Autonomous map building in outdoor environments

Niklas Wedin

Master Thesis
Centre for Autonomous Systems
Royal Institute of Technology

ABSTRACT

This report gives an analysis of the problem of building a map of an outdoor environment using an intelligent vehicle, also known as an autonomous system. Different characteristic parts of an autonomous system are described with a discussion of what kind of equipment that is necessary for map building. An odometry sensor and a laser range finder are selected as the primary sensors to build the map. GPS are used to give absolute position estimation. Two different kinds of map representations are covered. These two are created by either the feature based mapping method or by the grid based method. The grid based method is chosen as the method used for our solution. A simple implementation of this method is done and this illustrates that it is possible to create a map of an unknown environment but that the global resolution of the map depends on the accuracy of the position estimation of the GPS. This report also covers different circumstances and characteristics that are found in indoor respectively outdoor environment and what problems these could cause for map building.

Autonom kartframställning i en utomhusmiljö

Sammanfattning

Denna rapport är en studie av problemet med att skapa en karta av en utomhusomgivning med hjälp av ett självgående intelligent fordon, även kallat ett autonomt system. De delar som karaktäriserar ett autonomt system diskuteras. Dessutom diskuteras hur ett fordon bör utrustas om det skall kunna generera en karta av omgivningen. En odometrisensor samt laseravståndsmätning väljs som de huvudsakliga sensorer som skall användas för att mäta upp kartan. Försök med GPS har utförts för att ge fordonets absoluta position. Två olika karttyper som grund för metoden gås igenom, dessa skapas med en featurebaserad metod respektive en rutnätsbaserad metod. Den rutnätsbaserade metoden väljs som grund för lösningen. En enkel tillämpning med denna metod genomförs och visar att det är möjligt att skapa en karta av en omgivning men att den globala noggrannheten begränsas av den absoluta positionsangivelse vi kan få från GPS-systemet. I rapporten diskuteras även olika förutsättningar som skiljer utomhus- och inomhusmiljöer samt vilka svårigheter dessa olikheter kan vålla.

TABLE OF CONTENTS

PREFACE

This project was done as a Master of Science project for the Center of Autonomous Systems at the Royal Institute of Technology, Stockholm. This report could be used as an introduction for people that are involved with autonomous systems and that are interested in building autonomous maps. Among companies that are interested, the Swedish FMV can be mentioned. Also, great thanks to them for providing us with necessary equipment and material for this project. This report is not to be considered as a complete guide to the solution to the map building problem, which is a problem too big to handle in a master thesis. This report may however serve as an introduction to the problem and will discuss some aspects that need to be considered if an autonomous map system is of interest.

I would also like to thank several persons for supporting me in this work in different ways.

Mathias – for help with necessary software for GPS, but most of all for good company in the otherwise isolated “ExJobb” room.
Patric – for much help and support on feature based mapping.
Guido – for assisting and giving me a good start on some mathematical transformations.
John – for company and interchange of ideas “on-the-field”.
Gunnar – for pushing me on.
Carl – for cheering me up when things felt hard.
and of course *Henrik*, for assigning me with this project

1 INTRODUCTION

1.1 The problem

The issue studied in this project is the problem of autonomous construction of a map of an unknown environment. The problem involves selection of a suitable suite of sensors, selection of methods for map acquisition implementation of these on an operational platform, and evaluation of the methods under realistic conditions. It is of interest to discuss the different environmental characteristics that could be used to build the map and sometimes cause trouble when building a map. The map is in this case meant to be the layout of an outdoor environment that could primarily be used by humans to gain an overview of the otherwise unknown area. There are also other uses for a solution of this kind and this is discussed later on in this introduction. The hardware that is used in this report is a robot vehicle that is equipped for the task; figure 1.1 shows an image of this vehicle. Robot vehicles of this kind are also known as autonomous vehicles or autonomous systems.

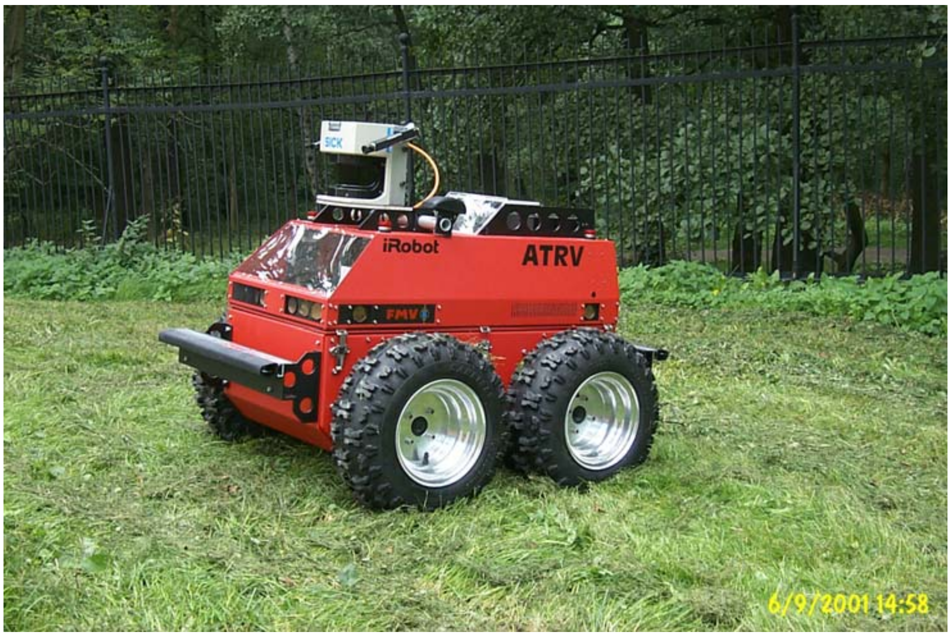


Figure 1.1 shows the vehicle used in this project is equipped for autonomous tasks in outdoor environments.

1.2 Autonomous systems

An autonomous system is considered as a system, which acts and interacts with the environment without the need for continuous guidance from a person. Often these systems are used when easy tasks should be performed but it is difficult or impossible to give the system continuous guidance. This could be a task such as moving from one point to another over a space where obstacles are blocking the straightest path. To be able to do this, a system is needed that has the capability to sense the environment and also move around and interact with the environment. This requires some intelligence to handle the inputs in a rational way so that the movements can be performed in order to reach the goal. The autonomous system used in this report is an ATRV (Autonomous Terrain Rover Vehicle). The autonomous system is mentioned short as the ATRV or the robot. A robot (or autonomous system) normally consists of a number of sensors to sense the environment, one or more actuators to interact with the environment and also one or more computers which control the system or is the "intelligence" of the system. The current setup for our system is given in Appendix. Further discussion concerning sensors for robots can be read in Johann Borensteins work [Borenstein, 1996]

Different actions performed by a robot could be considered as different behavior, where a more complex action or behavior could consist of a combination of various kinds of simpler behavior. Lately animal-behavior has been studied to gain knowledge from this field that could be used when programming robot behavior [Arkin, 1998]. As the hunt behavior of a predator can be seen as a series of chasing-biting-kicking behavior, the more complex behavior used in our solution can be broken down into smaller, simpler parts. This robot has a follow-me behavior that is active at the same time as a collect behavior. This makes the ATRV follow an object in front of it and also gather information about its surroundings.

An autonomous system that should be used for map building or navigation needs to have a way to detect its surroundings and also a way to move around. This is discussed in the section "Necessary properties of a map building system".

1.3 The project

The project is divided into a practical part and a theoretical part. The first part is to setup the hardware and software of the shipped robot to be an autonomous system that could be used for development of solutions of this and other tasks. This involves setting up all necessary software and also connecting some additional sensors. The additional sensors that are connected are one laser range measuring sensor[1] and an inertial sensor. Included in this part is also the primary testing of all necessary sensors. Most details concerning this part are mentioned in Appendix.

The second and more theoretical part of this project is to analyze the problem of developing a map of an unknown environment by using the equipped robot system. Some research has been done earlier to use an autonomous system to create a map, but mostly for indoor environments. The reviewed papers that describe maps in outdoor environment rely primarily on artificial features such as the radar reflecting landmarks used by Dissanayake, Newman, Clark, Durrant-Whyte and Csorba [Dissanayake, Newman, Clark, Durrant-Whyte and Csorba, 2001] or on simple landmarks such as only point features such as trees as used by Guivant&Nebot [Guivant&Nebot, 2001]. Experience from the research indoor has therefore been used as a starting point for this project, but the features that differentiate the indoor from the outdoor environments need to be discussed. This is done in the next chapter. In this project a vehicle equipped with odometry sensors, laser range finder and a GPS is used to extract information from the surrounding environment.

1.3.1 Limitations

The problem described is rather extensive, and therefore a number of limitations and assumptions are made. This was necessary because of the limited duration of this project.

The map is only considered as a two-dimensional map. That means that the map only contains information on where obstacles are located in the plane and no topological information is used. Since the map only shows one level this gives the limitations that all ground of the area is considered to be at the same level. A 2D map is normally a sufficient map for an area, since this is what we usually consider a normal map where we have a top view of the area. However, since the map cannot record information about hills and slopes a use of this solution could in certain areas result in map distortion. Other interesting characteristics of the environment will be discussed in chapter 2.

The outdoor environment makes it possible to use a GPS. GPS is a measuring system that gives absolute position. The theory for the GPS will be described in chapter 3. To simplify the project further the signal was considered to be continuous. That means that a position estimate from the GPS should be available at all times when used. Tall buildings and other obstructing materials could cause interrupts in the signal and this is discussed further in chapter 2.

[1] The laser range measuring sensor is abbreviated as LMS in this report.

1.4 Different use for this solution

Autonomous systems for map building could be used anywhere where it would be dangerous or impossible for a man to build a map. It could either be used as a map to be read by a man or as a map that the robot then can use to localize and move around in the now known area. A map building system could, for example be used by a robot that looks for bombs in a car park. By first driving around to make a map of an area it is then easy to investigate the area, car by car. The maps used for these different situations are not necessarily of the same kind. A map that is used by a man to orient himself in an unknown environment preferably has many details that show where different objects are situated, the complexity of the map is of limited relevance. More details is often better. But if a robot is using a map it is of importance that the map is well structured and relatively simple. A few simple structures could be enough for the robot to keep track of its movements and find its goal. Too many details will be difficult to evaluate and will require more computer power and thus slow the system down.

If an extension is made so that the map built is a three-dimensional map this solution could be used to map otherwise unexplored areas in deep sea or space.

1.5 Necessary properties of a map building system

Humans see the surrounding environment through their eyes and make a map over the area around them by walking around and discovering. But, how should this be done by a computer? We need a way of supplying the computer with necessary sensory perception to acquire information about its environment and we also need to give the computer a way to move around. But there are many more problems that need to be addressed to give us a system for autonomous map building. Some of these problems will be discussed below.

- *Perception of the environment* – For a robot to see the environment many different sensors can be used. The sensor used in our project is a laser range sensor that scans the horizontal plane in front of the robot. The measurements taken are the distance to the nearest hit object with an accuracy of 1 degree and +/- 5 cm in distance. This sensor is a good choice for the 2-D map since it gives information about distances in the horizontal plane. Other sensors that could be used for this are sonars or cameras. Sonars or ultra sonic range sensors are commonly used for obstacle avoidance and have a resolution of 25-30 degrees only at short range. O. Wijk [Wijk, 2001] has made a comparison of different grid mapping techniques where he uses sonar measurements for mapping. Amongst other he discusses the problems with memory requirements with big environments and also gives examples of maps built using these different techniques.

- *Transportation* – Most mobile robots have wheels. Although some progress has been made with legged robots the wheeled ones are much cheaper and in most cases sufficient. The ATRV is a 4-wheeled rover that uses skid steering. This means that the wheel axes are fixed and the robot turns by driving the wheels on one side faster than the wheels on the other side. This is also known as differential drive for 2-wheeled robots.
- *Map Representation* – There are many different ways to represent a map and some methods are described in the following section. Is it sufficient to store the location of all obstacles or is it important to also identify some specific features from others?

1.6 Different representations of the map

How a map is represented in the system affects the solutions in many ways and it is therefore necessary to understand what properties that are changed with the selection of representation. A solution that works well for one map representation is probably not the ideal solution if the map representation is changed.

There are two main ideas of map representations discussed in this section. The first idea uses a matrix or a bitmap to store information about objects that occupy different areas on the ground. The other idea instead uses a list of objects and their positions.

1.6.1 Grid representation

The grid representation uses a matrix to store information about different cells in the map. The map size is given by the size of the matrix and the area that is to be included in the map. For example, if the area that is to be covered is 100×100 meters and the size of the matrix is 100×100, the size of each cell is 1×1 meter. So, each element in the matrix corresponds to a space of 1×1 meter in the area that is mapped. Each element can then hold different information about the corresponding cell of physical space. The information that is stored is normally if this cell is occupied by an object or not, but the element could also contain information on which type of object that occupies this space or the probability that this cell is occupied or not. If we extend the matrix we could store several types of information for each cell. We could additionally add information about roll, pitch, yaw and altitude for each cell, which would give information about the topology of the environment. This could for instance be done by adding an inertial sensor to the system. This is not considered in this project.

The advantage of this solution is that this map gives a good visualization for humans. The matrix can easily be plotted as a bitmap that could show the information about the area on the map. The disadvantage is that if the map requires a high resolution over a big area the matrix gets big. This could cause problems if we are using a system with low memory or computing capacity.

1.6.2 Feature representation

Instead of using the fixed matrix a list of all objects in the environment is kept. The objects could be of different types and would correspond to different features[2] in the environment. The list contains information about the objects, such as type, position and orientation.

The advantages for this solution are its unlimited resolution and a smaller memory requirement. The position of all objects is possible to store with very high accuracy since we only need to set the X and Y positions for each object. If we have a limited number of distinct objects in an area the representation only needs a list of these objects to show a map over the area. The memory requirements will only grow with the number of objects and not with the size of the area.

The disadvantage is that good feature detection is a relatively difficult task for an autonomous system. Even if simple features can be detected it is difficult to separate specific features from others. The normal objects that we want to detect are walls or point shaped objects such as trees or lampposts. The point shaped objects are relatively simple to detect, but a wall is more difficult. We need to set a threshold to detect if the wall is a wall or a series of small point shaped objects. This is discussed further in chapter 5.

How the map is represented could affect both performance and clarity in an autonomous system that is localizing in the map or building it. A more efficient and structured way of representing a general map in a more hierarchical way is discussed in chapter 8.

1.7 Literature read during the work with this project

For an orientation of the subject and material to give ideas on how to solve the problem of map building, many papers have been reviewed. Some of those, that have been of most help, and that help understanding the concepts mentioned are described below. Even though

[2] Features are discussed further in chapter 5

many papers have the same main idea we have tried to distinct the literature into a number of specific issues. Several of the works mentioned will however fit under several sections and this partition may only be used as a guide.

1.7.1 Understanding sensors and autonomous systems

Borensteins work, ”Where am I” [Borenstein, 1996], has provided important information concerning the equipment of an autonomous system that is used for map building. His work describes most information necessary for map based localization. This is however not completely the same as map building but the sensor suite and the discussions are very useful. He gives a very in-depth description of different sensors that could be used to perceive the environment and also different properties of different autonomous systems.

Some further material on GPS was read. Abbot&Powell’s work, ”Land-Vehicle Navigation Using GPS” [Abbot&Powell, 1999], discusses how the GPS is used for position estimation and Misra, Burke & Pratts work, ”GPS Performance in Navigation” [Misra, Burke & Pratts, 1999], gives more understanding of the problems with the use of GPS. This material describes the fact that signals of the frequency that GPS uses cannot pass solid materials, which implies that GPS cannot be used indoors or even close to tall buildings or dense foliage that block the view. The papers above mentioned also discuss how the GPS position estimate is formed, which is not covered in our report.

1.7.2 General map building

Wijks thesis, ”Triangulation Based Fusion of Sonar Data with Application in Mobile Robot Mapping and Localization” [Wijk, 2001], discusses amongst others how sonars could be used to create a map of the environment. Several different types of maps are discussed and implemented on a robot for indoor use. Wijk has developed an algorithm for point feature detection using sonars, which gives an overview on what point features are and how they can be perceived. He also gives a brief comparison between different grid mapping techniques. Unfortunately no time was left to evaluate or describe these methods in our report.

Thrun has written a paper that covers map building, ”Robotic Mapping: A Survey” [Thrun, 2002]. This paper mainly focuses on indoor mapping but several ideas are of course applicable to an outdoor environment as well. He describes the problems of mapping such as measurement noise, the high dimensionality of accurate maps and the problem of dynamic environments. He also discusses the advantage of using probabilities since ”robot mapping is characterized by uncertainty and measurement noise”. Among the different map types mentioned is the Occupancy grid map, which is a map of the same type as the grid map implemented in our work, where obstacles are stored as probabilities in different positions on the map. He mentions that this method has a reputation of being extremely robust, but unfortunately it lacks a method for accommodating pose uncertainty, which lead us to believe that the quality of the pose estimation used is of high importance.

1.7.3 The SLAM problem

The SLAM problem is the problem of performing simultaneous localization and map building. This is described in section 5.5. Even though no time was left to implement the SLAM algorithm several papers have been reviewed that cover this idea.

Zunino has performed experiments with the SLAM algorithm on his automated vacuum cleaner in his thesis, ”Simultaneous Localization and Mapping for Navigation in Realistic Environments” [Zunino, 2001]. He has made an implementation of the stochastic mapping algorithm that was first published by Smith and Cheeseman in 1987. This is the SLAM algorithm that is described in our work. Zunino has also extended this work with algorithms for failure recovery and suggestions on geometric constraints that are used to enhance the map. The failures he describes are caused by data association errors, map slippage and unexpected perturbations of the robot.

The performance of the SLAM algorithm is highly dependant on the number of features present in the map. P. Jensfelt has made an implementation of a Hierarchic SLAM algorithm in his work, ”Approaches to Mobile Robot Localization in Indoor Environments” [Jensfelt, 2001]. He uses an idea where the map is divided into several sub maps for different rooms of a building. By doing this only properties of the current sub map need to be updated, thus improving performance. His work is described further in chapter 5.8.

Another paper that uses sub maps to improve the SLAM algorithm is Guivant & Nebot in the work, ”Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation” [Guivant & Nebot, 2001]. They have instead of using rooms as sub maps divided the map into equal squares, as a grid and have focused on improving the mathematics for this model. Information about the current map call and the eight neighboring cells are updated at each step. Further discussion of this work is also described in 5.8 as improvements of the SLAM algorithm.

1.8 The structure of this report

This report is divided into several different chapters that cover different parts of the solution. Some parts might seem unnecessary to some and therefore a short description of the contents of the different chapters hereby follows. If the report is read from cover-to-cover this section can be skipped. This report is best viewed in colour.

The introduction that follows is supposed to give background and introduction to the necessary concepts that are included in this work. After the project is outlined a short introduction will follow that will help people that are not familiar with autonomous systems to get the necessary ideas of what autonomous systems are and how they could be used. After this a discussion on what the solution could be used for and what problems that needs to be overcome to make this solution possible is discussed.

Chapter 2 discusses the characteristics of the different environments used for tests, in this project. This information is necessary to understand what conditions that should be adapted to.

Chapter 3 provides the theoretical basis for the solution presented in this project. Further theories about the two main ideas follow in chapters 4 and 5. These chapters also cover the practical implementations for these solutions.

Chapter 6 describes the different tests that were made over the course of the project and the results that were derived from them. This chapter includes numerous images that illustrate the work done.

Chapter 7 is the conclusions from the final tests in the previous chapter. This is probably considered the most important chapter by most researchers.

Chapter 8 discusses the future work that could be done to gain further progress in the problem of building autonomous maps.

Appendix is mainly supposed to be used as a reference to subjects that are not covered in this report but could be interesting for people that intend to repeat these experiments. Since this chapter covers the hardware used, this could serve as a good complement to the people that are new to autonomous systems, and that want an example of what an autonomous robot could look like.

All chapters of this document begin with a short introduction and the report could be read in part or in full.

1.9 Summary

This chapter has covered an introduction to the important properties of an autonomous system that is used for autonomous map building. Important properties of a map building system, such as that the robot can move around and perceive the environment, was mentioned. Different map types are described. Grid based maps stores the map as a grid where each cell is occupied or not. Feature based map types keeps the map as a list of lines or other features that have been detected in the environment. This chapter did also review relevant literature that was read prior to this project.

2 ENVIRONMENT

2.1 Indoor and outdoor environment in comparison

The problem of localization and map building in an unknown environment is not new. It has been addressed by many researchers but mainly in indoor environments. Knowledge and experiences can be drawn from these studies, but it is necessary to consider the differences that may be encountered when solving this problem for outdoor environments. None of the papers read prior to our work has this discussion and the implementations of map building in outdoor environments are highly simplified, such as Dissanayake, Newman, Clark, Durrant-Whyte and Csorba [Dissanayake, Newman, Clark, Durrant-Whyte and Csorba, 2001], that primarily use artificial radar reflecting poles that build the map and Guivant & Nebot [Guivant & Nebot, 2001] that only use trees to create their map. We have therefore presented a short list of suggested properties that could affect map building and also the difference between indoor and outdoor mapping.

- *Smoothness* – The smoothness of the surface over which the mobile robot travels may affect the sensor readings. If the traveled surface is flat, as is normally the case indoors, the sensors are stable and give stable readings, but if the surface is rough, the sensors may vibrate a lot, thus causing the readings to be less accurate.
- *Bounded/Unbounded* – The environment may be bounded or not. For an indoor setting the environment is bounded by the walls of the room. The rooms may be large but the walls are always bounding. If the mobile robot moves in any direction it will encounter a wall. This is not necessarily the case outdoors. The environment might be bounded locally near buildings but in the surrounding areas the environment might be considered as unbounded. This may cause problems since sensors for detecting the environment normally have a limited range, which "blinds" the mobile robot in environments that are too vast.
- *Topology* - Indoor environments are normally flat but an outdoor environment can have slopes or hills. This can in many ways affect the way that the environment is perceived. If the mobile robot is moving over a hill or is tilted one way or another this will also cause the sensors to be tilted in the same manner. If the environment is flat the sensors perception of walls at a specific distance is always true but in a non flat environment the sensors might be facing not straight to the wall but up into the air or down to the ground if the mobile robot is tilted.
- *Detectable Features* – For an indoor environment lines are a good and sufficient choice of features that could be used for map building since all rooms have walls, even if they may be occluded

at some points. For an outdoor environment the choice of features for map building is not that simple. In some areas there might be walls or at least line segments, but even if this is the case the appearance of vegetation that occludes the walls is common. Even if some areas have lines, others have not. Thus an additional feature type, such as points, or ”spots”, would serve as a good complement, for example in environments that contain trees or lampposts.

- *Clutter* – Objects may occlude otherwise detectable features. An indoor environment can be densely cluttered with objects such as shelves, chairs, desks and computers that hide walls that otherwise would be detected. An outdoor environment is normally more sparsely cluttered.

2.2 Different locations used for tests

Tests that were made in the scope of this project were performed in two different environments. The two environments are different in many ways. This gives knowledge of different environmental characteristics that will affect the resulting map. The two different locations may be characterized by the terms mentioned in 2.1 and a concise description of these locations and illustrating sketches follows hereby.

2.2.1 Kungsängen

The area in Kungsängen was the primary test area that was considered for this project. Unfortunately it proved difficult to use it for more than a few tests and therefore the area around BB was suggested as an additional test area. BB is mentioned further in the next section. This section will give a description of the characteristics and problems with the environment in Kungsängen.

Characteristics

This area is a test site used to simulate a small town with several buildings. The buildings are one to two floor wood or concrete ”shells” with thin walls and a number of doors and windows. The windows have no glass and the doors are open or simple wooden doors. Some of the buildings have extra walls, which extend one or two walls. This can be seen in the sketch below. The area is located in a slope with several different levels of altitude. The ground is covered with big rocks and gravel of sizes varying from 1 to 15 centimeters in diameter. This gives a very rough surface that causes any moving vehicle to vibrate a lot. Inside the town the area could be considered as bounded by the surrounding buildings. The small town has a surrounding of grass fields and forest. Several wrecked cars are placed near the buildings to simulate parked cars. The walls could be detected as line features that form the different buildings. This area is deserted and everything is static. A sketch of this area is shown in figure 2.1.

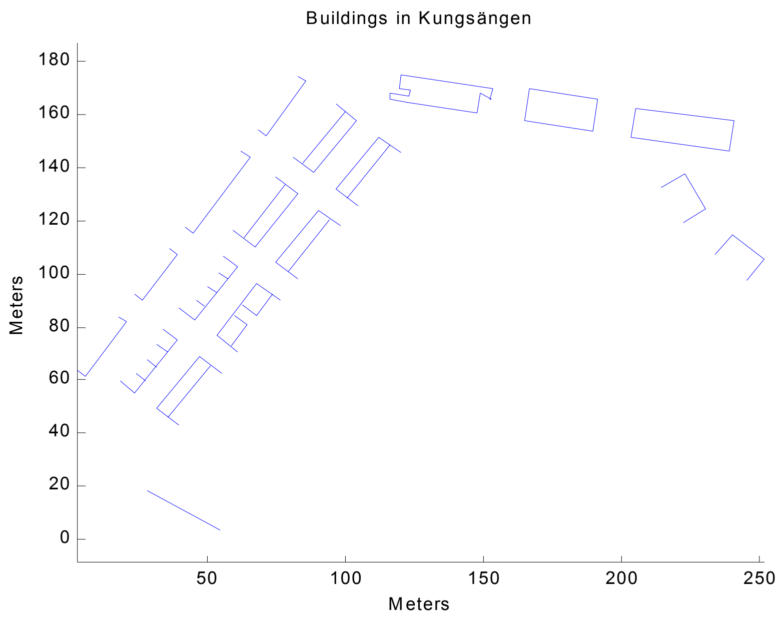


Figure 2.1. This sketch shows the test area in Kungsängen

Properties of this environment that could cause problems

The biggest problem in this environment is the rough surface. The roughness causes sensors to vibrate up and down giving readings that are not in level with the ground. The slopes and hills also cause problems since range sensors detect and reflect wrong values since the sensors are heading up into the air or on a high spot on a wall when the robot is moving uphill and heading down into the ground when moving downhill. This gives false readings that in the worst case could be perceived as walls that do not exist. Some of these problems could be avoided by integrating an inertial sensor. An inertial sensor detects accelerations in all different directions, including rotational acceleration. This sensor could therefore be used either to recalculate other sensor measurements according to this information or to ignore measurements that are considered too noisy because of excessive vibrations detected. Due to the limited time for this project an integration of an inertial sensor is not tested in this project.

2.2.2 BB

BB is the area around the building where Center for Autonomous Systems is situated. This serves as a good test site for simpler tests due to the convenience when transporting the robot to this location. It was also suggested that a second test area was needed, since the area in Kungsängen only was available a few of times. Several characteristics distinct this test area from the one in Kungsängen and this should be noted since this has given experiences of what kind of characteristics that could cause problems when autonomous map building should be performed. A simple sketch of the area is shown in figure 2.2.

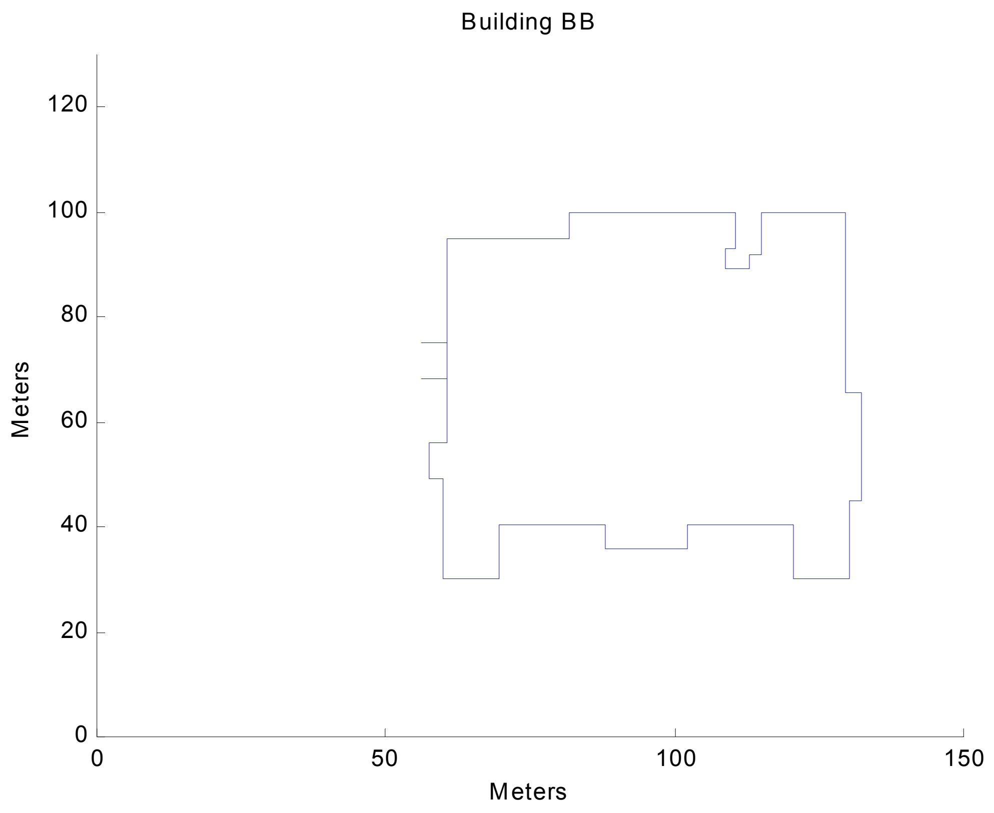


Fig. 2.2. The lines show the outer boundary of the building BB

Characteristics

Figure 2.2 shows the outline of one tall building. The roads surrounding the building are covered with asphalt thus giving a quite smooth surface, as long as the robot stays on the road. The area is rather flat, with some exceptions; therefore the assumption that the area could be considered as two-dimensional is almost fulfilled. Since we are always near the building, it should be possible to detect line segments to create a map of the building. But the surrounding environment is not that simple. The area is partly surrounded by a diffuse steel fence that is difficult to detect. Beyond the fence there is a forest where trees could be distinguished outside the fence. A grass slope and a wooden fence surround another part of the building. The slope and the fence could both be detected as a curved wall. The front side of the building has no close bounding features, but there are some point features (walls, lamppost) besides the road.

Properties of this area that could cause problems

A difficulty with the line features of the building is glass windows that could be perceived as open with use of some sensors. Even if the steel fence could be detected it is occasionally blocked by parked cars all over the area. The cars should somehow be detected as dynamic objects, since they only appear at some times, as bypassing pedestrians. The difference between these two objects is that a car probably is parked at the same spot when it is seen and is therefore probably detected as a static object, whereas a pedestrian is detected as a moving object.

The building BB is rather complex and is composed of many walls. Some sections of the building might cause specific problems. We have a loading ramp that is blocking but could be missed when using two-dimensional sensors since it is a platform hanging in the air. Objects like this that has different bounds at different height is hereafter considered as three-dimensional objects. These objects may cause problems since it is difficult to detect if they are blocking or not, by using normal sensors. Another 3-D object in this environment is a staircase in the front of this building. A staircase is a typical 3-D object since the perceived distance to this object varies as different heights of the object is studied.

The tall buildings obstruct part of the sky if the robot is near any of the walls. This could be a problem if we use sensors that need a good view of the sky.

2.3 Summary

This chapter has presented how different properties of the environment can affect the resulting map. The two different test sites in Kungsängen and around the building BB are described in detail with a discussion concerning their different characteristics. The tall buildings around the building BB could pose a great problem with the GPS since this could obstruct the GPS signals. The environment in Kungsängen has a rather clear view of the sky but instead the ground is really rough in this area, which could cause problems with the odometry sensor.

3 THEORY

This chapter will discuss many of the theories concerning the instruments and ideas that are of interest for autonomous map building. Some further theories and details concerning the two main ideas for autonomous map building, grid based map and feature based map, can be read in their corresponding chapters 4 and 5. These two chapters also cover the tested implementations for these ideas.

3.1 Different methods to create a map

There are two main theories on how to solve the map building problem, the grid based approach and the feature based approach. These are different in many ways and a comparison of the two methods is given below.

3.1.1 The grid based map method

The grid based approach uses the grid representation of the map as mentioned earlier. This map is updated as the robot moves around and measures the environment. The measurements from the LMS are directly mapped to a corresponding position in a matrix as relative to the robot position. The element that the measurement is placed in gets a raised certainty of an obstacle. As the robot moves around and new measurements are taken the certainties grow in the elements that have a high probability of containing an obstacle. When the mapping is complete the the elements of the matrix hold all map information. Elements, or cells, with high certainties show obstacles in the map. This could be walls, trees or anything that occludes the view of the robot. The area that lies between the vehicle and the measurements could be marked as free. The following image sequence in figure 3.1 will illustrate the updating of the grid map as the vehicle rotates in the environment.

The first picture shows the objects in a nearby environment. The dark green object, in the upper left corner, represents a building. The dark green object, in the lower right corner, represents a wall, the blue object, in the lower left corner, represents a car and the gray round object, in the upper right corner, represents a rock. The first picture has also an underlying grid that shows how the objects could occupy different cells in this grid, thus creating a grid map. The following pictures show what is visible to the vehicle and how unknown areas are explored as the robot rotates. The map is initially unexplored and all areas are unknown, which is represented by the color yellow, or light gray. In the second image the field-of-view of the laser range finder is shown in green color, the shaded area in front of the vehicle.

The green color could also represent areas that are known to be free or un-occupied. The points that are reported by the laser range finder are marked as occupied. This is done for the rock and the wall, no other objects are yet known. In picture three the building is also visible, and also part of the car. The darker green, or darker shade, shows the overlapping area between the field-of-view for the first and second scan. In this area the certainty is higher that the objects really exist and that the area in front of the objects are free. In the fourth picture almost all of the yellow ground has turned green and has therefore changed from unexplored to explored and free.

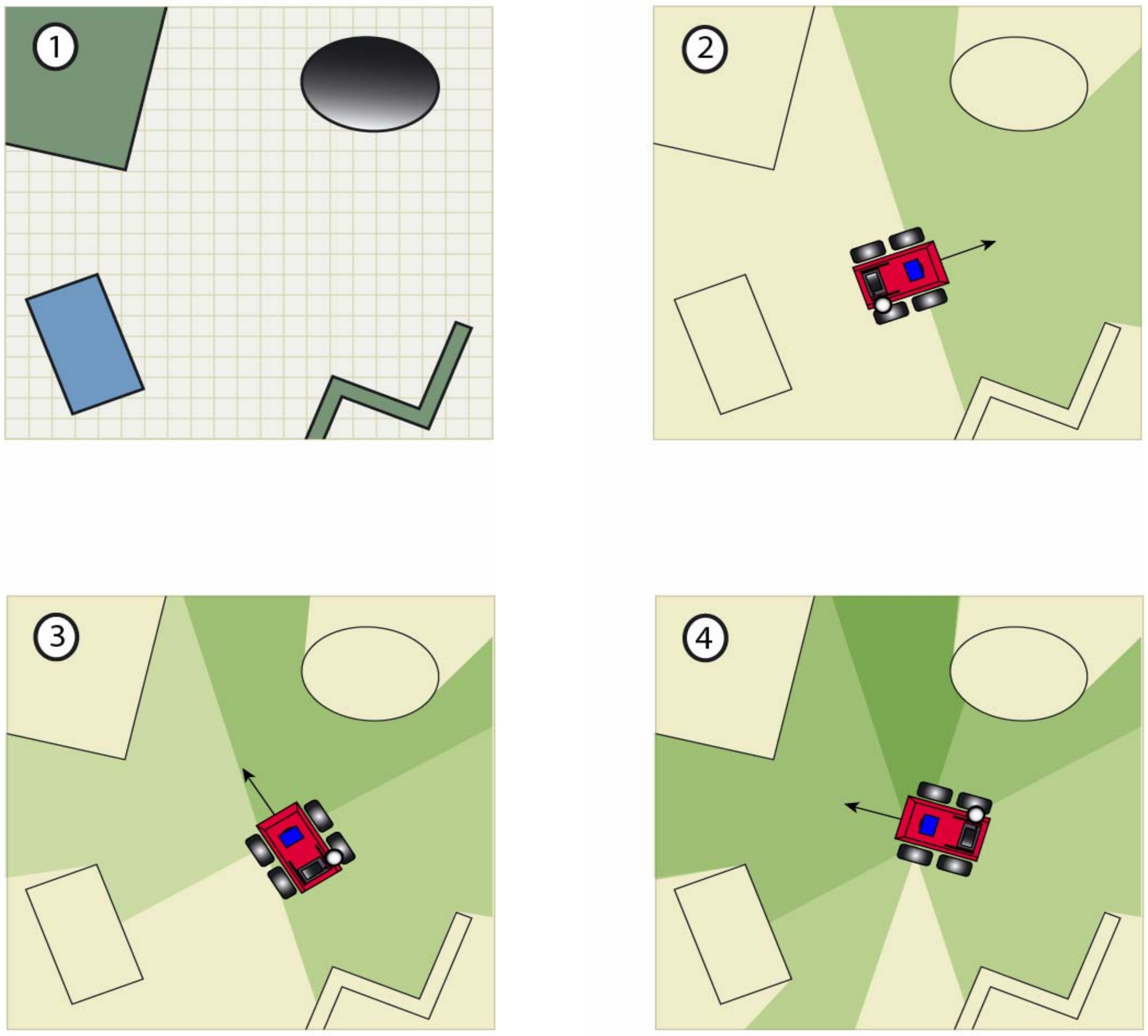


Figure 3.1 Four steps showing how unknown areas are gradually explored as the robot is rotated. The laser range finder is active in a semicircle in front of the vehicle. The field of view is shown in green (the darker shade) in front of the vehicle and also shows areas that are explored and reported as free. Unknown areas are colored in yellow (the brightest background color).

The different tones of green show that some areas have been visible several times, which would raise the certainty about these areas. Objects or open areas that have been measured several times are more reliable than objects or areas that have only been seen once. Still we have unexplored areas. By moving around the rock or the car the unexplored areas could be explored in a later step. In the grid map the objects and areas would be stored in cells corresponding to a specific area in the environment. These cells will after updating have information such as unexplored, free or occupied. This is known as an occupancy grid map. To get rid of noise in the matrix it can be filtered by deleting all low certainties prior to the final map. Elements with small certainties could be moving objects such as pedestrians or bad measurements as the robot was tilted by rough terrain. The details concerning the steps in the algorithm are described in the following chapter about grid based mapping.

3.1.2 The feature based map method

The feature based map is a more complicated probabilistic model of how the features in the map are connected and how the robot is moving relative to these features. This method uses a list or vector based map representation. As the robot moves around the scans from the LMS are evaluated to see if any features are present. This is called feature extraction. Normal features are lines or points. Line extracting could be done by using the Hough transform or by using least square estimation of the points in the scan. If a line feature is found it is appended to the list of features, but first after it has been compared to the earlier features in the list to see if it already exists. If it does it is discarded. This is performed in the map matching step. When the robot moves around it could use the list of features to align to the already existing walls to give a better estimation of its position. The problem of updating the robot position using the recently built map is called the SLAM problem. SLAM stands for Simultaneous Localization And Mapping. Many papers are written on this issue and some of these are discussed further in the end of chapter 5. Figure 3.2 below gives a schematic view of the SLAM problem.

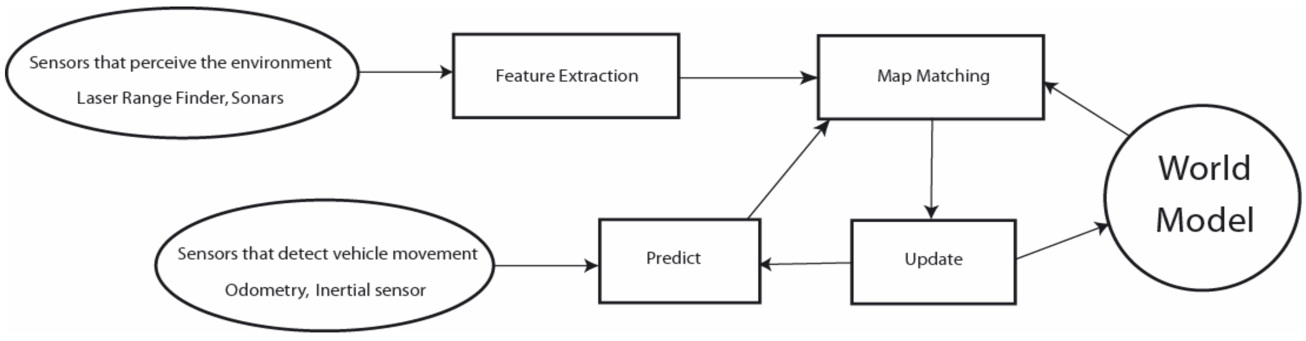

Sensors that perceive the environment
Laser Range Finder, Sonars
Feature Extraction
Map Matching
World Model
Sensors that detect vehicle movement
Odometry, Inertial sensor
Predict
Update

Figure 3.2 The SLAM algorithm in schematic steps.

Sensors provide information that could be used both to estimate the motion of the robot and information about the environment. Information about the environment is used to extract features that are matched to the already present features in the world model. Using this map and information from sensors that detect motion of the robot, its position can be predicted. Information about the robot is also used to match new features to older ones. This is described further in chapter 5.

As a comparison between the grid based map building and feature based map building it can be said that feature based mapping often stores less information about the environment, but in a more structured way. This makes it a good choice for robot localization, that is when a robot is moving autonomously between different points in an environment. The feature based has a number of relatively simple features that could be detected by the robot and therefore used to find a way between two points in the map. The grid based map has a detail level only limited by the quality of the sensors and the resolution of the grid. If the map is used to gain a picture of an unknown environment this is a good choice.

3.2 Other important theories

3.2.1 Odometry

Odometry is the sensor that is used to keep track of how the robot moves. The technical details concerning the odometry sensor are discussed in appendix A. Odometry gives the accumulated distance traveled and the total rotated angle since the system started. The turning radius of the system is determined by the difference in wheel speeds between the left and the right wheel pair. Since these values are calculated from the revolutions of the wheels these values are inaccurate if the surface is uneven or if the wheels slip. This is unavoidable in an autonomous system and therefore the errors from every small adjustment in rotation or translation add up to an increasing error. However, if the system is used over a flat area and wheel slippage is minimized by turning with a big radius this gives a good estimate of the position of the robot over limited distances.

To create a map the location and the orientation in the plane is needed. The orientation is given directly from the angle of the odometry. The localization is calculated from odometry by giving a start position and then adding up the small distances moved in each step, from the odometry. Evaluating the following simple system where X and Y are given start values does this.

$$\Delta d = d_{n+1} - d_n$$
$$X_{n+1} = X_n + \Delta d \cos\alpha$$
$$Y_{n+1} = Y_n + \Delta d \sin\alpha$$

where d_n is the accumulated distance, α is the angle and X_n and Y_n are the calculated positions in the plane.

3.2.2 The laser range finder

The laser range finder is the sensor used to give an estimate of the surroundings. The laser range finder is often called "the LMS" throughout this report. LMS is the specific type of sensor used that is manufactured by SICK. The LMS scans the environment in a semicircle in front of the sensor, with a speed of up to 5 full scans a second. The measurements that are stored in a vector are the distances to the nearest obstacles in all directions. Figure A.1 in appendix show one laser scan inside a building where for example the main features of the walls can be seen. In this project the LMS used was originally set to a resolution of 1 measurement/degree, which was therefore used throughout all tests. This gives a vector with 181 entries for the front semi circle. The LMS reflects of all non-transparent materials, but as glass is transparent to laser beams glass objects are not detected by the LMS, thus causing measurements through windows showing the distance to the nearest object outside the window. The positions for the surrounding objects relative to the LMS are calculated by the following formula:

$$x_a = laser(a) * \sin(a)$$
$$y_a = laser(a) * \cos(a)$$

where *laser(a)* is the *a*:th element in the laser vector. Since there is 1 measurement/degree, *a* is also the right oriented angle. The variables x_a and y_a are the coordinates for the nearest object in direction *a* relative to the LMS.

The LMS will only scan the surroundings in the horizontal plane of the robot, since it is mounted on top of the robot and scans a plane from right to left. This means that if the robot is not level the LMS might take measurements from the ground or from points high in the air. This is to be considered if the map building is performed in rough terrain.

The fact that the LMS only detects obstacles in the front semi circle implies that as soon as any object passes 90 degrees to the left or to the right of the robot, these objects can no longer be seen. This can cause problems if not handled correctly in the feature based map building since features that were recently seen disappear as they are out of sight.

3.2.3 Global Positioning System

The Global Positioning System or GPS is a sensor that can give absolute position estimates all over the world. It is a system that consists of 24 satellites that orbits the earth in cycles that guarantee that at least four satellites are visible from everywhere on earth. Each satellite transmits a signal that contains its position and the time for the transmission. The distance to each satellite can then be measured by multiplying the speed of light with the time elapsed since the transmission was made. If signals are received from at least four satellites it is possible to calculate the position of the receiver and the time difference between the receiver and the satellites. The position can be estimated in three dimensions, but in our tests the altitude is not used since the map is supposed to be two dimensional, thus only requiring information about position in the ground plane. The accuracy of the position depends on many different factors. The signals from the satellites are transmitted at a frequency of 1.575 GHz that is too high to bend around or pass through solid objects. This implies that GPS cannot be used indoors. This also implies that tall buildings and dense foliage might block the view of the satellites, as mentioned by Abott & Powell [Abott & Powell, 1999]. The accuracy is also determined by the number of satellites in view and their spatial distribution is the main factor as mentioned by Misra, Burke & Pratt [Misra, Burke & Pratt, 1999]. Their article also describes how the raw carrier phase from the satellites could be used to calculate the positions of the satellites, which is not covered in this report. The possible accuracy is better than +/-10 meters.

The position estimate can be improved by using a Differential GPS. That is an additional receiver connected to the GPS that receives signals from DGPS stations that are situated at known positions. These stations calculate the error between their real position and the estimated GPS position and transmit these distance corrections. The differential receiver is then used to correct the GPS signal, which in theory could give position estimates with sub meter precision. A more in depth overview of the Global Positioning System can be read in the article written by Enge & Misra [Enge & Misra, 1999]

The position is received in longitude, latitude and altitude. If a GPS receiver is used to build a map it is preferable to have the position in plane xy coordinates. This can be done by a series of geometrical transformations of the measurements. These transformations are not discussed further in this report but are described in detail by M. Sørensen [Sørensen, 2002].

3.3 Summary

The use of different sensors and the two main theories for map building were covered in this chapter. For grid based mapping the sensor data from the ranging sensor is projected into grid cells in the global map, where the reflections correspond to objects in the environment. As these grid cells are reobserved, the probabilities that this cell is occupied are increased. The feature based mapping method uses the information from sensors to extract features in the environment that are stored in a feature vector. When the position of the vehicle is estimated from this map, that is still under construction, this is known as a solution to the SLAM problem. The sensors that are used in this project are odometry, which estimates vehicle movement by calculations from wheel revolutions, laser range finder which is a sensor that measures the distances to objects around the vehicle, and the GPS sensor, which gives absolute position estimation by a calculation of distances to a number of transmitting satellites. The satellites transmit information about their position and the time of the transmission. At least four satellites need to be visible to calculate a position estimate.

4 GRID BASED MAP

4.1 What is a grid based map?

The grid map is a map that consists of a grid system of the area that is to be mapped. Each cell in this map contains a certain value that corresponds geometrically to a specific area in the environment. The cells in the grid map obtain values containing certainties that this cell is occupied or not. The cells correspond to a specific size in the area. This could be any size depending on the resolution of the map. The cells could be 1×1 meters, 25×25 centimeters, or any size that gives the required resolution of the map. This map type is also called a metric map. The map model that is used in the implementation is a simplified version of the occupancy grid mapping algorithm by Elfes and Moravec that is described in the paper by S. Thrun [Thrun, 2002]. This algorithm requires known pose information and therefore much effort has gone into optimizing the odometry to give as good readings as possible. These experiments are shown in the beginning of chapter 6.

4.2 How is the grid map constructed?

4.2.1 Overview

The grid map could be constructed in real time during the process of measuring the environment but this is in our implementation divided into two steps. The first step is to guide the autonomous vehicle around in the environment that is to be mapped. During this step data are collected from all sensors of interest and stored in files on the computer. The second step is then to use these data to create the grid map.

The necessary data to create the grid map is the odometry data that keep track of how the robot has moved and measurements from the laser range finder, the LMS. A matrix then needs to be created. This matrix is used to store information about the different cells in the map. The required resolution of the map and the size of the area that is to be mapped determine the size of the matrix. If the area is 100×100 meters and a cell size of 25 centimeters is required, the matrix must have the size 400×400. The elements in the matrix will then be updated for all steps to raise the certainty for the cells that are reported as occupied by the LMS.

4.2.2 Implementation

The algorithm implemented uses the odometry data and the LMS measurements to distinct which cells that are occupied. The LMS readings are considered reliable if their measurements show a distance below 50 meters. This is done to filter out the signals that are out of range for the LMS. The LMS should be able to report reflections from objects at distances up to 81 meters but 50 meter was set to the maximum range. All these measurements from the directions of the 180° semicircle are projected to cells in the grid with reference to the robot position. The cells that the measurements are mapped to get their corresponding values in the matrix raised. This implementation basically uses the Borenstein grid mapping technique, which raises the certainty with a fixed number that adds to the probability until a certain maximum value is achieved. This method is described in O. Wijks thesis [Wijk, 2001]. A check is made so that the probability cannot reach a higher value than 1. All these updates are then performed for every step as the ATRV moves around. An integer division of the mapped measurement determines the selection of which cell that should be updated. This operation is set to fit the cell size so that all measurements that fit within a certain cell affects that cell. It was our intent to try a more probabilistic model of this grid map but unfortunately no time was left for this.

The equations for the mapping of the measurements are

$$Gx_i = X_n + laser(i) * \sin(i + \alpha)$$
$$Gy_i = Y_n + laser(i) * \cos(i + \alpha)$$

where Gx_i and Gy_i are positions within the cell that should be updated, X_n and Y_n is the position of the robot at step n. The angle of the robot is α and i is the direction of the laser measurement. Figure 4.1 shows an illustration of the process of creating a grid map.

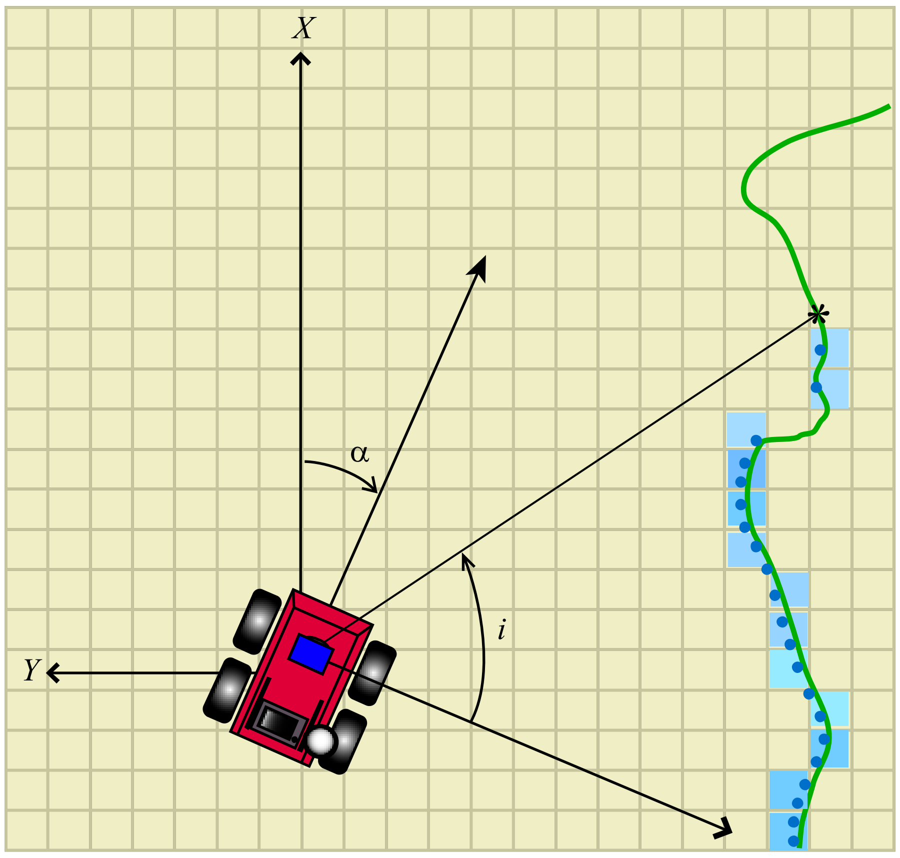


Figure 4.1. This figure shows the creation of a grid map. X and Y are the true positions for the robot and the angle α is the angular deviation from the start angle. The angle for the laser measurement is i. The green line (the curved line to the right) shows a detectable surface and the blue dots (along the line) correspond to the detected measurements. The grid map will be updated in the cells where the measurements are mapped.

The corresponding cells in the grid are then updated by the following step

$\mathbf{A}_{ij} = \max(1,\mathbf{A}_{ij} + c)$,

Where $\mathbf{A}ij$ is the grid cell closest to the position G from above and c is a constant value for increasing the probability of an obstacle in this position. Here, $1/c$ defines the number of times an obstacle needs to be found in a cell to gain a certainty of 1.

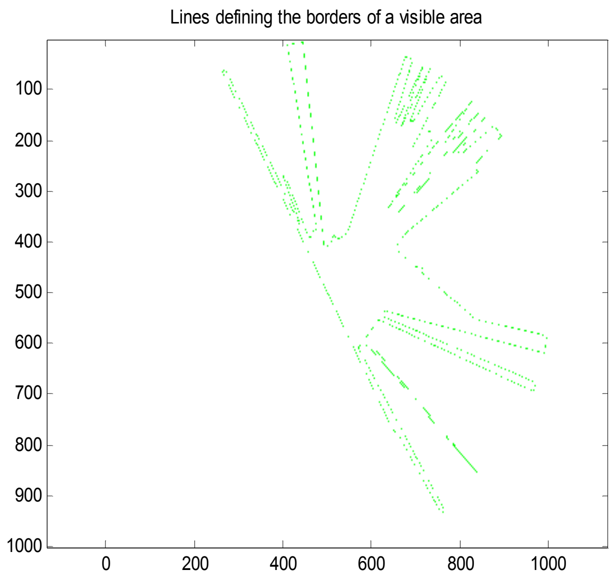
Equally the grid can be updated for cells that are detected as non-occupied by

$\mathbf{A}_{ij} = \min(0,\mathbf{A}_{ij} - d)$

Where d is the decreasing constant and $\mathbf{A}_{ij}$ are all cells that are within the detected range. Unfortunately this is not as simple as the detection of the cell that is occupied. Since the cell space between the robot and the occupied cells consists of several cells we need to find out which cells to update. Borenstein uses a line stepping algorithm to update the cells that lie on the line connecting the robot and the occupied cell. We have instead implemented an algorithm that updates all cells that lie within a full laser scan. To do this all 181 measurements are connected with lines to form a polygon as shown in figure 4.2a. A polygon-filling algorithm then sets all positions of a matrix that are inside this polygon. This matrix has the same scale as the matrix **A** and can therefore be subtracted directly from the matrix A, thus causing all cells that should be unoccupied to be updated. The polygon-filling algorithm is not described further in this report but can be found in almost any book on computer graphics. We have used the algorithm from the Book from Foley, van Dam, Feiner & Hughes [Foley, van Dam, Feiner & Hughes, 1995]

The update of non-occupied cells was implemented late in this project and therefore this is only used as an illustration in one of the later experiments.

Another matrix is added that only holds the position for the ATRV. This is only used for visibility. The ATRV position is also shown in the grid maps in chapter 6.



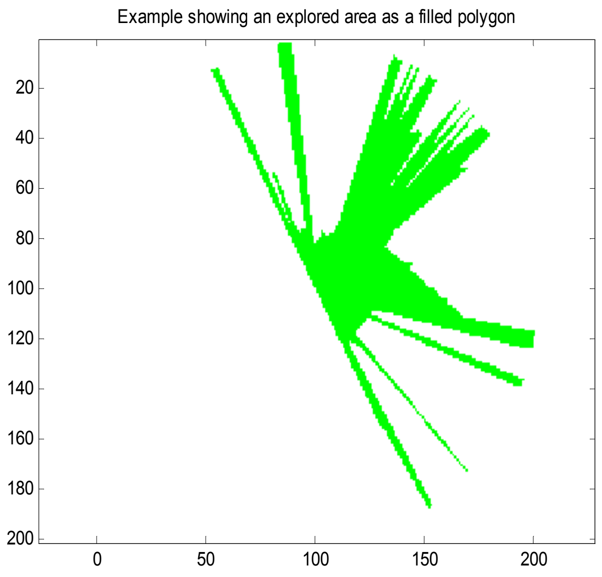


Figure 4.2a (to the left) shows the lines connecting the measurements from the laser range finder. Figure 4.2b (to the right) shows a matrix where all elements inside the polygon are set.

4.3 Specific problems when constructing a grid map

As with all mapping algorithms the biggest problem is uncertainties in localization of the robot. Knowledge about the localization of a robot requires knowledge about its position and its orientation. The algorithm used herein relies fully on the sensed localization for the robot and therefore localization errors will strongly affect the result. Performing pose estimation map matching of a grid map is a task that was too big to handle in the scope of this project. The cells are mapped relative to the robot localization and errors will therefore map wrong cells. The output that could be seen in the bitmap does however resemble the environment since these errors grow gradually as the robot moves; this is because of the odometry as discussed earlier. The error is mainly seen if the robot follows a closed loop from start to stop. The initial and final position in the map should then be the same, but this is rarely the case. This is known as the closed loop problem.

The closed loop problem is difficult to solve if we do not have any accurate sensors that can give absolute positions. Even if we had a way of detecting when the traveled path is closed the grid based map system is difficult to adjust to close the loop. To close the loop the map that is stored in the matrix need to be updated. In the feature based map it is rather simple to update the list of the objects to close the loop but in this case we need to update all cells in the map to make a closed path. This problem is computationally expensive if the size of the matrix is large, that is if the resolution of the map is high.

4.4 Advantages and disadvantages for the grid based algorithm

The grid mapping algorithm is easy to implement and is according to Thrun and Borenstein known to be extremely robust and this has made it a popular choice for map building. According to Thrun its major disadvantage is the lack of a method to accommodate pose uncertainty. To make this possible the robot need to have detected features in the environment, which is the case for feature based mapping. A further comparison between the grid based mapping and feature based mapping is discussed by Borenstein [Borenstein, 1996], in his discussion of different mapping techniques.

4.5 Summary

The chapter describes the grid based method. The grid based map is constructed by defining a grid or matrix to represent the mapped area. Each cell in the grid corresponds to a small space in the environment that is either occupied or not. The map is updated by projecting the range measurements into the different cells, thus marking these cells as occupied. This is often done by raising a value that corresponds to the cell and in this way storing probabilistic information in the map. As the vehicle moves around the cells are reobserved from different positions therefore raising their probabilities. The advantages of a grid based map are that it is robust and stores detailed information about the environment. The problems with the method are that a high detailed, large map requires much memory to store and handle.

5 FEATURE BASED MAP

5.1 What is a feature based map

In comparison with the grid based map where the map is represented as a geometrically static matrix where information about possible objects at different positions are stored, the feature based map is instead created in a more dynamic way. Different features that are detected in the environment are stored in a dynamic list, which is updated as the vehicle moves around and takes new measurements. The information about the features is not fixed and may change many times to update information about i.e. the position of the feature as well as other properties concerning the features. This makes this map more flexible than the grid based map but instead less robust. The dynamic nature of the feature based list will require a more active map building implementation than the grid based method, all reasons for this will be covered in the following sections.

5.2 What are features

The features mentioned here are objects or properties of objects that could be recognized and used to create a structural map. It is not yet common to use complete objects as features as they are too complex and difficult to recognize. The features used for feature based mapping are simple geometric objects, such as lines and points. These could represent walls and point shaped objects in our map, such as trees or lampposts. It would also be of interest to use abstractions of these features, such as doorways and corners, which would be of good use for robot localization, path planning and traversal to a given goal. For feature based mapping it is necessary that the features are simple enough to be detected and recognized by the available sensors of the vehicle in use. Even if it would be nice to have features for complete buildings, cars and moving humans and animals this is still a future issue since object recognition has not come this far yet. The features have certain properties that are of interest. The point features need only position information, perhaps with relative position estimation certainties, but the line features require more information to be fully determined. It is enough to have information about the position of the two endpoints of the line but it is also common to have properties such as center point, direction of the line and extension of the line, since these properties is a more natural choice for the normal sensory equipment. The laser range finder scans the surroundings with radial measurements over a semicircle, sonars are often mounted as a circumference of the vehicle and odometry naturally reports position and turning angle. As the localization of the vehicle is stored as position and orientation values it is also convenient to store the features in the same way, with probabilistic values for all properties.

The possible choice of features used for map building is dependent of the type of sensors used and what kind of features that are most common in the environment.

5.3 Detection of features

Feature detection, or feature extraction as it is normally named in papers covering feature based mapping, deals with the process of taking measurements from the environment using the present sensors and try to extract information about specific and/or possible features in the environment. The method used for this is highly dependent on the sensor equipment and the type of feature that is detected. This section covers general information about extracting point and line features using sonars and/or laser range finder.

5.3.1 Detection of point features

O. Wijk [Wijk, 2001] has developed the triangulation based fusion (TBF) method for point feature detection using sonars. This algorithm is also described by G. Zunino [Zunino, 2001]. This method describes how to get an estimated position to detect features containing vertical edges or points, such as doorposts and table legs. The principle of the algorithm uses the fact that sonars have a wide main lobe angle (~25°) which means that one sonar reading is not enough to find a vertical edge, but perhaps several sonars can do this. One measurement of a sonar only gives an arc that specifies that an echo was received from somewhere on this arc. But if another sonar hits the same spot from another direction the intersection of the new arc and the old arc is enough to determine the position of the point shaped object, see figure 5.1.

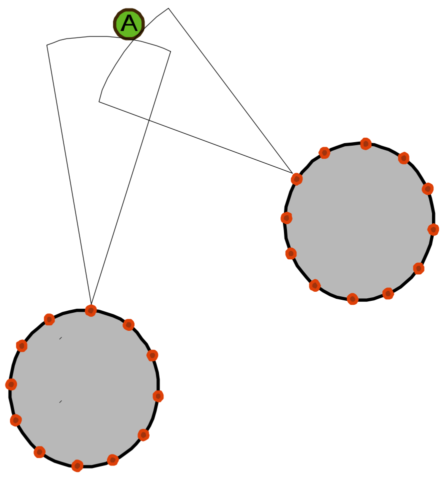


Figure 5.1. This figure shows the triangulation of two sonar measurements from the same point feature. The insersection between the two arcs gives an estimation of the position of the object. The robot is in this case round with sonars in all directions.

The same principle can be used with the laser range finder, but in this case the angular resolution is much higher which makes it possible to detect vertical edges just by studying local minima of the measurements. Another measurement from another position is however necessary to determine that the detected feature really is a point feature.

5.3.2 Detection of line features

Due to the higher density of the range measurements from the laser range finder it is possible to detect line features, such as walls. Two different methods to do this are described by P. Jensfelt [Jensfelt, 2001], and these are described below. This could also be used in combination then used as a two-step operation. Since the line is measured by sensors on the vehicle it is of interest to have properties that describe the line as relative to the vehicle, such as: distance perpendicular to the line and the angle of the line relative to the vehicle.

5.3.2.1 Least square method

One well-known method for fitting a line to a set of measurements is the Least Square fitting method. This method uses the principle that the perpendicular distance from all points to the line should be as small as possible. This is done by studying the equation of the line where all points of the line should satisfy the equation

$$r_i \cos(\varphi_i - \alpha) = \rho$$

Where r_i is the range measurements in polar coordinates, φ_i is the angle to the measurements and ρ and α define the distance to the line and the angle of the line relative to the measuring sensor. But since the measurements are not perfectly on the line, their distance to the line is described by the equation

$$d_i = r_i(\varphi_i - \alpha) - \rho$$

To find the line that most corresponds to the set of measurements, the sum of all squared distances, d_i, is evaluated as in the equation below.

$$\arg\min_{(\rho,\alpha)} \sum_i (r_i(\varphi_i - \alpha) - \rho)^2$$

The value ρ, α that minimize the sum give the distance and angle to the line that best corresponds to the given set of measurements.

This method is highly dependent on all measures in the set and is therefore sensitive to outliers and other disturbances.

5.3.2.2 Hough transformation

When the least squares method finds one line the Hough transform instead creates a space over several different line distances and angles where each cell in the space holds the number of occurrences of measurements that satisfy this line equation. These cells are called accumulator cells. The process of creating this space is computationally expensive when performed for all distances and angles and therefore this method is best used if a rough estimation of the direction and angle of the line of interest is already made. Then, it is possible to create a space within a certain distance range and within certain angular values. The resolution of the line distance and angle that is requested directly affects the size of this parameter space, since each cell corresponds to lines within a certain window, for example with a range deviation of 10 centimeters and an angular deviation of 2 degrees.

The parameter space could be defined as

$$C = \{(\rho,\alpha) | \rho_{\min} \le \rho \le \rho_{\max}, \alpha_{\min} \le \alpha \le \alpha_{\max} \},$$

meaning that only lines in a certain parameter region are studied. This space is then discretized to a grid by the quantization step $\Delta\rho$ and $\Delta\alpha$, that define the resolution of the line estimated. Now carrying out the Hough transformation fills this grid. This could for example be done by the following algorithm described by O. Wijk [Wijk, 2001].

```
for $(x_k,y_k) \in D$
        for  $\alpha = \alpha_{min}$; step $\Delta\alpha$; $\alpha_{max}$
                $\rho = x_k \cos(\alpha) + y_k \sin(\alpha)$
                if $\rho_{min} \le \rho \le \rho_{max}$
                        $A_{ij} := A_{ij} + 1$
```

Here x_k,y_k is the coordinates for the different measurements and A_{ij} denotes the cell that is nearest to the polar coordinates with α,ρ from the current step in the algorithm.

The accumulator cells with the highest values correspond to the lines that are detected from the measurements. Since the complexity grows with increased resolution, $\Delta\rho$ and $\Delta\alpha$ should be chosen large enough to cover the expected uncertainty of the expected lines with respect to position and orientation. An illustration on how the Hough transform can be used for indoor navigation can be found in the work by Forsberg, Larsson, Åhman & Wernersson [Forsberg, Larsson, Åhman & Wernersson, 1997].

5.4 How is the feature based map constructed

A feature map could be constructed manually, by measuring some interesting features that build the map that later could be used by a vehicle, or robot to localize within this environment without making new adjustments to the map. Feature maps are widely used for structured indoor environments or as Borenstein puts it: "The feature based approach is probably one of the widest used techniques when a mobile robot is supposed to move freely in an indoor environment." [Borenstein, 1996]. The basics for the feature based mapping technique, will be discussed in the following sections. The creation of a feature based map is an iterative sequence of several steps, which are described briefly below.

5.4.1 Data collection

Data collection is the process of taking measurements of the environment. Data is collected from all available sensors, for example the laser range finder and the sonars collect data about the surroundings and odometry and GPS collect data about how the vehicle is moving.

5.4.2 Feature extraction

When any features in the environment are detected, recognized and stored in the features list this is called feature extraction. As we mentioned above there are several different types of features that could be used to build a map and the algorithm for detecting and extracting features varies with the type of feature. But whatever feature type is used the result from this step is a list of all detected features from the sensor measurements of the data collection.

This could be considered all that would be needed to build a feature map. Now features are collected into a list that could serve as a representation of our environment. But the problem is, that in comparison with the grid based map where we had certain cells where information is available if that area is occupied or not, we now have a list of features at different positions in the area. This map will only grow and grow if we only add features every time they are seen. We need a way of matching new features to older ones and perhaps update their properties to adjust to new information that is available. We could also correct the position estimate of the robot using measurements to features in the surrounding. This problem is called the SLAM problem, or Simultaneous Localization And Map building. Many papers have been written on this issue but we will only cover the general idea of a solution to this problem. Some references to papers covering specific implementations of solutions to the SLAM problem will however be mentioned.

5.5 Simultaneous localization and map building

The general idea of the SLAM problem is to create a map of the environment that is used by the robot for localization and therefore also is used for the further mapping of the environment. But since both the localization of the robot and the map is unknown initially. This is a ”chicken and egg” problem, as stated by P.Jensfelt [Jensfelt, 2001] and O.Wijk [Wijk, 2001]. But since neither the map nor the localization can be performed at first, both has to be carried out simultaneously. To do this we need a model of both the robot and all features that build the map. These models need to be continuously updated as new features arise, and the localization of the robot (hopefully) gets more and more accurate. The solution that follows is a stochastic mapping algorithm, which is an Extended Kalman Filter approach to the SLAM problem. Much of the information about this algorithm is taken from G.Zunino [Zunino, 2001], who uses this method in his work. The Extended Kalman Filter will not be discussed more than short together with the following algorithms, but more information about Kalman filters can for example be found in the work by G. Welch and G. Bishop [Welch & Bishop, 1993].

5.5.1 Modeling the robot and the features

A simple model of the robot needs to be defined. The robot state estimate is denoted by $\mathbf{x}_r = [x_r\ y_r\ \theta_r]^T$, where x_r,y_r is the position of the robot and θ_r is the angle of the robot. For the model used the control input is $\mathbf{u} = [v_t,v_\theta]^T$, where v_t is translation velocity and v_θ is rotational velocity. A general dynamic model is then described by

$$\mathbf{X}_{rk+1} = \mathbf{f}(\mathbf{X}_{rk},\mathbf{u}_k) + \mathbf{q}_k$$

Where the noise process $\mathbf{q}_k$ is added to the nonlinear function $\mathbf{f}$ of the robot. $\mathbf{X}_{rk}$ and $\mathbf{u}_k$ are the current state and control input. Here, $\mathbf{q}_k$ is a noise process with zero mean and a covariance matrix Q_k. The noise is used to model the natural noise and dynamics that are left outside of this model. The time index for this model is *k*. The function **f** is defined as follows:

$$\mathbf{f} = \begin{bmatrix} x_r + v_t \cos(\theta_r + \frac{v_\theta}{2}) \\ y_r + v_t \cos(\theta_r + \frac{v_\theta}{2}) \\ \theta_r + v_\theta \end{bmatrix}$$

To be able to use feature based navigation an Extended Kalman Filter is used. The Kalman Filter makes it possible to estimate about the past, present and the future. This means that old values can be used to estimate new values. The Extended Kalman Filter has the support of linearizing a non-linear system. The representation of the system is

$$\mathbf{x}_{k|k} = \mathbf{X}_k + \eta_k$$

Where $\mathbf{x} = [x_r\ x_1\ x_2 \dots x_N]^T$ is the system state vector with $\mathbf{x}_r$ is the estimate of the robot position and $\mathbf{x}_i$ is the estimated features. **X** is the true state vector and η holds the error estimation. The covariance of the error estimate is given by

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}^{rr} & \mathbf{P}^{r1} & \cdots & \mathbf{P}^{rN} \\ \mathbf{P}^{1r} & \mathbf{P}^{11} & \cdots & \mathbf{P}^{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^{Nr} & \mathbf{P}^{N1} & \cdots & \mathbf{P}^{NN} \end{bmatrix}$$

Where $\mathbf{P}^{rr}$ is robot-to-robot covariance, $\mathbf{P}^{ri}$:s are the robot-to-feature covariance and the $\mathbf{P}^{ii}$:s are feature-to-feature covariance. Since the features normally are static they are updated by

$$\mathbf{x}_{i_{k+1}} = \mathbf{x}_{i_k}$$

for each time step with **P** as covariance. The Extended Kalman Filter estimation also updates the covariance matrix by new measurements, **z**, which is described later. The estimation is done by a prediction of the movements of the vehicle as given by the model used and also by feature updating, when features are reobserved. It is also necessary to add new features to the vector that have not earlier been observed. These three steps will be described in detail below.

5.5.2 Prediction from vehicle movement

The resulting vehicle state is given by the expectation value of the state transition model

$$\mathbf{x}_{r_{k+1|k}} = E\left[\mathbf{f}(\mathbf{X}_k, \mathbf{u}_k)\right] \approx \mathbf{f}(\mathbf{x}_{r_{k|k}}, \mathbf{u}_k)$$

where **q** is assumed zero since the noise has zero mean. The covariance for the robot $\mathbf{P}^{rr}$ and the covariance between the vehicle and the features $\mathbf{P}^{ri}$ are updated by the change in robot position.

$$\mathbf{P}^{rr}{}_{k+1|k} = \mathbf{J}_x \mathbf{P}^{rr}{}_{k|k} \mathbf{J}_x^T + \mathbf{Q}_k$$

$$\mathbf{P}^{ri}{}_{k+1|k} = \mathbf{J}_x \mathbf{P}^{ri}{}_{k|k}$$

$\mathbf{J}_x$ is the Jacobian of **f** with respect to **X** and $\mathbf{Q}_k = E[\mathbf{q}_k \mathbf{q}_k^T]$.

5.5.3 Integration of new features

Features that are detected in the environment are compared to the old ones in a validation gate, which defines if a new feature is recognized as an old feature. If not, a new feature is defined as $\mathbf{L}_{new}$ and added to the system vector

$$\mathbf{x}_{N+1} = \mathbf{m}(\mathbf{x}_{k+1|k+1}, \mathbf{L}_{new})$$
$$= \begin{bmatrix} x_{r_{k+1|k+1}} + \rho\cos(\theta_{r_{k+1|k+1}} + \theta) \\ y_{r_{k+1|k+1}} + \rho\sin(\theta_{r_{k+1|k+1}} + \theta) \end{bmatrix}$$

$$\mathbf{x}_{k+1|k+1} \longleftarrow \begin{bmatrix} \mathbf{x}_{k+1|k+1} \\ \mathbf{x}_{N+1} \end{bmatrix},$$

$$\mathbf{P}^{N+1N+1} = \mathbf{J}_{x_r} \mathbf{P}^{rr}_{k+1|k+1} \mathbf{J}_{x_r}{}^T + \mathbf{J}_z \mathbf{R} \mathbf{J}_z{}^T,$$

$$\mathbf{P}^{rN+1} = \mathbf{P}^{N+1r^T} = \mathbf{P}^{rr}_{k+1|k+1} \mathbf{J}_{x_r}{}^T,$$

$$\mathbf{P}^{N+1i}_{k+1|k+1} = \mathbf{P}^{iN+1}_{k+1|k+1}{}^T + \mathbf{J}_{x_r} \mathbf{P}^{ri}_{k+1|k+1}{}^T.$$

The equations above show the updates need to incorporate the new feature into the system. The matrices $\mathbf{J}_{\mathbf{xr}}$ and $\mathbf{J}_{\mathbf{z}}$ are the Jacobians of **m** with respect to the robot state $\mathbf{x}_r$ and to $\mathbf{L}_{new.}$

5.5.4 Reobservation of old features

If an observed feature is recognized as an old feature the state need to be updated for both the robot and all features in the map. Distance defined as

$$\bar{x}_i = x_{i_{k+1|k}} - x_{r_{k+1|k}}$$
$$\bar{y}_i = y_{i_{k+1|k}} - y_{r_{k+1|k}}$$

gives the observation model

$$\mathbf{x}_{ik+1} = \begin{bmatrix} \rho_{ik+1} \\ \theta_{ik+1} \end{bmatrix} = \begin{bmatrix} \sqrt{\bar{x}_i^2 + \bar{y}_i^2} \\ \arctan\dfrac{\bar{x}_i}{\bar{y}_i} - \theta_{r_{k+1|k}} \end{bmatrix} + \mathbf{n}_{zi} = \mathbf{h}_i(\mathbf{x}_{k+1|k}) + \mathbf{n}_{zi}$$

The white noise n_{zi} has covariance $\mathbf{R}_i$. If N features are observed the model becomes

$$\mathbf{z}_{k+1} = \begin{bmatrix} \mathbf{z}_{1k+1} \\ \vdots \\ \mathbf{z}_{Nk+1} \end{bmatrix}, \mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_N \end{bmatrix}, \mathbf{R}_{k+1} = \begin{bmatrix} \mathbf{R}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{R}_N \end{bmatrix}$$

And the update step of the EKF is

$$\mathbf{x}_{k+1|k+1} = \mathbf{x}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{z}_{k+1} - \mathbf{h}(\mathbf{x}_{k+1|k})),$$
$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_x)\mathbf{P}_{k+1|k},$$
$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{H}_x\mathbf{P}_{k+1|k}\mathbf{H}_x^T + \mathbf{R}_{k+1})^{-1}$$

where $\mathbf{H}_x$ is the Jacobian of **h.**

5.5.5 Data association

Data association is probably the most critical aspect of the SLAM problem. This is where the decisions are taken about if a feature is new or if it can be found among the previously detected features that define the map. To associate measurements to features a gating approach is used that incorporates both robot uncertainty and feature uncertainty. This creates the innovation matrix Si:

$$\mathbf{S}_i = \mathbf{H}_{x_i} \begin{bmatrix} \mathbf{P}^{rr} & \mathbf{P}^{ri} \\ \mathbf{P}^{ir} & \mathbf{P}^{ii} \end{bmatrix} \mathbf{H}_{x_i}^T + \mathbf{R}_i$$

where

$$\mathbf{H}_{x_i} = \frac{d\mathbf{h}_i([\mathbf{x}_r \mathbf{x}_i])}{d[\mathbf{x}_r \mathbf{x}_i]}$$

If the innovation is defined as $\nu_i = \mathbf{z}_i - \mathbf{h}_i(\mathbf{x}_i)$ the validation gate is defined by

$$\nu_i^T S_i^{-1} \nu_i \le \lambda$$

Where λ sets the limiting value that defines the space the measure-ments needs to be in to be recognized as the already known feature.

The description above gives the general parts of a solution to the SLAM problem. Other algorithms could also be added such as deletion of features that no longer are visible, and other ideas to improve the solution. Some of these ideas are described in section 5.8.

5.6 Specific problems when constructing a feature map

The specific problem of the feature based map is that it is necessary that correct features are found. Finding features can be really difficult in a cluttered environment since for examples lines that are defining walls are hidden behind furniture, bushes or other obstacles depending on the specific environment. It will then either be perceived as several features if the line is partially visible or not detected at all if the parts of the line are too small. When features are found the problem with data association arises. Is this a new feature or is it a feature that is already in the vector. If the validation gate is too big a feature that is new can be associated with an erroneous old feature, causing distortions in the map. If the validation gate is too narrow a feature that should be associated with an old feature is added as a new feature, which causes errors in the map, but also causes the map to be more complex. The possible quality of the feature extraction and data association is highly dependant on the choice of sensors and the quality of their signals which is also affected by the current environment, as described in chapter 2.

5.7 Advantages and disadvantages for the feature based method

The feature based map is more difficult to implement than the grid based map because of all computing extensive steps such as feature extraction and map matching. The map matching is probably the strongest advantage of the feature based method since this makes it possible to update the map and also estimate and correct the position of the vehicle to be more accurate and update the features to the actual environment to make them hopefully converge. Unfortunately data association is a critical part and convergence cannot be guaranteed. Borenstein states [Borenstein, 1996]: “Currently most work in map-based positioning is limited to laboratory settings and to relatively simple environments”.

5.8 Different ideas for feature based mapping

The method described above is a general stochastic mapping solution to the SLAM problem. This method suffers from the map scaling problem, which means that the map quickly grows complex and computationally heavy as the number of features increases. A direct implementation of stochastic mapping has an $O(n^3)$ complexity, where n is the number of features in the map. This creates problems in large environments where many features are present. P.Jensfelt [Jensfelt, 2001] has made an implementation that gives an approach to this problem.

Jensfelt has a minimalistic approach, which means that this map is as simple as possible. The number of features is decreased and therefore the map is not as detailed as it could be. This map is only used for robot localization, hence the detail is of limited interest, as long as the map can be used for pose estimation. The number of features is decreased by deleting features if they are not confirmed soon enough but also by deleting features that are too close to other features in the map. Jensfelt’s solution also uses a hierarchical map structure, where the map is divided into several sub maps. This makes it possible to update only the features that are present in the current sub map. The smaller size of the sub maps and the number of local features now makes computations less demanding. Jensfelt has used the different rooms on a floor as separate sub maps and transitions between sub maps are performed as the robot passes through a doorway.

Another solution to the map scaling problem that also uses sub maps is performed by Guivant & Nebot [Guivant & Nebot, 2001]. Their solution is a mathematical optimization of the SLAM algorithm by using a special form of the matrices during the prediction state and also by using compressed filters suitable since the map now is divided into local maps. The sub maps or local maps are in this case fixed quadratic areas that divide the full map. The features that are within the current map and within the 8 neighbor maps are updated at all steps. Their algorithms will not be presented further in this report, since this is only an overview to prove that it is possible to improve performance without a loss of quality.

5.9 Choice of mapping method used in this project

The statement in the end of section 5.7 and probably the knowledge about the more difficult algorithms needed for a solution the SLAM problem gave us the decision that this was not the first choice for an implementation of a map building algorithm for the less structured outdoor environment. The area is much bigger than a normal indoor environment and the number of features could become very large if a detailed map is required. The complexity of the feature based map grows with the number of features but the grid map only uses a predefined amount of memory and computing power with the selected resolution independent of the number of fine details in the map. Therefore an implementation of a grid based map was decided as the primary solution.

5.10 Summary

This chapter covers the theories behind the feature based mapping method and the SLAM algorithm. Point and line features are discussed. Point features could represent trees or lampposts in an outdoor environment, and line features could correspond to wall segments or other flat surfaces. The feature based map could be constructed by solving the Simultaneous Localization And Mapping problem, which also updates the position of the vehicle according to the map that is constructed. The SLAM algorithm consists of a number of steps. The map consists of a vector of all detected features and the vehicle itself. Features that are extracted are stored in the vector. By comparing the newly extracted features to the features that already are in the vector it is possible to associate some of these to features that are already stored, thereby only updating the map with the new information. This is known as the data associating problem and is probably the most critical and part of this solution since wrong decisions may result in a map with either too many features or features that are wrongly associated. Feature based maps have big advantages if used for localization of a vehicle, since this map represents the world as the vehicle sees it. Unfortunately this method is more complex, and therefore more difficult to implement than the grid based method.

6 TESTS AND RESULTS

This chapter shows a fraction of all tests that have been performed with the goal to build a map that resembles the environment as well as possible. The tests are performed at two different locations, either in Kungsängen or in the area surrounding the building BB1. These areas were described in detail in chapter 2. The current circumstances during all tests will be mentioned together with the control type used for the test.

Several tests have also been made to gather information on necessary properties that should be considered when performing autonomous map building. Some of these tests are also described in the following section.

During all tests the ATRV is controlled around a path through the environment either by a simple remote control system or by a follow program that makes the ATRV follow an object in front of it, this allows the user to lead the ATRV through an environment. The ATRV records sensor data during this phase and this is the data that later on is used for processing in i.e. MATLAB to create a map. No tests have been made that create a map in real time. This is discussed further in chapter 8.

Note that for all figures both x- and y- axes represent a meter scale if no other scale is noted in the figure or explaining text.

6.1 Kungsängen – first test of the system

The first test performed in Kungsängen was also the first test made with the fully equipped ATRV. The aim for this test was to record all sensor data so that an evaluation of all sensor performance could be made. The secondary aim was of course to have some data that could be used to build a map.

The ATRV was controlled by a simple control program that lets the user specify translation velocity in m/s and rotation velocity in radians/s. The ATRV runs at this speed until a key is pressed and new values are given. The ATRV was controlled to follow a path that is shown in the following picture.

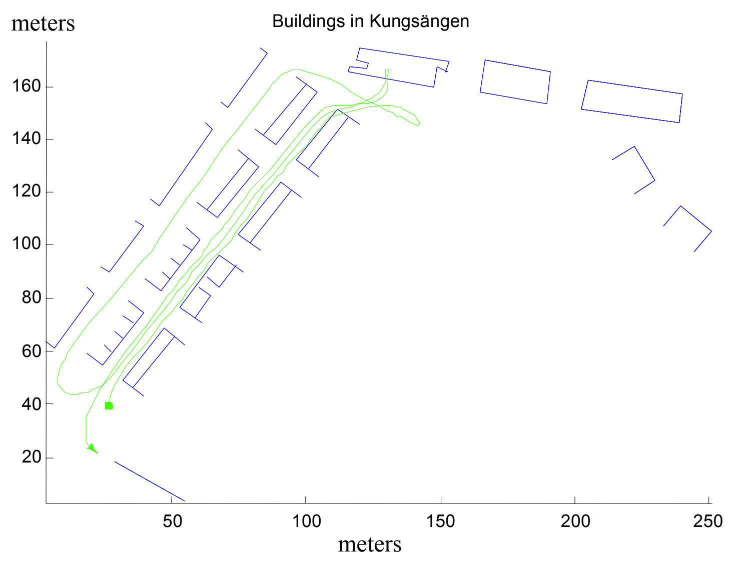


Figure 6.1. The path driven by the ATRV is visualized by the green line (the curved line that ends with an arrow)

As the ATRV was moved around it was realized that it turned to the right when the control was set to move it straight forward. By giving a constant rotation velocity of 0.1 rad/s it did almost move straight with a translation velocity of 0.7 m/s. The turnings will in later tests prove to be caused by improper tire pressure. Figure 6.2 shows the output of the odometry readings.

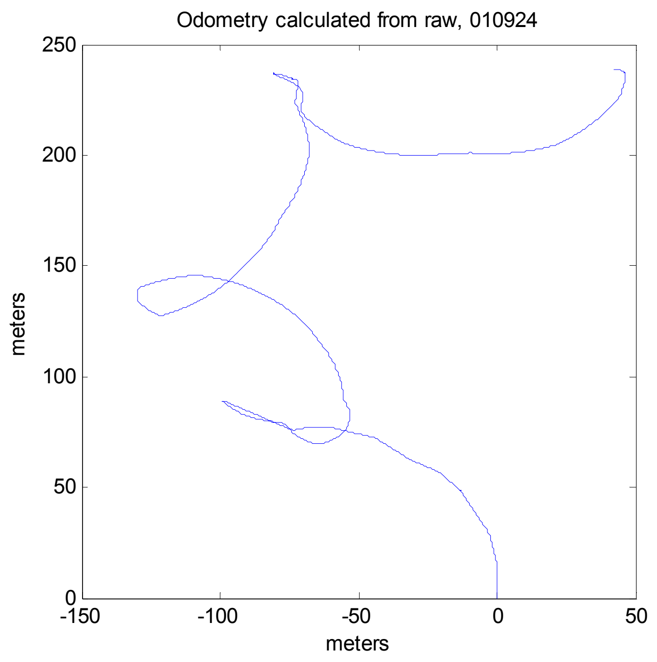
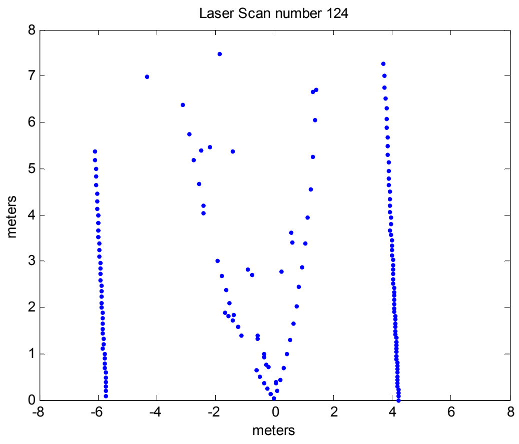


Figure 6.2. This picture shows the measured odometry from the first run in kungsängen.

The odometry values are as seen from the picture very strange. The starting point is at the bottom of the picture. When the ATRV was controlled to go straight instead of turning right, odometry data show that the ATRV is moving to the left. This strange behavior was studied in the following tests in section 6.2.

The laser readings did also show strange effects. Several obstacles were detected in front of the robot and in many different directions. Since the road was clear this is noise. Figure 6.3a and 6.3b shows these effects. The true walls that are detected are the straight lines to the left and to the right in 6.3a and the two perpendicular lines in 6.3b.



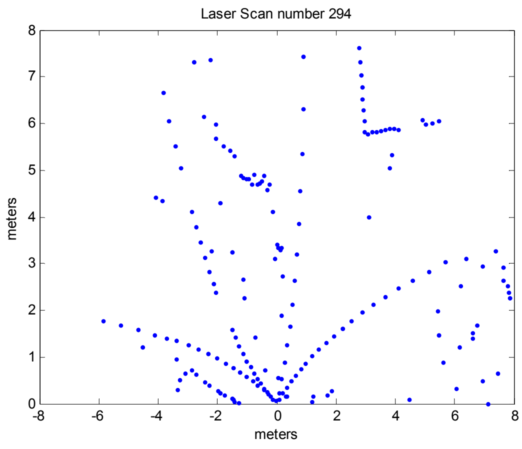


Figure 6.3a and figure 6.3b above shows the readings from the LMSat two different locations. The dots that form curved arcs in front of the ATRV is noise. In 6.3a the straight lines to the left and to the right is the true walls and in 6.3b the perpendicular lines in the upper right corner is the true walls.

As long as we have the errors in odometry and the noise in the LMS reading it is impossible to use these measurements to create a map of the environment. The scope for the following tests was therefore to find the source of these errors and try to minimize these errors to get the sensor values good enough to use for map building.

6.2 Corrections of errors in laser readings and odometry

Tests were performed in the corridors of BB1 to evaluate and solve the problems with the noise in the LMS measurements and to get the odometry data more accurate.

The noise in the LMS measurement proved to be caused by improper range settings. The settings were during the first tests in Kungsängen set to millimeter precision that gives a maximum range of approximately 8 meters. All measurements out of this range seem to project as noise in the range scale below 8 meters. When the precision was set to centimeters instead the maximum range value increases to 80 meters and the noise disappears. Since most measurements that now are out of range (above 80 meters) are detected as exactly 80 meters away the maximum range for the LMS is further on cut off at 50 meters. No measurements above 50 meters are considered as reliable readings.

The odometry was tested in the corridors. When the robot was programmed to go straight by turning slightly left the measurements from the odometry were recorded as the robot moved to the end of the corridor and turned and moved back, see figure 6.4a. The fact that the robot turned even as it was programmed to go straight proved to depend on inappropriate tire pressure. Since the ATRV has four air filled wheels, the air pressure affects the diameter of the wheels thus making the robot to turn if the air pressure is higher on one side than on the other.

By letting air out on the side that the ATRV was turning from, the robot now moves straight. See figure 6.4b.

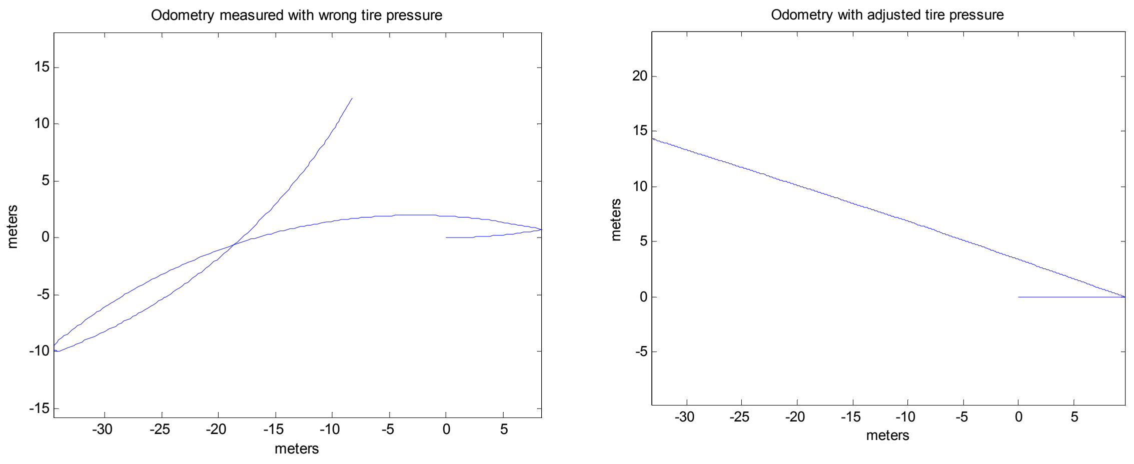


Figure 6.4a shows the odometry measurements as the robot moves back and forth in the corridor. Figure 6.4b shows how odometry has been adjusted by letting out air from the wheels on one side. In this test the robot does not turn back at the second end of the corridor.

6.3 Measurements around the building BB1, Test1

The following test was performed to evaluate how the sensors work in this second environment. This environment was used for every day testing since the location in Kungsängen only was available at two specific times during the work with this project.

The ATRV is controlled using the same type of program as in the first test in Kungsängen. The tire pressure is now adjusted so that the robot moves straight. The LMS is also set to the correct value, which gives it a maximum range of 80 meters with a precision of +/- 5 cm.

The issue of this test was to test the new odometry and laser settings and collect data that could be used to build the first grid map.

As seen in fig. 6.5a the odometry now reports straight lines as straight lines but the turning is still wrong. The graph should be a closed path since the robot starts and stops at the same location. By some tests it shows that the odometry reports 5.5 radians after turning a full circle, which should give 6.28 radians. This phenomenon can also be seen in fig. 6.4b. The orientation is wrong with a linear factor of 1.13. Figure 6.4b shows the odometry values corrected by this factor. The path is now almost closed.

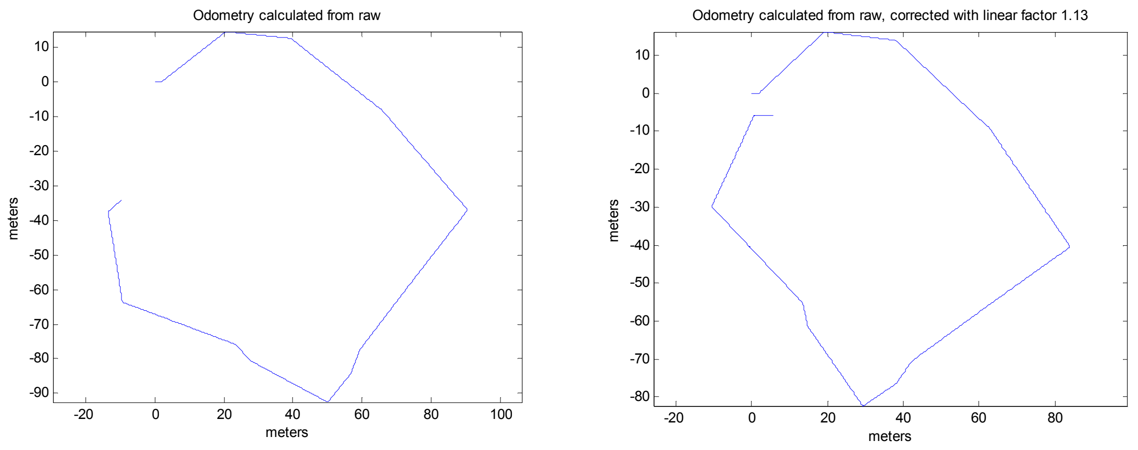


Figure 6.5a to the left shows the measured odometry as the ATRV has travelled clockwise around the building. Figure 6.6b to the left shows this with values corrected by a linear turning constant of 1.13.

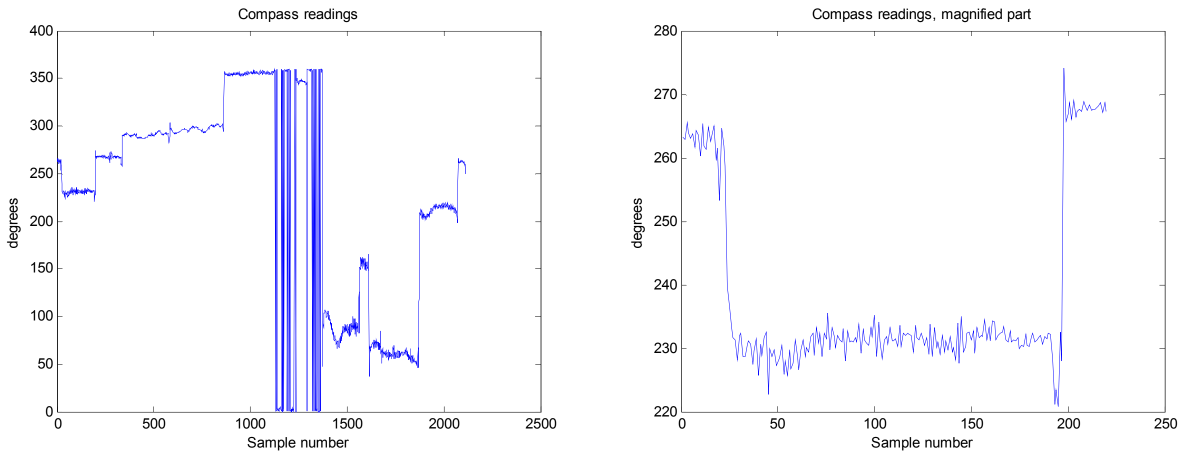


Figure 6.6a shows the measured compass readings. Figure 6.6b is a magification of the first part of the readings from the compass. This shows that the compass values has noise with an amplitude of up to 10 degrees.

The readings from the compass are shown in figure 6.6a. The compass values show noise with amplitude of up to 10°. Most noise seems to occur when the motors are running. Due to these problems the compass is not used for the pose tracking of the robot in this project. A further discussion of compass values is therefore not covered in this report.

The GPS was also tested and the measurements are shown in figure 6.7. Unfortunately the result from this is not looking good. At some points the GPS values look all right but most of the time it jumps as much as 30 meters in position. Some points do not seem to have a value at all. The reason for this is probably partly due to the tall walls of the building. This causes some satellites to be blocked thus giving a bad position estimate. But since even some readings far away from the building showed jumps this gave us the conclusion that a better GPS, preferably a differential GPS is needed if it is intended to give position estimates for map building.

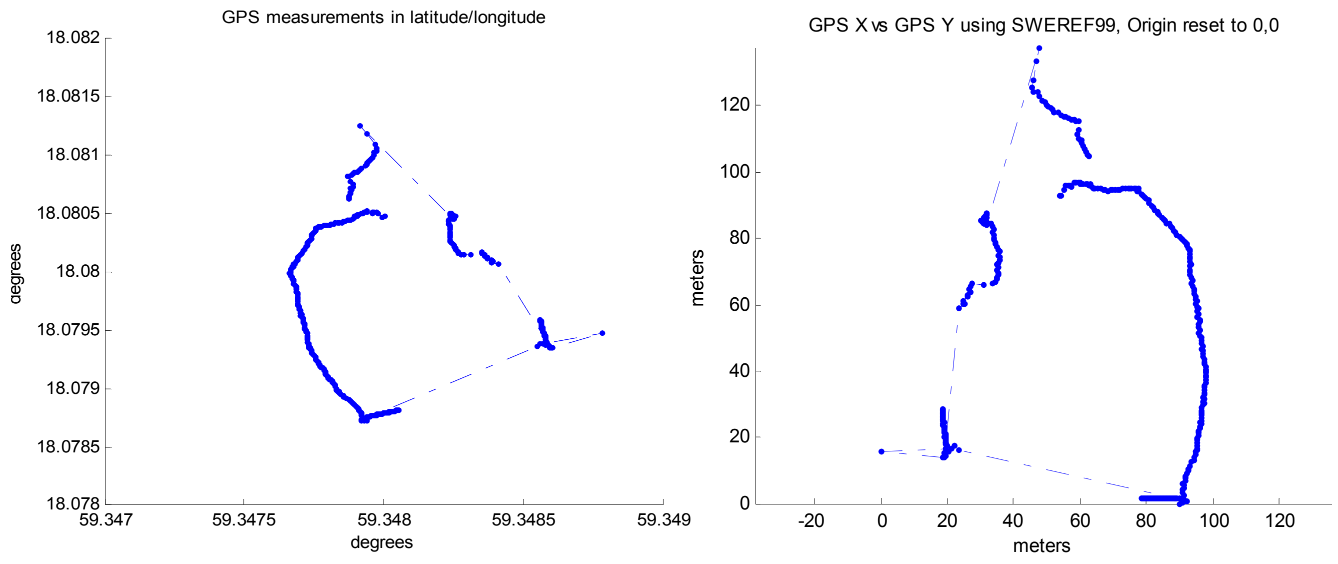


Figure 6.7a shows the GPS measurements as detected by the sensor. Figure 6.7b show these measurements transformed to metric coordinates. Both pictures shows that the GPS is instable with jumps and noise that makes this difficult to use for map building. The dot-dashed line shows areas where gps measurements are missing.

From the corrected odometry values and the scans from the LMS a grid map was constructed. This grid map is shown in figure 6.8. Because of limited computer memory the resolution was chosen to 50 cm. This means that the map divides the area into 50x50 cm squares that each holds the probability of a possible obstacle. In the map this is shown in different shades of black, where the darkest black indicates the highest level of certainty. The path for the robot is shown in red on the map. As seen in this map the path for the robot is not closed, as it should be. This is caused by the errors in odometry and gives a small mismatch in the map. The effect where the map is not closed is called the closed loop problem. It has been studied by many researchers and is considered a difficult task and a solution for this problem is out of the scope of this project.

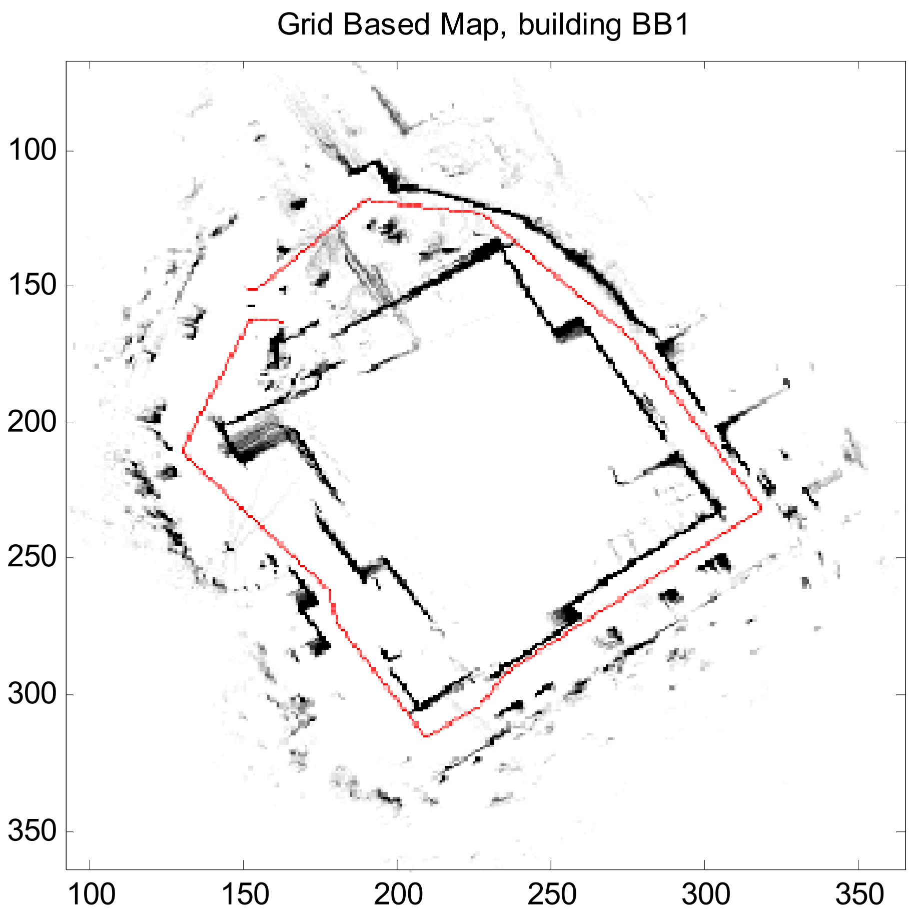


Figure 6.8. This figure shows the created grid map. The probability for obstacles are represented in different tones of black. The travelled path is also shown. The robot has moved clockwise along the path.

The robot has moved clockwise along the plotted path. This can be noted since some corners are missing. The LMS is pointing forwards and therefore some corners that are passed are missed. The map shown in figure 6.8 can be compared with the sketch showing the outline of the building in figure 2.2. This map is rotated 45° clockwise compared to the sketch in chapter 2.

6.4 New tests around the building BB1, test 2

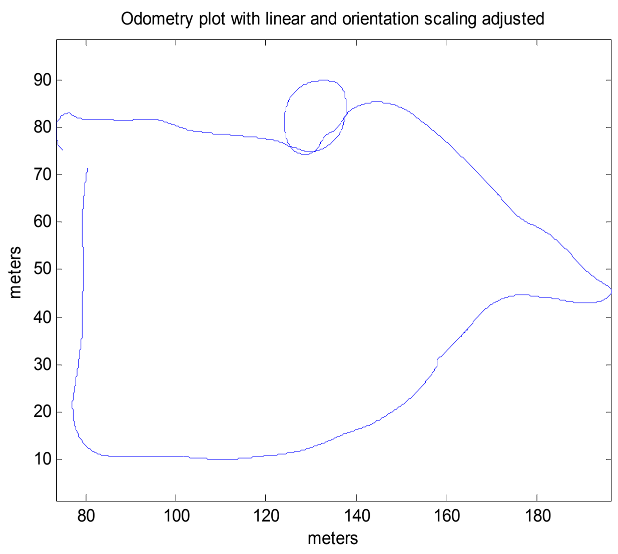
New tests were made in the surroundings of the building BB1. This time a new GPS is purchased that is equipped with a differential receiver that would improve the signal. A comparison between the new and the old GPS is made.

The ATRV is in this test controlled for the first time with the follow algorithm. This allows a person to walk in front of the robot and guide it through the area that is to be mapped.

The odometry is now adjusted by setting hardware parameters for orientation scaling so that the turning angle is correct. That means that when turning the robot a full revolution odometry shows 6.28 radians. The linear scaling is also calibrated so that one meter is measured as one meter.

Figure 6.9a shows the odometry readings for this run. The values are much smoother now as the robot follows the movement of a person. The robot has moved clockwise.

The old and new GPS are compared in figure 6.9b. The measurements from the old GPS are shown in cyan and the measurements from the new GPS are shown in red. The figure show that the new GPS is much more stable than the old one but still values are missing in a number of areas. The dashed lines are drawn where no readings are available. The conclusion for this is that the building is too tall when we are moving near the walls since the signal are good at the peak that is far away from the building. This was discussed in section 3.2.3. Unfortunately this means that we cannot use the GPS as an improving sensor for pose tracking in this test area.



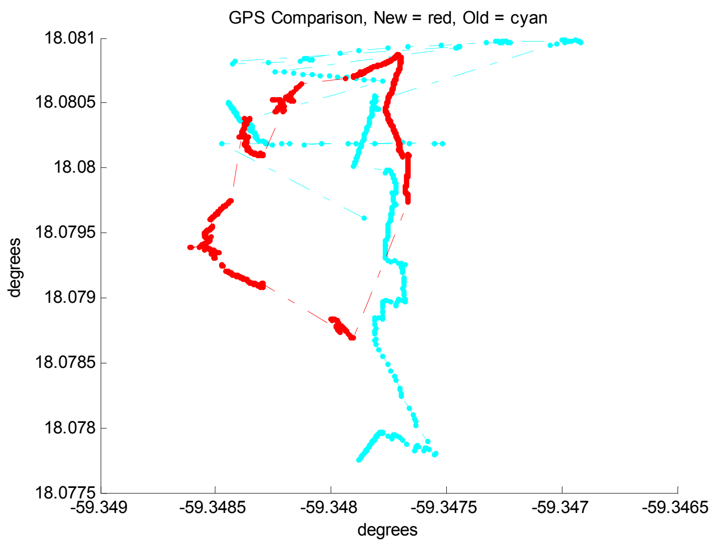


Figure 6.9a shows the odometry for this test and figure 6.9b shows the GPS measurements for both the old and the new GPS. The old GPS is shown in cyan (the lighter tone) and the new in red (the darker tone). The dashed lines are drawn were no GPS signal was detected.

The map in figure 6.10 was made in the same way as in the previous test by using only odometry. This map shows that the odometry values are not as good as they seem in figure 6.9a. Since the robot moves more freely now with the follow algorithm, many more adjustments are made than in the case with the remotely controlled system. All these adjustments add up to the odometry error. This is visible as angular distortions in the map. This map is turned 180° compared to figure 2.2.

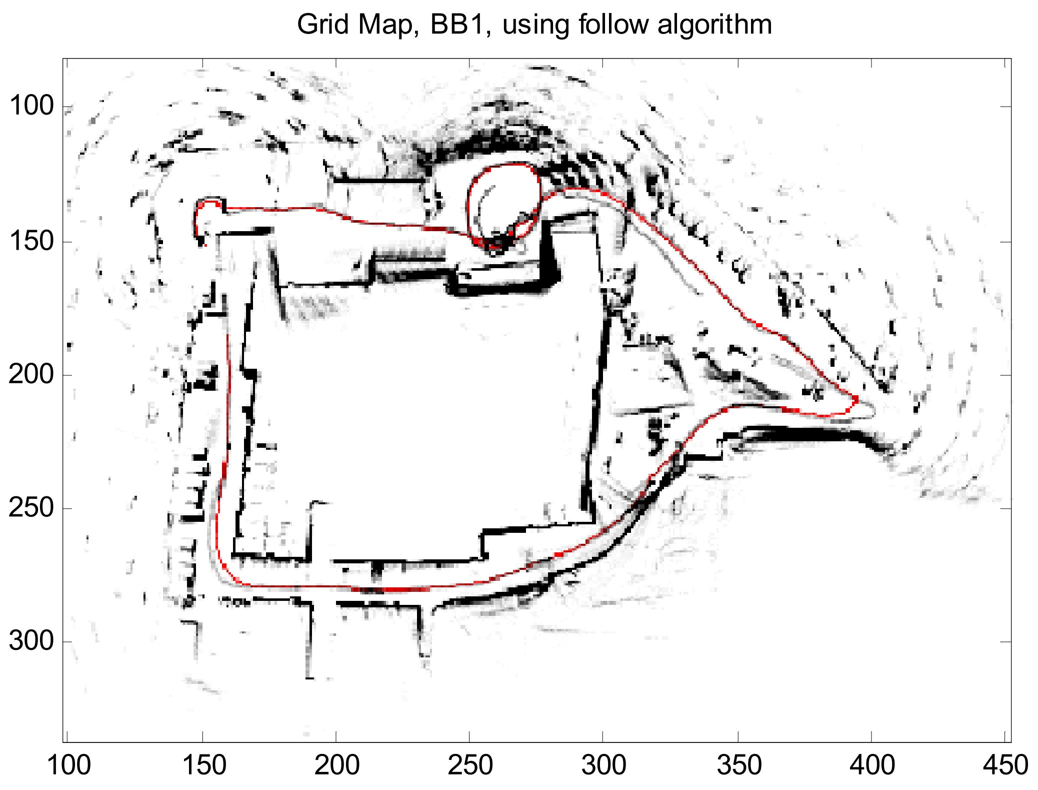
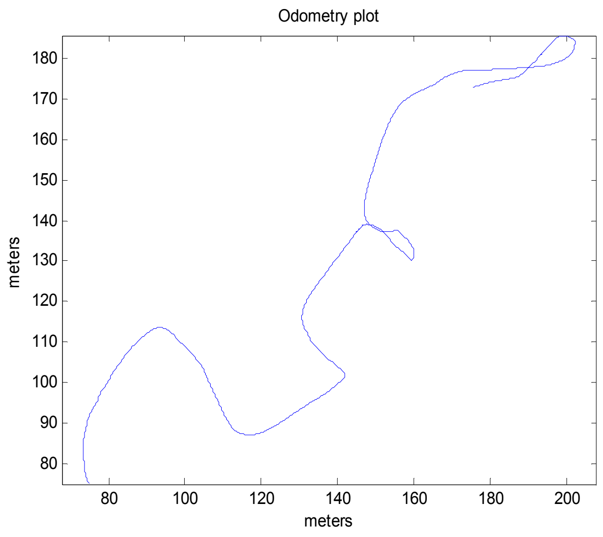


Figure 6.10. This is the grid map created from odometry and LMS measurements in the environment. The path of the robot is also and was traversed clockwise. The angular distortions was caused by odometry errors that are bigger than in the previous because of the follow algorithm.

6.5 Kungsängen – second and last test in this environment

This time much effort has gone into checking and making sure that all parameters were adjusted properly. The LMS is set to correct resolution, the tire pressure was adjusted and the linear and orientation scaling was set correct. This was supposed to be the final test, which would give the best possible measurement that could be used for future map building. The new GPS is also used in this test. Unfortunately the battery run out before all necessary tests were performed, but the partial measurements were used for the evaluation in this chapter. Experience is drawn from all mistakes and problems. These are collected in a checklist in the end of chapter A.

The control system for this test was the follow algorithm previously described. As noted in the previous test around BB1 the odometry gives poorer results with the follow algorithm. This effect is notable in this test, as the odometry values have closed loops where they should not be. This can be compared to the GPS measurements in figure 6.11b. Odometry readings are shown in figure 6.11a. The odometry errors are additionally increased by the rough ground surface in Kungsängen. The effects of the odometry errors are easily discovered in the grid map in figure 6.12. The GPS measurements are better than around BB1. The signal is never lost and the noise is less than earlier. The jumps that can be studied in the plot are probably caused when the number of satellites in view has changed thus changing the estimated position. Figure 6.11b is rotated 90 degrees clockwise compared to figure 6.11a.



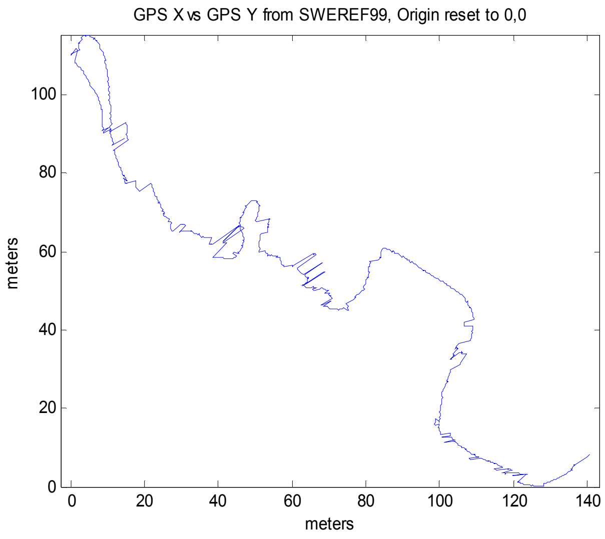


Figure 6.11a shows the measured odometry from the second test in Kungsängen. Figure 6.11b shows the GPS measurements transformed to metric coordinates. The jumps in the GPS signal is probably caused when the number of satellites in view change.

A grid map was built by using the odometry values and the measurements from the LMS, as in the earlier tests. The resulting map is shown in figure 6.12 on the next page. The error in odometry causes the lines in the map to overlap in many different directions. This can be noticed especially after sharp turns where the walls should be aligned but are not. The closed loop in the middle of the loop shows the most obvious error.

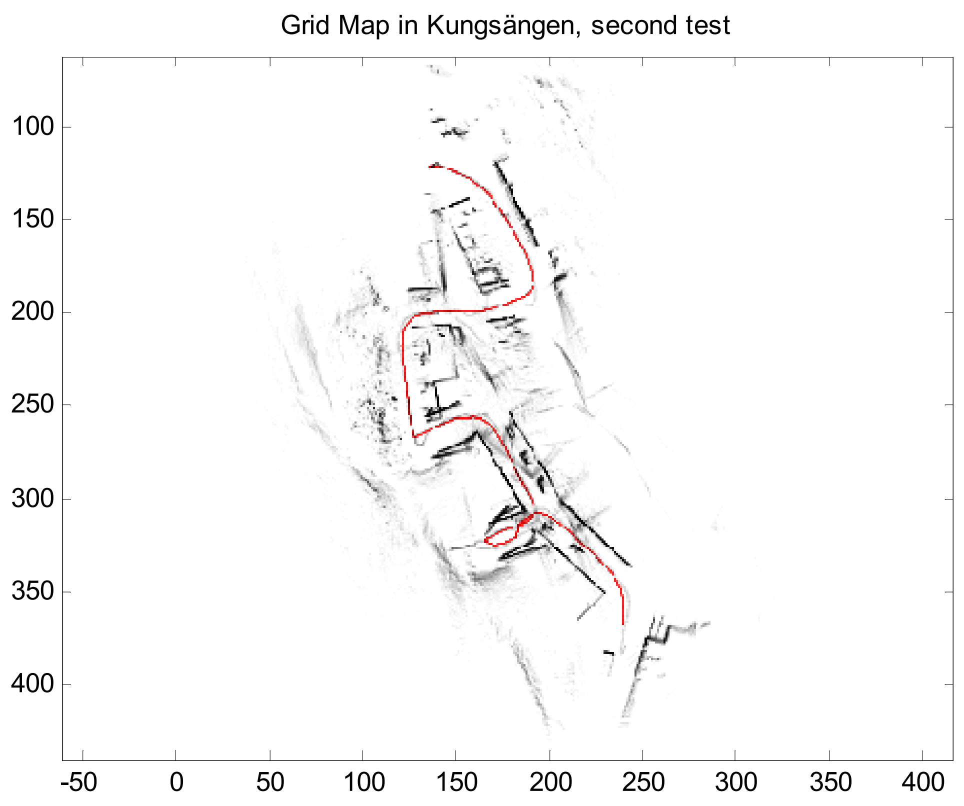
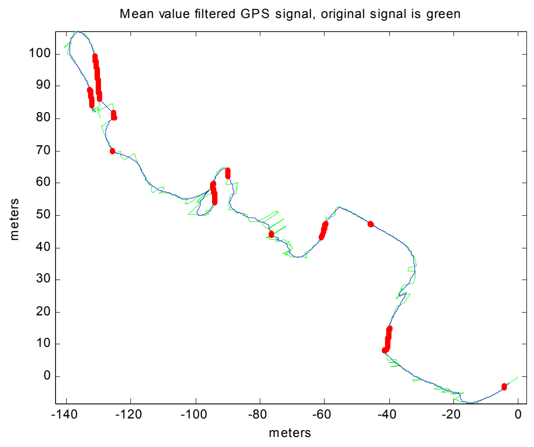


Figure 6.12 shows the created grid map from the second test in Kungsängen. The red line is the estimated odometry position for the robot. The black part shows the detected obstacles. The errors in position estimation is clearly visible in this map since the same wall is detected in many different locations.

Further work was performed to process the GPS signal so that it could be used together with odometry. Our idea is to use the GPS signal to align the odometry data to a number of fixpoints given by the GPS. The jumps shown in figure 6.11b could causes big trouble since this would cause the odometry to follow these jumps. The GPS signal is therefore filtered by a mean filter that smoothes the GPS signal in the desired filter window. The filter window was after some evaluation set to 60 samples, which mean that the new signal holds values that corresponds to the 60 surrounding measurements, thus giving the signal a smoother appearance. The filtered GPS signal is shown in figure 6.13a, where the original signal is shown in green (the jagged curve) and the filtered GPS signal is shown in blue (the smoother line). Figure 6.13b shows the variance distribution for all samples. The red (horizontal) line shows a limit of a variance of 2 meters. Regions that have low variance are more stable than regions of high variance and therefore measurements in these regions are more reliable. Regions that have a variance of less than 2 are marked in figure 6.13a as red (thicker) dots.



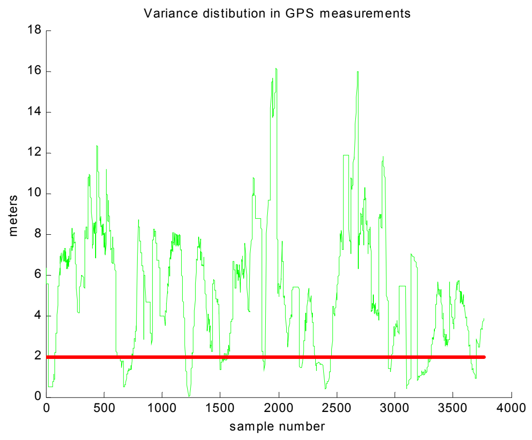


Figure 6.13a shows the mean value filtered gps signal in blue (the smooth curve). The original signal is green (the jagged curve). The red (thicker) dots shows regions with low variance. The level for the variance in these regions is below 2 meters, which is shown by the red (horizontal) line in figure 6.13b. This figure shows the variance for all samples.

An experiment was then performed, where the filtered GPS was used to align the odometry signal by adjusting its rotational angle and translation scale. It proved that the best results were achieved when the GPS signal only was used at few points to adjust the odometry signal. Since the odometry signal is continuous and rather accurate over small areas it is enough to "straighten" its path at a number of points. If the odometry is adjusted too often the continuity of the signal is broken down to look more and more like the GPS signal, but since we know that the GPS is noisy this is not a good idea. Figure 6.14 shows odometry and GPS signals combined to a new signal.

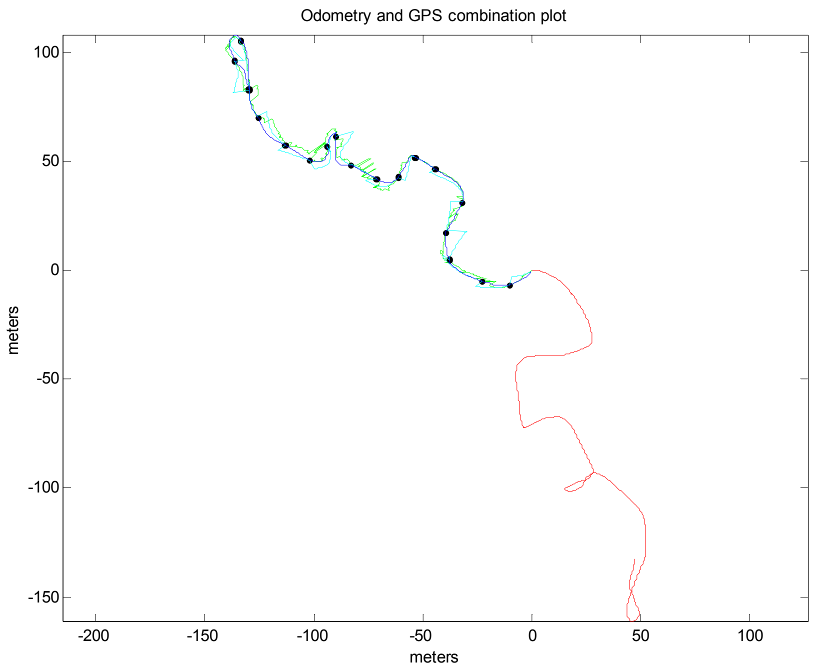


Figure 6.14 shows the combined signal from odometry and GPS in blue (the smooth line with black dots in the upper corner). The red signal (lower right) is odometry values. The green signal is GPS values and the light blue line segments show the aligned odometry values prior to scale and angle adjustments (the jagged lines behind the curve in the upper corner.

The new signal is constructed by adjusting the scale and rotation of the odometry so that it aligns with the GPS signal at a number of points. Figure 6.14 shows the odometry aligned to the GPS signal in eighteen intervals. The light blue (shown in background) line segments show the direction and scale of the odometry signal prior to the adjustments. The combined signal from odometry and GPS has been used for the grid map shown in figure 6.15.

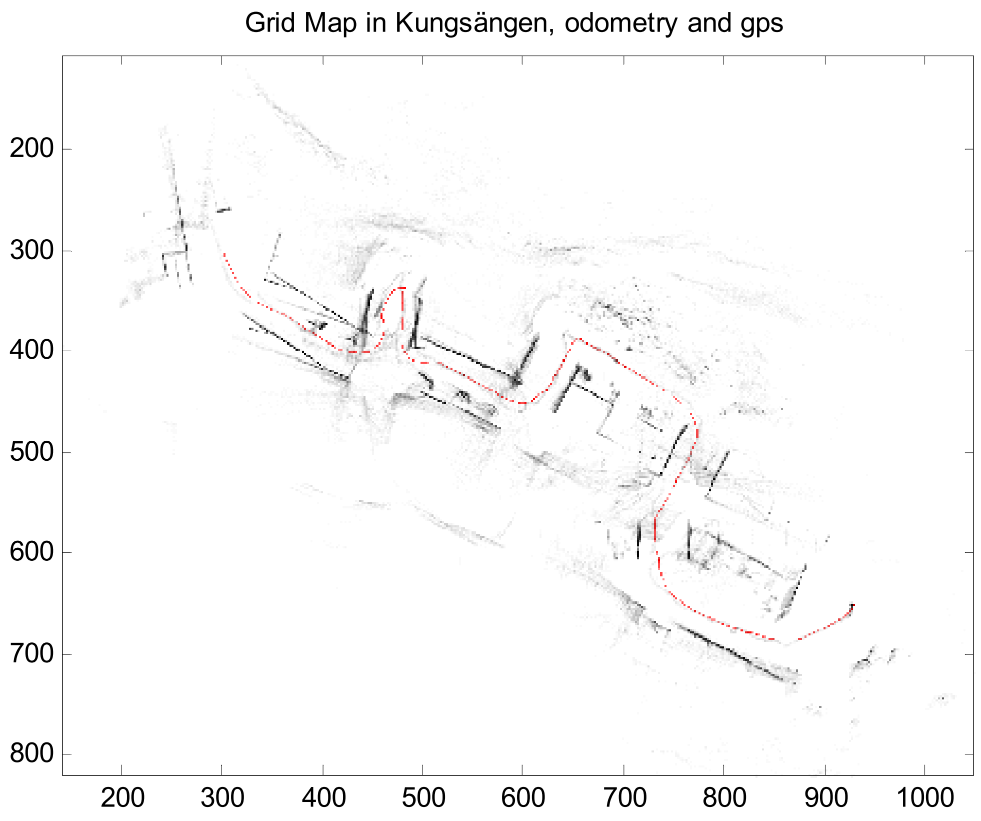


Figure 6.15 shows the gridmap constructed from GPS and odometry sensor data. The figure shows that the wall segments of the road are better aligned than in earlier plots. Each pixel represents 10x10 centimeters.

Even though the map now appears noisier than the last gridmap it is possible to see the effect the GPS signal had on the map. The walls of the buildings surrounding the road are aligned better than earlier. A comparison with the true environment can be done by viewing figure 2.1.

6.6 Areas marked as explored/unexplored

It is of great interest to mark areas that already have been explored since this gives the autonomous system the possibility to avoid areas that already have been explored. This experiment was performed very late in the project and therefore only a short test was made to show this idea. Figure 6.16 shows the result from this experiment.

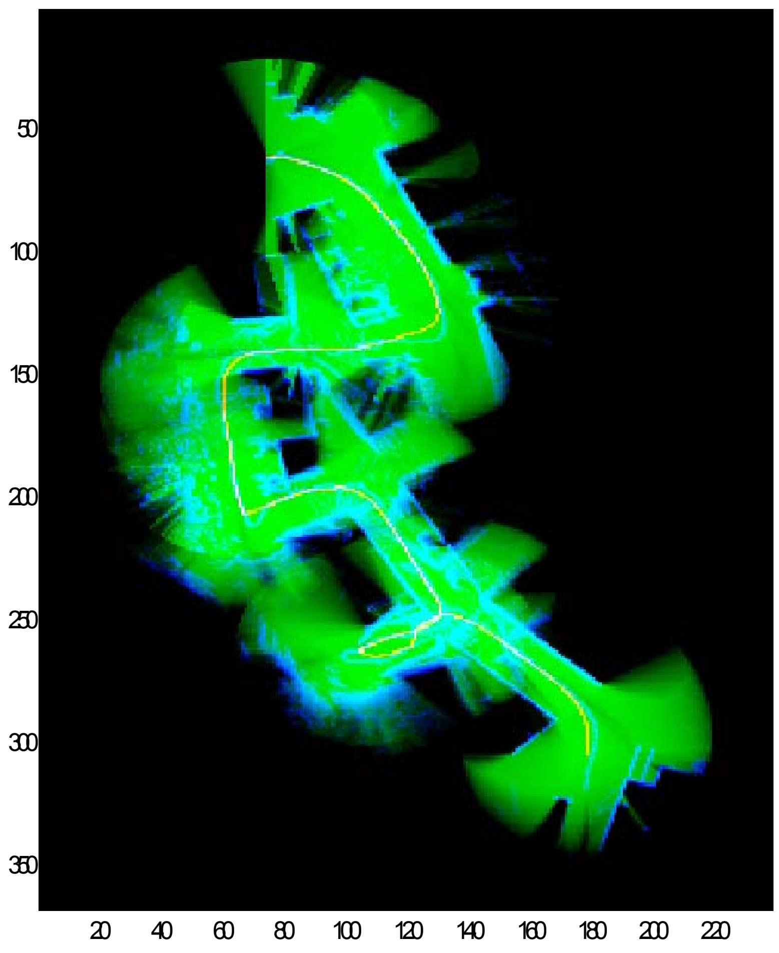


Figure 6.16 shows unexplored areas marked in black and explored areas marked in different levels of green. The saturation of green reflects the number of times different areas have been viewed by the robot. Each pixel in the map represent 50×50 centimeters.

To store information about areas that have been explored the region that lies in between the vehicle and the recorded laser range measurements are set. All these regions are added together and therefore regions that have been viewed more times are colored in a more saturated green color (lighter shade). This information could be used to find unexplored areas in the environment as discussed in appendix A. For simplicity is only odometry used for the position of the vehicle in this experiment. The grid size is set to 50 centimeters because of computer limitations.

6.7 Summary

This chapter describes the different tests that were performed to increase the systems map building capabilities. In the first sections many problems with hardware was detected, such that erroneous turning angles depended on the tire pressure of the vehicle and the importance of having the laser set to the correct range. The compass was tested for use with the system but its noisiness caused us to occlude this sensor. As expected the tall walls of the building BB caused the GPS sensor to give noisy and insufficient position estimates and therefore this sensor was only used for the final map in Kungsängen. The GPS and odometry measurements were combined in the late tests to show how this could be used to correct the diverging odometry measurements.

7 CONCLUSIONS

The problem in this project was to equip a robot with a system for mapping an outdoor environment. Our experiments show that it is possible to draw a map of an unknown environment, even if the quality of the map differs with the quality of the measurements. The problem was solved by equipping a vehicle with a laser range finder and a GPS that was used together with the odometry sensors of the vehicle to build a map using a grid based map method. The laser range finder showed a good reliability and showed precision even over distances of 50 meters. This gave the conclusion that the position estimation of the vehicle is of high importance for map building since this describes where the laser measurements are mapped. Locally, the odometry sensor gave a good precision but as wheel slippage caused the odometry to diverge over time it is necessary to complement odometry with at least one additional sensor to compensate for these errors. When combined with GPS it seems possible to draw the conclusion that the accuracy of the map globally depends on the accuracy of the GPS position estimation. Unfortunately, the GPS is not fail-safe. Tall buildings and other environmental circumstances change the number of satellites visible to the GPS, which causes the estimation of the position to change. These changes can be seen as ”jumps” in figure 6.11b. Since these variations sometimes are as large as a couple of meters this map is only acceptable in coarse maps and not in maps needed for sub meter precision. To solve this our conclusion is that more work is needed to further improve the map. One suggestion is that extra sensors should be applied, for example an inertial sensor that could work as a stabilizer to filter the position information. Another suggestion is that it would be of great interest to change the software so that new sensor data is matched to the data already stored in the map, as in the SLAM algorithm mentioned in chapter 5. Unfortunately no time was left for this. This project gave a good experience in the work of equipping a vehicle to perform a certain task, but it also gave the understanding that developing a robust autonomous system requires solutions to many problems that were not thought of and all of these take time to solve. Therefore many interesting aspects of this problem have unfortunately been left unexplored due to the limited time of this project. Some of these are however mentioned in the following chapter as future work.

8 FUTURE WORK

The problem of autonomous map building is quite extensive and only a few aspects of this problem are covered in this report. Many ideas have appeared during the work with this problem, either from reading or by experience. Some of these ideas will follow in this section, as suggestions for future work.

8.1 Improving the map building

Some work could still be done to just optimize the parameters used to give better results, but to get a better quality of the map and to give it a better and more intuitive structure here are some ideas that could increase performance.

8.1.1 Integration of additional sensors

The use of several sensors that work together could be a quite complex task. Although more sensors could dramatically improve the pose tracking of the robot, thus giving us a more accurate localization, which is necessary for the grid based methods. M. Sørensen [Sørensen, 2002] has made some work on pose tracking with the use of odometry, GPS and an inertial sensor. The inertial sensor can give additional information about pitch, roll, yaw and accelerations in different directions. Some work has also been done using stereo cameras to build a map of the terrain. All of these sensors give different kinds of information that could be used for map building and an integration of all would be interesting, even though the complexity of the system grows with every additional sensor as mentioned earlier.

8.1.2 Adding map matching

A great problem with our solution is that it lacks the possibility of matching the sensor data to the already stored map. If it was possible to adjust the map to older data this would help the map converge. By implementing the SLAM algorithm mentioned in chapter 5 it would be possible to solve this problem. Unfortunately this is left as further work for time reasons.

8.1.3 Combining grid and feature based mapping

By combining the ideas from the feature based mapping and grid based mapping it could be possible to make a better approach to feature detection. If local partial maps are created using the grid mapping technique, we can use the created “bitmaps” to detect and segment features using image-processing techniques. The features that are detected can then be stored in a list as for the feature based technique. The parts that contain elements that are not detected as features are stored as bitmap clusters in an unknown features list that later could be used to add new known features to the map building system. This would improve the detection of features since the grid based maps are constructed from several measurements from different angles and therefore make a more probabilistic base for feature detection. Detecting features from a grid based map is an image segmentation problem. M. Sonka describes this further [Sonka, 1998].

8.1.4 Hierarchic map structure

To make the map representations to be more dynamic the map could be broken down into smaller pieces. For the matrix representation the matrix could be divided into smaller sub maps that could be used individually. By only using the sub maps that correspond to the area where the autonomous system is situated and the sub maps of the closest environment the map is easier to handle. The limit of how large a sub map should be can be determined by the maximum range of the visibility sensors of the system.

The idea of a hierarchical system is even more natural to the list. This could also be used to give a more structured view of the map. The reason why a hierarchical structure is interesting is that the list of objects may eventually grow very large. By grouping objects together in smaller lists the material is easier to handle. One way is to have different lists for different area cells as in the matrix representation. But it is also possible to group the objects together in a more structured way. For example a *structure* is several connected walls, a *building* consist of a minimum of four walls that form a closed boundary. Further, a number of buildings could be a *town*, and a composition of towns makes a *world*. In this way the map is structured into several hierarchical layers that are connected to each other. The higher objects contain lists of the object type below. Only objects in the near environment are accessible through their parent objects when necessary.

8.2 Making the map building more autonomous

There are different levels of autonomy that could be reached when creating a map. The ideas that have appeared during work are gathered in the following list. The tests made in this project only use the first two levels of autonomy.

8.2.1 Different levels of autonomy

1. *Remotely controlled* The robot is remote controlled using a joystick or a simple control program to guide the robot through the area that is to be mapped. Measurements are taken continuously and the measurements are used to create the map.
2. *Following a leader* In this step the follow algorithm is used. This makes the robot follow the person in front of it whilst measuring the surroundings to create a map. This is the highest level of autonomy used in this report.
3. *Traversing through waypoints without crashing* By giving a number of way points in the environment the robot can traverse through these points using GPS and odometry. These waypoints form a path in an environment that is to be mapped. To make this possible it is necessary to implement an obstacle avoidance behavior that could guide the robot around obstacles that block the way between the waypoints. This makes the robot drive autonomously between the given points. It is also necessary that the localization of the robot is good enough for the waypoints to be reached. This is easily done if the GPS signal is reliable.
4. *Explore a specified area* This is the highest level of autonomy. By a given area specified by for example a GPS boundary, the robot explores this area until a complete map over the area is developed. That is, it is necessary to detect areas that are not yet mapped and explore these further. This requires that the robot have the property of using the map it creates. By using this map it is possible to go back to areas that are not yet completely mapped and add refinements and additions to the map. Marking areas that have been mapped and look for boundaries between mapped and unknown areas can make this possible. With feature based mapping this can be done by looking for corners. All lines should have corners that connect to other lines. If a line just stops this is probably the evidence that this area is not mapped completely. For the grid based mapping it is necessary to add a layer in the matrix that is to be marked for all areas that are explored as in chapter 6.6. Filling the space between an obstacle measured with the LMS and the LMS itself marks this space as explored. By processing this layer together with the layer of obstacles it is possible to detect boundaries to areas that are not yet explored. The figure below

shows where unexplored areas are detected. For a further illustration of boundaries to unexplored areas view figure 6.16.

8.3 Summary

This chapter discusses the ideas for future work that have come up during the work with this project. Different ideas that could improve map building is additional sensors or an implementation that uses map matching to adjust the position of the vehicle according to the created map. Additional autonomity would be of interest to improve the map building. It is discussed that if it is possible to mark explored areas it would be possible to give the vehicle the task to map a given area, that for instance could be bounded by measurements from the GPS.

9 REFERENCES

E. Abbott and D. Powell, *Land-Vehicle Navigation Using GPS*, *Proceedings of the IEEE,* Vol. 87, No. 1, January 1999

Arkin, *Behavior Based Robotics*, Ch.2, The MIT press, Cambridge, 1998

J. Borenstein, H.R. Everett, L. Feng, *Where am I? Sensors and Methods for Mobile Robot Positioning*, Michigan, April 1996

P. Enge and P. Misra, *Scanning the Issue/Technology –Special Issue on Global Positioning System*, *Proceedings of the IEEE*, Vol. 87, No. 1, January 1999

J. D. Foley, A. v. Dam, S. K. Feiner and J. F. Hughes, *Computer Graphics –principles and practice*, Addison-Wesley Publishing Company, 1995

J. Forsberg, U. Larsson, P. Åhman, and Å. Wernersson, *The Hough Transform inside the Feedback Loop of a Mobile Robot*, Robotics & Automation, University of Luleå

M.W.M. Gamini Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba, *A Solution to the Simultaneous Localization and Map Building (SLAM) Problem, IEEE Transactions on Robotics and Automation*, vol. 17, No. 3, June 2001

J.E.Guivant and E.M. Nebot, *Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation, IEEE Transactions on Robotics and Automation*, vol. 17, No. 3, June 2001

P. Jensfeldt, *Approaches to Mobile Robot Localization in Indoor Environments*, Doctoral Thesis, Department of Signals, Sensors and Systems, Royal Institute of Technology, Stockholm, 2001 …

P. Misra, B.P. Burke and M.M. Pratt, *GPS Performance in Navigation*, *Proceedings of the IEEE*, Vol. 87, No. 1, January 1999

J. Skansholm, *C++ direkt*, Studentlitteratur, Lund, 1996

M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis, and Machine Vision*, Ch.5, Second edition, PWS Publishing, 1998

M. Sørensen, *Pose Tracking*, Master Thesis, Department of Signals, Sensors and Systems, Royal Institute of Technology, Stocholm, 2002

S. Thrun, *Robotic Mapping: A Survey*, School of Computer Science, Carnegie Mellon University, Pittsurgh, February 2002

G. Welch and G. Bishop, *An Introduction to the Kalman Filter*, Department of Computer Science, Chapel Hill R.C. Gonzalez and

O. Wijk, *Triangulation Based Fusion of Sonar Data with Application in Mobile Robot Mapping and Localization*, Doctoral Thesis, Department of Signals, Sensors and Systems, Royal Institute of Technology, Stocholm, 2001

R.E. Woods, *Digital Image Processing*, Ch.7, Addison-Wesley Publishing Company, 1993

G. Zunino, *Simultaneous Localization and Mapping for Navigation in Realistic Environments*, Licentiate Thesis, Centre for Autonomous Systems, Royal Institute of Technology, Stockholm, 2001

List of other useful litterature

Manual ATRV2

Manual Compass: *C100™ Compass Engine, Technical Manual*, KVH Industries, Inc., 1998

Manual LASER: *LMS 200, Laser Measurement System*, SICK

Manual Garmin GPS: *GPS 35 LP TracPak™, Technical Specification, Rev. E*, GARMIN, March 2000

Manual Trimble GPS: *DSM12™/212™ Operation Manual, Rev. A*, Trimble, May 1998

Manual Inertial System: *DMU User's Manual, Rev. 1.2*, Crossbow Technology Inc.,January 1999

Manual Communication Interface: Comtrol Corporation, *RocketPort Multiport Serial Cards, Software Installation and Configuration Guide*, September 1998

Manual Mobility graphical interface:

APPENDIX A

This chapter will serve as a good reference when specific information is needed that is not covered previously. For example all details about the robot used is described, both the hardware used and what software that needs to be involved. Sometimes the sections give examples on how to deal with specific subjects and otherwise references will be presented to literature that is recommended if more in-depth explanations or details are necessary. This chapter will also cover a list of problems that was encountered and a short "Checklist" that was used previous to the more advanced tests in this project.

A.1 The robot

The following section will cover most details about the robot, ATRV2, which was used throughout this project. This will give a good overview of the possibilities and limitations that could be encountered when using this type of robot. For people using other robots this section can be seen as a guide for comparison with other used systems. This section can also serve as a help to those who mainly are interested in the theories involved but also need to know what kind of limitations that might be encountered in "real life".

A.1.1 Overview

The Robot chosen for this project is an All Terrain Robot Vehicle or ATRV. It is a robot platform that could be used for research and development of solutions to numerous different tasks. The ATRV has 4 fixed air pressurized big wheels that are controlled by independent motors. The size of the robot is 105×80×65 centimeters and it weighs approximately 120kg. It is possible to load a maximum of 100 additional kilograms onto the system. This robot was chosen for this project because of its capabilities of moving outdoors and the size that makes it possible to attach additional sensors and actuators to the system. The system is manufactured by iRobot and the model is called ATRV2.

A.1.2 Hardware

This section will describe the ATRV in detail. Several aspects of the different hardware inside the ATRV will be covered. Additional details about the hardware can be found in the manual of the ATRV vehicle.

A.1.2.1 Power

Four motors control the robot, one for each wheel. This gives a rather powerful and flexible solution that gives the ATRV the properties to carry a lot of weight and also rotate on the spot if necessary. The motors are powered by two 12-volt batteries that are connected in serial, thus giving a maximum voltage of 24 volts. With standard equipment the batteries have the capacity to run the system for 4 hours. It is possible to use the output from the batteries as a power supply to additional equipment. One way is to connect it to raw power that gives 12 or 24 volts or fused through the computer, which gives a lower current at 5 or 12 volts.

A.1.2.2 Actuators

Actuators are parts of a robot that performs something on the environment. It could be a robot arm that could grasp something from the environment. In this system the only actuator that is used is the drive system, which is considered as an actuator since the robot sends signals to the motor control that makes the robot move around in the environment. The drive system on the ATRV uses skid steering. Compared to a car, for example, the ATRV has fixed wheels that cannot be used for steering in the normal way. Instead the ATRV uses the independent motors to drive the motors on either side faster than on the other side. This causes the ATRV to turn with the rotational velocity proportional to the difference in speed between the left and right wheel pair. There is also the possibility to turn on the spot by driving one side forward and the other backward. This causes the ATRV to ”skid” about its center. This type of steering uses the same type of principle as differential drive. The only difference is that differential drive uses only two wheels for steering. Differential steering is commonly used on round robots. The advantage of both of these methods is that it is possible to turn ”on-the-spot”, which makes navigation easier, since it is not necessary to consider the minimum turning radius of a ”normal” steering system.

A.1.2.3 Sensors

The robot uses sensors to perceive the environment. Sensors can be compared to our ears and eyes. The shipped ATRV has several already mounted sensors to measure movements and distance to nearby objects. These are described in this section together with some additional sensors that have been used with the system.

Odometry

Odometry is a sensor that is used to keep track of the robot position by the movements of the wheels. Odometry is said to be a dead reckoning sensor. The odometry sensor is a system of encoders that are placed close to the wheels. The encoders count the rotations of the wheels in fractions of a full revolution and this can then be used to calculate how far the robot has moved and in which directions. Since this is done frequently over time this can also be used to provide information about the speed of the robot. The ATRV can present odometry data in two ways. One way is as translation and rotation separate, as total distance traveled in meters and total rotation angle in degrees. The other way is as X and Y position. In this mode the movements are automatically mapped to the XY-plane so that the position of the robot can be read directly. Unfortunately odometry is not perfect. As the robot turns the wheel slippage causes errors in the estimations about the robot movements and as this accumulates the X and Y positions will diverge from the real value over time. Even though this is the case odometry is of good use when estimating positions over limited distances.

Sonars

Sonars or ultrasonic range finders are used to detect nearby objects. The ATRV has 12 sonars, 6 on the front panel, 2 on the rear panel and 2 on each side. Sonars use an ultrasonic speaker that transmits a sound pulse in the direction of the speaker. A microphone near the speaker then receives the pulse and the time elapsed since the pulse was sent until the echo was received is measured. This can then be used to calculate the distance to the object that produced the echo. The sonar receives echoes from objects that are within ~+/-15 degrees from the centerline of the sonar and at a distance of up to 10 meters. The sonars on the ATRV report the distances in meters. Sonars are often used for collision detection and since the front panel has 6 sonars it is easy to decide which direction to turn if obstacles are encountered.

Compass

The Compass sensor on the ATRV is a C100 Compass engine from KVH Industries. It is used to sense the heading as compared to the magnetic north pole of the earth. The sensor is working at tilt angles up to 20 degrees. The control system of the compass report readings in degrees with an accuracy of 0.1 degrees and give new measurements at a speed of up to 1 Hz. This sensor is easily disturbed by other magnetic fields, which could cause problems at certain areas. It is possible to detect if the detected magnetic field is either too high or too low compared to normal values from the magnetic north pole.

Unfortunately the delivered software does not support these features. Another problem seems to be that the motors of the ATRV disturb the compass. This is noted since the compass signal is stable when the ATRV is standing still but that it gets very noisy, as the ATRV starts moving. This is shown in the following figure. It is possible to use an internal damping filter in the compass system that uses signals over a time up 24 seconds to give a more stable signal. This feature is neither included in the software. As it is now the compass can be of good use when the ATRV pauses and the compass stabilize. Further information concerning the Compass can be found in the C100 manual.

Laser Range Finder

This is probably the most important sensor used in this project, since it acts as the eyes of the robot. The Laser range finder is a LMS291 from SICK. The LMS can measure the distance to objects in the environment in a horizontal 180° area with a resolution of up to 0.5 degrees and at distances up to 81.9 meters, with a speed of up to 5 measurements per second. This gives us measurements that can be used to create a simple obstacle map of the environment in front of the robot. The following figure shows a plot from a laser reading. Some further theory about the LMS is found in chapter 3.2. Details about the hardware can be found in the LMS manual.

GPS

To gain a more absolute estimation of the robot position a GPS can be used. GPS stands for Global Positioning System and uses up to 12 satellites to estimate the position to the receiver. The receiver uses the distances and positions of the received satellites to calculate the position. There are many different systems that use GPS for navigation, with varying quality and performance.

Garmin GPS

This system was shipped with the ATRV and is a quite simple system that should produce position readings with an accuracy of up to +/-15 meters. This GPS is a Garmin GPS 35LP. It produces readings with latitude, longitude and altitude with an update rate of 1 Hz. This GPS is not differential. Further technical details can be studied in the Garmin GPS manual.

Trimble GPS

This is an additional GPS receiver that was added to the system to give better GPS performance. It is a DSM212H™ GPS from Trimble that is equipped with a differential receiver. The differential receiver uses radio signals from a known fixed position to further correct the estimation errors from the GPS. The brand of the differential receiver is AZTEC. The device is described further in the AZTEC manual. The GPS with differential correction gives measurements with an estimation error of less than 1 meter. This GPS has a maximum update rate of 10 Hz. For more details, see the Trimble GPS manual.

Inertial Sensor

This is an additional sensor that measures accelerations and angular velocity on the body in all three dimensions. This can be used to estimate the pose of the robot. For example measure if the robot is located in a slope and how the robot is tilted. The accelerometers can also be used to calculate how the robot has moved if the accelerations are integrated twice. The sensor used is a DMU-FOG by Crossbow. It is described in the DMU user's manual.

A.1.3 Computer/Software

All the sensors and actuators are controlled by a computer. The computer that is shipped with the system is a Pentium PC with dual 800 MHz processors. This PC is used to store and execute all programs and data on the ATRV. The computer runs with the LINUX operating system.

Communication

Connection to this computer is primary done by an added wireless network card. The card is an Orinoco WaveLAN card. The Wireless network can be run in either through a WaveLAN bridge or in Peer-to-Peer or ”AdHoc” mode. In bridge mode many computers can be connected at the same time to the system as long as they are within range of the bridge. In peer-to-peer or ”AdHoc” mode one computer can be connected directly to the ATRV. This is the preferred when the system is tested on the field.

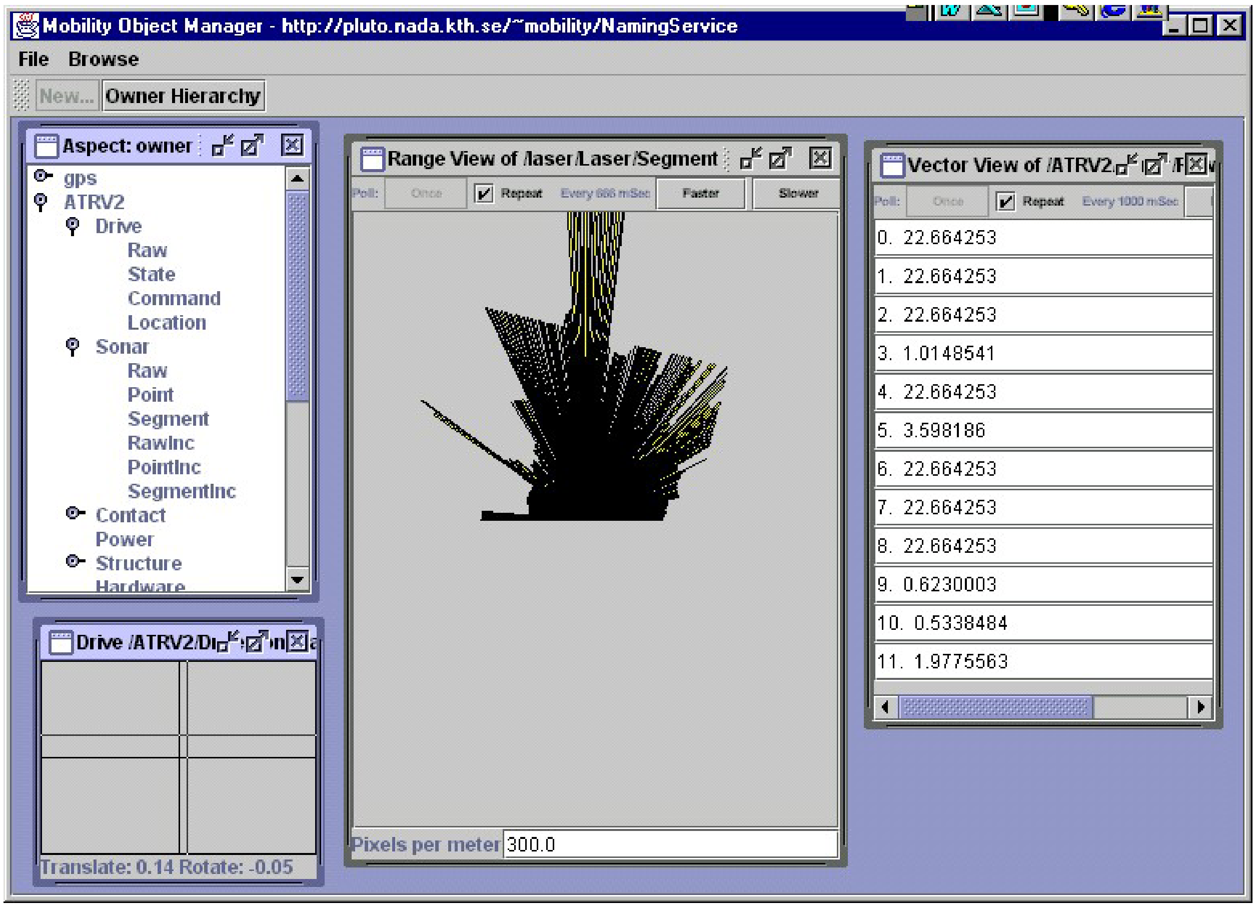
Connections

Connection of all sensors and actuators are done through the RocketPort. The RocketPort is a card mounted in the back of the computer with 8 serial communication ports. The speeds for these can be set independent of each other. The connectors for these ports are modular 4 pin connectors. Specifications about this card can be found in the RocketPort manual.

Software control

The software that is used to gather information from the sensors or to control the motors is based on COBRA. By using CORBA all sensors and actuators can be seen as Internet resources. A name server is running that is used to connect using names. Mobility is a package of C-programs that has the standard control for the Hardware on ATRV. Connection to the different sensors is done by starting the server program that interacts with the sensor. After this is done the server is connected to the name server. Then the sensor can be connected and controlled by simple C-programs that need the mobility package included. Some sample programs are included in the Mobility Package. It can seem quite difficult to get hang of how to control the different parts through mobility but with some testing it becomes clearer. The test program "simple-follow" is a good start. The names that are needed to connect to the different parts are easiest found through under properties for the part in Mobility's graphical interface MOM, which will be mentioned in the following section. Further reading on mobility and how it works with CORBA can be found in the mobility user's manual. For help on C-programming a good suggestion is the book by J. Skansholm [Skansholm, 1996].

MOM – the graphical interface for mobility



```
0. 22.664253
1. 22.664253
2. 22.664253
3. 1.0148541
4. 22.664253
5. 3.598186
6. 22.664253
7. 22.664253
8. 22.664253
9. 0.6230003
10. 0.5338484
11. 1.9775563
```

Pixels per meter 300.0

Translate: 0.14 Rotate: -0.05

Fig. A.1. This picture is a screen shot from MOM, the graphical interface to mobility

MOM is a tool that could to test and control sensors and actuators in real time. The GUI presents the different parts of the robot in the object selector. Only parts that have their servers started are shown in the object selector. By clicking on an object its properties are available. The ATRV2 object contains controls to motors, sonars and odometry. The information from these sensors can be viewed either graphically or numerically. The figure above shows the graphical range view for the Laser, the numerical view of the sonars and drive command view. The drive view is used to control the motors. By dragging the mouse in the drive window the ATRV can be moved. It is also possible to adjust settings using MOM. For example, it is possible to change the linear and orientation scaling for odometry.

A.2 Mounting of the laser range finder

A.2.1 Mounting

The Laser is mounted on top of the ATRV and an adapter board was manufactured in Plexiglas that allows the Laser to be mounted directly on the front mounting rack of the ATRV. The screws that fit the predrilled holes in the mounting racks are hard to find, but they can be found at “Siewert Skruv”. The Laser range finder is placed (relative to the center of rotation) at a height of 82 centimeters, centered and 31 centimeters in front of the center.
The power supply for the Laser should deliver a voltage between 18 and 27.5 volts. This voltage is used directly from the raw battery power. This voltage could reach slightly above 27.5 volts when fully charged and therefore the power to the Laser should be stabilized with a VICOR 24-24 stabilizer. The serial connection from the Laser is connected to the rocket port.

A.2.2 Cable pin configuration

The Laser has two connectors, one for power and one for serial communication. Both are 9-pin DSUB connectors. These are connected to the power and to the modular connector for the RocketPort as follows.

<u>Power</u>

DSUB9	Description
1	GND
3	Vcc (24V)

<u>Serial Communication</u>

DSUB9	Description	Modular
2	RS-232 Transmit	3
3	RS-232 Receive	4
5	GND	2

A.3 Mounting of the inertial sensor (DMU)

A.3.1 Mounting

The Inertial sensor is mounted inside the ATRV in front of the computer. Holes are drilled in the aluminum plate and screws fasten the inertial sensor. See picture above. The Inertial system is the black box in front of the computer. It is preferable to mount an inertial sensor as close to the center of rotation as possible, because of the lever effects that otherwise will occur. Since the computer already is placed in the center this is the best possible location. The inertial sensor is located at a height of 35 centimeters and 22 centimeters in front of the center of rotation for the ATRV. The inertial sensor has both power and communication connected to a 15-pin DSUB connector. Since the recommended voltage is between 18-40 volts the power connector is connected to raw battery power. The pin configuration will follow

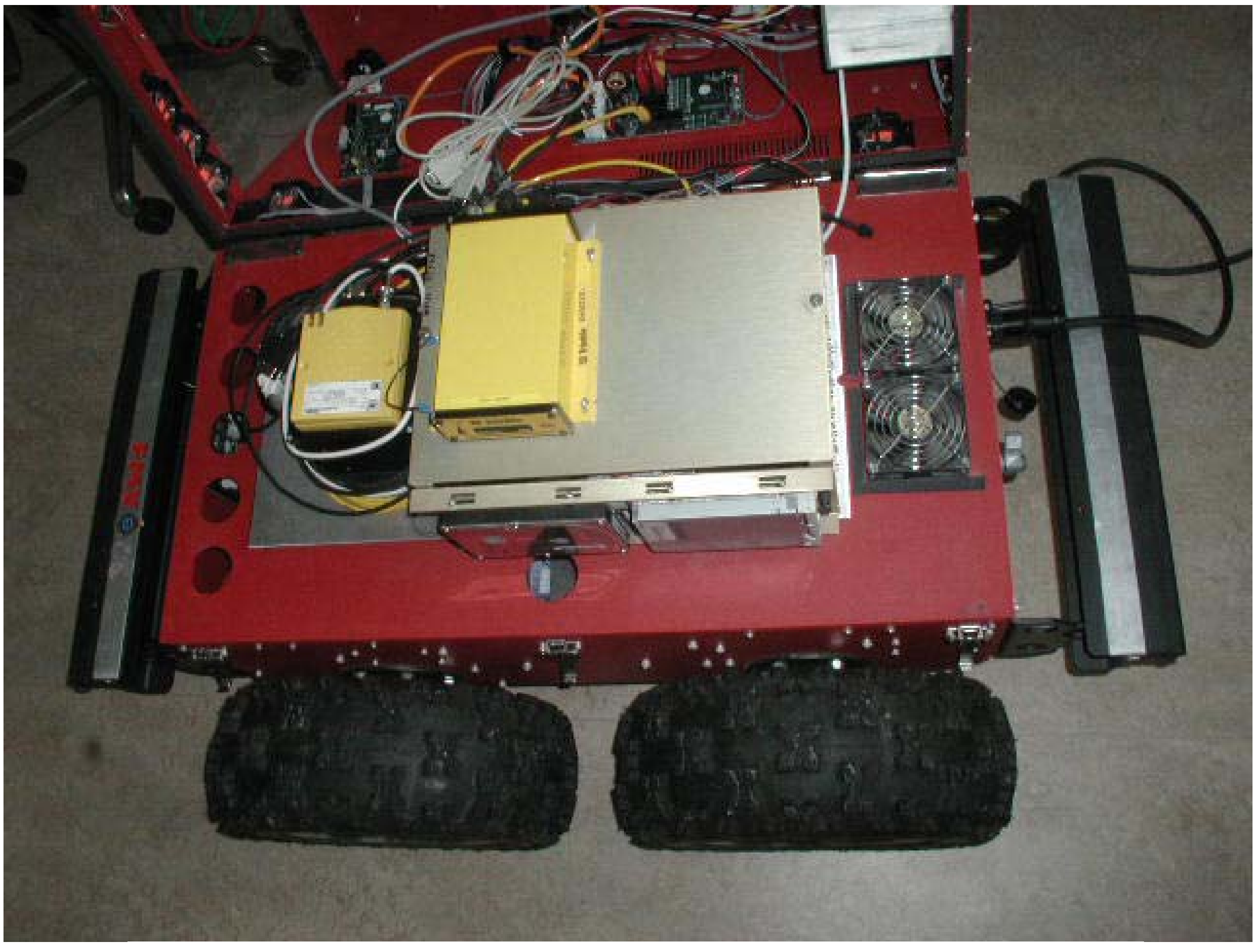


Figure A.2. This picture shows the ATRV with its lid open. The box In the center is the computer and the black box in front of it is the inertial sensor.

A.3.2 Cable pin configuration

The connector for the inertial system is of the type DSUB15. This is connected to the power and to the modular connector for the RocketPort as follows.

DSUB15	Description	Modular
1	RS-232 Transmit	3
2	RS-232 Receive	4
3	Vcc	
4	GND	2

A.4 Encountered problems

Several problems were encountered during the work with this problem. All these affect the results and some of them are possible to sort out but others are not. Most of them can be explained but not all of them can be avoided. Here follows a list of some of them

- *Bad GPS signal* – The first GPS showed very poor performance, even from the beginning. Not only is the estimation error large, the signal completely disappears in several areas. The second one showed better performance but still the same problems with some areas that looses signal. The readings were better in Kungsängen than around BB. The explanation to this is that BB is a tall building that covers part of the sky. Even though the robot was driven more than 10 meters from the walls these problems appeared. Obviously it is impossible to use GPS near high buildings, even if only one side is blocked.
- *Bad Odometry/Tire Pressure* – The first tests showed that the robot turned a lot even though the odometry reported that it was moving straight. This was early discovered and proved to depend on poorly balanced tire pressure. As the tire pressure was much lower on one of the wheels the robot turned in this direction.
- *Bad Odometry/Turning errors* – When the tire pressure was adjusted the robot was moving straight, but it still gave erroneous readings when turning. Turning 90 degrees only reported to odometry as turning 70 degrees. This was possible to adjust within mobility. These parameters are called orientation scaling. It also proved that the translation could be corrected by the parameter linear scaling.
- *Bad Laser readings* – The first tests outdoor with the laser showed many strange effects. The signal was disturbed for many values less than 8 meters. Changing the Laser resolution from millimeters

to centimeters easily solved this problem. This gives a maximum range of 81.9 meters instead of 8 meters. This is discussed further in the checklist in the following section.
- *Network Communication Errors* – Problem with the network causes programs to stop executing occasionally. This is reported by the ATRV as a "timeout error". No solution to this problem has been found.
- *Sudden Shutdown* – It has happened that the system suddenly stops for no obvious reason. The power is shutdown and data may be lost.

The problems that could be corrected ended up on the checklist that follows in the next section.

A.5 Calibration and checklist

This checklist was used during some of the later, more advanced tests with the robot. The checklist was developed after some frustration caused by hardware that was not set as it should or other parts that was improperly adjusted. This caused disturbed data and other unwanted problems. Use this checklist as a mainframe for a new checklist for use with other hardware or as a troubleshooting guide.

CHECK:

- *Laser range finder settings* – Check that the range settings are set to centimeters. This gives a maximum range of 81.92 meters with a resolution of +/- 5 cm. If the maximum range is set to 8.19 meters the readings will not only cut at greater distances but will also produce reflected disturbances of down to the minimum detected range. This could cause a system to detect obstacles close to the robot when this actually ought to correspond to a reading of an object far away.
- *Tire Pressure* – Since the ATRV uses air pressurized wheels on all four wheels a difference in air pressure could cause the ATRV to turn dramatically even though the odometry reports that the robot is moving straight ahead. By driving the ATRV on a straight line the wheel pressure can be used to adjust the drive path. This is no good scientific way to do this, so just start with about the same amount of air in all wheels. Then let some air out at the wheels on the left side if the robot initially turns right and then adjust back on the right side if it now turns left.
- *Odometry calibration* – Even though the robot is moving straight ahead it might report the wrong distance traveled. This can also be the case when turning. Turning 90 degrees using odometry might not result in a 90 degrees turn. Both these parameters can be adjusted, either by using MOM or by programming this in to the

program. The parameters are in MOM called Linear scale for translation and Orientation scale for rotations. The translation parameter can be found by driving the robot a known distance and see what odometry value that is reported by MOM. Then adjust the linear scale under properties and try again to make sure the value is correct. This can be done in the same way for rotation by for example drive the robot in a 360 degrees circle and study the value that is reported by mom. The constant values used in our tests are 6.99E-6 for linear scale and 1.41E-5 for orientation scale. Notice that wheel slippage may affect the odometry reading so this can only be used to make the initial values as good as possible.

- *Radio LAN configuration* – Make sure that the wireless network is working if plans are to use this to communicate between the robot and a LAPTOP. Make sure that it is possible to connect in Peer-to-Peer or "AdHoc" mode if the robot is going to be used away from the base station.
- *Check Battery Status on Robot and LAPTOP* – This might sound unnecessary but just in case. It is an easy check to make sure that the batteries are charged so that the tests do not need to be aborted because of the batteries in the Robot or in the LAPTOP. It could be a good idea to bring extra batteries if tests are performed for a longer period of time.
- *Make sure that all sensors are working* – Just make a short test to see that the readings look all right and that all necessary sensor data is recorded.

This might seem like an unnecessary list but it may save a lot of time and effort to check this at least briefly previous to any important test, especially if tests are performed far from any location where the problems can be solved.