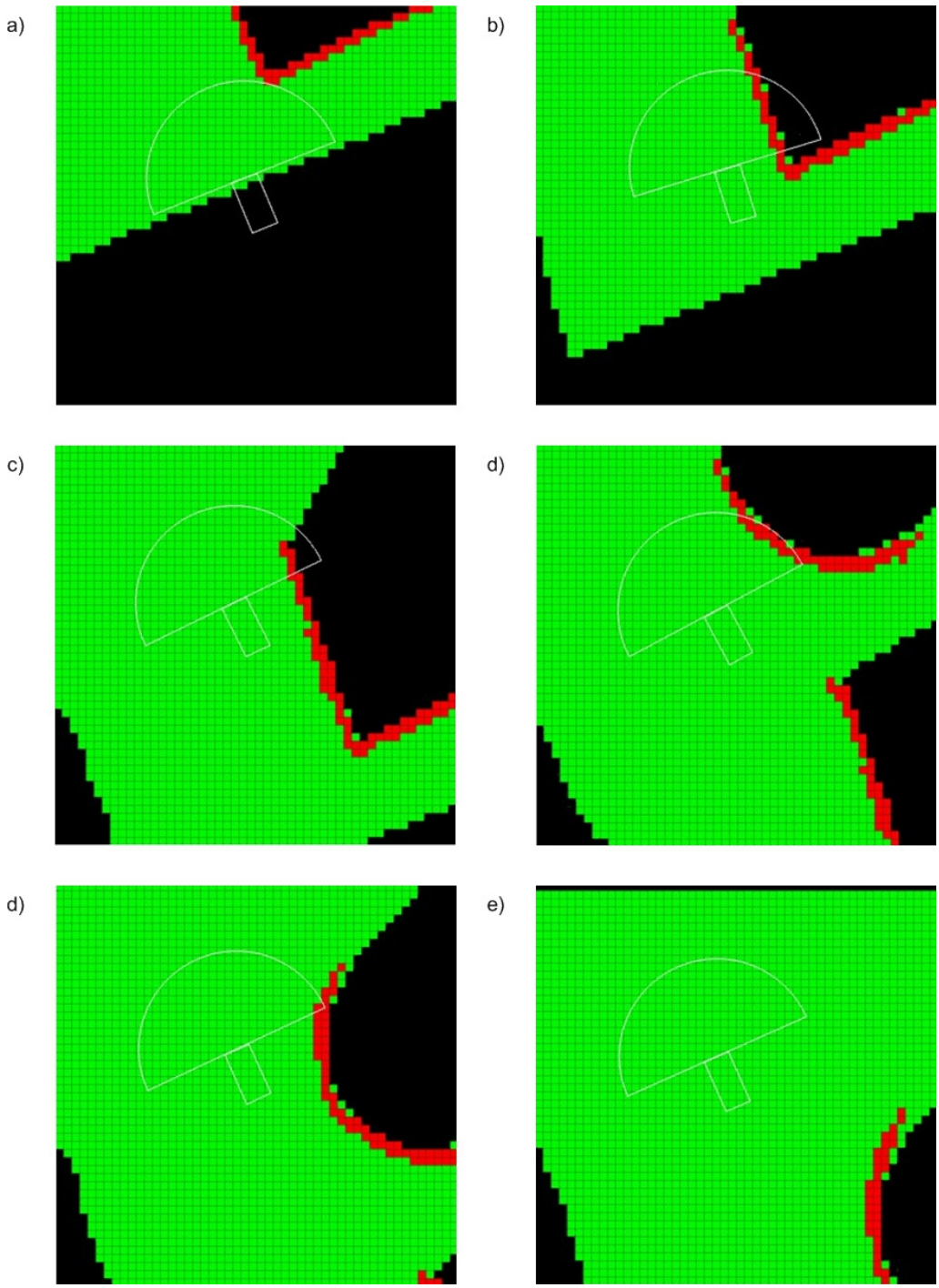

Figure 8.15 - Example of Confidence Mapping

For this experiment the ARP vehicle was driven autonomously at full speed following a fixed heading. During the run the UGV passed two small buildings. The first was rectangular and approximately 7m in width. The second building was circular and had a diameter of roughly 6m. For the first part of this run the UGV was driving across grass while the last few metres were on pavement.

Throughout the experimental phase of this research the basic 2D confidence mapping system has performed very well. The maps produced were always clear and detected obstacles have appropriate dimensions. Very little noise is shown while the UGV is in motion. Also the map always reacted rapidly when objects within the UGV field of view move. It was even possible to see when obstacles were being blown around in the wind.

8.3.3 Sources of Error

In the last subsection the high quality of the confidence mapping system was demonstrated. Following this success a series of experiments were conducted to find out how the mapping system degraded in the presence of errors in the underlying localisation system. The confidence mapping system was trialled using each of the particle filter, stand alone GPS and stand alone dead reckoning system to provide UGV localisation. Where possible the localisation equipment was deliberately subjected to error inducing conditions.

The example run illustrated in Figure 8.15 was repeated with the ARP vehicle using each of the trio of localisation systems in turn. The mapping system showed least noise when the stand alone dead reckoning system was used for localisation. Next best was the particle filter which infrequently showed small amounts of map smudging. Using the GPS was far worse with regular smudging as the location estimates drifted.

Figure 8.16 perfectly captures the situation with GPS localisation. The ARP vehicle was in approximately the same position as it was for Figure 8.15(d). However in this experiment jumps in the GPS location estimate have clearly occurred. At the time Figure 8.16 was generated the GPS location estimates were drifting in a south westerly direction. It can be seen that the estimates have rapidly drifted around 1.5m from when the UGV first sensed the wall of the small circular building.

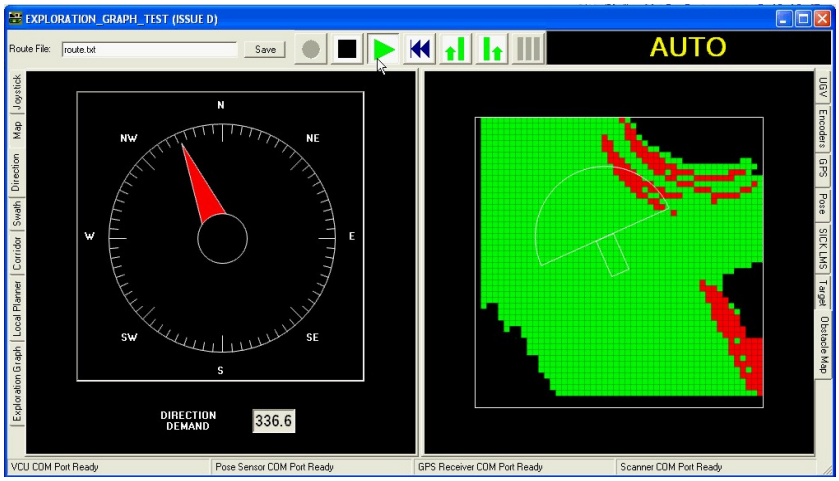

Figure 8.16 - Map Smudging caused by jumps in GPS Location Estimate

In additional runs using just the GPS receiver for UGV localisation other strange things have been observed. On several occasions drift in the location estimate has lead to the UGV being repositioned inside of obstacles. A situation has also been observed where the GPS location estimate has not kept up with the motion of the UGV for several seconds. This results in the obstacle map staying stationary while the confidence mapping algorithm is updating all the map cells within the UGV field of view. To the observer this odd behaviour makes it look as if one half of the obstacle map is scrolling while the rest is still.

From time to time during this research another source of error has been seen to have an effect on the obstacle mapping system. Placement of range points on the obstacle map uses the

heading measurement from the pose sensor regardless of which UGV localisation system is in use. The compass within the pose sensor is known to be sensitive to magnetic field sources. So whenever the UGV drives through a magnetic field the obstacle map is likely to be distorted.

In section 8.3.1 it was seen that a small oscillation in the heading measurement could cause smudging of the obstacle map. Driving the UGV through a magnetic field can cause a huge deflection of the heading measurement. While this is happening new range points will be added to the obstacle map at the deflected heading. This has the effect of drawing phantom duplicates of detected obstacles on the obstacle map.

Figure 8.17 illustrates the problem. This screen shot from NOVA was captured during a test of the local path planner. As the ARP vehicle rounded the corner of the factory building the pose sensor heading was deflected anticlockwise by 35°. This caused several phantom walls to be drawn on the obstacle map. Machinery in the corner of the building was later found to be generating a strong electro-magnetic field.

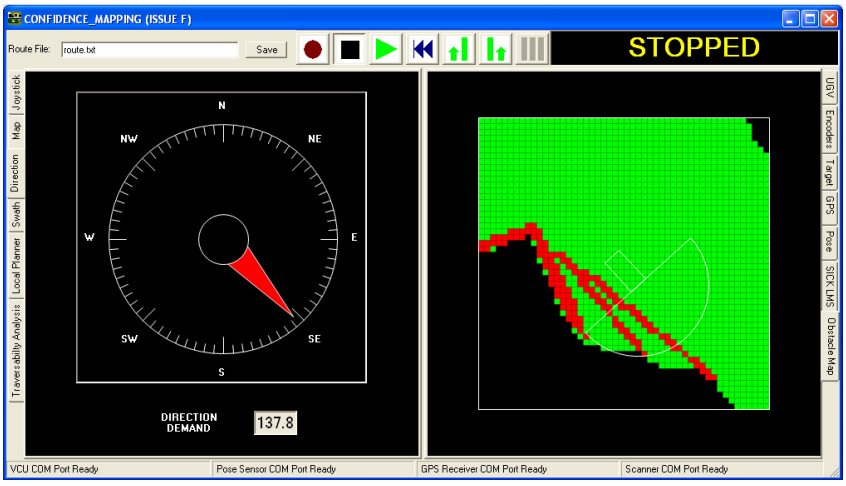

Figure 8.17 - Heading Deflected by Magnetic Field

Thus while the 2D confidence mapping system normally produces high quality maps it does have a few limitations. It works well with the particle filter or the stand alone dead reckoning system for UGV localisation. However GPS location estimates are too noisy to allow the map to be used reliably if that was the only UGV localisation method. Finally since the mapping system is reliant on the heading measurements to place range points on the obstacle map it is also susceptible to any error in this measurement. It has been demonstrated that when the UGV drives through a magnetic field the obstacle map is likely to be corrupted.

8.3.4 Mission Mapping

A final experiment was performed with the 2D obstacle mapping system to see if the idea of mission mapping was viable. The basic concept of mission mapping is to use a much larger obstacle map to record a complete UGV mission. This offers a number of advantages. Firstly the UGV can begin to avoid obstacles as soon as they are detected. Secondly it gives a complete obstacle map for a return journey or for other vehicles that need to use the route. It also shows the UGV operator a much clearer view of the area the UGV has driven through and the sensor range of the UGV.

The critical part of this experiment was to test if the mapping system could handle updating a much larger number of map cells. For this experiment the basic 10m x 10m confidence map was expanded. The 0.2m cell size was retained but the size of the map was increased to 40m x 40m. This in turn increased the number of cells in the obstacle map from 2500 to 40000.

Figure 8.18 shows an example UGV run using the larger obstacle map. This time the UGV was given a waypoint route to follow across a narrow grassy area next to the main UGV test area. Parts (a) to (e) of the figure are time ordered with around 10 seconds elapsing between snapshots.

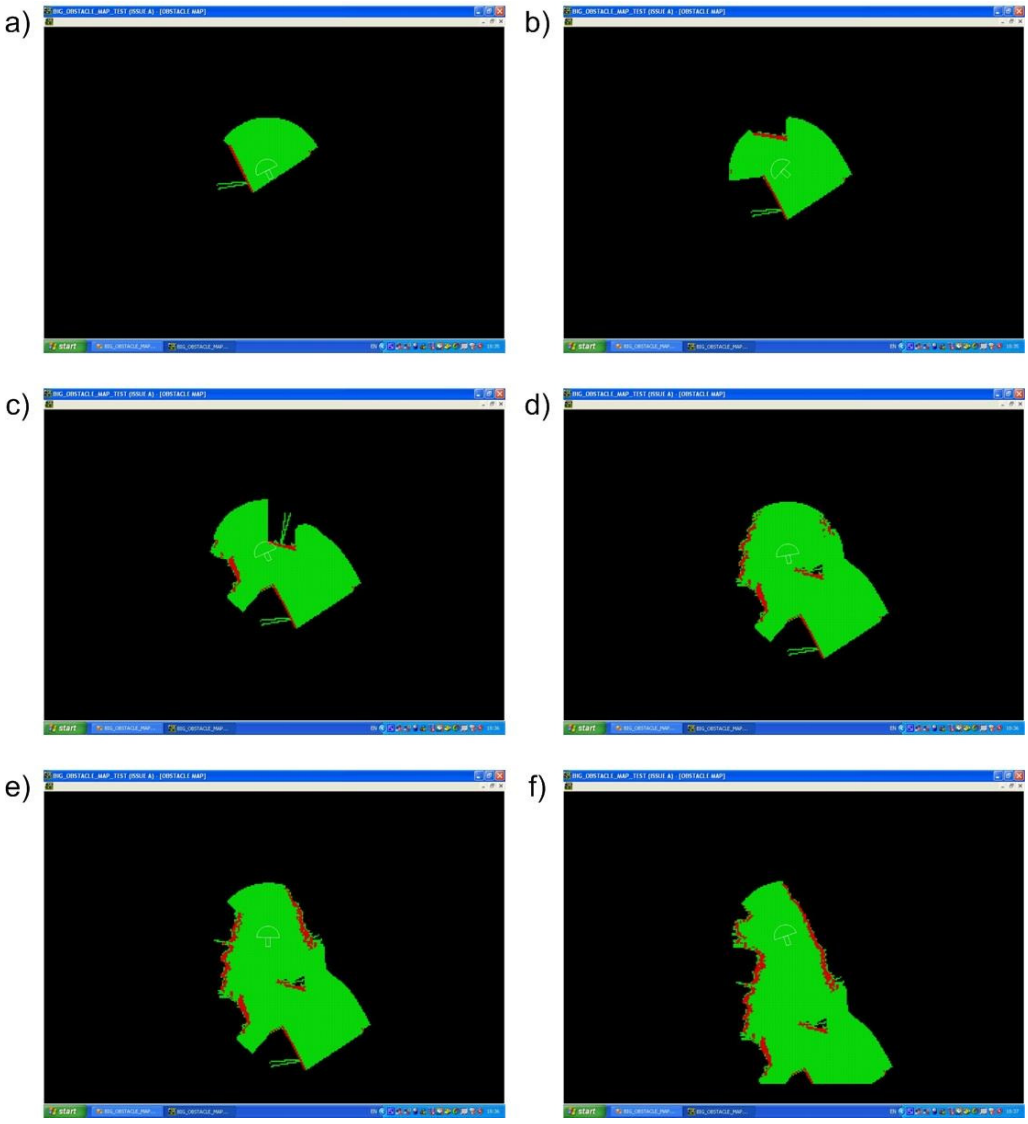

Figure 8.18 - Example of Mission Mapping

It can be seen that the UGV starts the example run next to a small building. After a few seconds it encounters a wall built from wooden pallets that was erected in the middle of the area. Then the UGV continues to drive forwards across the grass flanked on either side by bushes. It is clear to see that larger map gives much more information about the environment the UGV is driving through than the standard obstacle map.

During the experiment the speed of the obstacle mapping system was not noticeably slower. This implies that even bigger obstacle maps could be maintained using this system. Although

perhaps a better idea would be to store a larger map as a sequence of smaller tiles. This method has the potential to make processes of updating and scrolling the larger map much more efficient.

8.4 3D OBSTACLE MAPPING EXPERIMENTS

The experiments described in this section analyse the performance of the 3D obstacle mapping system. Initial experiments evaluated the PMD [vision] ® 1k-S range camera and highlight a problem that was encountered. This is then followed by a description of 3D mapping experiments using the Sick LMS 200 laser scanner. In later experiments the need for UGV pose compensation in the 3D mapping system is demonstrated. Experiments are then detailed to show the performance of the system when using pose compensation.

8.4.1 PMD [vision]® 1k-S Range Camera

Experiments with the PMD [vision]® 1k-S did not progress far before a major problem was encountered. Preliminary indoor tests with the camera showed that it could produce good albeit low resolution 3D images as shown in Chapter 5. Only when the camera was fully integrated with the ARP vehicle was the problem noticed.

Figure 8.19 shows a digital photograph of an indoor scene and corresponding 3D images from the PMD [vision]® 1k-S range camera. The two lower images were generated using CamVis Pro 2.0 a visualisation program provided with the range camera. Comparing the lower left image with the digital photo it is possible to make out the door frame shown by the blue pixels and even some of the objects to the left of the door.

This 3D image is not correct. In the lower right image the same scene is shown but has been rotated to give a different viewpoint. It can be seen that the pixels that represent the floor and windows all have approximately the same depth. The floor pixels should have a range of

depths stretching from the UGV up to the door. The window pixels should also have the same depth as the door frame.

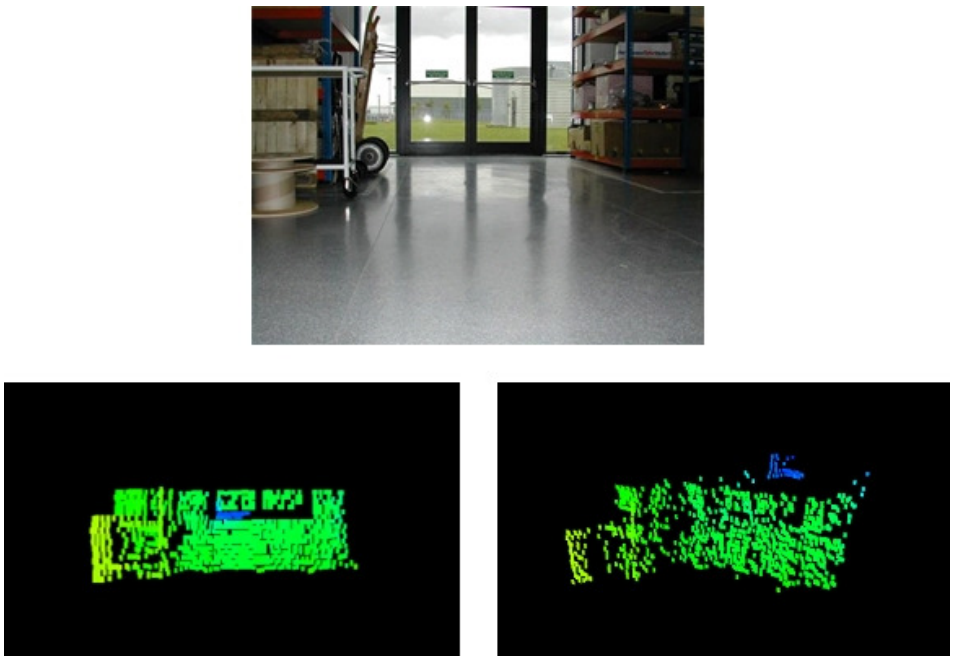

Figure 8.19 - Performance of PMD [vision]® 1k-S Indoors

Compare this with the image from the PMD [vision]® 1k-S in Chapter 5 that was taken in a place with low ambient light. In Figure 5.13 the perspective effect can be clearly seen in the floor pixels. It was initially thought the problem highlighted by Figure 8.19 was caused by reflections from the shiny floor surface. Hence another experiment was performed outdoors on a grassy surface.

Figure 8.20 shows a digital photograph of the outdoor scene and corresponding images from the PMD [vision]® 1k-S. As before the two lower images show the scene in 3D from two different viewpoints. The 3D images resemble a phantom wall approximately 4m from the camera with some random noise. During this experiment the sky was overcast and the range camera was angled down. So the camera was not exposed to direct sunlight.

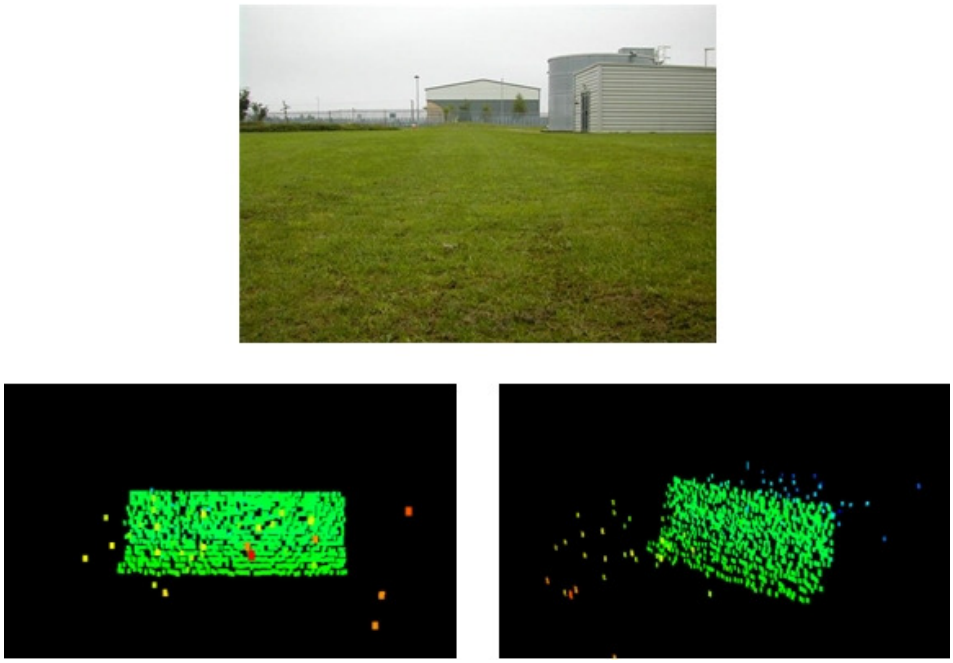

Figure 8.20 - Performance of PMD [vision]® 1k-S Outdoors

It seemed that the ambient light level was significantly affecting the range perception of the camera. Most likely the image sensor pixels were becoming saturated. Final confirmation of this was given by covering the active illumination sources on the camera. Even though the camera was not receiving any of the modulated light from the active illumination the 3D image of the scene remained unchanged. Typically in an area with low ambient light this test would result in the camera seeing nothing.

Literature from PMD Technologies GmbH states that one of the key features of the range camera is the active suppression of background intensities for each pixel [PMD 2005]. PMD sensors see only their own active illumination and are not affected by ambient light including sunlight. For this reason they are especially suited to outdoor applications. This did not seem to be the case.

In an attempt to reduce the amount of ambient light getting to the image sensor filters were fitted in front of the lens. The aim was to make a band pass filter centred on the 870nm

frequency of the active illumination source. In theory this could greatly reduce the sensitivity of range camera to sunlight. Some improvement was seen using the filters but to achieve the required performance from the camera a much higher quality band pass filter was required.

PMD Technologies have now replaced the PMD [vision]® 1k-S with an upgraded range camera. Unfortunately the replacement camera did not arrive in time to be tested for this research. However PMD Technologies have recently demonstrated use of a range camera outdoors to track the motion of vehicles on a road from a moving car.

8.4.2 Mapping without Pose Compensation

Work on the 3D obstacle mapping system continued despite the problems encountered with the PMD range camera. The remainder of the mapping experiments used the Sick LMS 200 laser scanner in a push broom configuration and the corresponding sensor model. It should be noted that the obstacle mapping system is not sensor specific. Thus many of the experiences reported in this section would apply equally if a working version of the range camera was being used in place of the laser scanner.

Initially all evaluation of the 3D obstacle mapping system was performed on the flat tarmac roadways around the main test area. This simplified the test process as the pose of the UGV did not have to be considered. In the sensor model the pose of the UGV was ignored when transforming range points from the sensor coordinate frame to the coordinate frame for the obstacle map.

The first major experiment with the 3D obstacle mapping system was to evaluate the statistic used to determine if voxels on the obstacle map were traversable. In Chapter 7 two subtly different traversability statistics were proposed. The first of these considered the maximum difference in height between the range points currently recorded in each voxel. As such it

offered a type of short term memory for the voxel that should be able to respond in a dynamic environment.

When testing the first traversability statistic it was found that the voxel memory length needed to be tuned to match the speed of UGV. Using the initial memory length genuine obstacles kept disappearing as voxels discarded older range points. During these early tests the UGV was deliberately limited to a low speed. This limited speed was measured at $0.27ms^{-1}$ which is roughly half of the maximum speed.

Given that the laser scanner was tilted at 11° below the horizontal the intersection of the scanning plane with the ground was around 2.2m in front of UGV. The scanner has an angular resolution of 0.5° which implies an approximate 2cm lateral separation between range points on flat ground at the centre of the scanner field of view. Thus up to 14 range points from each scan may fall into a 20cm wide voxel if the UGV was driving diagonally across the obstacle map. With the laser scanner sweeping at a rate of 37.5Hz theoretically as many as 550 range points could fall into a voxel as the UGV drove past.

Various memory lengths were trialled with the voxel map. In each case a threshold of 100mm was used with the traversability statistic to determine if a voxel contained an obstacle. When the memory length was under 250 range points the voxel map did not show all obstacles and individual voxels would often turn from untraversable to traversable.

In a final run of this experiment up to 500 range points were allowed in the memory of each voxel. A snapshot of the obstacle map traversability summary from this run demonstrated that voxels at centre of sensor field of view did have the highest number of range points. In fact voxels with over 300 range points were common. To the sides of the UGV path the number of range points recorded was much lower.

Next the second traversability statistic proposed in Chapter 7 was evaluated. This statistic considered the maximum difference in height between all range points that had ever been observed in a voxel. Hence the second traversability statistic is equivalent to maintaining a full history of the range points from each voxel but without the overhead of storing the associated data. Using this traversability statistic the performance of the mapping system was equivalent to using the first statistic and a memory length of at least 250. For the remaining experiments with the 3D obstacle mapping system the full history traversability statistic was used.

Figure 8.21 is a photograph of the tarmac test area used for the traversability statistic experiments. The UGV was typically driven autonomously along the roadway parallel to the kerb. Both the sign post in the centre of the photograph and bushes to the left were within the range of the laser scanner during the each run of the experiment.

Figure 8.21 - Flat Tarmac Test Area

A snapshot of the corresponding obstacle map generated by the 3D mapping system is shown in Figure 8.22. The kerb at the side of the UGV is clear to see as are the bushes at the bottom right of the map. At the time the snapshot was taken the UGV was alongside the signpost. This is represented by the small clump of untraversable voxels next to the kerb. In repeated

runs of this experiment a small number of isolated untraversable voxels would also be seen on the grass verge. It is likely that these were a product of the laser striking tall blades of grass.

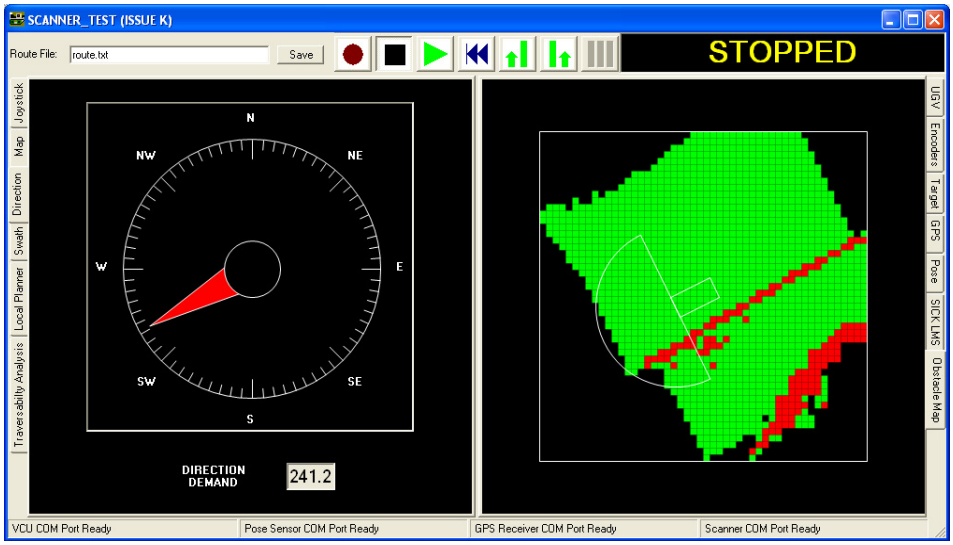

Figure 8.22 - Obstacle Map from Flat Tarmac Area

To get a better idea of what the UGV was sensing as it drove along another experiment was performed. This time all of the range points added to the obstacle map were recorded and plotted in 3D space. Due to memory constraints only 10 second blocks of the visualisation were plotted as they contained in excess of 135000 range points.

Figure 8.23 shows a plot generated by driving the ARP vehicle through the tarmac test area originally pictured in Figure 8.21. The features from the digital photograph are clearly reproduced in the model. To the far left of the plot the texture of the bushes can be seen. To the right of the bushes is the grass verge which itself has some discernable texture. The kerb is shown as a large square edged step in the plot and finally to the right of the kerb is the smooth tarmac surface.

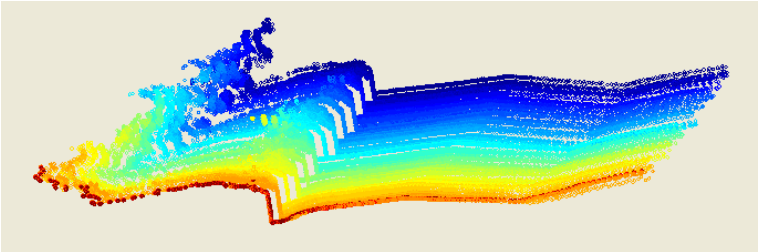

Figure 8.23 - 3D Model of Flat Tarmac Area

Once it was established that the 3D obstacle mapping system was working correctly on the flat tarmac the UGV was moved onto a grassy test area. This new test area had a relatively flat surface with a few small undulations and holes. In this area the mapping system was evaluated to see how it coped with the undulations. The purpose of the evaluation was to see if the 3D obstacle mapping system was robust enough to allow the UGV to roam around the full extent of the main test area.

For this experiment a start point was marked on the grass and the UGV was driven autonomously on a fixed heading from that point. Figure 8.24 is a digital photograph of the grassy test area taken from the start point. A snapshot of the obstacle map a few seconds into a run is also shown in Figure 8.25. It can be seen that the obstacle map has several bands of phantom untraversable voxels. Repeated runs of this experiment with the UGV following different headings produced similar results.

Bands of untraversable voxels appeared each time the ARP vehicle drove over an undulation in the ground. When the undulations pitched the UGV forwards or back by a few degrees a band of untraversable voxels stretching across the whole map would appear. This is what caused the upper of the untraversable region in Figure 8.25. In contrast if an undulation tilted the UGV to the left or right a band of untraversable voxels would appear on the corresponding side of the UGV. This is the cause of the other untraversable region in Figure 8.25.

Figure 8.24 - Grassy Test Area

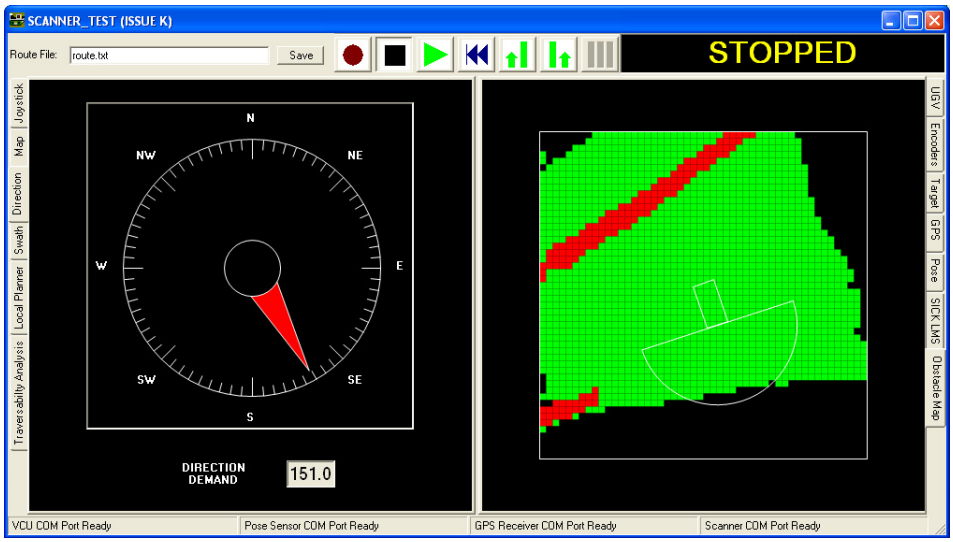

Figure 8.25 - Obstacle Map from Grassy Test Area

Figure 8.26 shows a 3D plot from a run corresponding to the one that generated the obstacle map in Figure 8.25. Looking at scan lines in the plot that are coloured yellow, green and blue it is possible to see a hump in the range points especially on the right hand side. The hump has a height well over the 100mm obstacle detection threshold. This hump would correspond to the upper untraversable region shown in Figure 8.25.

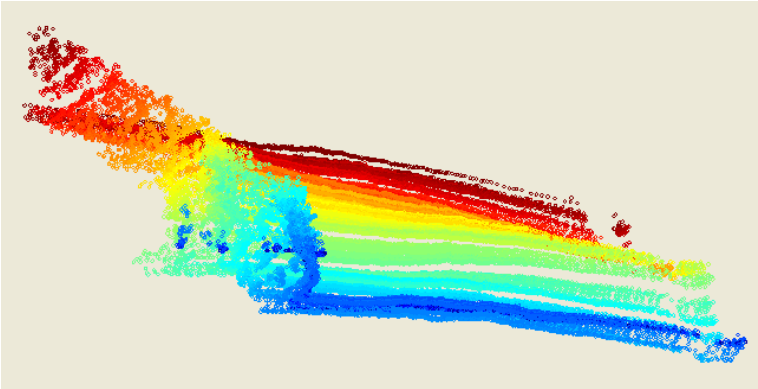

Figure 8.26 - 3D Model of Grassy Test Area

The hump shown in Figure 8.26 is the product of the UGV driving over a small undulation. Since the range points are being generated several metres in front of the UGV the effect of the undulation is amplified. It is estimated that a change in the UGV pitch as small as 3° could cause phantom untraversable voxels to appear on the obstacle map. Bigger pitch changes would be likely to cause wider bands of untraversable voxels.

If the UGV pose was correctly compensated for as new range points were added to the obstacle map then all of the phantom untraversable voxels should disappear. These experiments demonstrate that unless the UGV is operating on a smooth flat surface it is essential to use UGV pose compensation in the sensor model. The next subsection describes a similar set of experiments that use the complete sensor model and so attempt to compensate for the UGV pose when adding range points to the obstacle map.

8.4.3 Mapping with Pose Compensation

After the need for UGV pose compensation was demonstrated the UGV was returned to the flat tarmac test area. The 3D obstacle mapping system was then evaluated using the sensor model complete with UGV pose compensation. In this model pitch and roll measurements from the UGV pose sensor were used as input for the compensation process.

Figure 8.27 shows the obstacle map generated by the first run using UGV pose compensation. The map was covered with phantom untraversable voxels. Initially it was thought that the UGV pose compensation process contained an error but this was found not to be the case. In a subsequent run a 3D plot of the tarmac test area was generated from the range data. This plot is shown in Figure 8.28.

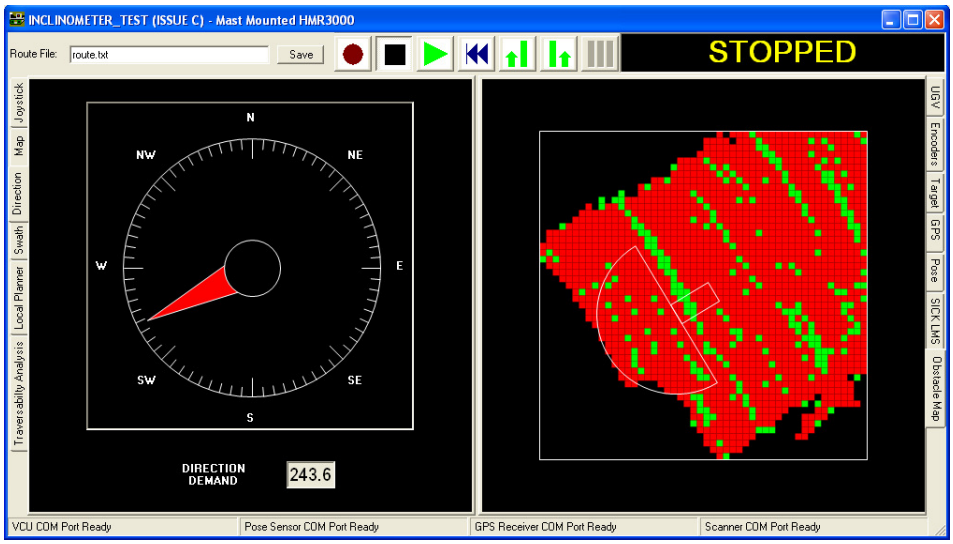

Figure 8.27 - Flat Tarmac Area Mapped with Pose Data from HMR3000

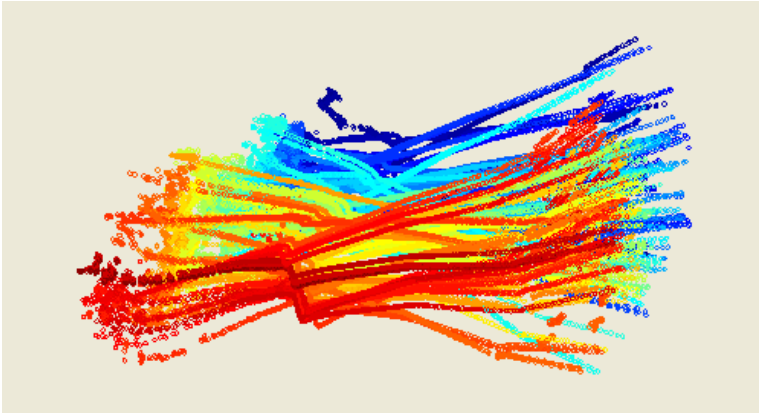

Figure 8.28 - 3D Model of Flat Tarmac with Pose Data from HMR3000

Compare Figure 8.28 with Figure 8.23 the corresponding plot that was generated without UGV pose compensation. Individual scan lines look to be an appropriate shape. The texture of the

bushes and the step of the kerb can be made out. It is just that the scan lines are all at different angles. This pointed to a problem with the pose sensor data.

Pitch and roll measurements from the Honeywell HMR3000 pose sensor were recorded as the UGV drove along flat tarmac. Figure 8.29 and Figure 8.30 show these recorded pose measurements. Even though the ARP vehicle was driven along smooth tarmac the pitch and roll measurements were observed to swing wildly. It can be seen that the range of the pitch and roll oscillations was greater than ±30°.

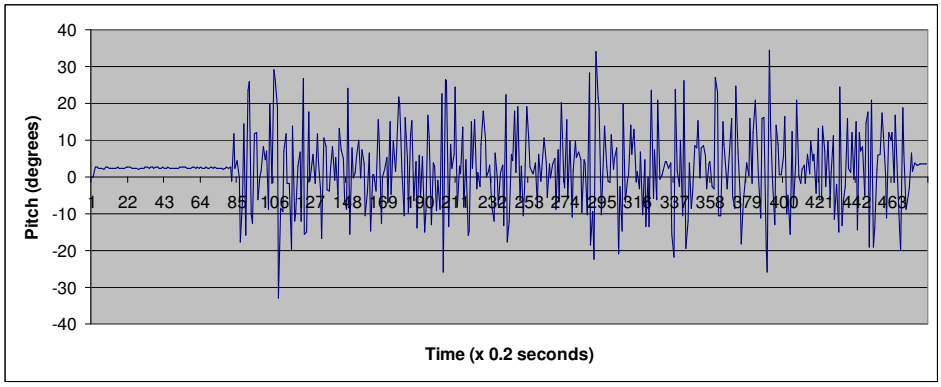

Figure 8.29 - UGV Pitch data from HMR3000

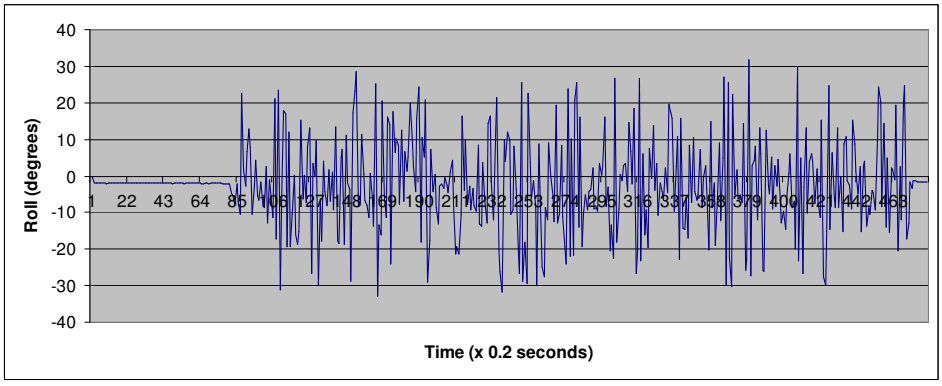

Figure 8.30 - UGV Roll data from HMR3000

These oscillations in the pitch and roll measurements are attributed to vibration. Since the ARP vehicle has tracks it does vibrate as the weight of the UGV shifts from spud to spud on the tracks. There were two options to overcome this problem. A different pose sensor could be used. Alternatively pose measurements could be smoothed in a similar way to the heading measurement smoothing described in section 8.3.1.

Engineers at Remotec UK recommended the VTI Technologies SCA121T inclinometer as an alternative pose sensor that they had experience in using. On this basis a new inclinometer was procured and integrated with the ARP vehicle. Figure 8.31 shows the inclinometer mounted on the chassis of the ARP vehicle.

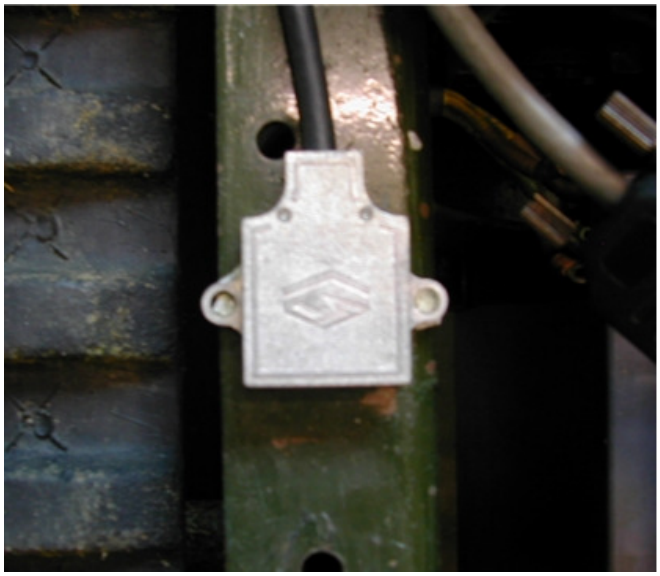

Figure 8.31 - VTI Technologies SCA121T Inclinometer

Pose measurements from the new inclinometer were recorded as the UGV was driven around on flat tarmac. The recorded measurements are shown in Figure 8.32 and Figure 8.33. It can be clearly seen that the noise associated with this sensor is a factor of 5 smaller than the Honeywell HMR3000.

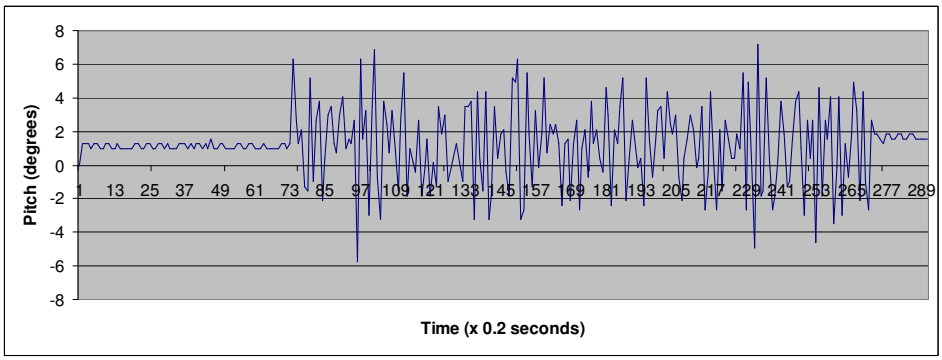

Figure 8.32 - UGV Pitch data from Chassis Mounted SCA121T

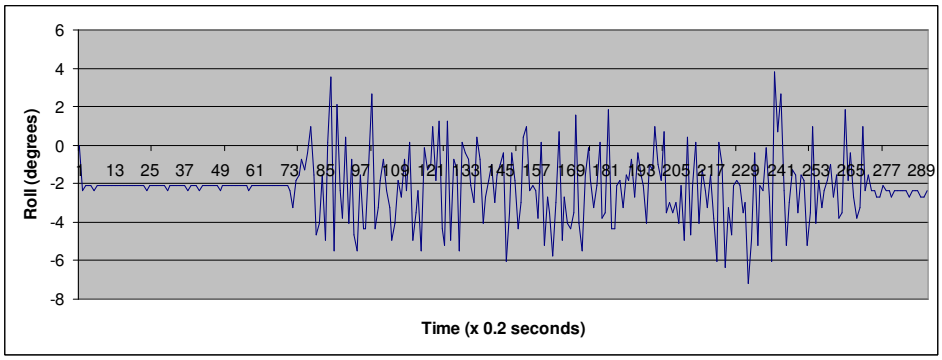

Figure 8.33 - UGV Roll from Chassis Mounted SCA121T

The 3D obstacle mapping system was then tested using the pose measurements from the new inclinometer in place of those from the Honeywell HMR3000. Figure 8.34 shows a snapshot of the obstacle map that was generated driving along the flat tarmac test area shown in Figure 8.21.

Compare Figure 8.34 with the earlier Figure 8.22 which shows a map of the same ground generated without UGV pose compensation. The noise in the pose measurements from the new inclinometer was still causing small regions of phantom untraversable voxels. So a method to smooth the noise in the pose measurements was also required.

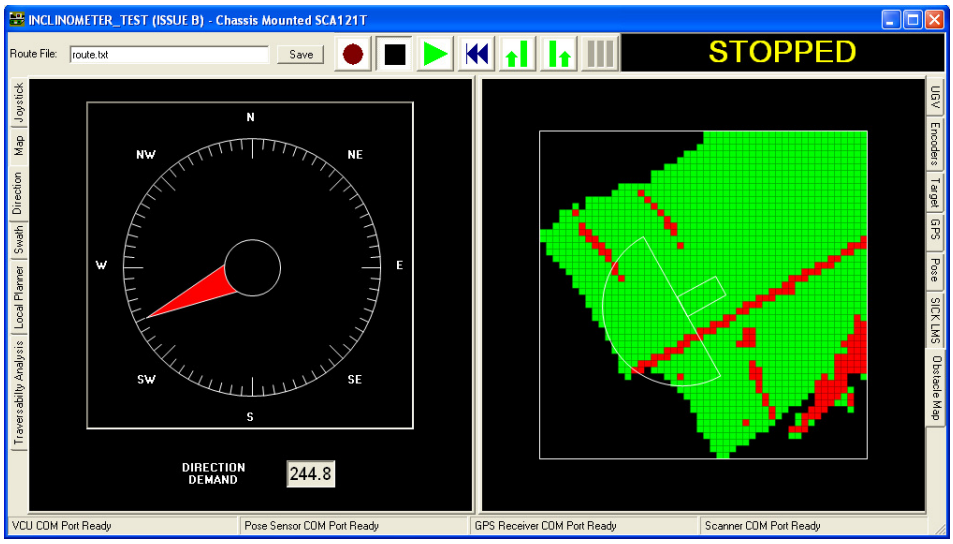

Figure 8.34 - Flat Tarmac Area Mapped with Pose Data from SCA121T

8.4.4 Smoothing Pose Measurements

Initially single pole recursive filters [Smith 1997] were implemented to smooth the UGV pitch and roll measurements. A description of this type of filter is given in section 8.3.1. To set up the filters the UGV was repeatedly run alongside the kerb in the flat tarmac test area. On each run the gain of the pitch and roll smoothing filters was fine tuned until the phantom untraversable voxels were banished from the obstacle map. A filter gain of 0.08 was found to work best for both the pitch and roll filters.

The effect of the smoothing process on the pose measurements was captured by recording both the raw and smoothed data during a run. Figure 8.35 and Figure 8.36 show the recorded raw pose measurements with the corresponding smoothed measurements. It can be seen that the smoothing process reduces the range of the pose measurements to around ±1° of the true value.

For the next experiment the UGV was returned to the grassy test area that was pictured in Figure 8.24. As before the UGV was set to drive autonomously on a fixed heading across the grass and again the performance of the 3D obstacle mapping system was evaluated. It was

hoped that the maximum acceptable filter gain found driving the UGV on tarmac would also work when driving the UGV on grass.

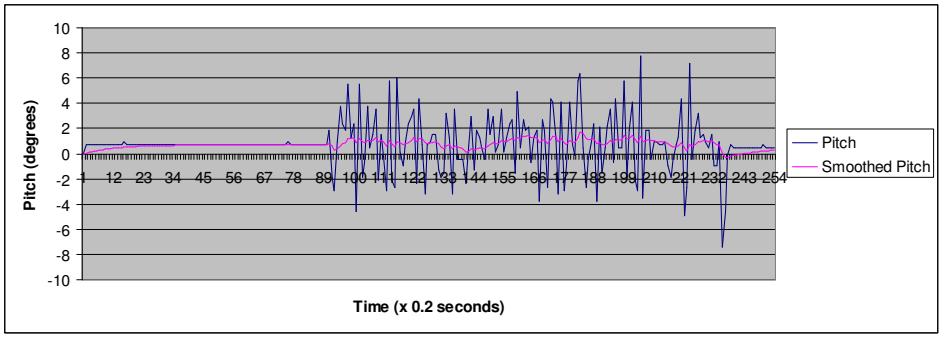

Figure 8.35 - Pitch Data Smoothed using Single Pole Filter with 0.08 Gain on Tarmac

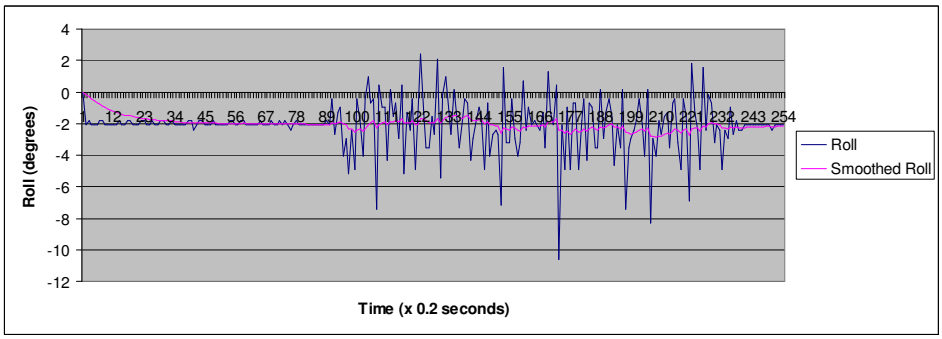

Figure 8.36 - Roll Data Smoothed using Single Pole Filter with 0.08 Gain on Tarmac

Unfortunately the obstacle maps generated were not much better than that shown in Figure 8.25. Phantom bands of untraversable voxels still appeared as the UGV drove over undulations in the ground. Once again the raw and smoothed pose measurements were recorded to see if a cause could be found. Figure 8.37 and Figure 8.38 show the pose measurements recorded as the UGV drove across the grassy test area.

The two plots show that over the undulating grass the smoothing filters are doing a fairly good job of following the general trends in the pose measurements. Closer inspection shows that in

a few places the smoothing process may have been thrown by sensor noise. For example in the plot of UGV pitch measurements there was a general dip soon after the UGV began to move. Noise around this dip causes the smoothing filter to report a pitch that is perhaps 3° shallower than the true value.

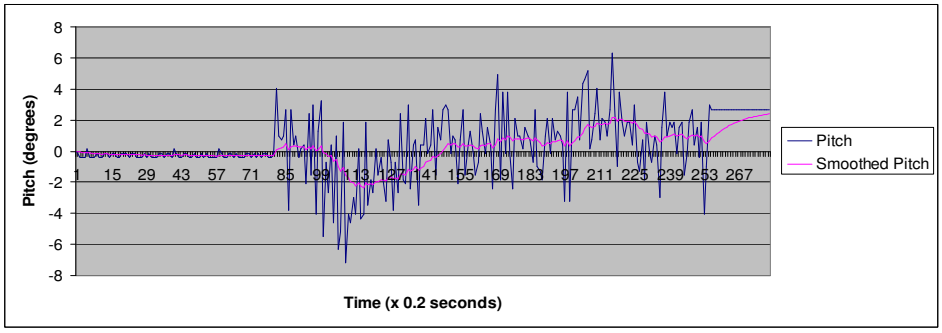

Figure 8.37 - Pitch Data Smoothed using Single Pole Filter with 0.08 Gain on Grass

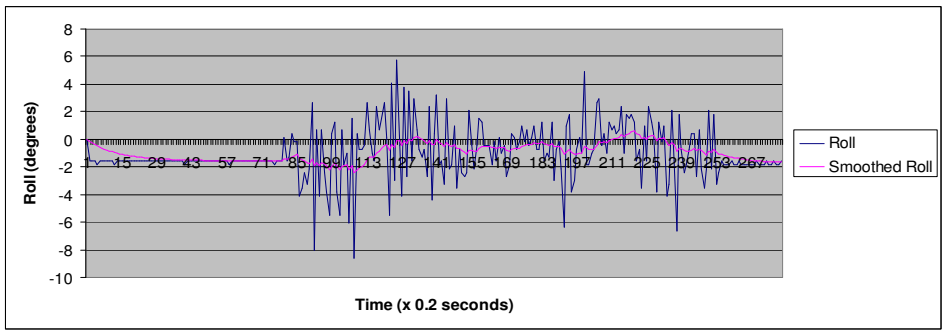

Figure 8.38 - Roll Data Smoothed using Single Pole Filter with 0.08 Gain on Grass

In an attempt to improve the performance of the 3D obstacle mapping system on the grassy test area a variety of different filter gain values were tried. Other types of smoothing were also implemented and tested. A series of experiments were conducted using a 2-Pole Cheyshev filter [Smith 1997]. The Cheyshev filter was specifically designed to remove the high frequency noise component from the genuine pose measurement. None of the other filters produced significantly better results than the basic single pole smoothing filter.

Ultimately to show the full potential of the 3D obstacle mapping system a UGV pose sensor with a better signal to noise ratio is required. The VTI Technologies SCA121T is much better than the Honeywell HMR3000 but still not good enough. There are definitely inertial navigation systems that could give the required performance but at a cost beyond the budget of this research.

8.5 LOCAL PATH PLANNER EXPERIMENTS

In this section a series of experiments are detailed that test the local path planner within NOVA. The first experiment in the series demonstrates that the basic local planner works as intended. Several issues with the basic planner are then highlighted and solutions that overcome these issues are also shown.

Additional experiments then focus on extensions that were developed for the basic path planner. First the effects of changing the lookahead of the basic path planner are investigated. Then a demonstration of the adjustable planner lookahead concept is given. Alongside this the effect of the path planner claustrophobia setting is illustrated.

8.5.1 Basic Local Path Planner

During the experimental phase of this research the local path planner was tested extensively. It was used to control the ARP vehicle for many kilometres of autonomous operation around the test site. The path planner was tested with both the 2D and 3D obstacle mapping systems. Although tests with the 3D system were limited due to the issues with UGV pose compensation discussed in the last section.

Figure 8.39 illustrates the basic path planner in operation. For this experiment a series of wooden pallets were erected in the main test area to serve as obstacles to be avoided. The ARP

vehicle was set up to drive on a fixed heading towards these pallets. Between adjacent pallets the spacing was approximately 5m.

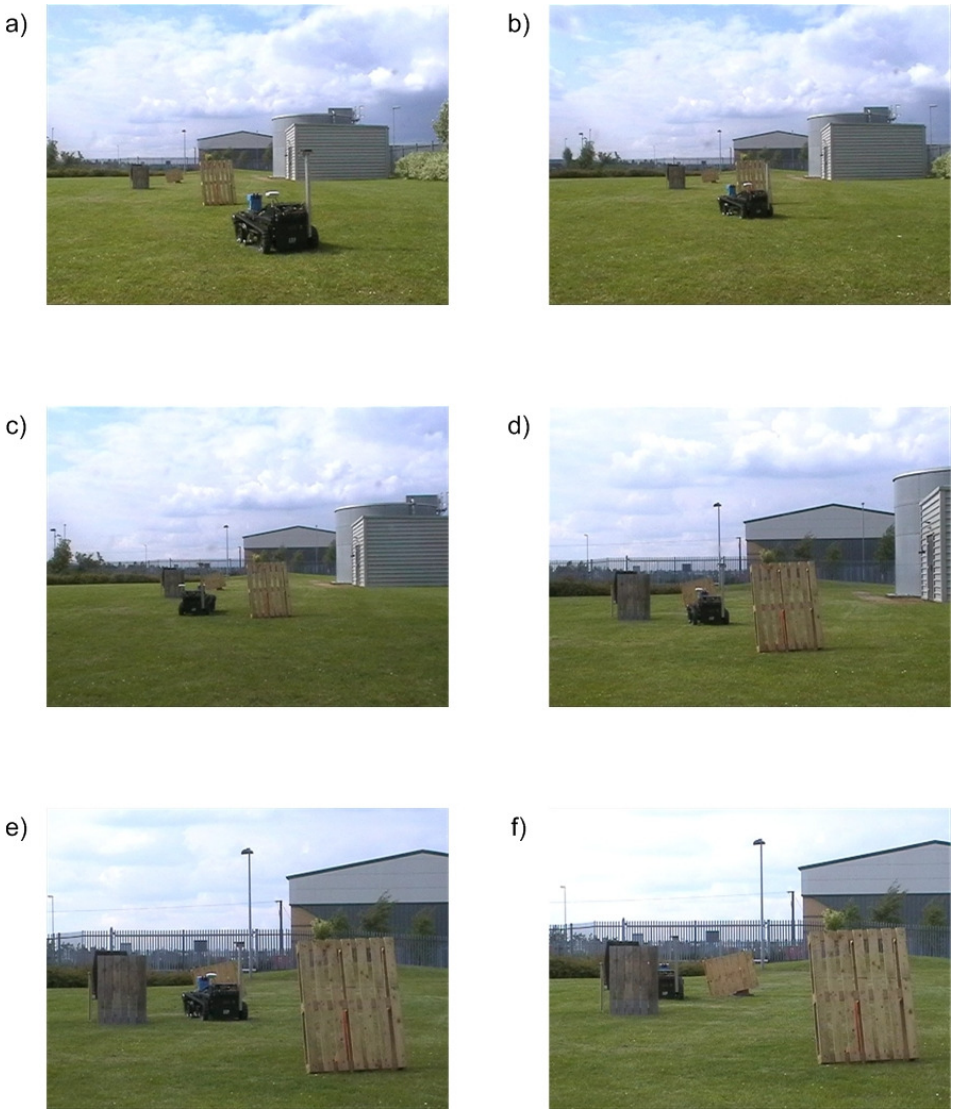

Figure 8.39 - Example of Obstacle Avoidance

Figure 8.39 shows frames from a digital video camera that was used to record the experiment. Frame (a) captures the set up at the start of the experiment. In frame (b) the local path planner has turned the UGV to the left to avoid the first pallet. Frame (c) shows that the UGV successfully avoided the first pallet and was in the process of turning to avoid a second pallet. By frame (d) the local path planner has guided the UGV away from the second pallet and the

UGV is progressing forward. In frame (e) the local path planner is turning the UGV to avoid the third pallet. The final frame shows the UGV having avoided the third pallet and continuing on the desired heading.

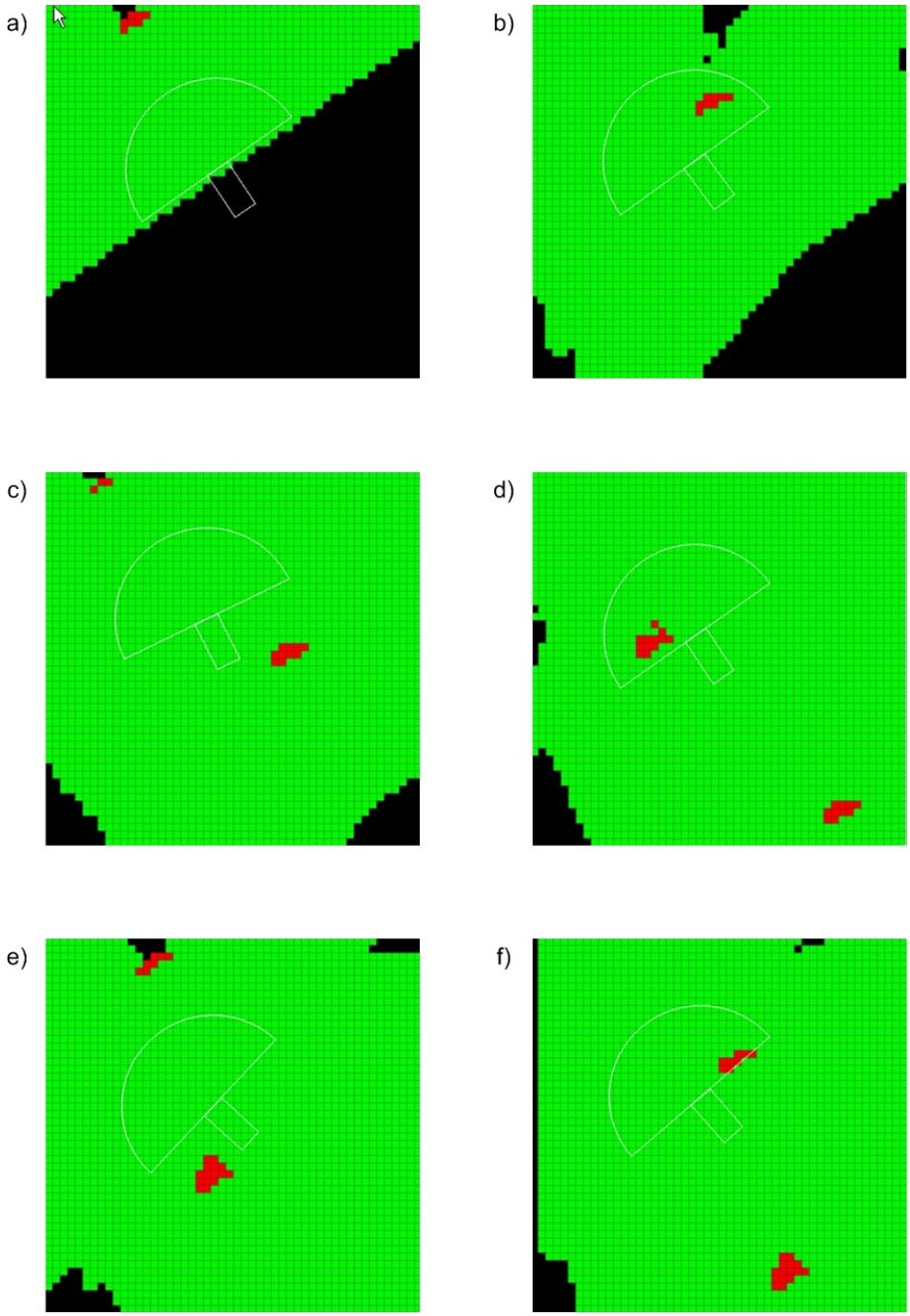

Figure 8.40 - 2D Confidence Mapping for Obstacle Avoidance

This experiment used the 2D obstacle mapping system to detect and map the obstacle course. Figure 8.40 shows a time ordered sequence of snapshots from the obstacle map that were generated during a run though this obstacle course. These snapshots roughly correspond to the video frames in Figure 8.39.

The snapshots from the obstacle map show that the UGV correctly detected each of the pallets and the objects that were supporting them. They also demonstrate that the spacing between the pallet obstacles was about half the width of the obstacle map. So as the UGV completed the avoidance manoeuvre around one pallet the path planner would begin to avoid the next.

In Chapter 7 it was stated that the behaviour of the local path planner is based on a Polar Obstacle Distance Histogram that is generated from the obstacle map. A radial swath search process is used to find the distance the UGV can safely move in each direction across the obstacle map from the current location. The polar histogram then collates these distance values. Figure 8.41 illustrates the changes to the polar obstacle distance histogram used by the local path planner as the UGV avoids a pallet in the obstacle course.

Figure 8.41(a) shows the polar histogram created by the local path planner before the pallet is detected. No obstacles have been seen so the UGV can safely move in any direction in the ±90° range around the ideal heading. Snapshot (b) shows the polar histogram soon after the pallet first appears on the obstacle map. The red bars on the histogram indicate that these directions are blocked by the pallet. Distance to the pallet is indicated by the length of the red bars.

Due to the presence of the pallet the path planner had to take avoiding action. It can be seen that the target heading value for the UGV decreases by 15° between snapshots (a) and (b). This indicates that the path planner selected a clear swath to allow it to pass to the left of the

pallet. The UGV then turned to the target heading and drove along the clear swath. It can be seen from snapshot (b) that the radial swath with angle -15° does represent the minimum turning angle to avoid collision with the pallet.

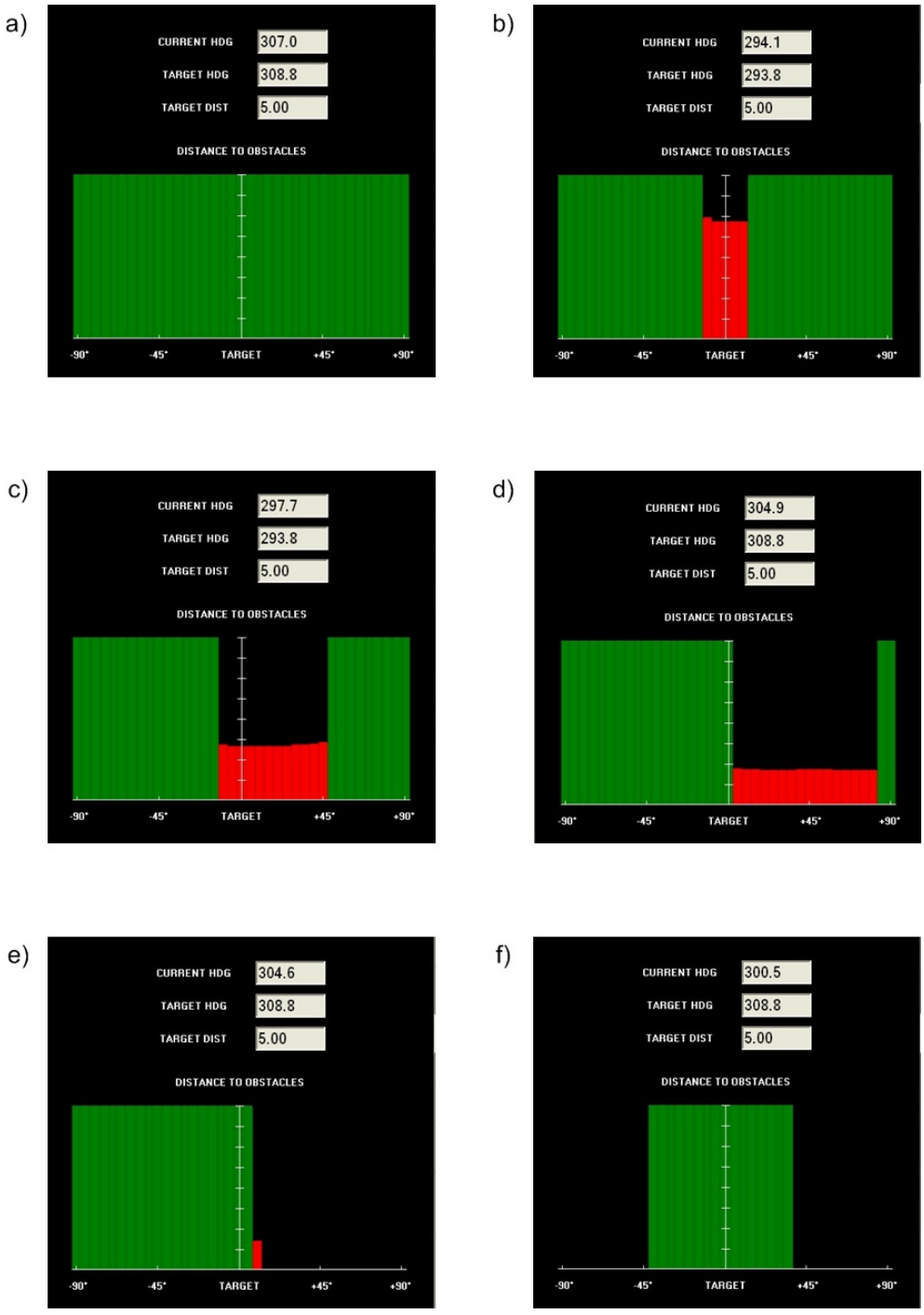

Figure 8.41 - Polar Obstacle Distance Histogram for Obstacle Avoidance

Figure 8.41(c) shows how the polar histogram had changed after the UGV had driven around 3m along the clear swath. It can be seen that the red bars are shorter as the UGV was closer to the pallet although still on course to safely avoid collision. Also as a consequence of the UGV being closer to the pallet the valley of red bars in the histogram is much wider. This implies that more extreme avoidance behaviour is generally required the closer the UGV is to an obstacle. Finally it should be noted that the centre of the histogram valley has shifted to the right confirming that the UGV was passing to the left of the pallet.

Eventually by following the clear swath the UGV passes the edge of the pallet. This moment is captured in Figure 8.41(d). At this point the path planner can see that the ideal heading is once again clear so the path planner selects this path. This is confirmed by the fact that the target heading value in snapshot (d) is the same as that for snapshot (a) before the pallet had been detected.

The final two parts of Figure 8.41 show what happens to the polar histogram as the UGV passes by the pallet. First the valley of red bars in the histogram expands to include all of the swaths on the right side of the histogram. Then the majority of these red bars disappear as has happened in snapshot (e). This is because the length of the radial swaths that lead towards the pallet drops below the length of the UGV and so the swaths are considered completely blocked.

Once the UGV is alongside the pallet radial swaths that lead off at extreme angles to the left as well as the right are considered blocked. Figure 8.41(f) illustrates this phenomenon. This happens since the UGV turns on the spot. Thus if a large angle neutral turn was attempted the close proximity of the pallet means that the UGV would collide with the pallet regardless of the turn direction.

The experiment detailed above demonstrates that the competence of the local path planner to guide the ARP vehicle around obstacles. However during extensive trials with the path planner a few issues were observed and overcome. These issues and the associated solutions are the topic of the next few subsections.

The first issue was that the UGV would occasionally stop at the side of an obstacle and declare that it was completely blocked. It was also noticed that the heading selected by the local path planner could oscillate. This would lead to the UGV turning one way to avoid an obstacle then turning back the other way a few seconds later. Generally this type of heading oscillation happened very infrequently but on several occasions the path planner got stuck continuously turning left then right.

The final issue was expected when the planner was first designed. Due to the long lookahead of the basic path planner the UGV would struggle to navigate through more complicated arrangements of obstacles. This confirmed the need to investigate the effects of changing the planner lookahead as was proposed in Chapter 7.

8.5.2 Radial Swath Shape and Angle

The first problem with the basic local path planner was related to the shape of the swaths used when searching for clear paths across the obstacle map. In Chapter 7 these swaths were described as extending from the back of the UGV forward across the obstacle map. To ensure that the UGV does not get dangerously close to obstacles all swaths include a safety margin. This safety margin describes a small amount of empty space around the perimeter of the swath.

Figure 8.42(a) illustrates what a clear swath that avoids an obstacle might look like. In the diagram the safety margin around the UGV is coloured red. If for some reason the UGV turns slightly towards the obstacle while it is following the clear swath the UGV would stop and turn

to a larger avoiding angle. The problem occurred if an obstacle was detected within the length of the UGV.

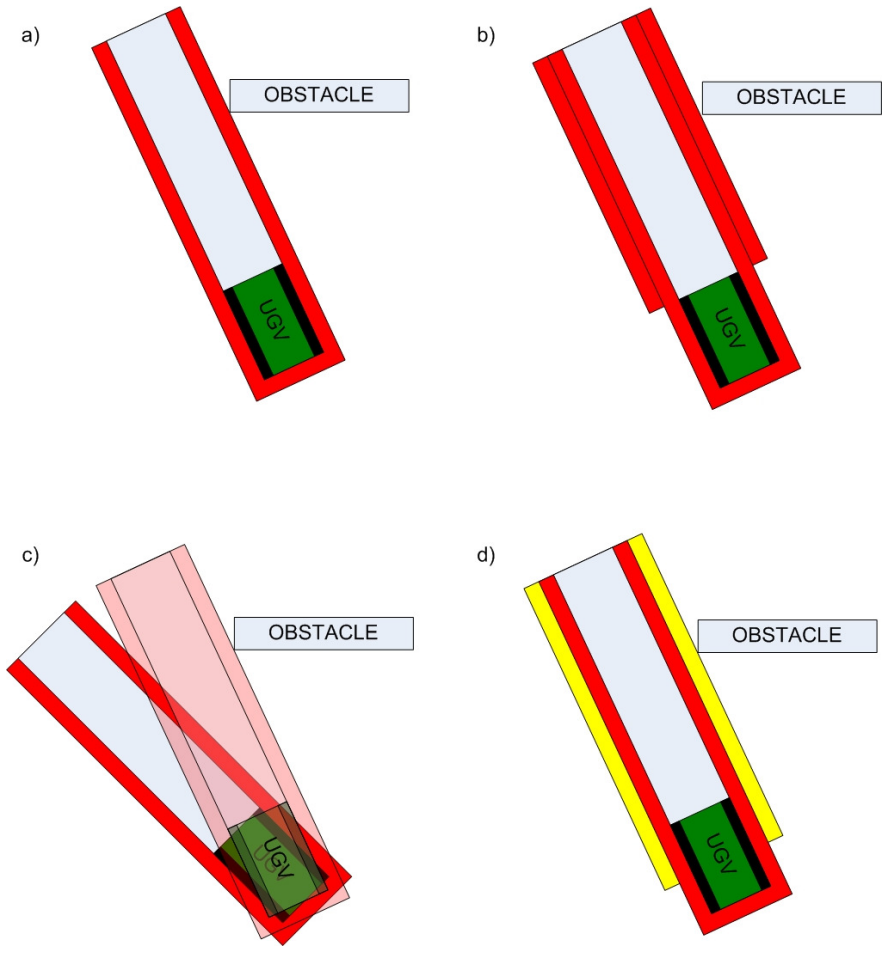

Figure 8.42 - Different Swath Shapes for the Local Path Planner

The basic path planner deliberately selects a swath that avoids obstacles with the minimum angular deviation. Thus typically the selected swath will brush one side of an obstacle. Each time the UGV stopped next to an obstacle it was found that the obstacle was inside the safety margin within the length of the UGV. So the UGV heading must have deviated slightly from that of the selected swath as the UGV approached the obstacle. Having an obstacle this close to the turning centre of the UGV meant that path planner could not find a radial swath that avoided the obstacle.

A number of different techniques were tested to see if this problem could be overcome. First a double width safety margin in front of the UGV was tried. Figure 8.42(b) shows the idea. Next a method was implemented that only allowed the path planner to select a clear swath if there was another clear swath between the obstacle and the selected swath. This is illustrated in Figure 8.42(c).

Both of these methods had the desired effect of forcing the UGV to take a wider path around obstacles. However both still fell foul of the same problem as the original swath shape. When the UGV straightened up after avoiding an obstacle it would sometimes end up with the obstacle inside the safety margin within the length of the UGV.

Finally a solution was found. It employs an extra wide safety margin in front of the UGV but this time it extends forward from the UGV centre of rotation. The longer outer safety margin prevents the UGV from straightening up too early after avoiding an obstacle. This idea is shown in Figure 8.42(d). In the latest version of NOVA this has been used in conjunction with the idea shown in Figure 8.42(c) for a belt and braces solution.

An experiment was also performed to look at the viability of forcing the UGV to select much wider angles when avoiding obstacles. The operation of the path planner was modified so it no longer selected the best swath. That is the clear radial swath with the smallest angular deviation. Instead the planner considered the peaks in the Polar Obstacle Distance Histogram. The planner would select the swath at the centre of the histogram peak that contained the best swath.

This modification made the behaviour of the path planner very similar to that reported for popular Vector Field Histogram (VFH) method [Borenstein 1991]. Whenever an isolated obstacle was encountered the path planner would turn the UGV by around 45° to avoid the

obstacle. Little benefit was seen from this behaviour so it was not adopted. However the experiment implies that all of the concepts developed in this research could be used with a VFH style path planner.

8.5.3 Planner Heading Hold

Next a series of experiments were conducted to find a solution to the problem of oscillation in the heading selected by the local path planner. The aim was to develop a method that forced the path planner to hold a selected heading for a number of planner cycles. However first it was necessary to isolate the true cause of the oscillation.

NOVA was expanded to record the path planner heading variables and allow them to be analyzed after each experiment. Specifically at each time step the heading of the UGV was recorded along with the heading of the next waypoint and the heading of the radial swath selected by the local path planner. The heading of the next waypoint is known as the ideal heading since it is the route the UGV will follow if there were no obstacles. The heading of the radial swath selected by the path planner is known as the sub-goal heading as it represents a sub-goal on route to the next waypoint.

Figure 8.43 shows the path planner heading variables that were recorded as the basic path planner guided the ARP vehicle around a single pallet obstacle. For this experiment the UGV was set up to autonomously follow an ideal heading that took the vehicle directly towards the centre of the obstacle. The plot shows what normally happens and is free of path planner oscillation.

When the path planner encounters the pallet the sub-goal heading changes and the clear swath with the smallest angular deviation is selected. This sub-goal heading is held until the UGV is clear of the pallet at which point it should return to the ideal heading. The UGV heading

should always follow the demands of the local path planner and turn to the sub-goal heading as quickly as possible. In practice it takes some time for the UGV to actually perform the turn so the UGV heading lags behind the sub-goal heading.

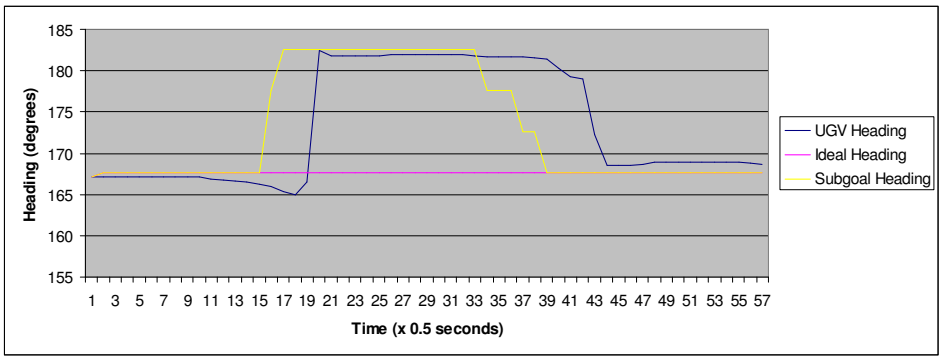

Figure 8.43 - Path Planner Heading Variables during Obstacle Avoidance

In a repeat run of the experiment a path planner oscillation was seen. Figure 8.44 shows the path planner heading variables recorded during the run. It is clear to see that the path planner initially chose a path to the right of the obstacle. Once the UGV had turned to the right and began to move the planner selected a path on the left of the obstacle. By the time the UGV had come to a standstill the planner again chose a path to the right of the obstacle. This brief oscillation made the UGV turn to the left then immediately back to the right.

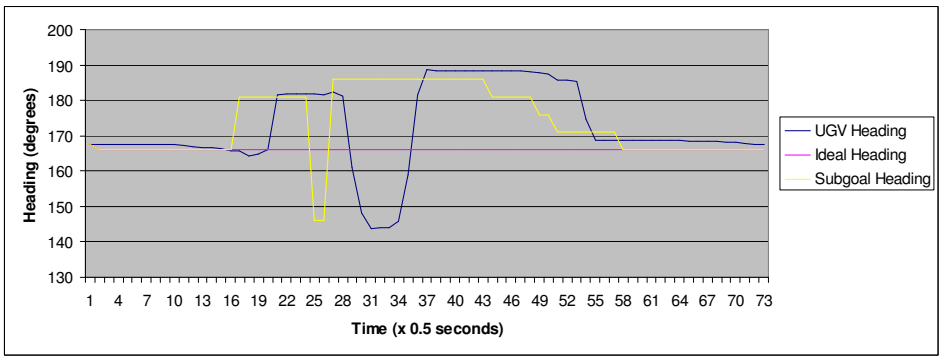

Figure 8.44 - Example Sub-goal Heading Oscillation

The oscillation was caused by the method the planner uses to select an obstacle free swath. Swaths are selected by the planner with the order of preference 0°, -5°, +5°, -10°, +10°, etc. Thus if the path planner selects the best swath then when the UGV begins to move it finds that the selected swath is blocked the next best swath will be selected. If the UGV was roughly in the centre of the obstacle when this happens it is likely that sub-goal heading selected by the planner would switch from one side of the obstacle to the other.

An acceptable solution was found to overcome the oscillation. Essentially the path planner tries to hold every selected heading for a period of time. If the planner finds that the selected heading is blocked during this period the path planner will not run as normal. Instead the planner checks to see if there is another clear swath adjacent to the originally selected swath.

During testing a deviation of ±10° from the originally selected swath heading was allowed. If a clear swath was found in this small angular range it would be selected by the path planner. Otherwise the path planner would run as normal and be allowed to select the best swath. Every time the path planner ran as normal the heading hold period would be reset.

Figure 8.45 demonstrates this idea. To generate this plot the UGV was once again set up to drive towards the pallet. This time the local path was configured to hold every selected heading for a period of 30 seconds. After a few seconds the UGV encounters and avoids the pallet. This requires a turn of 15° to the right. The planner is forced to run as normal to select this swath. A few seconds later the path planner finds that the selected heading is again blocked. This time a clear swath is found 5° further to the right. In this case the oscillation protection modification handles the update of the selected heading. Around 30 seconds after the initial obstacle avoidance manoeuvre the planner finally returns to a heading close to the ideal heading.

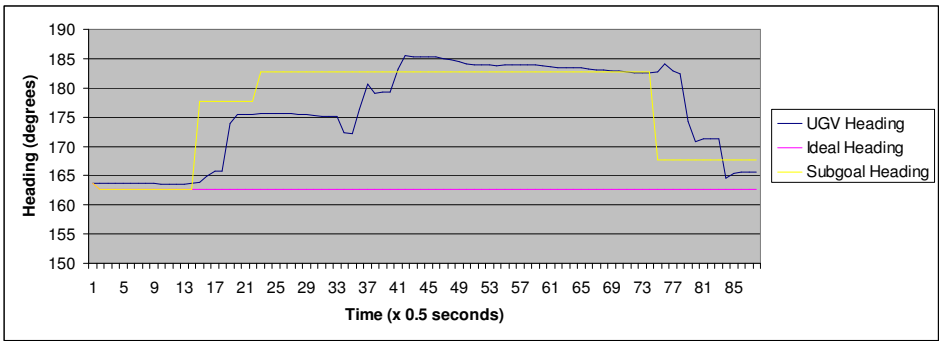

Figure 8.45 - Path Planner Heading Variables with Heading Hold

Figure 8.46 shows a plot of the path hold variable during the same experiment. This variable simply counts the number of path planner cycles for which the selected planner heading has been held. It can be seen that the count is reset when the path planner first encounters the obstacle as the path planner was run to find an avoiding swath. The count is however not reset a few seconds later when the selected heading is adjusted by the oscillation protection. Finally 30 seconds after the initial avoidance manoeuvre the count is reset as the hold period came to an end and the path planner selected a new heading.

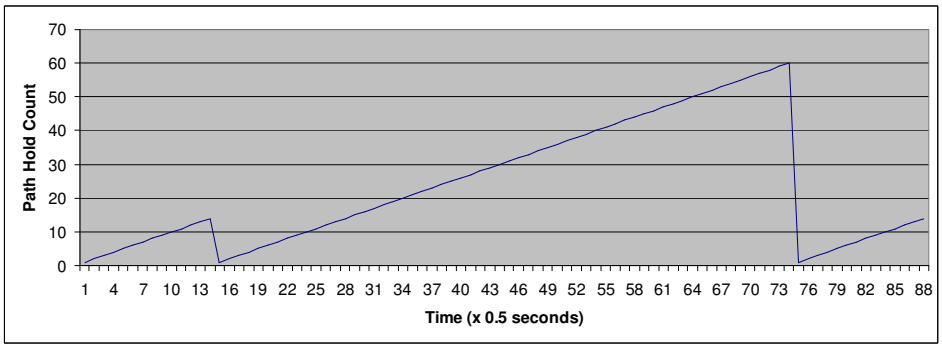

Figure 8.46 - Path Hold Count during Experiment with Heading Hold

In reality holding the selected path planner for 30 seconds is an extreme example. The experiment was repeated to fine tune the length of the path hold period. It was found that the

optimum path hold duration was 5 seconds. This covers the time it typically takes for the UGV to turn and start moving again after encountering an obstacle.

8.5.4 Planner Lookahead

In Chapter 7 it was stated that many path planners had their lookahead tuned to suit the experimental environment. So an experiment was performed to look at the effects of changing the length of the path planner lookahead. For this experiment a start square was painted on the ground for the UGV. A wooden pallet of width 0.75m was then erected at a distance of 7m in front of the painted square. Figure 8.47 shows this setup.

Figure 8.47 - Single Pallet Obstacle Course

This experiment required a small modification to the basic local path planner. The radial swath search process was limited so that it only searched swaths across the obstacle map as far as the specified lookahead distance. Normally this search would extend to the edges of the map. For simplicity the planner lookahead was made a parameter that could be set on the NOVA GUI.

The ARP vehicle was run repeatedly from the painted start square towards the pallet with an ideal heading that would lead to a direct collision. For each run a different planner lookahead

was specified. Results from a selection of the runs are shown in Figure 8.48. This diagram was generated by recording the relative location of the UGV during each run.

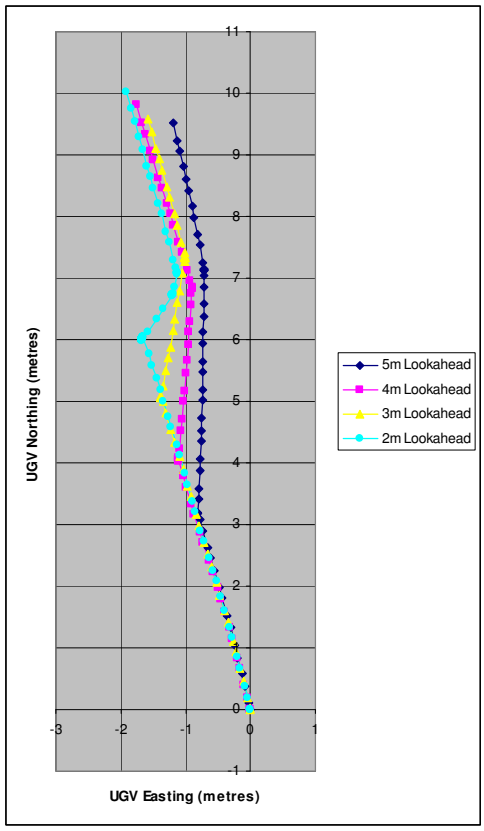

Figure 8.48 - Effect of Lookahead on Obstacle Avoidance

Two observations can be made about these results. In general the shorter the planner lookahead the closer the ARP vehicle would get to the pallet before taking evasive action. As a consequence of being closer to the pallet the UGV had to turn a greater angle to avoid collision with the pallet.

Now consider the obstacle course illustrated in Figure 8.49. This narrow and twisty course was built using a number of wooden pallets. As before the UGV was set up to run autonomously on a collision course with the pallets. When using a planner lookahead of 2m the UGV could

successful negotiate the course. If the lookahead was any greater the UGV would generally stop and declare that it was blocked.

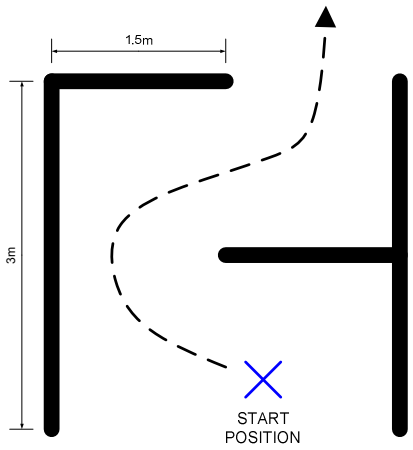

Figure 8.49 - Twisty Obstacle Course

So for the basic obstacle course it is clearly better to have as great a lookahead as possible. For the more complex obstacle course the UGV needed to use a very short planner lookahead to find a path between the pallets. These observations show that a path planner with an adjustable lookahead would be really useful for a real world UGV.

8.5.5 Adjustable Planner Lookahead and Claustrophobia

The basic local path planner was then expanded to include the adjustable lookahead feature described in Chapter 7. A series of experiments were conducted to evaluate this feature. These experiments also explored the effects of changing the claustrophobia setting. This setting dictates the amount the planner lookahead should adjust when the UGV encounters obstacles.

In this subsection only one of the later experiments in the series is described. An obstacle course was devised that could be negotiated in two ways. The first way was to bypass the

obstacles and follow a clear path alongside the course. The second was to drive through the middle of the obstacle course negotiating each obstacle. A plan of this course is shown in Figure 8.50.

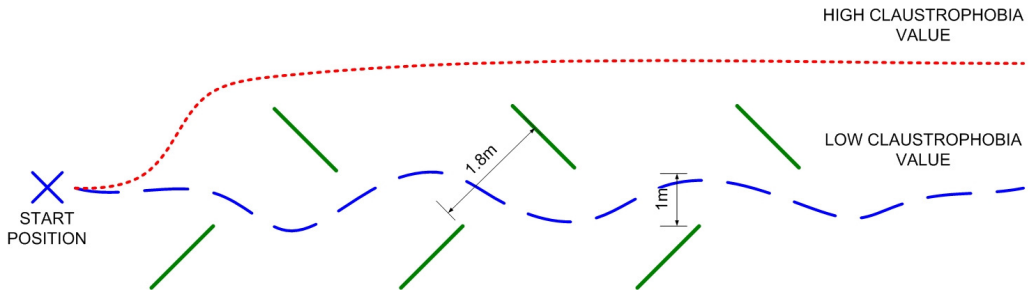

Figure 8.50 - Plan of Claustrophobia Obstacle Course

The proposed obstacle course was constructed using tarpaulin fences and a start location was marked on the ground to ensure the UGV always started from the same point. Then the UGV was run on a fixed heading towards the obstacle course. This was repeated with a variety of different values for the claustrophobia setting.

Figure 8.51 shows what happened when the claustrophobia setting was given a high value. In frame (a) the ARP vehicle encounters the obstacle course and selects a heading to avoid the first fence. By frame (b) the path planner has seen the second fence and selects a wider avoiding angle. This guides the UGV across the front of the obstacle course until it sees that the ideal heading is again clear at the location shown in frame (c). The UGV then bypasses the rest of the obstacle course as shown in frame (d).

A high claustrophobia setting implies that the UGV would have to be surrounded by obstacles in order for the planner lookahead to drop. So throughout the example run shown above the lookahead was maintained at the maximum value. As a consequence the planner did not see the gap between the two tarpaulin fences as a possible path. It could only bypass the obstacle

course. The UGV did seem to portray the signs of claustrophobia as it had an aversion to entering tight spaces.

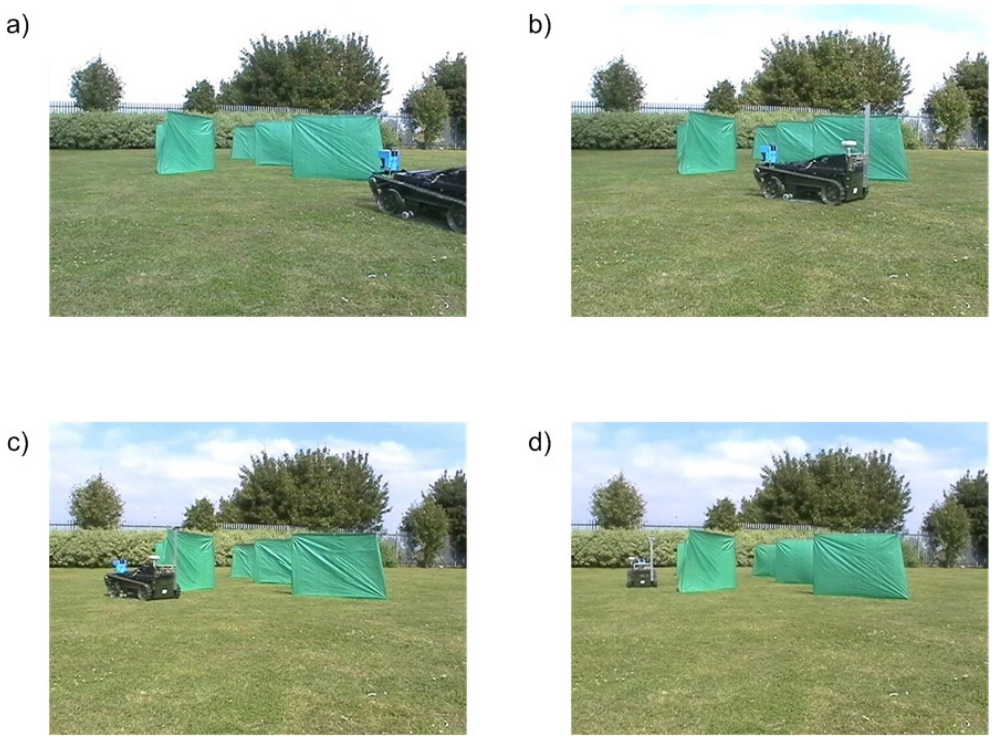

Figure 8.51 - Experiment with High Claustrophobia Setting

Figure 8.52 illustrates a repeat run of the obstacle course with a much lower claustrophobia setting. As can be seen from snapshot (a) the ARP vehicle was allowed to get much closer to the first fence before the path planner took evasive action. Avoiding the first fence took the UGV straight towards the second fence as shown in snapshot (b). In this run when avoiding the second fence the path between the fences is seen as a viable option. Snapshots (c) to (e) show that the UGV then encountered and avoided each of the fences in the obstacle course. The UGV emerges at the far end of the obstacle course in snapshot (f).

The path planner heading variables that were recorded during this run are shown in Figure 8.53. This plot clearly demonstrates that the UGV required three left turns and three right turns to negotiate the obstacle course. Figure 8.54 shows the corresponding recording of the planner

lookahead. Comparing this with the plot of heading variables it can be seen that the planner lookahead was at the maximum value until the UGV encountered the first fence. By the time the path planner avoided this fence the lookahead had been reduced to the lowest value. The lookahead then remained at 2m until the UGV had passed the last fence.

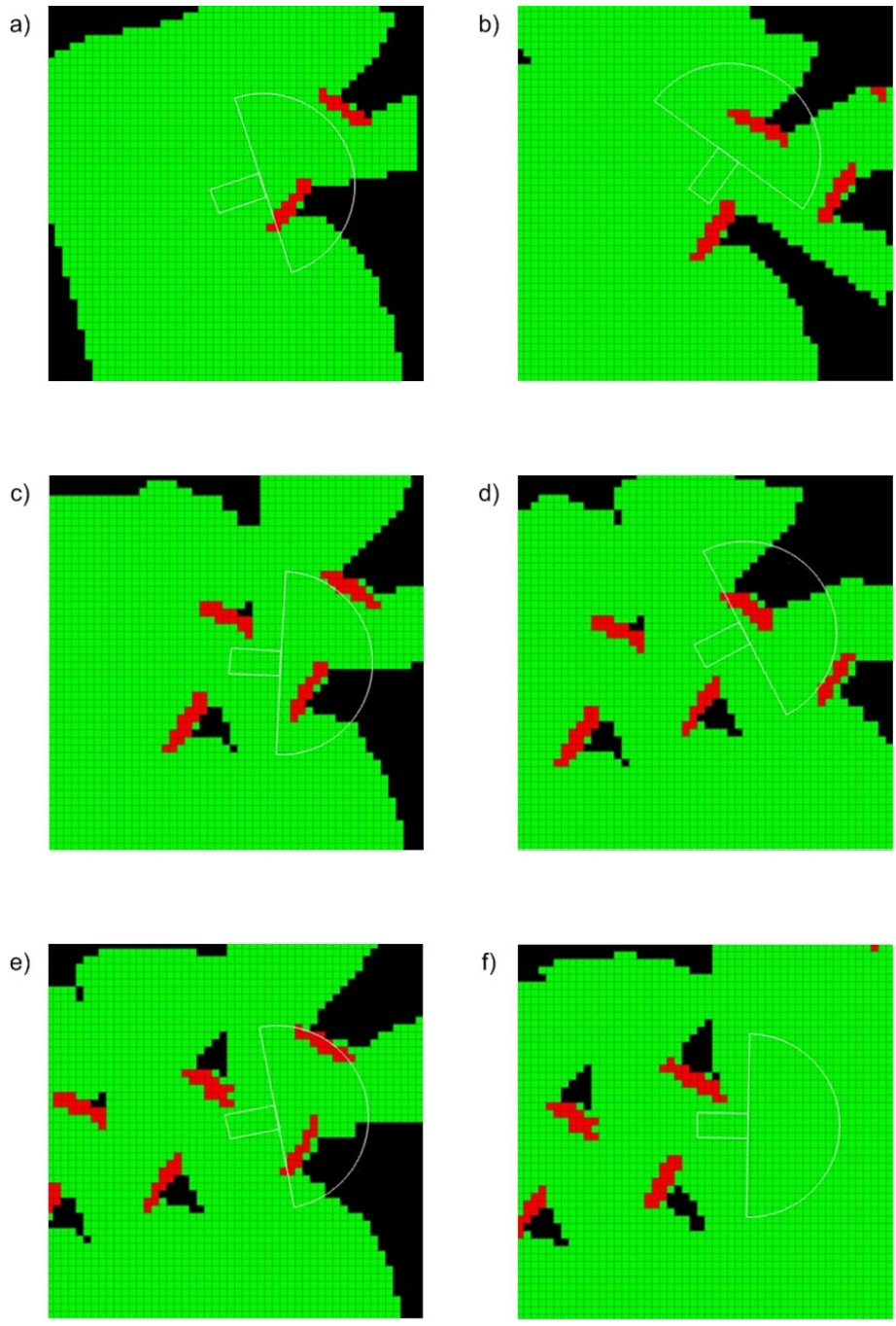

Figure 8.52 - Experiment with Low Claustrophobia Setting

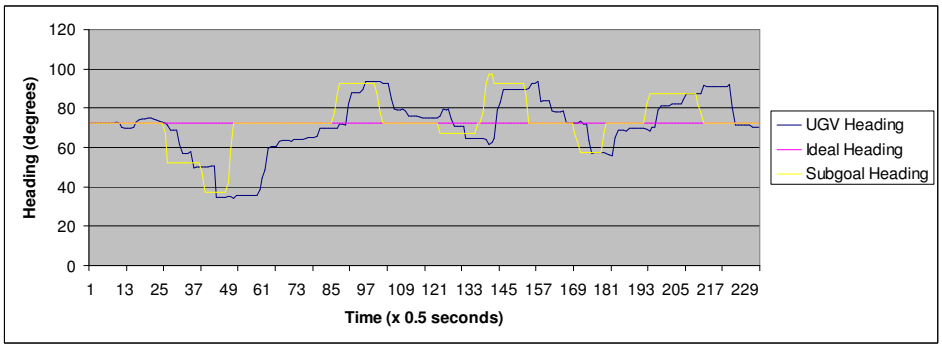

Figure 8.53 - Heading Variables from Low Claustrophobia Experiment

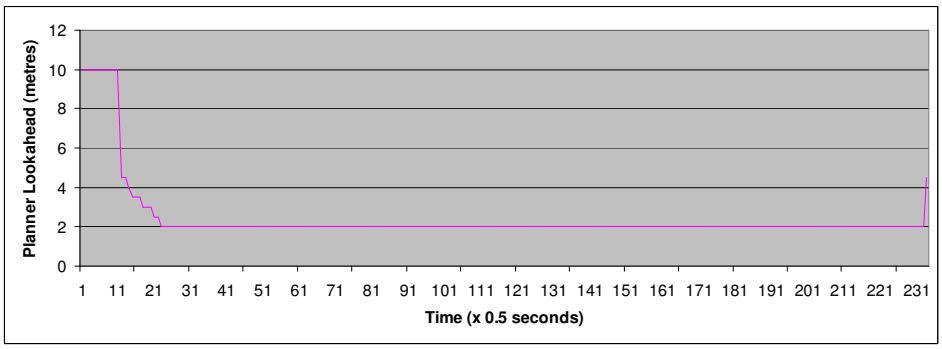

Figure 8.54 - Planner Lookahead from Low Claustrophobia Experiment

In general for claustrophobia settings above 0.5 the UGV would always bypass this obstacle course. This equates to the planner only adjusting the lookahead if more than 50% of the radial swaths across the obstacle map are blocked. Clearly this value is specific to this obstacle course.

After trialling the path planner with this obstacle course and also isolated obstacles the claustrophobia setting was given a default value of 0.75. It can be seen in Figure 8.41 that when the UGV is avoiding an isolated obstacle using the maximum lookahead up to 50% of the radial swaths can be blocked. The default claustrophobia setting was deliberately made larger than this so lookahead adjustment would only take place if the UGV was genuinely

surrounded by obstacles. Since the claustrophobia setting is configurable on the NOVA GUI it can be changed if the operator requires the UGV to enter a tight space or doorway.

8.6 EXPLORATION GRAPH EXPERIMENTS

This section describes the final series of experiments conducted in this research. The aim of these experiments was to evaluate the novel exploration graph concept. Initial experiments looked at the use of the exploration graph to escape from arrangements of obstacles that formed a cul-de-sac. The complementary idea of obstacle flooding was also demonstrated as an aide.

Further experiments were then performed to test extensions to the exploration graph idea. First exploration graphs with a rationalised number of decision nodes were trialled. Then the minimax planner lookahead was tested as a means to ensure the cul-de-sac escape behaviour was robust even if the path planner claustrophobia setting was low.

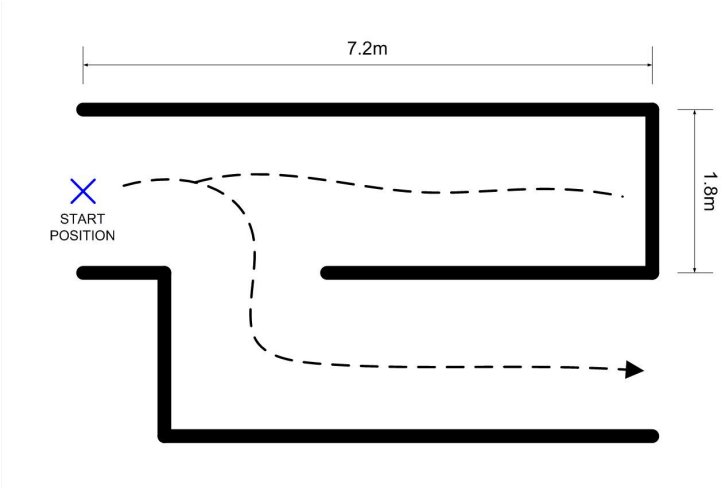

Figure 8.55 - Plan of Cul-de-sac Obstacle Course

To allow the cul-de-sac escaping behaviour to be demonstrated a new obstacle course was devised. Figure 8.55 shows a plan of this obstacle course. Essentially it is a corridor that has a dead end. The only way to escape is to back up out of the dead end and follow a branch off to

the right. Figure 8.56 shows the final obstacle course constructed using flexible tarpaulin fencing.

Figure 8.56 - ARP Vehicle Entering Cul-de-sac Obstacle Course

8.6.1 Without Obstacle Flooding

Before these experiments began it was expected that without the aid of obstacle flooding the UGV would not be able to escape from a cul-de-sac. So the aim of this initial experiment was really to show that use of obstacle flooding was necessary. The UGV was first parked at the entrance to the cul-de-sac obstacle course. Then it was set to autonomously drive towards the dead end on a fixed heading. As expected the UGV did not escape from the cul-de-sac. However it was really close to achieving this goal.

A trace of the UGV location during the experiment is shown in Figure 8.57. It can be seen that the UGV followed a roughly straight path into the obstacle course. When the dead end was detected the UGV correctly began to reverse through the decision nodes in the exploration graph. The UGV continued to reverse until it reached a decision node adjacent to the corridor branch. Then the UGV drove back into the cul-de-sac.

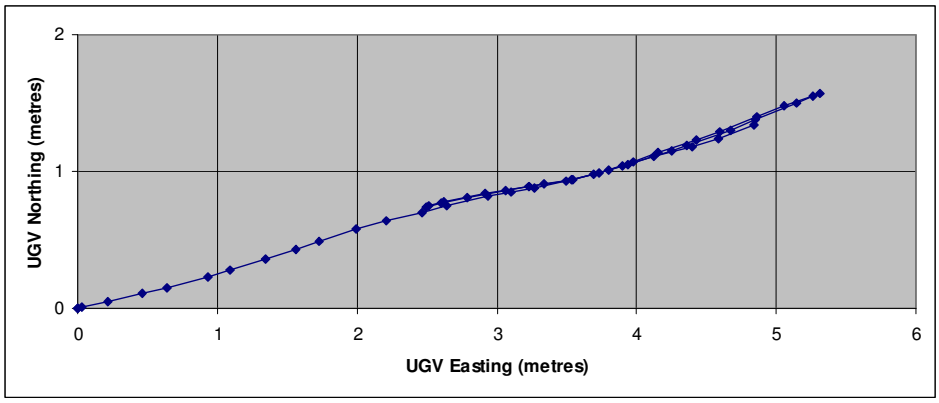

Figure 8.57 - Relative UGV Position Record

Figure 8.58 is a plot of the rank of the path planner decisions during the run. Initially the rank is 1 which shows the UGV was following the best path. When the cul-de-sac is detected the rank drops to 0. This indicates that the path planner was blocked and was reversing the UGV. After a few seconds the decision rank jumps briefly to a value of 2. This implies that for one cycle of the path planner the second best path had been selected.

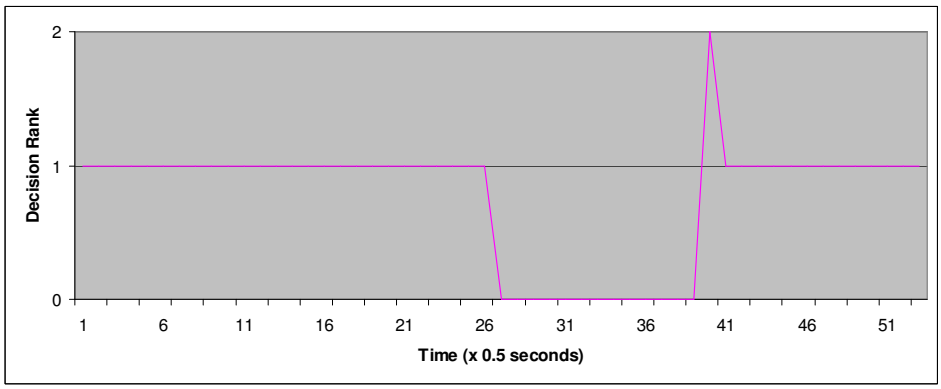

Figure 8.58 - Decision Rank Record without Obstacle Flooding

So the path planner reversed until it found a decision node that had an unexplored path. The planner then selected this alternate path for one cycle. This path would have corresponded to the corridor branch. After selecting the alternate path the path planner continued to select the best path on subsequent cycles. This then guided the UGV straight back into the cul-de-sac.

Figure 8.59 shows a plot of the path planner heading variables during the run. It can be seen that just before the path planner guided the UGV back into the cul-de-sac it indicated that the UGV should turn right 40°. Before the UGV had a chance to make the turn the path planner again selected the ideal heading leading the UGV back into the cul-de-sac.

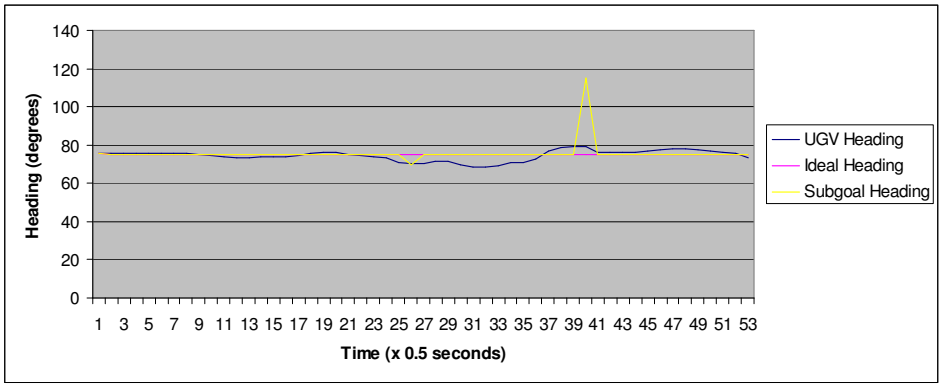

Figure 8.59 - Path Planner Heading Variable Record without Obstacle Flooding

This experiment was performed without the heading hold feature developed for the basic path planner. It is conceivable that using the heading hold the UGV could be coerced to escape from the cul-de-sac. There is however no guarantee that this behaviour would be repeatable. Obstacle flooding promised to be a much more reliable solution.

8.6.2 With Obstacle Flooding

Next the initial exploration graph experiment was repeated using an expanded version of NOVA that included the obstacle flooding feature. With the addition of obstacle flooding the ARP vehicle could repeatedly escape from the cul-de-sac. Figure 8.60 shows an example run.

In Figure 8.60 frame (a) the UGV is entering the obstacle course. Frame (b) shows the moment when the UGV detected the end of the cul-de-sac and began to reverse through the decision nodes. The point where the path planner finds a decision node with an unexplored path is captured by fame (c). This alternate path corresponds to the corridor branch which the

UGV immediately turned to face as frame (d) shows. After this the path planner guided the UGV along the corridor branch and out of the obstacle course. This is shown by the final two frames of Figure 8.60.

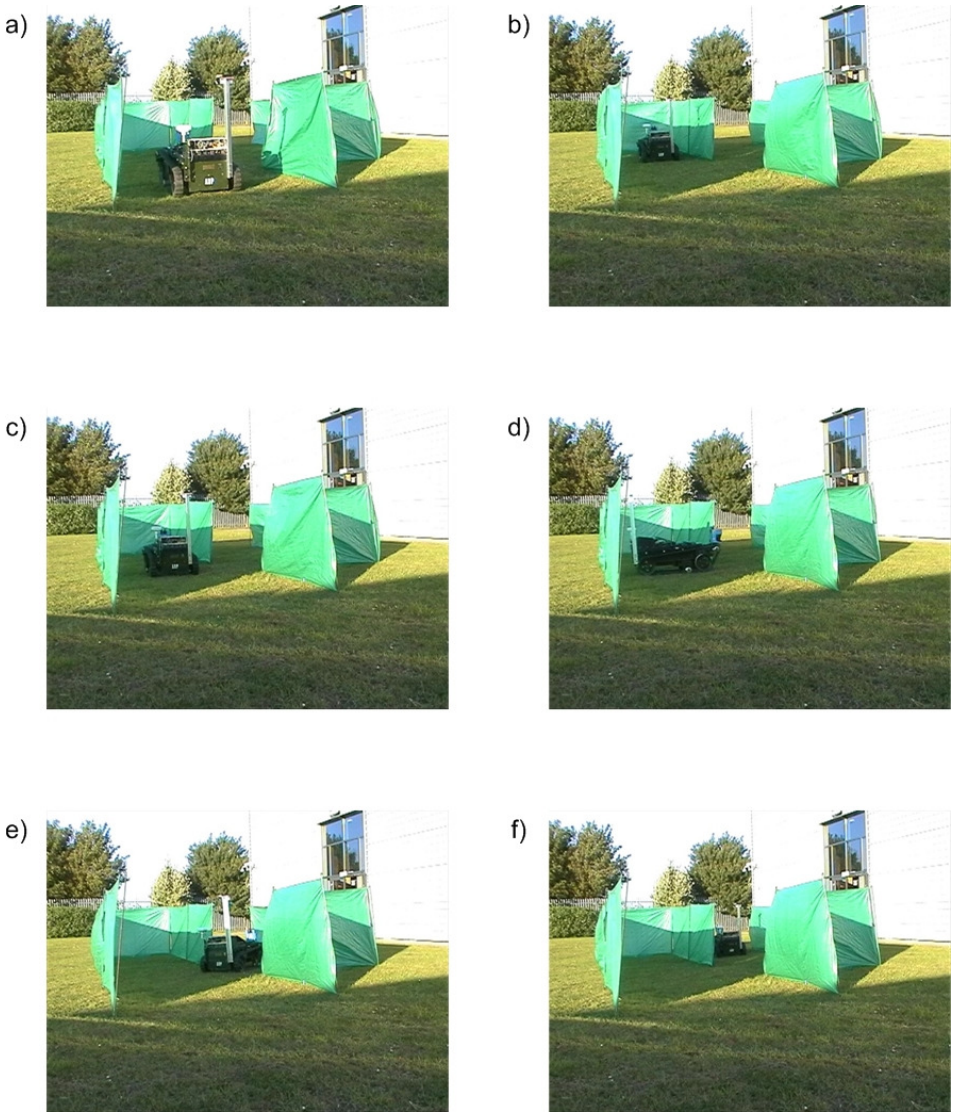

Figure 8.60 - ARP Vehicle Escaping from Cul-de-sac

In a separate run through the cul-de-sac obstacle course the state of Polar Obstacle Distance Histogram was recorded. Figure 8.61 shows a sequence of snapshots of the polar histogram. Snapshots (a) to (d) are from when the UGV was first driving into the cul-de-sac.

Backtracking through the decision nodes is captured by snapshots (e) and (f). The remaining snapshots show the UGV escaping along the corridor branch.

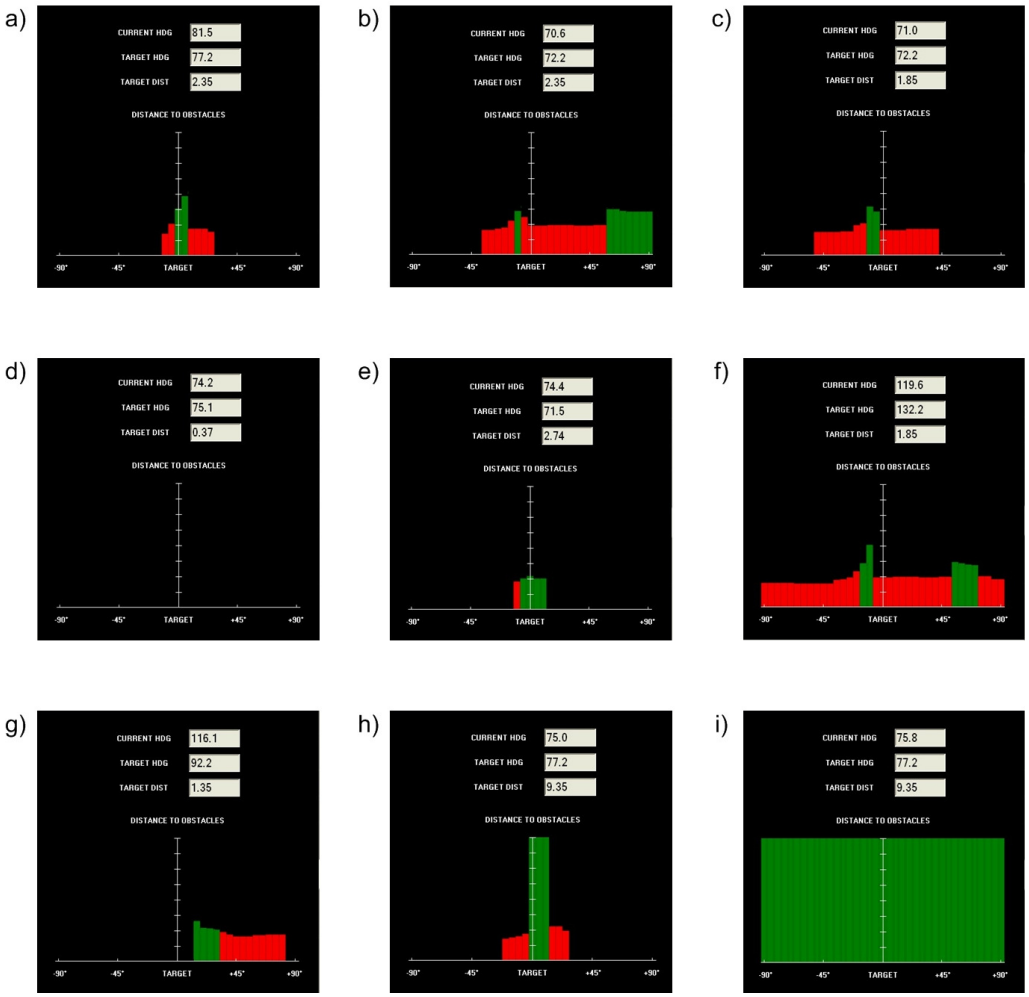

Figure 8.61 - Polar Obstacle Distance Histogram from Cul-de-sac Escape

Snapshot (a) in Figure 8.61 shows the polar histogram as the UGV enters the obstacle course. There is a single narrow path that can be followed. After a short distance the UGV reaches the corridor branch. This is captured in snapshot (b). Here the UGV has two possible path options the first is roughly straight ahead and the other is to the right. The path planner selects the best path and so continues straight as shown in snapshot (c). Ultimately this leads the UGV into the cul-de-sac and as snapshot (d) illustrates the path planner is completely blocked.

To escape the cul-de-sac the UGV reverses through the decision nodes. At each node the path planner checks the number of paths. If a node has an unexplored path the UGV will stop reversing. Figure 8.61(e) shows the polar histogram as the UGV is initially reversing. Only the path leading to the cul-de-sac is detected. Eventually the UGV stops at a decision node adjacent to the corridor branch. Here the polar histogram shows the two possible paths as indicated in snapshot (f).

Obstacle flooding prevents the UGV from selecting the path straight ahead. As a result the UGV selects the path to the right. Snapshot (g) shows the polar histogram as the UGV negotiates the corner. Once round the corner the UGV turns back to the ideal heading and follows the corridor branch as seen in snapshot (h). After a short distance the UGV escapes the obstacle course. Snapshot (i) shows the moment when the polar histogram indicates that there are no longer obstacles around the UGV.

So the obstacle flooding feature stopped the path planner from guiding the UGV back into the cul-de-sac as was observed in the last experiment. The effect of obstacle flooding is best observed by looking at the obstacle map as the UGV negotiates this obstacle course. Figure 8.62 shows a sequence of time ordered snapshots of the obstacle map from a run through the obstacle course.

Figure 8.62(a) pictures the point where the UGV first enters the obstacle course. In snapshot (b) the path planner has just detected the cul-de-sac. Snapshot (c) shows the UGV beginning to reverse through the decision nodes. The blue cells on the obstacle map are the false obstacles generated by the flooding process. Every time the UGV reaches another decision node the flooded area expands.

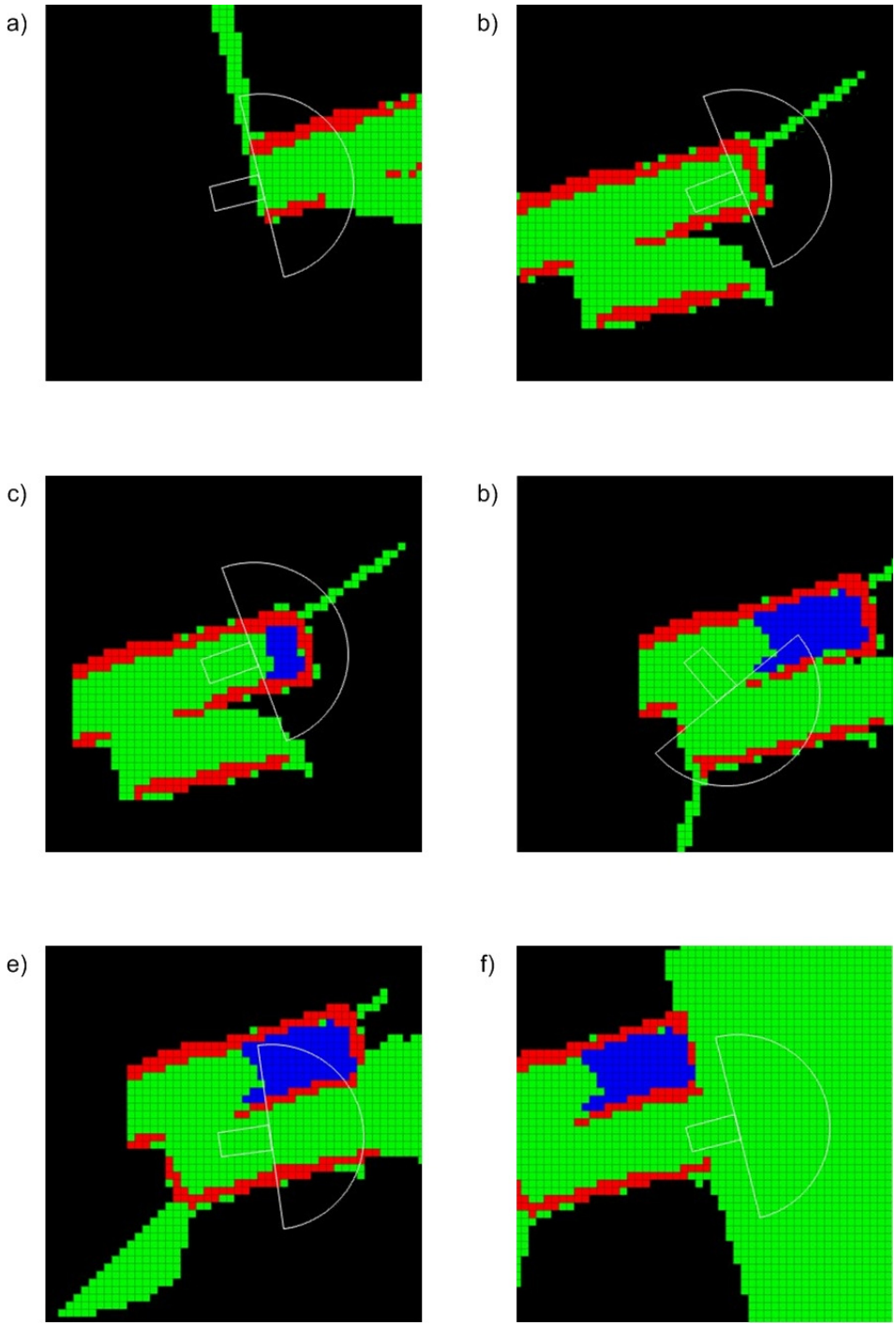

Figure 8.62 - Escaping the Cul-de-sac using Obstacle Flooding

Flooding continues until the path planner sees a decision node with an alternate path and the UGV turns to face this path as shown by snapshot (d). On subsequent cycles the path planner treats the flooding as an obstacle to be avoided. This has the effect of forcing the path planner

to guide the UGV along the branch. The final two snapshots in Figure 8.62 show the UGV escaping from the obstacle course.

Figure 8.63 shows a plot of the path planner decision rank during the cul-de-sac escape. Compare this with Figure 8.58 the similar plot from the experiment without the obstacle flooding feature. The key difference is that the path planner repeatedly selects the second best path in this experiment.

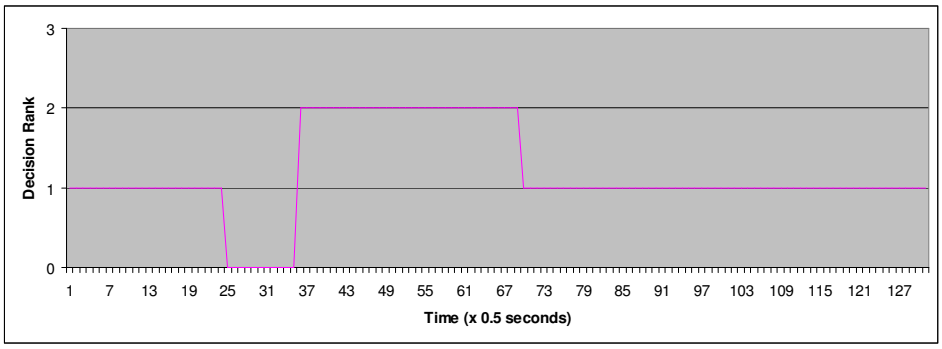

Figure 8.63 - Decision Rank Record using Obstacle Flooding

At the point where the UGV has reversed to a decision node with an unexplored path the best path will lead back into the cul-de-sac. Obstacle flooding prohibits selection of this path. The second best path corresponds to the corridor branch. So the plot indicates that the path planner selected to follow the branch. Once the UGV had negotiated the corner the branch becomes the best path. This was because the path planner could no longer see the path that led to the cul-de-sac.

The path planner heading variables recorded during the cul-de-sac escape are shown in Figure 8.64. It can be seen that the path planner turns the UGV right 85° when first selecting the corridor branch. Then as the UGV negotiates the corner the path planner gradually returns the UGV to the ideal heading.

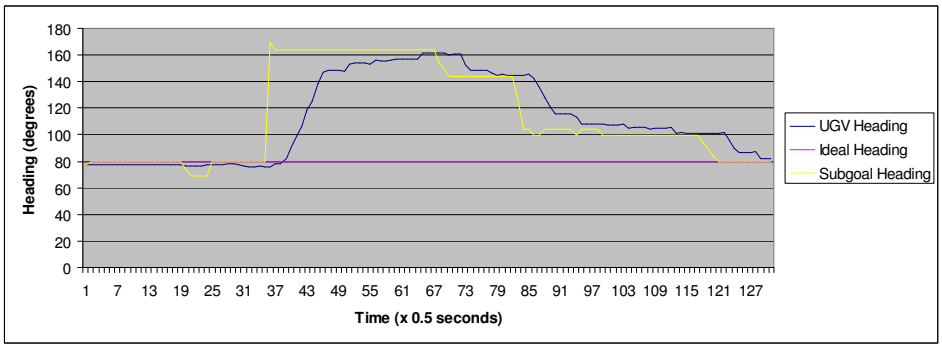

Figure 8.64 - Path Planner Heading Variable Record using Obstacle Flooding

The results above demonstrate that exploration graphs are a viable way to allow an autonomous vehicle to recover from driving into a cul-de-sac. To facilitate this use of exploration graphs the idea of obstacle flooding is of great help. Once a path has been identified as leading to a dead end if it is flooded with false obstacles the path planner cannot accidently guide the UGV back into the cul-de-sac. As such the path planner is forced use the exploration graph to find an alternate path.

8.6.3 Rationalised Decision Nodes

After it was shown that exploration graphs could be used to escape from cul-de-sacs extensions of the basic concept were considered. In Chapter 7 it was proposed that creating a decision node for every cycle of the path planner was unnecessary. The critical decision nodes were those at places just before the number of alternate paths decreased. These nodes represent the last opportunity that the path planner has to select a particular path before committing to another. As such it was suggested that the number of decision nodes in the exploration graph could be rationalised.

Figure 8.65 shows a plot of the number of decision nodes in the exploration graph during a run through the cul-de-sac obstacle course. Periods where the number of nodes was constant

correspond to when the UGV was turning. Whenever the UGV was moving forward a new node was created for each cycle of the path planner.

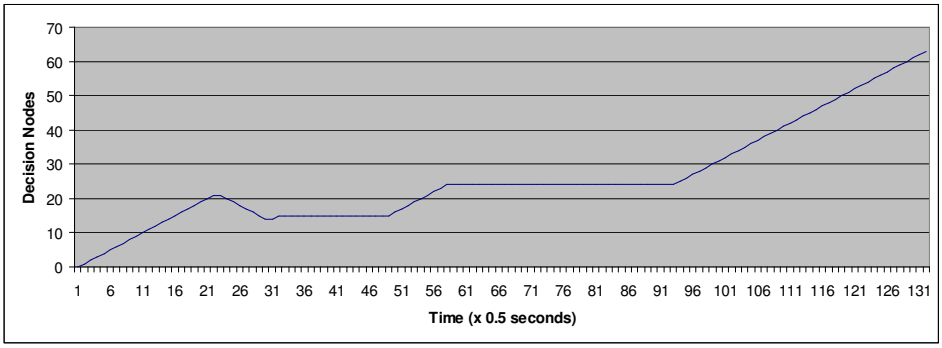

Figure 8.65 - Decision Nodes Generated during Cul-de-sac Escape

At one point in the plot the number of decision nodes decreased. This happened when the UGV was escaping from the cul-de-sac. As the UGV backtracked through the exploration graph in search of an alternate path the fully explored decision nodes were discarded. It can be seen that immediately after the UGV stops backtracking a turn is triggered. This turn leads the UGV down the corridor branch.

Two degrees of decision node rationalisation were trialled. First a basic rationalisation algorithm was implemented. This kept all decision nodes that were created immediately before the number of alternate paths increased or decreased. Second the full rationalisation algorithm was implemented. This only kept decision nodes that were created immediately before the number of alternate paths decreased. Trials using the leaner exploration graphs again used the cul-de-sac obstacle course.

Figure 8.66 shows the number of decision nodes recorded during a run of the obstacle course using the basic rationalisation process. The first node is the root node that is recorded at the

start of each UGV run using exploration graphs. This is shortly followed by a second node that corresponds to the path planner reducing the lookahead to an appropriate distance.

When the UGV begins to move forward three more decision nodes are added to the exploration graph. The first of these was generated when the path planner first saw the corridor branch. This is then followed by the node that represents the last time the UGV saw the corridor branch. The third node represents when the path planner detects the cul-de-sac.

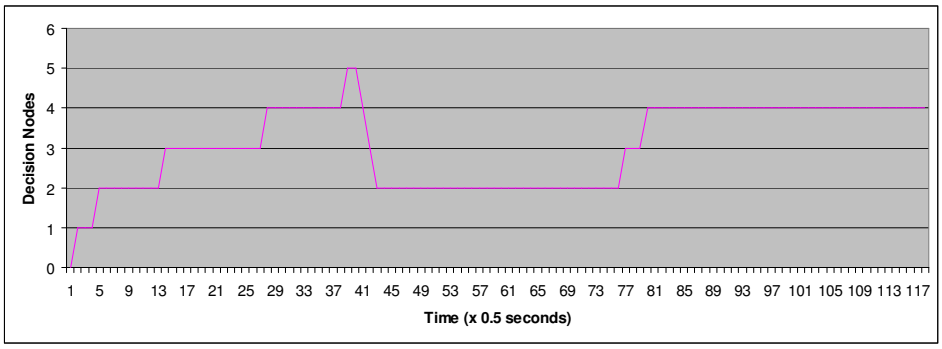

Figure 8.66 - Decision Nodes Generated with Basic Rationalisation

After detecting the cul-de-sac the UGV backtracks through the decision nodes discarding those that have been fully explored. When a decision node with an unexplored path was found the UGV stopped and turned to face this path. Once the UGV had completed the turn a new decision node was recorded. This node was to replace the node that had the unexplored path indicating that the path planner had now selected an alternate path. A final decision node was then created when the path planner lost sight of the path into the cul-de-sac.

It was observed that when using the basic rationalisation the cul-de-sac escape behaviour was not as reliable. Often when the UGV returned to the second decision node adjacent to corridor branch it could not see the alternate path. This was because the UGV was in a slightly different position. In this case the UGV normally backed up to the first decision node adjacent

to the corridor branch and then correctly turned down the branch. On rare occasions the path planner failed to find the alternate path.

Using the full decision node rationalisation process the cul-de-sac escape behaviour was much flakier. The path planner only had one chance to see the corridor branch when backtracking through the decision nodes. If the UGV was not in exactly the right position when it returned to the decision node for the branch the path planner did not see the alternate path. If the path planner missed the corridor branch the UGV would continue backtracking to the root decision node.

This experiment illustrates that if exploration graphs are to be rationalised the location of the decision nodes that are recorded is very important. It may be best to record a selection of decision nodes adjacent to each alternate path. Recording more decision nodes means that there are more opportunities to see alternative paths when backtracking through those decision nodes. It would be relatively simple to modify the rationalisation process so the last few decision nodes were recorded next to each unexplored path.

8.6.4 Claustrophobia & Minimax Lookahead

The aim of the final experiments with exploration graphs was to ensure compatibility between the adjustable path planner lookahead feature and the cul-de-sac escape behaviour. In Chapter 7 it was suggested that if the path planner had a low claustrophobia setting the lookahead would fall to the minimum value in the presence of any obstacles. This could lead to a situation where the path planner did not notice alternate paths as the lookahead was too short. Up to this point all exploration graph experiments had been performed using the default claustrophobia setting of 0.75.

Figure 8.67 shows a record of the path planner lookahead during a typical run through the cul-de-sac obstacle course using the default claustrophobia setting. As the UGV enters the course the lookahead tumbles eventually reaching the minimum value of 2m. The lookahead increases again when the corridor branch is reached. This is because there was more open space around the UGV. After the branch is passed the lookahead falls back to the minimum distance as the path planner guides the UGV towards the cul-de-sac.

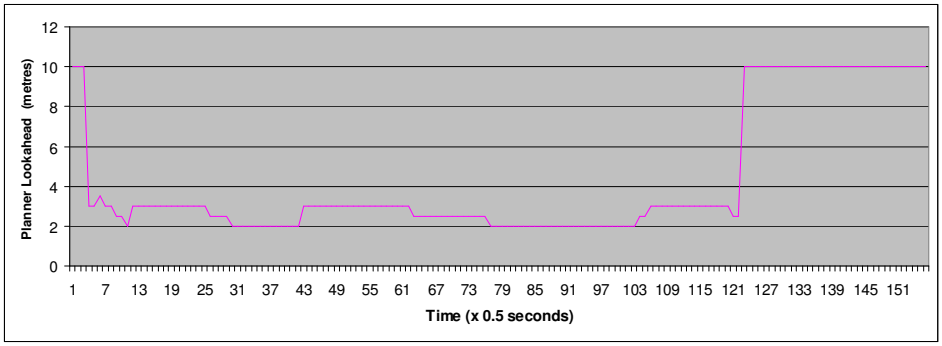

Figure 8.67 - Lookahead for Cul-de-sac Escape with High Claustrophobia

When the UGV returns to the decision nodes adjacent to the corridor branch the lookahead climbs again because of the extra space. However once the UGV has negotiated the corner the lookahead falls back to the minimum as the UGV drives along the narrow branch. Finally when the UGV emerges from the far end of the obstacle course the planner lookahead rises back to the maximum distance.

The original cul-de-sac escape experiment was repeated with the path planner claustrophobia setting given a very low value. Thus throughout this experiment the path planner lookahead was effectively fixed at the minimum distance of 2m. As a result in repeated runs of the obstacle course the UGV often failed to escape. Figure 8.68 illustrates what happened when the UGV did not escape.

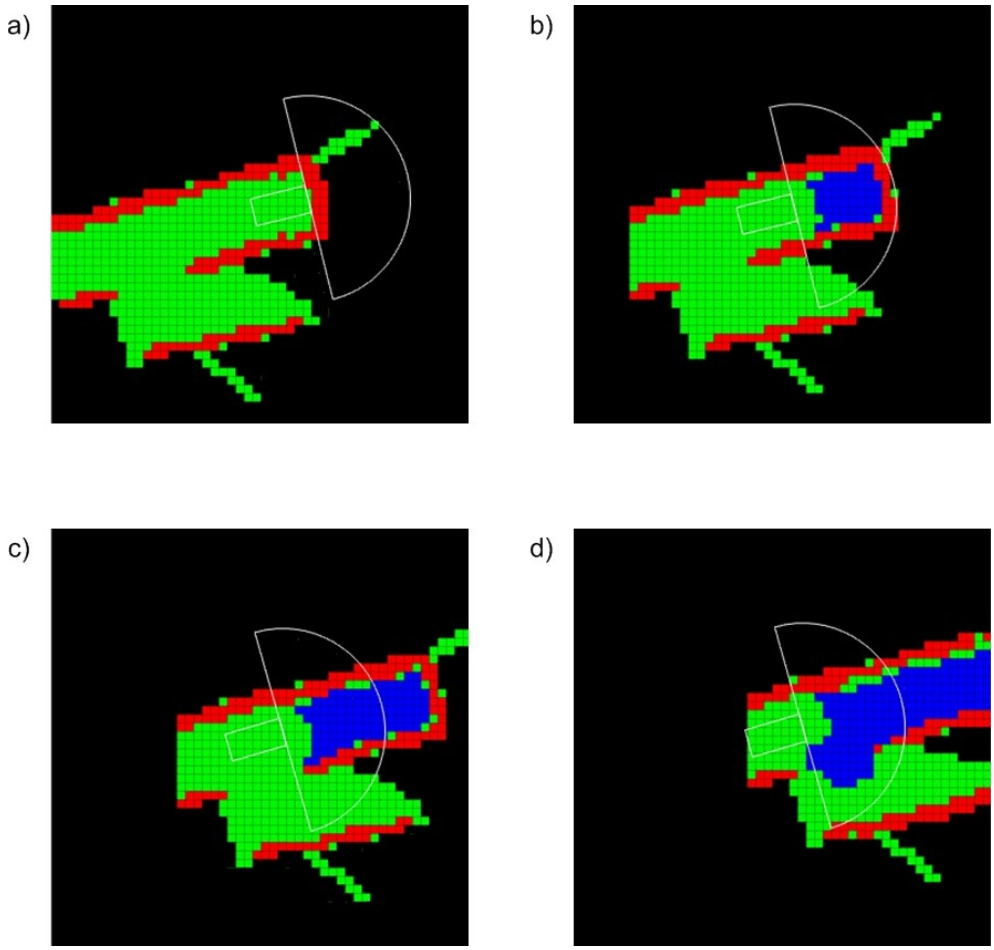

Figure 8.68 - Effect of Low Claustrophobia Setting

A sequence of snapshots from the obstacle map is shown in Figure 8.68. The point where the path planner detects the cul-de-sac is captured by snapshot (a). Snapshot (b) shows the UGV beginning to backtrack through the decision nodes correctly flooding the path into the cul-de-sac. The point where the UGV reached the corridor branch is shown in snapshot (c). However due to the short lookahead the path planner does not see the alternate path. The UGV thus continues to back up and floods the corridor branch with obstacles as shown in snapshot (d).

At the critical moment where the UGV would normally turn down the corridor branch the path planner could only see one possible path. On the corresponding polar histogram this path was represented by a peak around 90° wide. This included radial swaths that led into the cul-de-sac and those that led to the corridor branch. With a planner lookahead of 2m the UGV needed to

be very close to the fencing on the right hand side of the corridor if the path planner was to notice the branch.

In Chapter 7 a solution was proposed for this type of problem. This solution was to use the minimax lookahead for the path planner. The minimax lookahead is the minimum path planner lookahead that allows the maximum number of alternate paths to be seen. For this final series of experiments the path planner was expanded to determine the minimax lookahead in each path planning cycle. The adjustable lookahead algorithm was also modified to prevent the path planner lookahead from ever falling below the minimax lookahead.

Figure 8.69 illustrates the effect of using the minimax lookahead during the cul-de-sac escape. For this run through the obstacle course the claustrophobia setting was dropped to 0. Thus without the aid of the minimax lookahead the planner lookahead would have been 2m while the UGV was within the obstacle course.

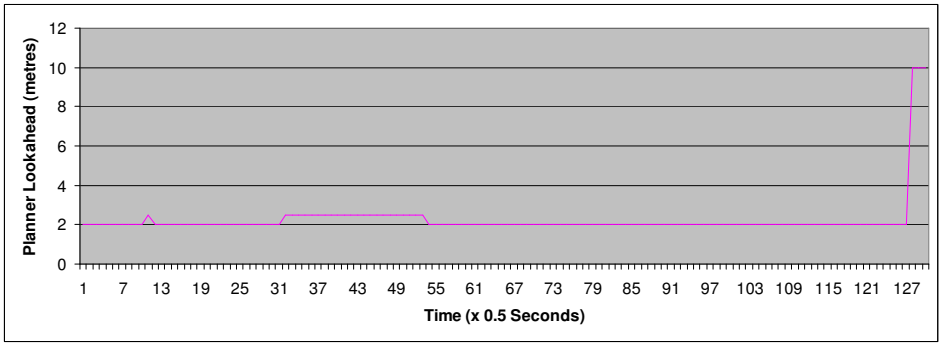

Figure 8.69 - Lookahead for Cul-de-sac with Minimax

At three points during the run the planner lookahead climbed to more than the 2m minimum value. The first time coincided with the UGV passing the corridor branch on the way into the cul-de-sac. The second was when the UGV backtracked to the corridor branch. This allowed the UGV to see and select the alternate path corresponding to the branch. Both of these

increases in the planner lookahead were due to the minimax lookahead being greater than 2m. So without using the minimax lookahead the UGV would have failed to escape from the course. The final time the planner lookahead raised to more than 2m was at the end of the run when the UGV escaped from the obstacle course.

This experiment was repeated numerous times with a variety of different claustrophobia settings. Whenever the minimax lookahead was used the UGV would successfully negotiate the cul-de-sac obstacle course regardless of the claustrophobia setting. So it can be said that use of the minimax lookahead is essential when using a path planner with adjustable lookahead.

8.7 EVALUATION OF RESULTS

Here in the concluding section the experimental results from this research are evaluated. Experiments have concentrated on the localisation, obstacle mapping, and path planning systems within NOVA. The novel exploration graph concept has also been extensively trialled.

In the following subsections the results from each series of experiments are summarised. The positives and negatives shown in each system within NOVA are highlighted. Suggestions are also made for potential improvements to the different systems.

8.7.1 Localisation

The localisation experiments demonstrated that the particle filter is a viable means to fuse the absolute and relative localisation systems on the UGV. It was shown that the particle filter overcame the error accumulation that afflicts the relative localisation system. The experiments also illustrated that the particle filter was not susceptible to the location drifts associated with GPS. This however did require a minor modification to the basic filter design to ensure that the filter was not updated while the UGV was stationary.

Several experiments looked at simplifying parts of the particle filter algorithm. It was demonstrated that the particle filter quickly degenerates if the particle set resampling step is omitted from the algorithm. Additionally it was shown that the GPS error model can be simplified as error data from the GPS receiver seemed somewhat optimistic.

The particle filter localisation system has many positive points. First it gives absolute position estimates for the UGV like the GPS receiver but was much more accurate in tracking the UGV location over short distances. Also when the associated GPS data has errors the particle filter was not greatly disturbed. The only minor negative point is that over short distances the particle filter seems not quite as accurate as the underlying relative localisation system.

If more time were available a full scale evaluation of the particle filter could be conducted. The effect of different particle set sizes could be evaluated. Alternatively the performance of the filter could be trialled against an implementation of the popular Kalman filter. This however would be a research project in itself.

8.7.2 Obstacle Mapping

As part of this research two separate obstacle mapping systems were developed. One of these systems was a complete success. The other worked but was a source of much frustration caused by the various hardware problems that were encountered.

It was demonstrated the 2D obstacle mapping system could produce very accurate maps of the area around the UGV. These showed little noise and scrolled to match the movement of the UGV. Later experiments also illustrated that the idea of mapping entire UGV missions was a possibility requiring only minor changes to the basic mapping system. A potential next step would be to expand this mission mapping capability to store many square kilometres of terrain maps in an efficient manner.

Experiments were additionally conducted to evaluate the performance of the 3D obstacle mapping system. This system also had the capability to produce accurate maps of the environment that scrolled to match the movement of the UGV. However hardware problems limited the extent of experimentation with this mapping system.

The 3D system was evaluated using a planar laser scanner in a push broom configuration. Planned experiments using a 3D range camera in place of the laser scanner were not performed due to technical problems with the camera. Also the 3D obstacle mapping system was limited to operation on smooth flat surfaces. This was due to unforeseen noise problems with the two pose sensors trialled on the ARP vehicle.

Given more time and funding the work with the 3D obstacle mapping system could be extended. Procurement of a better means of pose sensing would allow a full evaluation of the 3D mapping system in rough terrain. It would be desirable if the heading measurement of the replacement pose sensor was also not susceptible to deflection by magnetic fields. In the real world the reliance of the mapping system on a compass for heading measurements could cause problems.

If a true 3D range sensor was added to the ARP vehicle complete images of the environment could be captured. This would allow the development of a proper 3D confidence mapping system. The PMD [vision]® 1k-S range camera was originally acquired with that in mind. Due to the time spent dealing with hardware issues a lot of improvements for the 3D mapping system could not be implemented. Additional experiments were planned to investigate different ways to summarise the range points in each voxel and determine traversability.

8.7.3 Local Path Planner

Experiments with the basic local path planner have demonstrated that planning a path by searching radial swaths across the obstacle map was an appropriate idea. Using this system it was shown that the UGV was capable of avoiding obstacles that lay on the ideal heading towards the next waypoint. This path planner was tested with both the 2D and 3D obstacle mapping systems that were developed for NOVA. So it could be used with each of the range sensor models developed in this research.

A few minor limitations of the basic path planning system were observed. This led to the shape of the planner search swath being changed to ensure the UGV did not drive too close to obstacles. Also to overcome an infrequent heading oscillation an algorithm was devised to make the path planner hold a path for a configurable period of time.

Next the idea of lookahead was investigated with the path planner. It was shown that different situations require different lookahead distances. To that end a modified version of the path planner using a novel automatically adjusting lookahead was trialled. The lookahead of this planner changed in line with the obstacle density around the UGV.

Alongside this the notion of claustrophobia for the path planner was introduced. It was shown that with a high claustrophobia setting the UGV could be made to take a wide line around obstacle fields. Alternatively the UGV would get up close to obstacles and try to plot a route between them if the path planner claustrophobia setting was low.

The final version of the path planner could operate successfully with isolated obstacles and in tight obstacle cluttered situations. If there were more time the next step would be to test the system over longer distances and in natural environments. Several potential improvements to the path planner have also been conceived.

The first improvement would be to expand the planner to use a 360° polar histogram in place of the current 180° version. Simplifying the path planner to only consider paths in the range ±90° for the ideal heading was unnecessarily restrictive. Secondly an investigation could be conducted into the use of the maximax lookahead to control the planner lookahead adjustment process. The maximax lookahead would be the maximum lookahead that finds the maximum number of alternate paths for the UGV. This should ensure that the path planner always selects an optimum lookahead.

8.7.4 Exploration Graphs

In the final series of experiments it was demonstrated that an exploration graph can be used to recover the UGV from cul-de-sacs. The path planner was expanded for these experiments to record decisions as nodes in an exploration graph. Then when a dead end was encountered the UGV would backtrack through the decision nodes looking for one with an unexplored alternate path.

It was shown that obstacle flooding was a necessary feature to prevent the path planner from guiding the UGV back into a cul-de-sac once it had escaped. An example run illustrated that obstacle flooding fills the cells of the obstacle map that lay on the path leading into the cul-de-sac with false obstacles. These false obstacles then force the path planner to guide the UGV along an alternate path.

Additional experiments looked at improvements to the basic exploration graph concept. Rationalisation of the exploration graphs seemed like a good idea so each alternate path would be indicted by a single decision node. However it was found that the location of the decision nodes that are retained is critical. The UGV could get into a situation where it returned to a decision node but failed to see the alternate path that it indicated. It was concluded that retaining multiple decision nodes is a better idea around the alternate paths.

It was also demonstrated that with a low claustrophobia setting the local path planner could miss alternate paths as the planner lookahead had dropped too low. The idea of the minimax lookahead was tested as a minimum planner lookahead. It was shown that by modifying the path planner to use the minimax lookahead alternate paths were never missed.

If more time were available there are several more improvements that could be made to the exploration graph concept. For dynamic environments it may be a better choice to record the heading selected by the path planner rather than just the number of times the UGV had visited a decision node. If obstacles move then the current method would fall down but if the heading was recorded the true next best path could be found. It would also be nice to try exploration graphs together with a full mission mapping system.

Another potential extension is to use forward simulation with the path planner and exploration graph concept. Forward simulation could allow the path planner to build a path to the edge of the obstacle map using a string of radial swaths. At each cycle the path planner would first find the best path. If the best path did not extend to edge of the obstacle map the planner would imagine that the UGV had driven a short way along the best path then run again. This process would be repeated until a path to the edge of the map was found or it was found that the path was blocked. Forward simulation would allow the path planner to exploit more of the information on the obstacle map.

9 CONCLUSIONS

This research has seen the development of NOVA the Nottingham Off-road Vehicle Architecture. NOVA is a complete UGV control architecture and it includes elements for UGV localisation, obstacle mapping & autonomous navigation. NOVA has been extensively tested onboard a UGV known as the Autonomous Route Proving (ARP) vehicle. This vehicle was also constructed during this research.

In the following pages the entire research program is evaluated. The evaluation has been split into three sections. First a summary of the research is given. Next the novel contributions of the project are highlighted. The final section outlines ideas for further work to build on the foundation that this research provides.

9.1 SUMMARY OF PROJECT

This section provides a summary of the research conducted in this project. There were three phases in the project. The UGV was built first. Following this the control software was designed, implemented and then debugged. The experimental phase was the last part of the research and was used to test the advanced features of UGV autonomy. The major tasks in each phase of the research are detailed here in a rough chronological order.

9.1.1 Construction of UGV

The project began with the construction of ARP vehicle. This entailed refurbishing a Mk 7 Wheelbarrow EOD vehicle donated by Remotec UK. The entire drive train of the vehicle has been replaced from drive motors to tracks. A modern control system was then integrated with the chassis of the ARP vehicle. This included an electronic Vehicle Control Unit (VCU) to control the drive motors and the NOVA Computer to run the UGV control architecture. Next an array of localisation equipment was added to the UGV including a GPS receiver, pose

sensor and wheel encoders. Finally two perception sensors were integrated with the UGV. These were a Sick LMS 200 laser scanner and a PMD [vision]® 1k-S range camera.

9.1.2 Development of NOVA

Following the construction of the UGV a framework for NOVA was developed. This began as a basic architecture to fuse data from all of the sensor equipment. From this the ARP Interface GUI software was implemented. The ARP Interface is a program for the NOVA Computer that allows manual control of ARP vehicle. It also displays feedback from the various sensors onboard the UGV. This shows if each sensor is working correctly and allows the perception sensors to be calibrated.

Next NOVA was expanded to allow autonomous UGV operation. First a system was added to perform localisation of the UGV. After this an autonomous navigation system was devised that could follow an operator specified route. The perception sensing was then incorporated to create a system that could detect and map the obstacles in the environment. Finally local path planning was added to provide an obstacle avoidance capability that made use of the obstacle mapping system. Many refinements were made as the architecture took shape.

The first form of localisation for the UGV was a dead reckoning system. This used the wheel encoders and digital compass onboard the UGV. Average encoder change was used as a measure of distance travelled at each time step. The latest distance travelled was incorporated along the current UGV heading to update the relative location estimate. In this way the system recursively generated an estimate of the UGV location relative to the start location.

A GUI was then developed to allow routes for the UGV to be specified as a sequence of waypoints. To navigate the route the UGV visited each waypoint in turn always following the ideal heading towards the next waypoint. Autonomous system controls allowed the movement

of the UGV to be stopped and started as required. Finally a direction control was also devised for the GUI to allow the UGV to be driven in a specified direction. This allowed the UGV to be used without any prior knowledge of the test area.

As part of the autonomous navigation system a series of route proving behaviours were developed. These were all based on the concept of path recording. This created a sequence of waypoints that described the route the UGV had driven along. Using the recorded path the UGV could backtrack along the previously driven route. Also parallel swaths of the recorded path could be generated. This process shifted the recorded path to the left or right as required so the UGV could widen a proven route. A behaviour that performed an area search was also implemented. This generated a coverage plan to guide the UGV to cover a specified area with a series of parallel swaths.

Next the UGV localisation system was refined. The idea was to integrate the GPS receiver with the dead reckoning system. Originally the naïve plan was to use GPS as the main means of localisation for the UGV with dead reckoning acting as a backup when GPS location estimates were unavailable. Preliminary tests showed that GPS was too noisy for this plan to be successful. A particle filter was thus devised to fuse the location estimates from the dead reckoning system and GPS receiver.

After this a 2D obstacle mapping system was created for NOVA. This was based on idea of confidence mapping. A 2D sensor model was developed first to transform range data from the laser scanner into confidence map coordinates. Then a means to update the map was devised. For each frame of sensor data a momentary map is built and used to incorporate new range points into the confidence map. If a range point from the sensor falls into a map cell the confidence that an obstacle is present increases. Ray tracing is used to decrease the confidence

value for cells that lie in the clear space between the range points and sensor. This system produces very accurate and fast updating dynamic obstacle maps.

It was hoped that the UGV could be developed to a state where it could operate in truly rough terrain. This is not possible with a 2D obstacle mapping system. To this end a 3D mapping system was also devised. This included 3D sensor models for use with the Sick LMS 200 laser scanner mounted in a push broom configuration and also the PMD [vision]® 1k-S range camera. These models featured compensation for the pose of UGV when transforming range data from the sensor to the obstacle map. This was essential for the UGV to map and negotiate rough terrain.

The 3D mapping system was based on the idea of voxels. Range points from the 3D perception sensor were stored in appropriate voxels. For each voxel traversability statistics were generated to describe the range points that had fallen within. These traversability statistics were used to determine if the UGV could safely traverse the voxel or if an obstacle was present.

Next the local path planning system was developed. This was a refinement of the autonomous navigation system that made use of the obstacle proximity information held in the obstacle map. The basic local path planner searches for obstacle free paths from the UGV to the edge of the obstacle map. This is done using a radial swath search process that searches paths at 5° angular increments across map. From the search a Polar Obstacle Distance Histogram is constructed to record the obstacle free length of each radial swath. The path planner then uses the polar histogram to select the clear swath with minimum angular deviation from the ideal heading to the next waypoint. As before the ideal heading is provided by the basic autonomous navigation system.

It was recognised that the lookahead of the path planner was important. Situations were conceived where the basic planner concept would struggle as a clear swath to edge of map would not be found. As a result a method was devised to dynamically adjust the path planner lookahead based on the density of obstacles around the UGV. If the number of blocked radial swaths was more than a given threshold the planner lookahead was reduced. It was noticed that this threshold specified how claustrophobic the UGV would become. Higher threshold values would stop the lookahead being reduced so easily and force the UGV to stay further from obstacles.

The novel exploration graph concept was then devised to allow the path planner to recover the UGV if it drove into a cul-de-sac. The exploration graph is built from decision nodes that are recorded each time the path planner runs. These nodes record the details of the decision made by the path planner and the location of the UGV at the time. If the UGV does drive into a cul-de-sac the path planner reverses the UGV through the recorded decision nodes. At each decision node the path planner checks if an alternate path exists. When an unexplored alternate path is found the path planner guides the UGV along this path instead.

To facilitate the use of exploration graphs the complementary idea of obstacle flooding was also developed. After encountering a cul-de-sac the obstacle map is flooded with false obstacles as the UGV reverses through the decision nodes. Flooding is confined to map cells that lie on the path that led into the cul-de-sac. The presence of the flooding prevents the path planner from guiding the UGV back into cul-de-sac. It has the effect of forcing the UGV along an alternate path.

The final development of the research was the idea of the minimax path planner lookahead. This is a special planner lookahead that represents the minimum lookahead with the maximum number of alternate path choices. Having a planner lookahead that allows the UGV to see all

of the alternate paths is essential to the successful operation of the exploration graph concept. So the adjustable path planner was modified to use the minimax lookahead as the minimum allowed lookahead in any situation.

9.1.3 Experimental Phase

When the majority of NOVA had been developed the research was relocated to the test site. The open space at the test site allowed the ARP vehicle to be run around for the first time as previously it had been limited to short indoor tests. First the equipment onboard the ARP vehicle was evaluated. Then each part of NOVA was tested using the UGV. This allowed all observed problems in the software to be fixed.

After this experiments were performed with the ARP vehicle to validate NOVA. First the localisation system was subjected to scrutiny. Experiments showed that the particle filter could successfully fuse the location estimates from the dead reckoning system and GPS receiver. Furthermore it was demonstrated that the particle filter did not suffer from the errors associated with those two underlying localisation systems.

Additional experiments illustrated that both the 2D and 3D obstacle mapping systems could produce accurate maps using the Sick LMS 200 laser scanner. Unfortunately hardware problems were encountered with the 3D mapping system. These limited the operation of the UGV to flat ground when using this system. Two pose sensors were trialled on the ARP vehicle but neither was good enough to provide accurate pose information in light of the vibration of the UGV. Also it was found that the PMD [vision]® 1k-S range camera did not work well outdoors despite the claims of the manufacturer. The image sensor within the camera was found to become easily saturated by natural light.

It was then shown that the basic path planner was capable of avoiding sparse obstacles but for more complicated or narrow obstacle courses a shorter planner lookahead was required. To show that this problem could be overcome the adjustable lookahead path planner was demonstrated alongside the effects of the claustrophobia setting. With a high claustrophobia setting the UGV had a tendency to try and bypass obstacle courses. In contrast with a low claustrophobia setting the UGV would always attempt to negotiate complex obstacle courses that lay on the ideal heading.

The final series of experiments evaluated the cul-de-sac escape behaviour provided by the exploration graph concept. It was shown that when used with obstacle flooding the UGV could successfully use the exploration graph to escape from a cul-de-sac. In the very last experiments it was demonstrated that using the minimax planner lookahead guaranteed that the path planner never missed alternative paths when escaping. It was shown that without the minimax lookahead the adjustable lookahead could drop too low so the path planner could simply not find a viable alternate path to escape the cul-de-sac.

9.2 CONTRIBUTIONS

The contributions of the research are detailed in this section. All of the research work conducted in this project was original. NOVA does not contain any components based on designs that have been copied straight from published materials. Everything reported in this thesis was developed specifically for this research.

The first major contribution of this research has to be the construction of the ARP vehicle. This UGV was developed with an autonomous route proving application in mind. To that end the ARP vehicle is a complete UGV with a suite of localisation & perception equipment. It has been successfully used to validate the features of NOVA the associated UGV control architecture. Now it is available as a platform for continued research.

During this research a complete UGV control architecture was developed. Thus the second contribution of this research must be the development of NOVA. NOVA provides all the necessary components for control of an autonomous route proving vehicle. This includes systems for UGV localisation, obstacle mapping and autonomous navigation with obstacle avoidance. Within NOVA there are a number of truly novel aspects.

One of these novel aspects is at the heart of the UGV localisation system. NOVA uses a particle filter to fuse location data from wheel encoders, a pose sensor and a GPS receiver. Experiments in this thesis demonstrate that the particle filter location estimates are resistant to the errors associated with the underlying equipment. The particle filter provides absolute position estimates with higher accuracy than GPS alone and without the accumulation of error associated with the dead reckoning system.

Another novel part of the research is the adjustable path planner lookahead. It is typical for the lookahead of a path planner to be tuned to the environment it is going to be operating within. The method developed in this research adjusts the planner lookahead according to the local obstacle density. This density is measured by the proportion of blocked radial swaths that are found when searching for paths across the obstacle map. In more cluttered areas the planner will reduce the lookahead in an attempt to find paths that lead between obstacles. A control for the adjustable lookahead method has been developed that specifies how quickly the lookahead should fall in the presence of obstacles. This control defines how claustrophobic the UGV will behave.

The concept of an exploration graph was also introduced in this research. An exploration graph is built from a series of nodes that specify the decisions made by the path planner and the location of UGV at the time. If the UGV encounters a cul-de-sac it can use the exploration graph to escape. This is done by reversing the UGV through the nodes of the exploration

graph in search of an unexplored alternate path. When an unexplored path is found the path planner will guide the UGV along this alternate path.

Another novel concept devised in this research is that of obstacle flooding. This was developed to aid the use of exploration graphs. When the path planner is escaping from a cul-de-sac the obstacle map is flooded with false obstacles. The flooding is positioned so that it blocks the path into the cul-de-sac that the UGV has just driven along. This prevents the UGV from ever venturing back into the cul-de-sac.

The final novel development in this research was that of the minimax lookahead. This was another idea to complement the exploration graph concept. It was shown that when using the adjustable lookahead it was possible that the path planner would not see all of the alternate paths at a particular location. In response an algorithm was devised to find the minimum lookahead that saw the maximum number of paths or minimax lookahead for short. Use of the minimax lookahead with the adjustable lookahead system ensured that during the experiments the UGV always escaped from a cul-de-sac using the first alternate path.

9.3 FURTHER WORK

Given the opportunity there are many improvements that could be made to the ARP vehicle and NOVA. This research should be considered a foundation as it provides the building blocks for a great deal of other work. The following subsections suggest possible improvements for the ARP vehicle and extensions for NOVA. A number of new research projects could stem from these ideas.

More experimentation could also be done with the current implementation of NOVA and the ARP vehicle. There would be much to learn from testing in a large natural area. Also a period

of consultation and evaluation with potential users of this type of technology would be very informative.

9.3.1 Improvements for the ARP vehicle

An obvious first improvement for the ARP vehicle would be to fit the vehicle with specialist equipment for the autonomous route proving task. A lane marking system would be required. This could be implemented using a tank of coloured dye and a pair of nozzles that periodically spray spots on the ground. Additionally a metal detector or ground penetrating radar could be mounted at the front of the UGV. As potential landmines were detected by the sensor their location could be marked using a different coloured dye spot. Something similar was constructed in another project [Walton 2006].

In light of the hardware problems encountered when testing the 3D obstacle mapping system a new pose sensor could be procured for the ARP vehicle. A multisensory solution using accelerometers, gyroscopes and a compass in an integrated package would be best. This kind of inertial navigation system would provide compensation for errors in the measurements from one sensor type using the other sensors. A measurement accuracy of better than 1° for heading and 3° for pitch and roll would be required for accurate obstacle mapping.

Work with the 3D mapping system could be continued if a better pose sensor was fitted to the ARP vehicle. It is hoped that a 3D confidence mapping system could ultimately be developed for NOVA. In order to do this well the ARP vehicle should also be equipped with a true 3D perception sensor. This could be done by mounting the existing Sick LMS 200 laser scanner on a spinning or tilting mechanism as has been done in other research [Wulf 2003]. Alternatively if funds were available either a two axis laser scanner or a different range camera could be trialled.

It would be really useful if a whole range of different sensors were evaluated and compared on a UGV. At present researchers only have manufacturer claims to go on. That does not mean that the sensor will work well in the intended environment as was seen with the PMD [vision]® 1k-S range camera and the Honeywell HMR300 pose sensor in this research. The ARP vehicle would make a good platform for UGV sensor evaluation. Also NOVA has been developed in a way that would allow new sensors to be trialled with only minor software modifications required.

9.3.2 Extensions for NOVA

Proposed extensions for NOVA can be split into two categories. The first category is related to the maps that are used by the UGV. During this research there was no need to use geographical maps to allow waypoints for the UGV to be specified. This was simply because the test site was quite small. However to allow the ARP vehicle to be used over larger areas NOVA would need to be integrated with a global mapping system. When the UGV acquired an absolute position fix the back drop of the NOVA map tab could adjust to show a geographical map centred on the UGV.

Another extension would be to develop a full mission mapping capacity. Mission maps offer much more information about the environment. This information can be stored for when the UGV returns or transmitted to following vehicles. Experiments in this research showed that it was viable to build large obstacle maps. The next step would be to research methods to map a number of square kilometres. This could use an obstacle map composed of small tiles held in a larger structure.

For autonomous route proving it is important to know exactly which ground has been covered by mine detection or clearance equipment attached to the UGV. So NOVA could also be extended to build a statistical model of the ground coverage. This would be integrated with a

mapping system to show the area covered at a particular confidence level. NOVA could then ensure that all ground was covered with a specified confidence by directing the UGV to perform additional proving of the low confidence areas.

Other proposed extensions for NOVA are related to multi vehicle control. The ARP vehicle operates largely autonomously once the initial waypoint route has been specified. Hence another control layer could be developed for the operator base station that facilitates control of multiple ARP vehicles. The new operator layer would allow telemetry from each vehicle to be monitored. It could also act as a conduit for sharing information between the vehicles such as obstacle map data or proven routes through the environment.

The next step with a multi vehicle system would be task sharing. Consider a situation where a large area of ground needed to be proven. The area coverage task could be divided between several vehicles. Each vehicle would get a separate territory. Communication throughout task could be used to establish if all vehicles were able to maintain their commitments. If a vehicle failed the territories could be dynamically resized to ensure complete coverage.

Finally the idea of robot convoys with either a manned or autonomous lead vehicle gives another possible extension. The lead vehicle would record the route that was followed. This is already done within NOVA. The route could then be transmitted to following vehicles to use. Rules would need to be created to ensure the vehicles in the convoy maintained formation.

REFERENCES

[Abouaf 1998] Abouaf, J., "Trial by Fire: Teleoperated Robot Targets Chernobyl", IEEE Computer Graphics and Applications, Vol. 18, Issue 4, July / August 1998, pages 10-14.

[Apostolopoulos 2000] Apostolopoulos, D.S. et al, "Technology and Field Demonstration of Robotic Search for Antarctic Meteorites", International Journal or Robotics Research, Vol. 19, No. 11, 2000, pages 1015-1032.

[Arkin 1998] Arkin, R.C., "Behaviour-Based Robotics", The MIT Press, 1998.

[Arulampalam 2002] Arulampalam, S. et al, "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking", IEEE Transactions on Signal Processing, Vol. 50, No. 2, 2002, pages 174-188.

[Bares 1999] Bares, J. & Wettergreen, D., "Dante II: Technical Description, Results and Lessons Learned", International Journal of Robotics Research, Vol. 18, Issue 7, July 1999, pages 621-649.

[Batavia 2002] Batavia, P.H., Roth, S.A. & Singh, S., "Autonomous Coverage Operations in Semi-Structured Outdoor Environments", Proceedings of the 2002 IEEE/RSJ

International Conference on Robots and Systems, IROS 2002, Lausanne, Switzerland, October 2002.

[Bewsher 2001] Bewsher, G. & Newnham, P., "An Industrial Perspective of the United Kingdom's Mine Detection, Neutralisation and Route Marking System", Proceedings of SPIE Vol. 4394: Detection and Remediation Technologies for Mines and Minelike Targets VI, 2001.

[BGS 2008] "Grid Magnetic Angle Calculator", British Geological Survey, retrieved 6th February 2008 from: http://www.geomag.bgs.ac.uk/gifs/gma_calc.html

[Borenstein 1991] Borenstein, J. & Koren, Y., "The Vector Field Histogram – Fast Obstacle-Avoidance for Mobile Robots", IEEE Transactions on Robotics and Automation, Vol. 7, No. 3, 1991, pages 278-288.

[Borenstein 1994] Borenstein, J. & Feng, L., "UMBmark – A method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots", University of Michigan, 1994.

[Borenstein 1996] Borenstein, J., edited, "Where Am I? Sensors and Methods for Mobile Robot Positioning", University of Michigan, 1996.

[Bostelman 2001] Bostelman, R., Juberts, M., Sandor, S., Bunch, R. & Evans, J., "Industrial Autonomous Vehicle Project Report",

Intelligent Systems Division, National Institute of Standards and Technology, 7 June 2001.

[Bostelman 2005] Bostelman, R., Hong, T.H. & Madhavan, R., "Towards AGV Safety and Navigation Advancement - Obstacle Detection using a TOF Range Camera," Proceedings of the International Conference on Advanced robotics (ICAR) 2005.

[Braid 2006] Braid, D., Broggi, A., & Schmiedel, G., "The TerraMax Autonomous Vehicle concludes the 2005 DARPA Grand Challenge", Proceedings of the Intelligent Vehicles Symposium, Tokyo, Japan, 13-15 June 2006.

[Breitenbach 2000] Breitenbach, C., "The design and implementation of an interactive user interface for mobile vehicles", MEng Dissertation, School of Mechanical, Materials, Manufacturing Engineering & Management, University of Nottingham, 2000.

[Breithaupt 1999] Breithaupt, J., "Physics", Macmillan, 1999, pages 237 - 250.

[Brenneke 2003] Brenneke, C., Wulf, O. & Wagner, B., "Using 3D Laser Range Data for SLAM in Outdoor Environments", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA. 27-31 October, 2003.

[Brooks 1986] Brooks, R., "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, Vol. 2, Issue 1, 1986, pages 14-23.

[Burgard 1996] Burgard, W., Fox, D., Hennig, D. & Schmidt, T., "Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids", Proceedings of the 14th National Conference on Artificial Intelligence, Menlo Park, August 1996.

[Carroll 2005] Carroll, D.M. et al., "Development and Testing for Physical Security Robots", SPIE Proceedings Vol. 5804: Unmanned Ground Vehicle Technology VII, Orlando FL, 29-31 March, 2005.

[Caruso 1999] Caruso, M.J. & Withanawasam, L., "Vehicle Detection and Compass Applications using AMR Magnetic Sensors", Solid State Electronics Center, Honeywell, 1999.

[Casper 2003] Casper, J. & Murphy, R.R., "Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center", IEEE Transactions on Systems, Man and Cybernetics Part B, Vol. 33, Issue 3, 2003, pages 367-385.

[Ceranka 2003] Ceranka, S. & Niedźwiecki, M., "Application of Particle Filtering in Navigation System for Blind", Proceedings of

IEEE Seventh International Symposium on Signal Processing and its Applications, Vol. 2, July 2003, pages 495-498.

[Chang 1999] Chang, T. et al, "Concealment and Obstacle Detection for Autonomous Driving", Proceedings of IASTED International Conference on Robotics and Applications, Santa Barbara, CA, October 1999.

[Colon 2002] Colon, E. et al, "An integrated robotics system for antipersonnel mines detection", Control Engineering Practice, Vol. 10, No. 11, 2002, pages 1283-1291.

[Coombs 2000] Coombs, D. et al, "Driving Autonomously Offroad up to 35 km/h", Proceedings of IEEE Intelligent Vehicles Symposium, Dearborn, MI, 4-5 October 2000.

[Corkill 1991] Corkill, D.D., "Blackboard Systems", AI Expert, Vol. 6, No. 9, September, 1991, pages 40-47.

[CSI 2005] **"**Mini MAX Reference Manual", Part Number 875-0124-001, CSI Wireless Inc., January 2005.

[Dae-Sung 2005] Dae-Sung, S. et al, "A Probabilistic Approach for Mobile Robot Localization under RFID Tag Infrastructures", International Conference on Control, Automation, and Systems, KINTEX, Goyang, Korea, 2005, pages 1797-1801.

[Daily 1987] Daily, M.J., Harris, J.G. & Reiser, K., "Detecting Obstacles in Range Imagery", Proceedings of the DARPA Image Understanding Workshop, Vol. 1, February 1987, pages 87-97.

[Daily 1988] Daily, M. et al, "Autonomous Cross-Country Navigation with the ALV", Proceedings of IEEE Conference on Robotics and Automation, Philadelphia, PA, April 1988, pages 718-726.

[Dellaert 1999] Dellaert, F. et al., "Using the Condensation Algorithm for Robust, Vision-based Mobile Robot Localization", Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, 1999.

[Digimap 2008] "Ordnance Survey 1:10000 Scale Raster Map Tile TF11SW", EDINA Digimap Collections, retrieved 18th February 2008 from: http://edina.ac.uk/digimap

[Dudek 2000] Dudek, G. & Jenkin, M., "Computational Principles of Mobile Robotics", Cambridge University Press, 2000.

[Dunlay 1987] Dunlay, R.T., Hennessy, S.J. & Morgenthaler, D.G., "Obstacle Avoidance for the Autonomous Land Vehicle: Perception", Unmanned Systems, Vol. 6, Summer 1987, pages 36-45.

[Einramhof 2007] Einramhof, P., Olufs, S. & Vincze, M., "Experimental Evaluation of State of the Art 3D-Sensors for Mobile Robot Navigation", Proceedings of the 31st Annual Workshop of the Austrian Association for Pattern Recognition (OAGM 07), Krumbach, May 2007.

[Elfes 1989] Elfes, A., "Using Occupancy Grids for Mobile Robot Perception and Navigation", IEEE Computer Magazine, Vol. 22 No. 6, June 1989, pages 46-57.

[ESA 2008a] "Satellite navigation today", European Space Agency, retrieved 2nd February 2008 from: http://www.esa.int/esaNA/ESA91YZK0TC_index_0.html

[ESA 2008b] "How does EGNOS work?", European Space Agency, retrieved 30th January 2008 from: http://www.esa.int/esaNA/GGGQI950NDC_egnos_2.html

[Fayad 2001] Fayad, C., "Development of techniques for global / local path planning of an autonomous mobile robot in dynamic environments", PhD Thesis, School of Mechanical, Materials, Manufacturing Engineering & Management, University of Nottingham, 2001.

[Firschein 1997] Firschein, O. & Strat, T.M., eds., "Reconnaissance, Surveillance, and Target Acquisition for the Unmanned

Ground Vehicle: Providing Surveillance 'Eyes' for an Autonomous Vehicle", Morgan Kaufman, 1997.

[Gage 1995] Gage, D.W., "UGV History 101: A Brief History of Unmanned Ground Vehicle (UGV) Development Efforts", Unmanned Systems Magazine, Vol. 13, No. 3, Summer 1995.

[Garnett 2008] Garnett, S., Corporate Services West Yorkshire Fire & Rescue, e-mail communication on 17 January 2008.

[Gordon 2005] Gordon, E., "GRUE: An Architecture for Agents in Games and Other Real-Time Environments", PhD Thesis, School of Computer Science, University of Nottingham, 2005.

[Hamner 2006] Hamner, B., Singh, S. & Scherer, S., "Learning Obstacle Avoidance Parameters from Operator Behaviour", Journal of Field Robotics, Vol. 23, No. 11/12, 2006, pages 1037-1058.

[Harmon 1987] Harmon, S.Y., "The Ground Surveillance Robot (GSR): An Autonomous Vehicle Designed to Transit Unknown Terrain", IEEE Journal of Robotics and Automation, Vol. RA-3, No. 3, June 1987, pages 266-279.

[Havlík 1998] Havlík, Š., "Robotic systems for humanitarian demining operations", Advanced Robotics: Beyond 2000, The 29th International Symposium on Robotics, N.E.C., Birmingham, UK, 27 April – 1 May 1998.

[Hebert 1988] Hebert, M. & Kanade, T., "3-D Vision for Outdoor Navigation by an Autonomous Vehicle", Proceedings of the DARPA Image Understanding Workshop, April 1988, pages 593-601.

[Hebert 1992] Hebert, M. & Krotkov, E., "3D measurements from imaging laser radars: how good are they?", International Journal of Image and Vision Computing, Vol. 10, No. 3, April 1992, pages 170-178.

[Honeywell 1999] "HMR3000 Digital Compass Module", Datasheet, Solid State Electronics Center, Honeywell, December 1999.

[Honeywell 2005] "HMR3000 Digital Compass Solution User's Guide", Aerospace Electronics Systems, Honeywell, 2005.

[Hong 2002] Hong, T. et al., "A Hierarchical World Model for an Autonomous Scout Vehicle", Proceedings of SPIE Vol. 4715 Aerosense Conference, Orlando, Florida, 1-5 April, 2002, pages 343-354.

[Hu 2000] Hu, H. & Gu, D., "Landmark-based Navigation of Industrial Mobile Robots", International Journal of Industry Robot, Vol. 27, No. 6, 2000, pages 458-467.

[Iovine 1998] Iovine, J., "Robots, Androids, and Animatrons", McGraw-Hill, 1998, pages 4-7.

[iRobot 2007] **"**iRobot Roomba Vacuum Cleaning Robot 5th Generation: 500 Series Owners Manual", iRobot Corporation, 2007.

[Johnson 2007] Johnson, R.A. & Wichern, D.W., "Applied Multivariate Statistical Analysis", 6th Edition, Pearson Prentice Hall, 2007.

[Jong 2002] Jong, C.D. et al, "Hydrography", Series on Mathematical Geodesy and Positioning, Delft University Press, 2002.

[Kelly 1994] Kelly, A.J., "Essential Kinematics for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-14, 1994.

[Khatib 1986] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", International Journal Robotics Research, Volume 5 No. 1, 1986, pages 90–98.

[Knuth 1998] Knuth, D.E., "The Art of Computer Programming: Volume 2 Seminumerical Algorithms", 3rd Edition, Addison-Wesley, 1998, pages 122-123.

[Koren 1991] Koren, Y. & Borenstein, J., "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation", Proceedings of the IEEE Conference on Robotics and Automation, Sacremento, CA, 7-12 April 1991, pages 1398-1404.

[Krotkov 1989] Krotkov, E., "Mobile Robot Localization Using A Single Image", Proceedings of the 1989 IEEE International Conference on Robotics and Automation, Vol. 2, 14-19 May 1989, pages 978-983.

[Krutz 1997] Kurtz, J., Frederick, B. & Neubert, J., "Obstacle Avoidance Technologies for Robotic Platforms", Proceedings of the Association for Unmanned Vehicle Systems International Conference AUVSI 97, 1997, pages 374-382.

[Langer 1994] Langer, D., Rosenblatt, J. K. & Hebert, M., "A Behaviour-Based System for Off Road Navigation", IEEE Journal of Robotics and Automation, Vol. 10, 1994, pages 776-782.

[Langer 1997] Langer, D. & Thorpe, C.E., "Sonar-Based Outdoor Vehicle Navigation", Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon, Kluwer Academic Publishers, 1997, pages 159-185.

[Lopez 2007a] Lopez, R., "The Hunt for Improvised Explosive Devices", AUVSI Unmanned Systems, Vol. 25, No. 4, July / August 2007, pages 35-38.

[Lopez 2007b] Lopez, R., "Unmanned Systems at the Cutting Edge of Scientific Research", AUVSI Unmanned Systems, Vol. 25, No. 6, November / December 2007, pages 19-21.

[Luger 1999] Luger, G.F. & Stubblefield, W.A., "Artificial Intelligence – Structures and Strategies for Complex Problem Solving", 3rd Edition, Addison-Wesley, 1999, pages 99-106.

[Lux 1991] Lux, P.W. & Schaefer, C.H., "Range Imaging for Autonomous Navigation of Robotic Land Vehicles", Signal Processing, Vol. 22, 1991, pages 299-311.

[Madsen 1998] Madsen, C. & Andersen, C., "Optimal Landmark Selection for Triangulation of Robot Position", Journal of Robotics and Autonomous Systems, Vol. 13, No. 4, 1998, pages 277-292.

[Mallinson 1996] Mallinson, P. & Daniels, D.J., "Impulse Radar Mine Detection", Proceedings of IEE Conference on Detection of Abandoned Land Mines, Edinburgh, UK, 7-9 October, 1996.

[Matthies 1995] Matthies, L. et al, "Obstacle Detection for Unmanned Ground Vehicles: A Progress Report", Proceedings of IEEE Intelligent Vehicles Conference, September 1995.

[Meier 1998] Meier, E.B. & Ade, F., "Detection and Tracking in Range Image Sequences by Separation of Image Features", Proceedings of IEEE International Conference on Intelligent Vehicles (IV'98), Stuttgart, Germany, 28-30 October 1998, pages 176-181.

[Möller 2005] Möller, T. et al, "Robust 3D Measurement with PMD sensors", Proceedings of the First Range Imaging Research Day at ETH Zurich, 2005.

[Moorehead 1999] Moorehead, S. et al, "Autonomous Navigation Field Results of a Planetary Analog Robot in Antarctica", International Symposium on Artificial Intelligence, Robotics and Automation in Space, Noordwijk, Holland, 1999.

[Moravec 1985] Moravec, H. & Elfes, A.E., "High Resolution Maps from Wide Angle Sonar", Proceedings of the IEEE International Conference on Robotics and Automation, March, 1985, pages 116-121.

[Murphy 1993] Murphy, K.N. et al, "Ground Vehicle Control at NIST: from Teleoperation to Autonomy", Proceedings of the Seventh Annual Space Operations, Applications and Research Symposium, Houston, TX, 3-5 August, 1993.

[Murphy 1996] Murphy, K. & Legowik, S., "GPS Aided Retrotraverse For Unmanned Ground Vehicles", Proceedings of the SPIE 10th Annual AeroSense Symposium, Conference 2738, Navigation and Control Technologies for Unmanned Systems, Orlando, FL, April 1996.

[Navarro-Serment 1999] Navarro-Serment, C.J.J, Paredis, P.K. & Khosla, P.K., "A Beacon System for the Localization of Distributed Robotic

Teams", Proceedings of the International Conference on Field and Service Robots, Pittsburgh, PA, 29-31 August, 1999.

[NRC 2002] National Research Council Committee on Army Unmanned Ground Vehicle Technology, "Technology Development for Army Unmanned Ground Vehicles", National Academic Press, 2002.

[Oggier 2004] Oggier, T., Seitz, P. & Blanc, N., "Miniaturized all-solid-state 3D camera for real-time range imaging", Proceedings of Performance Metrics for Intelligent Systems (PerMIS '04), Gaithersburg, Maryland. August 2004.

[Ojeda 2000] Ojeda, L. & Borenstein, J., "Experiments with the KVH C-100 Fluxgate Compass in Mobile Robots", Proceedings of the IASTED International Conference on Robotics and Applications 2000, Honolulu, Hawaii, 14-16 August, 2000.

[Olin 1991] Olin, K.E. & Tseng, D.Y., "Autonomous Cross-Country Navigation: An integrated Perception and Planning System", IEEE Expert, Vol. 6, Issue 4, August 1991, pages 16-30.

[OS 2005] "Transformations and OSGM02 User Guide", Ordnance Survey, January 2005.

[OS 2007] "A guide to coordinate systems in Great Britain", Ordnance Survey, December 2007.

[Oskard 1990] Oskard, D.N., Hong, T. & Shaffer, C.A., "Real-Time Algorithms and Data Structures for Underwater Mapping", IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, No. 6, 1990, pages 1469-1475.

[Oudijk 1997] Oudijk, R., "The development of an intuitive user interface for telepresence operation of remote control vehicles", MEng Dissertation, Department of Manufacturing Engineering and Operations Management, University of Nottingham, 1997.

[Owens 2004] Owens, K. & Burgess, P., "Semi-Autonomous Landmine Clearing and Task Mapping: System Design and Experimental Results", Proceedings of the International Conference on Computing, Control and Communications Technologies, 2004.

[Perry 2008] Perry, C., "Teams Ponder Lessons from 2007 Urban Challenge: Winners, Losers Both Have Lots to Learn", AUVSI Unmanned Systems, Vol. 26, No. 1, January 2008.

[Pilarski 1999] Pilarski, T. et al, "The Demeter System for Automated Harvesting", Proceedings of the 8th International Topical Meeting on Robotics and Remote Systems, April 1999.

[Planas 2004] Planas, M., "A Short Guide to GPS", Souterrain Archaeological Services Ltd, 2004.

[PMD 2005] "PMD [vision]® 1k-S 3D Video Range Camera", Datasheet, PMD Technologies GmbH, 2005.

[Pook 1998] Pook, K.P. et al., "A Testbed for Evaluating Robot Team Interaction under Operator Supervision", Proceedings of the UXO Forum 98, Anaheim, CA. 5-7 May 1998.

[Rice 1995] Rice, J.A., "Mathematical Statistics and Data Analysis", 2nd Edition, Duxbury Press, 1995.

[Ryder 2006] Ryder, C., "A Special Kind of Courage 321 EOD Squadron – Battling the Bombers", Revised Edition, Methuen, 2006.

[Schaub 2004] Schaub, G., Pfaendner, A. & Schaefer, C., "PRIMUS: Autonomous Navigation in Open Terrain with a Tracked Vehicle", Proceedings of SPIE Vol. 5422: Unmanned Ground Vehicle Technology VI, Orlando, Florida, USA, 13-15 April, 2004, pages 156-165.

[Schwartz 2000] Schwartz, I., "PRIMUS Autonomous Driving Robot for Military Applications", Proceedings of SPIE Vol. 4024: Unmanned Ground Vehicle Technology II, Orlando, Florida, USA, 24-25 April, 2000, pages 313-323.

[Shoemaker 1998] Shoemaker, C.M. & Bornstein, J.A., "The Demo III UGV Program: A Testbed for Autonomous Navigation Research",

Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference, Gaithersburg, MD, 14-17 September, 1998.

[Shoop 2006] Shoop, B. et al., "Mobile Detection Assessment and Response Systems (MDARS) A Force Protection, Physical Security Operational Success", Proceedings of SPIE Vol. 6230: Unmanned Systems Technology VIII, Defense and Security Symposium, Orlando, FL, 17-30 April, 2006

[Simmons 1997] Simmons, R. et al, "Xavier: Experience with a Layered Robot Architecture", ACM SIGART Bulletin, Vol. 8, Issue 1-4, 1997.

[Smith 1997] Smith, S.W., "The Scientist and Engineer's Guide to Digital Signal Processing", California Technical Publishing, 1997.

[Sobottka 1998] Sobottka, K. & Bunke, H., "Obstacle Detection in Range Image Sequences Using Radial Slope", Proceedings of 3rd IFAC Symposium on Intelligent Autonomous Vehicles (IAV'98), Madrid, Spain, 25-27 March 1998, pages 535-540.

[Stentz 1995] Stentz, A. & Hebert, M., "A Complete Navigation System for Goal Acquisition in Unknown Environments", Autonomous Robots, Vol. 2, No. 2, August 1995.

[Talukder 2002] Talukder, A. et al, "Fast and Reliable Obstacle Detection and Segmentation for Cross-Country Navigation", IEEE Intelligent Vehicles Symposium, Versailles, France, 2002.

[Thurn 2005] Thurn, S., Burgard, W. & Fox, D., "Probabilistic Robotics", The MIT Press, 2005.

[Thurn 2006] Thurn, S., et al, "Stanley: The Robot That Won The DARPA Grand Challenge", in Journal of Field Robotics, Vol. 23, No. 9, September 2006, pages 661-692.

[Ulrich 1998] Ulrich, I. & Borenstein, J., "VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots", Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belguim, 16-21 May 1998, pages 1572-1577.

[Urmson 2006] Urmson, C. et al, "A Robust Approach to High-Speed Navigation for Unrehearsed Desert Terrain", Journal of Field Robotics, Vol. 23, No. 8, 2006, pages 467-508.

[Vishay 2003] "601-1045 Full 360° Smart Position Sensor", Datasheet, Vishay, July 2003.

[Walton 2006] Walton, M. et al., "Developing an Intelligent and Integrated Unmanned Ground Vehicle System: A Case Study", AUVSI Unmanned Systems North America, Orlando, FL, 29-31 August, 2006.

[Webb 2000] Webb, P., Fayad, C. & Breitenbach, C., "The Integration of an Optimised Fuzzy logic Navigation Algorithm into a Semi-Autonomous Robot Control System", Proceedings of the International Workshop on Recent Advances in Mobile Robots, Leicester, UK, 29 June 2000.

[Weingarten 2004] Weingarten, J., Gruener, G. & Siegwart, R., "A State-of-the-Art 3D Sensor for Robot Navigation", Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan. Sept-Oct 2004.

[Welch 2006] Welch, G. & Bishop, G., "An Introduction to the Kalman Filter", Technical Report TR 95-041, Department of Computer Science, University of North Carolina, 24 July, 2006.

[Withanawasam 1998] Withanawasam, L., "Getting Stable Output from HMR3000 Compass Module", Application Note AN-208, Solid State Electronics Center, Honeywell, 1998.

[Woodman 2007] Woodman, O.J., "An Introduction to Inertial Navigation", Technical Report 696, Cambridge University Computer Laboratory, 2007.

[Wulf 2003] Wulf, O. & Wagner, B., "Fast 3D-Scanning Methods for Laser Measurement Systems", Proceedings of 14th

International Conference on Control Systems and Computer Science (CSCS14), Bucharest, Romania, 2-5 July 2003.

[Yamauchi 2005] Yamauchi, B., "Wayfarer: An Autonomous Navigation Payload for the PackBot", Proceedings of AUVSI Unmanned Vehicles North America, Baltimore, Maryland, June 2005.

[Ye 2002] Ye, C. & Borenstein, J., "Characterization of a 2D Laser Scanner for Mobile Robot Obstacle Negotiation", Proceedings of IEEE International Conference on Robotics and Automation, Washington DC, 10-17 May 2002, pages 2512-2518.

[Ye 2003] Ye, C. & Borenstein, J., "A New Terrain Mapping Method for Mobile Robots Obstacle Negotiation", Proceedings of SPIE Vol. 5083: Unmanned Ground Vehicle Technology V, September 2003, pages 52-62.

[Ye 2004] Ye, C. & Borenstein, J., "A Method for Mobile Robot Navigation on Rough Terrain", Proceedings of IEEE International Conference on Robotics and Automation, New Orleans, LA, 26 April – 1 May, 2004, pages 3863-3869.

[Ye 2006] Ye, C., "An Optimal Reactive Navigation Approach for Mobile Robots in Unknown Environments", PhD Thesis, School of Mechanical, Materials & Manufacturing Engineering, University of Nottingham, 2006.