DEGREE PROJECT, IN MECHATRONICS , SECOND LEVEL
STOCKHOLM, SWEDEN 2014

Indoor Positioning and Localisation System with Sensor Fusion

AN IMPLEMENTATION ON AN INDOOR AUTONOMOUS ROBOT AT ÅF

JOHN-ERIC ERICSSON AND DANIEL ERIKSSON

KTH ROYAL INSTITUTE OF TECHNOLOGY

SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

KTH Industrial Engineering
and Management

Master of Science Thesis MMK 2014:101 MDA 475

Indoor Positioning and Localisation System with Sensor Fusion

John-Eric Ericsson
Daniel Eriksson

Approved 2014-12-22	Examiner Martin Törngren	Supervisor Sagar Moreshwar Behere
	Commissioner ÅF	Contact person Roger Ericsson

Abstract

This thesis will present guidelines of how to select sensors and algorithms for indoor positioning and localisation systems with sensor fusion. These guidelines are based on an extensive theory and state of the art research. Different scenarios are presented to give some examples of proposed sensors and algorithms for certain applications. There are of course no right or wrong sensor combinations, but some factors are good to bear in mind when a system is designed.

To give an example of the proposed guidelines a Simultaneous Localisation and Mapping (SLAM) system as well as an Indoor Positioning System (IPS) has been designed and implemented on a embedded robot platform. The implemented SLAM system was based on a FastSLAM2 algorithm with ultrasonic range sensors and the implemented IPS was based on a WiFi RSS profiling method using a Weibull-distribution. The methods, sensors and infrastructure have been chosen based on requirements derived from wishes from the stakeholder as well as knowledge from the theory and state of the art research. A combination of SLAM and IPS is proposed, chosen to be called WiFi SLAM, in order to reduce errors from both of the methods. Unfortunately, due to unexpected issues with the platform, no combination has been implemented and tested.

The systems were simulated independently before implemented on the embedded platform. Results from these simulations indicated that the requirements were able to be fulfilled as well as an indication of the minimum set-up needed for the implementation.

Both the implemented systems were proven to have the expected accuracies during testing and with more time, better tuning could have been performed and probably also better results. From the results, a conclusion could be drawn that a combined WiFi SLAM solution would have improved the result in a larger testing area than what was used. IPS would have increased its precision and SLAM would have got an increased robustness.

The thesis has shown that there is no exact way of finding a perfect sensor and method solution. Most important is, however, the weight between time, cost and quality. Other important factors are to decide in which environment a system will perform its tasks and if it is a safety critical system. It has also been shown that fused sensor data will outperform the result of just one sensor and that there is no max limit in fused sensors. However, that requires the sensor fusion algorithm to be well tuned, otherwise the opposite might happened.

KTH Industriell teknik
och management

Examensarbete MMK 2014:101 MDA 475

Inomhus Positionerings och Lokaliserings System med Sensor Fusion

John-Eric Ericsson
Daniel Eriksson

Godkänt 2014-12-22	Examinator Martin Törngren	Handledare Sagar Moreshwar Behere
	Uppdragsgivare ÅF	Kontaktperson Roger Ericsson

Sammanfattning

Examensjobbet presenterar riktlinjer för hur sensorer och algoritmer för inomhuspositionering och lokaliseringssystem med sensorfusion bör väljas. Riktlinjerna är baserade på en omfattande teori och state of the art undersökning. Olika scenarion presenteras för att ge exempel på metoder för att välja sensorer och algoritmer för applikationer. Självklart finns det inga kombinationer som är rätt eller fel, men vissa faktorer är bra att komma ihåg när ett system designas.

För att ge exempel på de föreslagna riktlinjerna har ett “Simultaneous Localisation and Mapping” (SLAM) system samt ett Inomhus Positioneringssystem (IPS) designats och implementerats på en inbyggd robotplattform. Det implementerade SLAM systemet baserades på en FastSLAM2 algoritm med ultraljudssensorer och det implementerade IPS baserades på en Wifi RSS profileringsmetod som använder en Weibullfördelning. Metoderna, sensorerna och infrastrukturen har valts utifrån krav som framställts från önskningar av intressenten samt utifrån kunskap från teori och state of the art undersökningen. En kombination av SLAM och IPS har föreslagits och valts att kallas WiFi SLAM för att reducera osäkerheter från de båda metoderna. Tyvärr har ingen kombination implementerats och testats på grund av oväntade problem med plattformen.

Systemen simulerades individuellt före implementationen på den inbyggda plattformen. Resultat från dessa simuleringar tydde på att kraven skulle kunna uppfyllas samt gav en indikation av den minsta “set-upen” som behövdes för implementering.

Båda de implementerade systemen visade sig ha de förväntade noggrannheterna under testning och med mer tid kunde bättre kalibrering ha skett, vilket förmodligen skulle resulterat i bättre resultat. Från resultaten kunde slutsatsen dras att en kombinerad WiFi SLAM lösning skulle förbättrat resultatet i en större testyta än den som användes. IPS skulle ha ökat sin precision medan SLAM skulle ha ökat sin robusthet.

Examensjobbet har visat att det inte finns något exakt sätt att hitta en perfekt sensor och metodlösning. Viktigast är dock viktningen mellan tid, kostnad och kvalitét. Andra viktiga faktorer är att bestämma miljön systemet skall operera i och om systemet är säkerhetskritiskt. Det visade sig även att fusionerad sensordata kommer överträffa resultatet från endast en sensor och att det inte finns någon maxgräns för antalet fusionerade sensorer. Det kräver dock att sensorfusionsalgoritmen är väl kalibrerad, annars kan det motsatta inträffa.

Acknowledgements

It has been a great opportunity for us to do this master thesis. It is a thesis that has involved all elements of a mechatronic product. From hardware design to software and control algorithms. We have both evolved in terms of knowledge as well as persons during this half a year. We would therefore like to thank ÅF for the possibility to do this master thesis and we owe our deepest gratitude to Roger Ericsson and Carl-Johan Sjöstedt for their support!

During the thesis we have received much help from engineers at ÅF, help that made it possible for us to finish the project. Persons we want to give a special thanks to is Michael Nilsen for his help with the FPGA-platform, Lovisa Gibson for her help with mounting encoders and Daniel Olsson for his help with the 3D-printer.

Contents

List of Figures

List of Tables

List of Abbreviations

AOA	Angle Of Arrival
AP	Access Point
BBA	Basic Belief Assignment
BiRLS	Bi-Loop Recursive Least Square
BLE	Bluetooth Low Energy
CDF	Cumulative Distribution Function
CPR	Counts Per Revolution
DARPA	Defence Advanced Research Project Agency
DoD	Department of Defence
DST	Dempster Shafer Theory
EKF	Extended Kalman Filter
ESM	Electronic Support Measure
FPGA	Field-Programmable Gate Array
GUI	Graphical User Interface
GPS	Global Positioning System
IEKF	Iterative Extended Kalman Filter
IFF	Identification-friend-foe
IMU	Inertial Measurement Unit
IPS	Indoor Positioning System
IR	Infrared light
JDL	Joint Directors Laboratories
LOS	Line Of Sight
MCU	Microcontroller
ML	Maximum Likelihood
NLOS	No Line Of Sight
NNSS	Nearest Neighbor(s) in Signal Space
PDF	Probability Distribution Function
PWM	Pulse Width Modulation
RLS	Recursive Least Square
RF	Radio Frequency
RFID	Radio Frequency Identification
RPM	Radio Propagation Model
RTOS	Real-Time Operating System
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SDM	Signal Distance Map
SEIF	Sparse Extended Information Filter
SLAM	Simultaneous Localisation and Mapping
SoC	State of Charge
TDOA	Time Difference Of Arrival
TOA	Time Of Arrival
TOF	Time Of Flight
UKF	Unscented Kalman Filter
UWB	Ultra Wide Band
VLC	Visible Light Communication
WLAN	Wireless Local Area Network

Chapter 1

Introduction

Since the introduction of smartphones to the public, the evolution of smart mobile devices has increased significantly. The computation power you hold in your hand can be used to a lot of interesting things. That has lead to that the application of smart mobile devices has increased. Not longer is the interest aimed to smart mobile phones, but also smart sensors, cars etc. One of the applications that have increased significantly is the possibility to locate your self, your car or other mobile devices with the Global Positioning System, GPS. The accuracy for GPS is pretty good for the applications it is used for. "Well designed GPS receivers have been achieving horizontal accuracy of 3 meters or better and vertical accuracy of 5 meters" at a 95 percent confidence interval [1]. This is good enough to find yourself in a map, take you from point A to point B, tracking a unit of interest etc.

The GPS however, needs a Line Of Sight, LOS, between the satellites and the device to be able to calculate the location. This is something that indoor environments lack of, because the No Line Of Sight, NLOS, that can be caused by walls and ceilings. One solution to this has been to extend the positioning by using triangulations of cellular towers. This solution works well to roughly locate a unit, but will not be able to locate which floor a unit is located inside a building etc. Therefore, the research interest in finding a solution for an accurate location method in indoor environment has risen the recent years.

1.1 Background

As mentioned above the indoor positioning and tracking methods is a very prominent research subject, for example Electronik Tidningen wrote about it in February 2014[1]. That's because of the many new innovating applications the indoor positioning and tracking methods give way for. This is something that ÅF[2] has seen an increased demand of at their customers. For example, tools need to be tracked in the assembling industry to guarantee the right assembly of the product. Concerning consumer products, applications improving integration of mobile phones and computers with each other is a prominent area of development which may need feedback of product localisation. Another application would be the possibility for a shopping mall to see the shopping pattern of their consumers during different times of the day or for a logistic company to quickly find high value goods and track the path of these goods.

[1] `http://www.etn.se/index.php?option=com_content&view=article&id=58761&via=s`

[2] `http://www.afconsult.com/sv/om-af/om-af/organisation/division-technology1/`

The list can be long and is a indication that accurate indoor positioning and tracking is a very prominent research subject.

Today, several different methods for indoor positioning already exist, but they all struggle with the difficult indoor environment, like attenuation and reflections. The methods can be divided in three major areas namely, Radio Frequency (RF) communication, Light communication, with Visible Light Communication (VLC) and Infrared light communication (IR) included, and Vision based positioning. The RF area can further be divided in four main methods namely, Wireless Local Area Network (WLAN) based positioning, Radio Frequency Identification (RFID) based positioning, Ultra Wide Band (UWB) based positioning and Bluetooth based positioning.

Except the different positioning[1] methods, there exist some tracking and navigation[4] methods as well, like Inertia tracking[2][3]. Navigation and localisation[2] systems which are solely based on inertial feedback are very expensive due to the extreme precision needed in the hardware for inertial navigation. Fusing multiplex sensor inputs one can filter signals in such a way that biased variation disappear. This has been successfully performed by the use of Simultaneous localisation and mapping (SLAM).

By combining different positioning methods with tracking and localisation methods the accuracy can be improved. This because different methods has different source of errors. By combining them the errors of the combined method can be refined. To combine different methods successfully, state of the art data fusion algorithms needs to be used. The algorithms uses different probabilistic functions in order to weigh the data from different inputs. Some algorithms weighing inputs higher from sources that are known to have better accuracy in the environment, when others compare the measured data to a pre-calculated model.

A conclusion drawn after the State of the Art research was that the major data fusion algorithms today are based on either ramifications of Bayesian Inference or Dempster-Shafer theory.

1.2 Task Formulation

ÅF has seen that in industry today companies choses sensors and methods for indoor positioning and localisation ad-hoc. That is, they chose a sensor they have knowledge about or happen to have in stock. Because of the great interest in the research society, ÅF wants to start and widen their knowledge about the subject. The aim of this thesis is therefore to improve ÅFs knowledge database in the subject of indoor positioning and localisation.

Because the area is wide and the subject is under study of many, a lot of different methods and applications exists. The thesis will therefore be limited to only evaluate methods for indoor office environment during daytime. However, the theoretical study will look at solutions for other areas to widen the possibilities to find methods that can be applied to the area of interest. The research will present recommendations on how to choose sensors and how to combine them with data fusion algorithms in order to improve the accuracy for the wanted application. Some questions to answer will be; What accuracy can be needed in different applications? Is it better to choose sensors with built in sensor fusion algorithms? Can there be too many different sensors/methods? How do you fuse data in the best way dependent of the source?

[1] A client use external help to position it self.

[2] A client do not use external help to locate it self.

[3] Historical record of a movement.

[4] Navigation, the way of relocating yourself to reach a specific position.

With aid of the findings in the theory and state of the art part, methods, hardware (sensors) and data fusion algorithms have been chosen to implement on ÅFs robot platform called the Digital Lobster, see Figure 1.1. It is an in-house project at ÅF to let their engineers increase their knowledge between different assignments. After the theory and state of the art research was done requirements, like the accuracy, were decided for the Digital Lobster and the project.

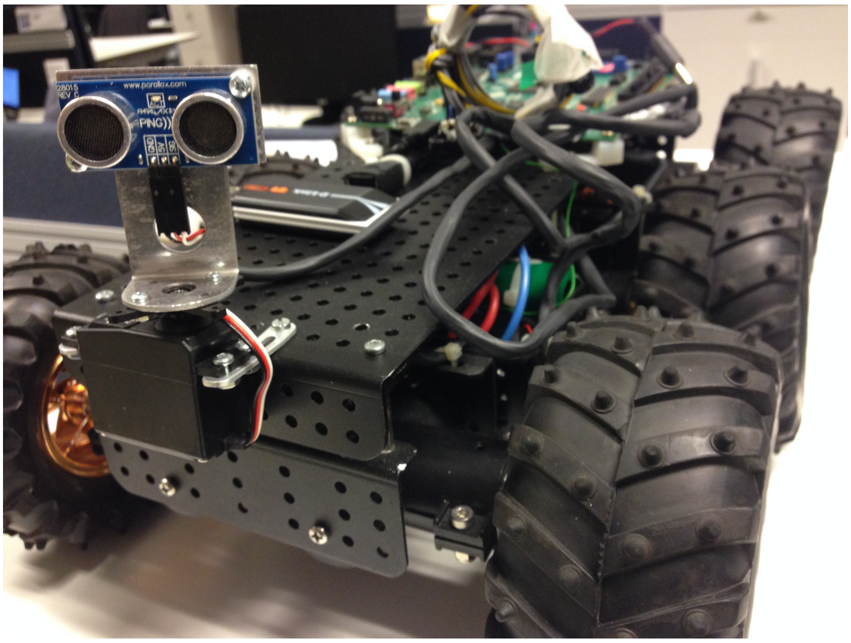


Figure 1.1: ÅFs robot platform the Digital Lobster, before the re-design.

The next phase, the design and implementation parts, were divided in to three main deliverables,

- Simulation results of the proposed methods and algorithms.
- The Digital Lobster shall be able to communicate its position with a certain accuracy, stated in the requirements section 3.1.
- The Digital Lobster shall be able to autonomously navigate to given coordinates with a certain accuracy, stated in the requirements section 3.1.

The success of these deliverables will be presented in the results part of the report. The position of the Digital Lobster will be measured in a local two dimensional Cartesian coordinate system placed at the 7:th floor at ÅFs headquarters in Solna, Sweden. The error will be measured with the Euclidean distance,

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \sqrt{\left(x_q - x_p\right)^2 + \left(y_q - y_p\right)^2} \tag{1.2.1}$$

where **p** is the wanted position and **q** is The Digital Lobsters actual position. During this phase the thesis will evaluate if a positioning method fused with a localisation method can improve the accuracy and performance of the positioning and localisation.

To summarize the thesis goal with one quote;

> *"Compile a collection of guidelines based on State of The Art research for selecting sensors, algorithms and sensor fusion methods for an indoor positioning system and a simultaneous localisation and mapping system, which are implemented and tested to prove the usefulness of these guidelines."*

1.3 Delimitation

Both SLAM and Indoor Positioning System (IPS) can be implemented in a various of areas. It is therefore important to limit these areas by some delimitations. The delimitations will also help in deciding requirements and further to chose sensors and methods to fulfil these requirements.

Early in the thesis it was decided that the robot platform shall only operate at daytime and in an office environment. That leads to a more dynamic environment, with people moving around, and with disturbances as sunlight. Sunlight is important to mention, because some range scanning sensors, as will be discussed in section 2.3, struggles with sunlight, like IR or Laser range scanners. At the same time no consideration is needed to navigate in the dark, which will lead to problem for a vision based range scanning solution.

To limit the test area even more, a selected office environment was chosen, namely the open area at floor seven in the ÅF head office building in Solna, Stockholm, Sweden. The area was chosen based on its closeness to the development area for the platform, for its good surface, simplifying the estimation of the vehicles rotational and translational dynamic model and also for the environmental dynamics, referring to obstacles and people moving. A drawing of the area can be found in section 4.3. Late in the project it was realised that the chosen sensor setup together with the chosen turning model of the platform weren't performing as good as expected and the area for testing SLAM had to be limited even more. The new testing map for SLAM was limited in size but also in dynamics, like less obstacles and more "clean" walls.

Limitation in knowledge about the embedded platform used at the Digital Lobster before the master thesis forced a change of embedded platform. The new embedded platform had limited calculation and storage capabilities. It was therefore early decided that it shall be no requirement to calculate the main algorithms on the platform. Instead a server based solution should be allowed.

An important wish from the stakeholder ÅF was to limit the cost of the project. To limit the complexity and time needed was another wish from the stakeholders. This will lead to a more basic and non advanced sensor setup. A less advanced and complex sensor setup will in turn lead to more basic maps from the SLAM and IPS algorithms. A two dimensional map in the x and y plane was therefore decided to be used instead of a three dimensional map.

1.4 Methodology

Present work done by researchers world wide has been analytically evaluated and matched to the questions depicted in the problem formulation. By the use of the encapsulated information a qualitative method has been used to first select the best SLAM and IPS algorithms fitting to the delimitation's. Also qualitative methods have been used to select a fusion strategy for SLAM and IPS. The performance of the selected algorithms have been evaluated by a quantitative method.

1.5 Ethics

Before going on, let's stop and think about ethics. The technology and methods presented in this thesis, like SLAM and IPS, are very useful. However, they can affect our daily life, in both negative and positive ways. For example, how would you feel;

- Meeting a driver on the highway, reading a book in 110 km/h and don't have any control over what is going on?
- Sitting in a airplane without a pilot? What will happen if an accident occur?
- Saying hi to a robot nurse in the hospital dragging beds or delivering blood samples?
- To know that someone knows exactly were you are? That is happening today outdoors with GPS, but now you are not even "safe" indoors.
- To know that stores uses the knowledge of your position and monitor your shopping pattern? Then they adapt their marketing.

Questions like those above have been kept in mind during the whole master thesis. If the accuracy and performance of the developed algorithms would be of that quality so the above questions is possible to occur, a discussion would have been risen to address the possibilities to build in accuracy errors. That is, if the accuracy of the IPS and SLAM system would be so good that it was possible to track a person within centimeters indoors, the possibility to build-in an accuracy error into the system would have been discussed. Also, if the system would have been sold to a customer or implemented in a larger system, such as a car on the road network, discussions with third parties would have been conducted. Examples of discussion subjects would have been limitations of usage with a user wanting to track its customers and legal discussion with authorities. However, due to knowledge about possibilities with the resources available in the project and the outcome of the algorithms performance, the above was just kept in mind and no actions needed to be taken. The system was also never planned to be sold by ÅF so no discussion with third parties have been conducted.

It is believed by the authors that the biggest obstacle to fully implement SLAM and IPS isn't connected to technology in the future, but to the trust of people. Before any self driving car can be used by the public, the public has to start and trust the technology as well as that clear laws of how to use the technology has to be written.

1.6 Outline of the Report

The rest of this thesis report consists of five chapters. First, chapter 2 will talk about related work and methods used today for indoor positioning system, data fusion algorithms and localisation and mapping methods. The chapter will also discuss how to combine methods with data fusion algorithms and evaluate the different methods of data fusion and give some recommendations of how to choose methods, hardware and algorithms to fulfil the requirements for some interesting applications.

Chapter 3 handles the design chosen for the Digital Lobster limited by the requirements, which also will be presented in this chapter. That is, which methods and algorithms have been chosen, why they have been chosen and explain them more in depth. The chapter will also present simulation with result, that have been performed before implementing the designed system.

Chapter 4 will discuss the platform, new implementations on the platform, which hardware that have been used and how the hardware together with the system have been implemented. Also description of how the implemented functionality have been tested is explained in the later parts of this chapter.

Chapter 5 will present and discuss the results and explain which factors affecting the final result. The final chapter, chapter 6 will conclude the report and will focus on ideas and possibilities to improve the performance of the Digital Lobster in the future.

Sections about SLAM and sensor fusion have been written by Daniel Eriksson and sections about IPS and Range scanning hardware have been written by John-Eric Ericsson. The remaining sections have been written in collaboration between the two authors.

Chapter 2

Theory and State of the Art

The following chapter will present a state of the art study in the subject of localisation using SLAM and in the subject of Indoor Positioning using different technologies.
The chapter will cover many different areas and should be seen as a summary reflecting the different methods existing but not a tutorial. That in mind, a previous knowledge about probabilistic theory is necessary and some knowledge about Kalman and Particle filters are recommended.

2.1 Principles of Sensor Fusion

Early development of microcomputers started in the 1970s as a subgroup of the standard computer science division. Since then a lot has happened in the area of micro-computing and a lot of subgroups has formed in the research area of embedded systems. One of the pioneers of developing this new technology was Intel Corporations who developed the Intel 4004 MCU [3] in 1971, one of the first microcontrollers.

In the same pace as the development of more powerful computers has progressed, the machine perception of the real world has also been likewise considered in research. Sensor input has always been an area of interest, both to feed control inputs to controlling computers, feed control signals back to control algorithms in the MCU and in more present days, even to corporate multiple signals inputs into smart designs for more robust and accurate machines.

The great, both military and social interest in sensing technology, together with the increasing computational power has led to that great research efforts has been spent into the area of "Sensor Fusion" [4]. Increased computational power has led to a big increase of sensor fusing possibilities. By merging multiple independent sources of data, an increase of accuracy of an unknown entity may be achieved by the use of probabilistic laws and by filtration using the best source of information.

Uncertainties of perception has existed since the first sensor was created and all measurements made by any instrument in the real world are conditioned with a certain amount of uncertainty, more or less. This uncertainty is described by the laws of probability. Probabilistic theory has been used much longer than micro-computers and the theory behind propagation of uncertainties has been established for quite a long time. New ways of implementing the theories have though been developed during the 20-th century in combination with computer science, e.g. the Kalman filter, Dempster-Shafer theory and Fuzzy Logic [5] [6] [7]. By the use of sensor fusion, this uncertainty can be greatly reduced by smart usage of sensors complementing each other.

The problem in Sensor fusion is to propagate belief distributions through time. Two principal methods have mainly been discussed and have also been evaluated in this paper. These methods are the Kalman filter algorithms based on objective Bayes inference which are compared against the Dempster-Shafer theory, which is an extension of a subjective case of Bayes inference.
The biggest disadvantage of the Bayesian recursive filtering algorithms (Kalman filter algorithms) is the necessity of a prior distribution. Either if the prior is pre-defined or if it, by some means is approximated to a pre-set value. In comparison, the Dempster-Shafer Theory of evidence is derived from the foundation of the Bayesian theory but the filter is designed in a non-recursive way which make it possible to propagate probabilities without any knowledge of the priors.

To clarify the subject we are reasoning about, data fusion needs a proper definition. Though, it does not exist any global unified and accepted terminology for the definition of data fusion. So to clarify a common definition of data fusion, we have chosen to follow U.S. Department of Defence (DoD) Joint Directors of Laboratories (JDL:s) definition. The department developed a lexicon [8] with the specific definition for data fusion described by the following words.

> *"A process dealing with the association, correlation and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete, and timely assessments of situations and threats, and their significance. The process is characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results."*
>
> ([8], chapter 2)

The definition is composed by a defence organisation. Among the definitions that exist, this one has gained a lot of popularity among researcher and describes the intent of a data fusion algorithm, even though its a very military influenced definition.

To handle a generalized convention of a methodology concerning data fusion, a four level process model for data fusion has been developed by the U.S. military institution JDL. The method have gained great popularity among data fusion researchers. [9]. The initially developed process levels stated by JDL was revised in 1999 to include a six level method see Steinberg et al. [10], but the foundation is laid on the initially developed four level model and are as described in table 2.1.

Table 2.1: The JDL Data fusion model for standardised methodology, adapted from [10]

JDL Data Fusion Model	
Level 1:	Object Refinement
Level 2:	Situation Assessment
Level 3:	Threat Assessment
Level 4:	Process Assessment

The first level is considering the task of pre-processing sensor data before it reaches the fusing algorithm. Parts of the first level is also focusing context to perform analysis and evaluate if data belongs and relates to the same time instance for fused data which has a relationship. This is discussed by Blackman [11] and may be one of the most important tasks in sensor fusion. For

a further analysis Steinberg et al. [10] describes the procedures involved in more detail.
Smith and Singh [12] describe the first step as “Data Registration”, “Data Association”, “Position Attribute Estimation” and “Identification”. The data registration is focusing on the task to transform reviewed data to a common coordinate system. The data association is a first step to connect the data subjected for fusion to the origination of it by the use of only the measurement data. The Position/attribute estimation is related to the process estimating a target’s state by the use of received measurements. This process is usually performed by the use of Fuzzy Logic,Kalman or Particle filters which will be discussed further in the following chapters. The last part of level one is the “Identification” step where classification of the origin of each measurement and specifies the association.

The second level in the JDL abstract data fusion level system is dedicated to “Situation Asessment”. The level named “Situation Asessment” is referring to a step where kinematic and temporal data is fused together to get a sort of situation awareness. All of the levels are originally developed by a military authority and will have definitions with a lot of connection to defence. A direct implication from that is the JDL definition of level two which is described as a measures to indicate warnings to plan for actions.

The third level described as “Threat Asessment” evaluates the severity of the present situation which has been estimated during the situation awareness phase in level two.

The fourth level is a system monitoring phase which evaluates and controls that the total process is optimised for its application. The process corrections can be considering setting up a priority of different targets, as Waltz and Llinas describes in [13]. Another application compensated by the “Process Asessment” could be to move the sensor to increase the coverage area, this is analyzed by Xiong and Svensson (2000) [14].

2.1.1 Bayesian Inference and the Kalman Filter

Bayesian inference refers to the process used to propagate probabilistic data from specific physical variables together with observed data. Bayesian inference refers to the probabilistic reasoning discipline belonging to a group of data fusion algorithms that uses prior knowledge filtered together with observations to make an inference about the data in the present observed space.
Due to the characteristic conditioned problem formulation, the derivation of the method originates directly from Bayes rule. Equation 2.1.1 presents the standard formulation of Bayes rule for conditional probability, where E in this case is an arbitrary event whose probability is estimated given another event H. By the use of the law of total probability, the can be derived according to equation 2.1.1.

$$P(E|H) = \frac{P(E,H)}{P(H)} = \frac{P(H|E)P(E)}{P(H)} \tag{2.1.1}$$

Because of H’s independence of the stochastic variable E, the probability $P(H)$ is often assigned as a normaliser, see equation 2.1.2.

$$P(E|H) = \eta \cdot P(H|E) \cdot P(E) \tag{2.1.2}$$

From the derivation of Bayes law, a recursive filter can be created(see table 2.2) using the conditioning characteristics. The Bayes filtering algorithms has its origin directly from the derivation of Bayes theorem where one can see that Bayes filter is a general form of a recursive filter. In the following part, the Kalman filter will be discussed which will be derived as a special case of Bayes filter. In Bayes filter an arbitrary probabilistic distribution is assumed to calculate the beliefs. In Table 2.2 we can see how Bayes rule is implemented in a filter algorithm.

The algorithm is calculating the belief of a state x at any time instance t. In line one, the algorithm is called with the prior belief for state x_{t-1} , the control input u_t and the measurement z_t as the function inputs. Line two is iterating for every state in the state vector x. In line tree the “prediction belief” is calculated using the prior distribution together with the process model. Line tree can also be seen as a implementation of the law of total probability to marginalise the previous states. In line four, the prediction estimation is used to calculate the updated belief implementing the correction of the measurement input.

Table 2.2: Bayes Filter Algorithm

```
1 :  Algorithm Bayes Filter($bel(x_{t-1}), u_t, z_t$)
2 :    for all $x_t$ do
3 :      $\overline{bel}(x_t) = \int p(x_t|\ u_t, x_{t-1}) \cdot bel(x_{t-1}) dx_{t-1}$
4 :      $bel(x_t) = \eta \cdot p(z_t|\ x_t) \cdot \overline{bel}(x_t)$
5 :    end for
6 :    return$bel(x_t)$
```

The Kalman filter is a linear, Gaussian implementation of the Bayesian filter. By assuming the definitions of the Gaussian probability density function (PDF) in equation 2.1.3 as the probabilistic represntation and also asuming the linear state in equation 2.1.4, the algorithm in table 2.3 can be derived. The probability density function in equation 2.1.3 can be calculated using the covariance of a multivariate PDF Σ and a best estimate which is represented by μ. For the state in equation 2.1.4, **A** and **B** represents the behaviour of the system, ε is an additive disturbance represented with a white Gaussian noise.

$$p(x) = \det{(2\pi\Sigma)}^{-\frac{1}{2}} \cdot \exp\left\{ -\frac{1}{2}(z-\mu)^T \Sigma^{-1} (z-\mu) \right\} \tag{2.1.3}$$

$$\mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{B}_t u_{t-1} + \varepsilon_t \tag{2.1.4}$$

Table 2.3 is presenting the Kalman filter algorithm developed by Kalman 1960 [15]. As described by Thrun and Burgard [16], the Kalman filter is using line two and three in the algorithm to estimate the predicted best estimate and the respective covariance. In line four, the Kalman gain is calculated which is representing how much the updated measurement in line five, is trusting in the measurement or if it believes in the model. Finally in line six, the updated covariance is calculated.

To get the filter adapted for real physical disturbances in both the control signal u_t, and the measurement readings, specific noise is added in both the two posterior estimations to have the real signal as good represented by the models as possible. The variable $\mathbf{R_t}$ is representing the process model noise and $\mathbf{Q_t}$ is corresponding to the measurement model noise. Both models are distorted by a white Gaussian to in the best way represent random noise occurring in the signals and measurements.

Table 2.3: Kalman Filter Algorithm

```
Algorithm Kalman Filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)
    $\bar{\mu}_t = \mathrm{A}_t\bar{\mu}_{t-1} + \mathrm{B}_t\mathrm{u}_t$
    $\bar{\Sigma}_t = \mathrm{A}_t\Sigma_{t-1}{\mathrm{A}_t}^T + \mathrm{R}_t$
    $K_t = \bar{\Sigma}_t {C_t}^T (C_t\bar{\Sigma}_t {C_t}^T + Q_t)^{-1}$
    $\mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t)$
    $\Sigma_t = (I - K_tC_t)\bar{\Sigma}_t$
return $\mu_t, \Sigma_t$
```

Bayesian inference in combination together with generalised Kalman filter is one of the most used algorithms for data fusion when a prior distribution exist. For practical consideration, this is not always the case and in context it might be impossible to calculate the prior distribution. Methods of guessing the prior distribution exists but do severely affect the algorithm convergence in complex environments. Then other methods exist to calculate posterior belief based on non-prior distributions .

2.1.2 Dempster-Shafer Theory - DST

Dempster-Shafer theory in contrast to the Bayesian inference is a subjective probability formulation for probabilistic inference. It is also in its characteristics invented in such a way that it doesn't have the need for a prior probability distribution.

DST has during the past 20 years been used a lot together with sensor fusion, especially to cases where a prior distribution doesn't exist, see Klein [17] or Wu [9]. The DST has the characteristic of answering a common question which relates to a series of subjective questions with local corresponding probabilities. The DST is described as a theory which persists of both evidence but also probable reasoning principles. Shafer describes the framework as a combination of both disiplines. "It is a theory of evidence because it deals with weights of evidence and with numerical degrees of support based on evidence. It is a theory of probable reasoning because it focuses on the fundamental operation of probable reasoning: the combination of evidence" [18] [19].

The DST is often described as a method for "ignorance handling". The meaning of ignorance is referring to the ability of the DST to distinguish between the evidence that supports a proposition and the lack of evidence that refutes the proposition. This is the way that DST handles the situations where a prior distribution isn't existing, by introducing the ignorance into calculations, which will give a better redundancy for convergence.

Figure 2.1 shows an example of how support, plausibility and evidence refuting the statement can behave and characterise the possibilities of the probability. As seen, the evidence supporting the statement is not in majority, but the plausibility makes room for a majority even if it is not verified.

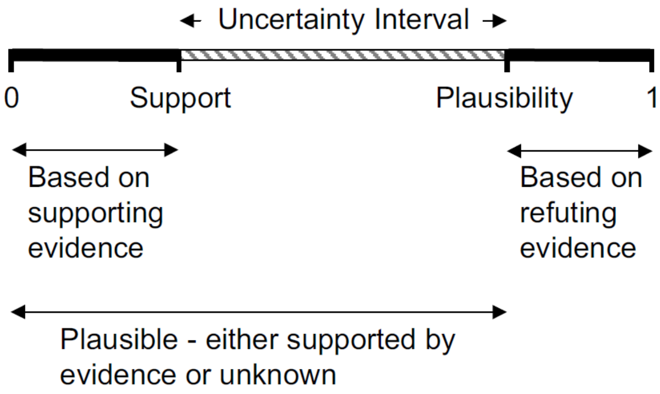

Uncertainty Interval
0
Support
Plausibility
1
Based on supporting evidence
Based on refuting evidence
Plausible - either supported by evidence or unknown

Figure 2.1: Dempster-Shafer theory interval for proposition (Adapted from Lawrence A. Klein, Sensor and Data Fusion, Bellingham, Washington, 2004).

Figure 2.1 is illustrating the DST evidence calculations in a simple bar figure. Evidence supporting the proposition is the black part colored to left in the figure which together with the uncertainty interval forms the Plausible evidence. To the right, the evidence given for refuting the proposal is allocated.

In Figure 2.2 , the process of DST data fusion algorithm is illustrated. In the most left boxes, the different data inputs which are supposed to undergo fusion are processed and given a clear declaration. By proceeding further to the right in the graph, the sensor 1 to N data is transformed into a mass distribution space. The process performed in this step can be compared to the Gaussian approximated distribution in the case of a Unscented Kalman filter. The mass-transform is mostly performed by expert-models which often are created through sampling tests creating a model, which can be both time consuming and a complicated task [19].
The resulting mass distribution is then combined using Dempster's rule of combination for evaluation. From the output result of the combination rule, logic is used to make a decision.

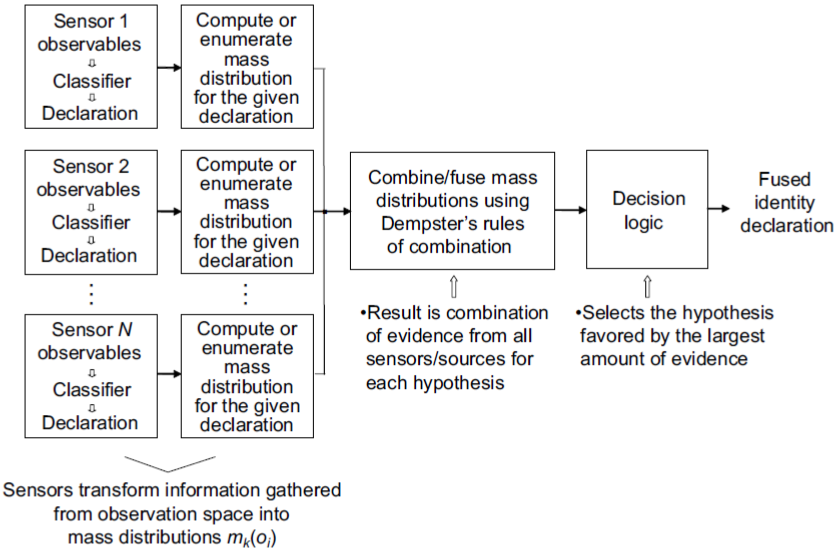

Sensor 1 observables
Classifier
Declaration
Compute or enumerate mass distribution for the given declaration
Sensor 2 observables
Classifier
Declaration
Compute or enumerate mass distribution for the given declaration
Sensor N observables
Classifier
Declaration
Compute or enumerate mass distribution for the given declaration
Combine/fuse mass distributions using Dempster's rules of combination
Decision logic
Fused identity declaration
•Result is combination of evidence from all sensors/sources for each hypothesis
•Selects the hypothesis favored by the largest amount of evidence
Sensors transform information gathered from observation space into mass distributions $m_k(o_i)$

Figure 2.2: Dempster-Shafer data fusion process (Adapted from E. Waltz and J. Llinas, Multisensor Data Fusion, Artech House, Norwood, MA [1990]).

The formal definition of the Dempster-Shafer theory is derived at start from the definition of a Powerset. The Powerset is defined by first defining the Universal set, for this case abbreviated as X. The vocabular definition of the Powerset is, the set of all the subsets of X including the empty set, see equation 2.1.5.

$$\begin{aligned} X &= \{a, b\} \\ P(X) &= 2^X = \{\emptyset, \{a\}, \{b\}, X\} \end{aligned} \tag{2.1.5}$$

By definition, the theory of evidence [6] defines a belief mass to each element of the Powerset, called the basic belief assignment (BBA) according to equation 2.1.6. The BBA can also be described as a specific mapping of the Powerset to masses which will form a proper distribution.

$$m : 2^X \to [0, 1] \tag{2.1.6}$$

The delimitation's of the BBA is firstly that the sum of all the masses connected to the Powerset sums up to one. The cause is to allow every data input subjected for fusion to have the same weight and be evaluated fairly, see equation 2.1.8. Secondly for the cause of simplicity, the value of the zero space shall be zero [20] see equation 2.1.7.

$$m(\emptyset) = 0 \tag{2.1.7}$$

$$\sum_{A \in 2^X} m(A) = 1 \tag{2.1.8}$$

The basic parts needed to describe Shafer's theory of evidence are first "an adequate summary of the impact of the evidence on a particular proposition" [17]. This is derived by defining the belief (or support) as seen in equation 2.1.9.

$$bel(a_1) = S(a_1) = m(a_1) \tag{2.1.9}$$

Equation 2.1.10 is describing an example calculating the support for that the target is either a_1, a_2 or a_3.

$$\begin{aligned} S(a_1 \cup a_2 \cup a_3) = m(a_1) + m(a_2) + m(a_3) + m(a_1 \cup a_2) + \\ m(a_1 \cup a_3) + m(a_2 \cup a_3) + m(a_1 \cup a_2 \cup a_3) \end{aligned} \tag{2.1.10}$$

Secondly there is a need for a definition of how well the negotiation of a particular target is supported. This is described by defining the so called Plausibility of the target. The Plausibility is defined in equation 2.1.11 and is defined as one minus the negotiation of the targets support function.

$$Pl(a_1) = 1 - S(\bar{a}_1) \tag{2.1.11}$$

To combine different independent sets of measures, Dempster developed a rule called the rule of combination to propagate the belief or support for the independent probabilities. The rule is deriving the appropriate common shared belief for different sources, and specifically ignores all conflicting (non-shared) beliefs by the use of a normalizing factor K. Equation 2.1.12 and 2.1.13 derives the relation between the different mass function, the “joint mass” which is expressed as the orthogonal sums of the individuals.

$$m_{1,2}(\emptyset) = 0 \tag{2.1.12}$$

$$m_{1,2}(A) = (m_1 \oplus m_2)(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \emptyset} m_1(B) \cdot m_2(C) \tag{2.1.13}$$

Where K in equation 2.1.14 is a normaliser which measures the conflict between the two mass sets.

$$K = \sum_{B \cap C = \emptyset} m_1(B) \cdot m_2(C) \tag{2.1.14}$$

The constant K is often used as a measurement for how much the different combinations is conflicting each other. By the use of this measurement, the user can derive if the probabilities are trustworthy or not.

By the use of Dempsters rule of combination together with the data fusion process in Figure 2.2, and a set of subjective questions evaluated by expert models, Dempster-Shafers theory can be used to fuse data when a prior distribution doesn’t exist.

2.1.3 Fuzzy Logic

Fuzzy logic was developed in the 1960’s by Lotfi Zadeh [7] [17] with the intention to transform data, not belonging to a delimited domain into specific delimited domain. The ideas were to wipe out the strict binary belongings as represented by regular set theory and represent it with a so called “Fuzzy set” which represents an entity with a more linguistic representation and maps it to a continuous belonging.

The theory of Fuzzy Logic uses the fact that many systems can’t directly be estimated by a sensor and has to be evaluated by an “expert” [21]. To evaluate and group the expert’s knowledge into algorithms, Fuzzy Logic is routinely used to transform its knowledge into a computer algorithm.

Figure 2.3 shows a graphical description of how a fuzzy set can be implemented. The temperature value is evaluated according to tree functions, one cold, one warm and one hot function. The experts evaluate the state from an estimated temperature. At this specific instance the temperature gets evaluated along a continuous temperature scale and gets assigned to some fuzzy values, “not hot” (*red arrow*), “slightly warm” (*orange arrow*) and “fairly cold” (*blue arrow*).

In this case, pointed out by the grey line. The different assigned fuzzy variables can then be used to code algorithms for system control, often with the use of "if"-statements.

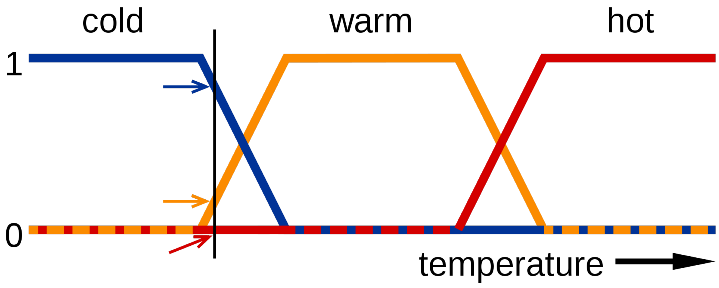


Figure 2.3: Wikipicture Fuzzy Logic

The use of Fuzzy Logic is often not directly implemented as a sensor fusion algorithm, but in combination, it has often been implemented as for the use of estimating non binary readings. In more concrete, Fuzzy Logic has been subjected to sensor fusion in combination with Kalman filtering algorithms [22] by estimating different noise levels in the Kalman filter. Kalman filter algorithms are modelled as Gaussian distributions with a modelled unbiased white noise [16].
The noise is characterising the property of how the KF adapts to the environment and how easily it diverges from the true value. The thought of the Fuzzy Logic implementation is to by an "expert" evaluate how the noise-level is best adjusted to get the best filtering value as possible.

2.1.4 Comparison of Probabilistic Inferences

Description of the main theories behind sensor fusion has above been presented by different approaches using different theories. The most popularly used methods, unrelated to their area of usage are divisions of the Bayesian filtering algorithms and the Dempster-Shafer theory. In later years, the fuzzy logic applications have also been an increasingly used method for evaluation of expert models.
In a wider context, Dempster-Shafer theory and Bayesian inference are the main pillars in modern probabilistic fusion theory. They are principally competing about different domains and are mainly used in different situations with different preferences. That means, that if a model and a prior distribution exists the Kalman filter is best used to filter the data. If a prior distribution and/or a model don't exist, the Dempster-Shafer theory is argued to be the best method to apply. As Klein is describing in "Sensor and Data Fusion" about the Dempster-Shafer theory, "Dempster-Shafer theory estimates how close the evidence is to forcing the truth of a hypothesis, rather than estimating how close the hypothesis is to being true" [23].

In comparison, Shafer express the limitations of the Bayesian inference as a general limitation that "Bayesian theory cannot distinguish between lack of belief and disbelief. It does not allow one to withhold belief from proposition without according that belief to negation of the proposition." [24]

To deal with data fusion, a lot of calculation power is needed which is a matter of consideras-tion selecting the best algorithm. The complexity of Dempster-Shafer theory algorithm and the Bayesian filtering algorithms is generally in the Bayesian algorithms favour, but according to Leung and Wu [25], as they reported, the comparison of the computational complexity is very depending of the implementation of the algorithms. For the Bayesian filters, the conditional probability is calculated at first when a new feature gets available. In comparison to the Dempster-Shafer theory, the disjunction of all the probabilistic propositions is calculated in each iteration, which makes the load heavier. However as Klein depicts [17] the implementation of DST gets simpler in the case the decision space has to be redefined. A summary concluding the preferences dividing Bayesian Inferrence and Dempster Shafer Theory in to two separate categories are presented in table 2.4.

An example made by Waltz and Llinas [13] presented in Klein [17] explains how fusion of identification-friend-foe (IFF) and electronic support measure (ESM) sensor data is performed both by Bayesian and DST. The result shows that the Bayesian algorithm takes less time to complete than the DST algorithm. Although they also mention that the difference is not of significant matters for their application, but there is a significant difference in computational power needed.

During the latest years has improvements been made to better adapt for dynamically changes of the fusion environment. The majority of this techniques have been based on expert models and especially Fuzzy Logic. J.Z Sasiadek and Q. Wang [22] used Fuzzy Logic in a linearized Kalman filter where the Fuzzy logic was used to adapt the noise models in the Kalman filter dynamically during the run. It showed a good response which improved the performance of the Kalman filter. Cohen and Edan [26] have written a report about sensor fusion mapping combined with sensor data fusion. They implemented multiple range sensors and used Fuzzy Logic to evaluate the combined belief of each pixel in a binary grid map. The result was a working algorithm to employ sensor fusion without a priori distribution and no specific system model.

Table 2.4: Comparison of the different data fusion algorithms

	Data Fusion Algorithms	
Item	**DST**	**Bayesian Inference**
Computational burden	Medium	Medium
Required Prior distribution	No	Yes
Implementation complexity	High	Medium
Calculates plausibility	Yes	No

2.2 Localisation - State of the Art

The interest of autonomous vehicles and robots has almost been into research for about 30 years, but due to computational power requirements and sensor accuracy, real autonomous robots and vehicles with SLAM were first tested in real applications as soon as about 10 years ago. The problem of perceiving a dynamical environment and adapting to it for control of motion is a complex task which requires a huge amount of data processing. The different methods and approaches for localisation will be discussed in this chapter, which will include short descriptions of the most frequently used filtering algorithms. The written report presumes that a previous understanding of basic probabilistic inference and probabilistic theory exists. The methods analyzed will sum up to an overall picture of the solutions existing for autonomous navigation using recursive SLAM.

The DST is here excluded considering the generalisation that a prior knowledge of the state exist which are the case of a robot operating in a two or three dimensional environment. Different filtering algorithms will be described including EKF, UKF and particle filters. Foundation of mapping structures will also be discussed and analysed. Finaly the chapter will be concluded describing models used for robot movement and common sensor models used together with localisation.

Localisation is the way of determining your position given a specified environment. This report will mainly focus its effort on localisation in a indoor environment. Although, localisation has been performed long before computational power was satisfactory to perform it autonomously. In early stages of in flight navigation, heading and speed was monitored and the localisation of the aircraft was estimated by a navigator by dead reckoning.

The simplest way to localise yourself is to by dead reckoning perform such a task. All kind of sensors are more or less induced by noise and disturbance, so to better estimate your localisation both dead reckoning and correction by another kind of reference measurement is used to better localise your position. This could either be a radio beacon, but in the case of indoor navigation, references to nearby objects are usually used as a secondary data for sensor fusion.

2.2.1 EKF - The Extended Kalman Filter

The Kalman filter was shortly presented in the previous theoretical part as a method for data fusion. It is restricted to applications with Gaussian distributions and strict linear process models as presented by Thrun [16] . However, the world is not created by linear behaviours. So in many cases the simple Kalman filter will lead to algorithm divergence and bad performing filtered values. A solution to this problem was presented by Kalman [5] [15]. Kalman used the prior work of Brook Taylor to linearize the modelled functions and then include it in his algorithm for the Kalman filter. For multivariate cases, first order Taylor expansion are used to calculate the Jacobian matrix which are used in the algorithm.

In short the filter is working very much alike the Kalman filtering algorithm, beside the use of Jacobian matrices calculating the linearized expressions.

$$g'(u_t, x_{t-1}) := \frac{\partial g(u_t, x_{t-1})}{\partial x_{t-1}} =: G_t \tag{2.2.1}$$

$$h'(x_t) := \frac{\partial g(x_t)}{\partial x_t} =: H_t \tag{2.2.2}$$

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + g'(u_t, \mu_{t-1})(x_{t-1} - \mu_{t-1}) \tag{2.2.3}$$

$$h(x_t) \approx h(\mu_t) + h'(\mu_t)(x_t - \mu_t) \tag{2.2.4}$$

In Table, 2.5 it's seen that the prediction step in line two has changed from a state matrix to a linearized function. The linearized function is derived in equation 2.2.1 and 2.2.3 where the function $g()$ is representing the first order Taylor expansion. The capital G is representing the Jacobian matrix which is the direct result of multivariate linearization. The same reasoning is valid for the function $h()$ which is representing the measurement linearization. In this case the capital H is also representing the Jacobian partial derivative matrix.

Table 2.5: The Extended Kalman filter algorithm

```
1 : Algorithm Extended Kalman Filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)
2 :     $\bar{\mu}_t = g(u_t,\mu_{t-1})$
3 :     $\bar{\Sigma}_t = \mathrm{G}_t \Sigma_{t-1} {\mathrm{G}_t}^T + \mathrm{R}_t$
4 :     $K_t = \bar{\Sigma}_t {H_t}^T (H_t \bar{\Sigma}_t {H_t}^T + Q_t)^{-1}$
5 :     $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$
6 :     $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$
7 : return $\mu_t, \Sigma_t$
```

The algorithm is then returning a new posterior which continuously runs through an iterative process. The parameters **R** and **Q** are representing the same noise additives as in the Kalman filtering case. The matrices are represented by a user determined value of a white noise covariance matrix which represents the imperfections in the estimator. The only difference in the Kalman gain from the Kalman filter is the use of the Jacobian matrix **H** instead of the state space model.

The Extended Kalman Filter was developed by Kalman during the first part of the 1960:s and have been implemented in lots of applications of sensor fusion since then. Hugmark (2013) [27] implemented a vision algorithm fused together with IMU data for position estimation of an embedded system. Another application was made by Xiong (2013) [28] using the EKF to estimate the "State of Charge" (SoC) for Lithium-Ion batteries fusing the state of the battery.

2.2.2 The Unscented Kalman Filter - UKF

The UKF is also an recursive filtering method for data fusion as the EKF with the difference in how it linearise nonlinearities. The Unscented Kalman filter is also based on the Bayes filtering algorithm, but is designed to make an approximation of nonlinearities without using Taylor expansions. The filter is restricted to Gaussian distributions in the same way as for the extended Kalman filter.
Instead of approximating nonlinearities, the filter is using sampled points which are sampled through the nonlinear models and remodelled after sampling. The points are chosen in characteristic way which can be calibrated for better performance dependent of the specific environment of operation.

The basic principle of the method it's built upon, is that instead of using a continuous distribution during the mapping sequence, individually sampled points are chosen to represent the distribution. Points are passed through the nonlinearity and are approximated by a Gaussian distribution after the nonlinear transformation which is illustrated in Figure 2.4.

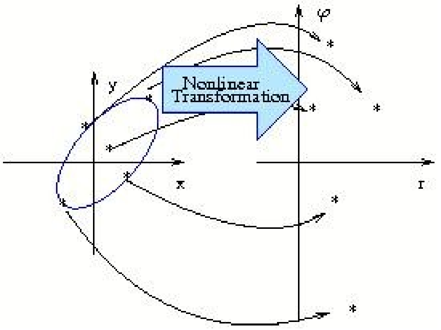


Figure 2.4: Unscented Kalman filter sigma points transformed through a nonlinear function.[16]

The following parts are describing the selection of the individually sampled points representing the distribution , the so called "sigma"-points and how they propagates through nonlinear process models and are represented by a Gaussian distribution.
The design of the Sigma points is done in specific way which has been tested and designed for best performance. To adapt for nonlinearities with different performance, the constant kappa in equation 2.2.6 is used to adjust for bigger or lesser spread of the "sigma"-points. The particles are beside that selected in a specific way to get the best adaption to the transforming entity according to the following rules.

The index is representing the specific particle number, and the zero index is representing the middle particle. The numbers of particles are always proportional to the number of dimensions which are filtered and are equal to the index n. The second and the third equation in 2.2.5 is existing to place one particle on each of the sides of the middle particle.

$$\begin{aligned} \chi^{[0]} &= \mu \\ \chi^{[i]} &= \mu + \left(\sqrt{(n+\lambda)\Sigma}\right)_i \text{ for } i = 1, ..., n \\ \chi^{[i]} &= \mu - \left(\sqrt{(n+\lambda)\Sigma}\right)_{i-n} \text{ for } i = n+1, ..., 2n \end{aligned} \tag{2.2.5}$$

Where λ is the equation in reference 2.2.6 combining the two dimensional constants κ and α describing the spreading distance of the sigma points.

$$\lambda = \alpha^2(n+\kappa) - n \tag{2.2.6}$$

Each Sigma point is also subjected to two weights respectively. The first one are used calculating the mean and the second recovering the covariance from the Gaussian which can be seen in equation 2.2.7 to 2.2.8 and can be seen in its final form in equation 2.2.11.

$$w_m^{[0]} = \frac{1}{n+\lambda} \tag{2.2.7}$$

$$w_m^{[0]} = \frac{1}{n+\lambda} + (1 - \alpha^2 + \beta) \tag{2.2.8}$$

$$w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n+\lambda)} \text{ for } i = 1, \ldots.., 2n \tag{2.2.9}$$

$$\Upsilon^{[i]} = g(\chi^{[i]}) \tag{2.2.10}$$

From the mapped sigma point in equation 2.2.10, a Gaussian distribution, μ' and Σ' is extracted through some arbitrary modelled process function $g()$ instead of using Taylor expansions.

$$\begin{aligned} \mu' &= \sum_{i=0}^{2n} w_m^i \cdot \Upsilon^{[i]} \\ \Sigma' &= \sum_{i=0}^{2n} w_c^{[i]} (\Upsilon^{[i]} - \mu')(\Upsilon^{[i]} - \mu')^T \end{aligned} \tag{2.2.11}$$

In Table 2.6 the basic implementation of the UKF algorithm is presented and described in context in the following part. In line 2 to line 5 the prediction step is calculated. Line 2 calculates the specific sigma points which then are sampled in line 3. Line 4 and 5 calculates the predicted mean and covariance. Line 6 is calculating new sigma points together with the estimated covariance. The sigma points are then sampled in line 7. The predicted uncertainty is calculated in line 8, together with the crosscovariance the Kalman gain is then calculated in line 11.

The Kalman gain is used in the same way as in the KF and EKF to grade how much affect the innovation would have on the updated model. As we can see in the parenthesises in line 12, the inovation is calculated. It represents the difference between the expected measurement of the range measurement and the real measured range. By the use of the calculated Kalman gain and the Inovation, the updated mean and covariance is then finaly calculated in line 12 and 13.

Table 2.6: The Unscented Kalman Filter Algorithm

```
Algorithm Unscented Kalman Filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)
    $\chi_{t-1} = \begin{pmatrix} \bar{\mu}_{t-1} & \bar{\mu}_{t-1} + \gamma\sqrt{\bar{\Sigma}_{t-1}} & \bar{\mu}_{t-1} - \gamma\sqrt{\bar{\Sigma}_{t-1}} \end{pmatrix}$
    $\bar{\chi}_t{}^* = g(u_t, \chi_{t-1})$
    $\bar{\mu}_t = \sum_{i=0}^{2n} w_m{}^{[i]} \bar{\chi}_t{}^{*[i]}$
    $\bar{\Sigma}_t = \sum_0^{2L} w_c{}^{[i]} \left(\bar{\chi}_{i,t}^* - \bar{\mu}_t\right) \left(\bar{\chi}_{i,t}^* - \bar{\mu}_t\right)^T + R_t$
    $\bar{\chi}_t = \begin{pmatrix} \bar{\mu}_t & \bar{\mu}_t + \gamma\sqrt{\bar{\Sigma}_t} & \bar{\mu}_t - \gamma\sqrt{\bar{\Sigma}_t} \end{pmatrix}$
    $\bar{Z}_t = h(\bar{\chi}_{i,t})$
    $\hat{z}_t^k = \sum_0^{2L} w_m{}^{[i]} \bar{Z}_t^{[i]}$
    $S_t = \sum_0^{2L} w_c{}^{[i]} \left(\bar{Z}_t^{[i]} - \hat{z}_t^k\right) \left(\bar{Z}_t^{[i]} - \hat{z}_t^k\right)^T + Q_t$
    $\bar{\Sigma}_t^{x,z} = \sum_0^{2L} w_c{}^{[i]} \left(\bar{\chi}_{i,t}^* - \bar{\mu}_t\right) \left(\bar{Z}_t^{[i]} - \hat{z}_t^k\right)^T$
    $K_t{}^i = \bar{\Sigma}_t^{x,z} S_t{}^{-1}$
    $\mu_t = \bar{\mu}_t + K_t{}^i \left(z_t{}^i - \hat{z}_t^{j(i)}\right)$
    $\Sigma_t = \bar{\Sigma}_t - K_t{}^i S_t \left[K_t{}^i\right]^{`T}$
return $\mu_t, \Sigma_t$
```

2.2.3 The Particle Filter or Sequential Monte Carlo

The Particle filter is a filtering algorithm directly implementing Bayesian recursive equations to calculate a posterior from prior distribution and input data. In comparison to the Kalman filter, the particle filter has the feasibility of using non-Gaussian prior distributions to present multi-modal posteriors.

The great advantage with the ability to represent multimodal posterior is the possibility to make multiple belief estimations of both localisation and mapping. This will be shown to be a great advantage in the favour of robustness of the filtering algorithm. The possibility has a direct effect on convergence because of the ability to represent multimodal beliefs, which was absent in the Kalman filters.

The filtering algorithm is based on two phases, in the same way as for the Kalman filter. The prediction has alternated its name to a so called “diffusion step”, but in principle it makes an estimate of the next state in the same way as for the KF and introduces a specific amount of disturbance to compensate for process model noise. The update step has been renamed to "re-sampling step".

In the diffusion phase, the particles are sampled through the process model and are presented as a predicted distribution. By receiving the second filtering parameter data, the update phase takes place. Then by using the data, the proposed distribution is corrected for the new measured data.

A basic implementation of the particle filter is presented in Table 2.7 which is an adaption from Thrun et al. [15]. From an initial particle distribution given by χ_{t-1}, particles are sampled through the specific process model (the diffusion step) to represent the proposal distribution, calculated in line 4. In line 5, the particle weights are calculated according to the probabilility of the measurement given the state. Line 8 to 11 performs the importance sampling step. Detailed explanation of the particle filter can also be found in Doucet and Johansen from 2008 [29].

Table 2.7: The Particle filter algorithm

```
 1 :  Algorithm Particle Filter($\chi_{t-1}, u_t, z_t$)
 2 :      $\bar{\chi}_t = \chi_t = \emptyset$
 3 :      for m = 1 to M do
 4 :          sample ${x_t}^{[m]} \sim p(x_t | u_t, {x_{t-1}}^{[m]})$
 5 :          ${w_t}^{[m]} = p(z_t | {x_t}^{[m]})$
 6 :          $\bar{\chi}_t = \bar{\chi}_t + \left\langle {x_t}^{[m]}, {w_t}^{[m]} \right\rangle$
 7 :      endfor
 8 :      for m = 1 to M do
 9 :          draw i with probability $\propto$ ${w_t}^{[i]}$
10 :          add ${x_t}^{[i]}$ to $\chi_t$
11 :      endfor
12 :      return $\chi_t$
```

The trick of the particle filter is the re-sampling step, and to explain it further, the following part will illustrate it in a more detailed way. Specific particle weights are calculated in line 4 as described above. The weight is best illustrated as in Figure 2.5 where the particles (sampled from the "proposal distribution", named g.) has specific weights which are represented in the bottom of the picture. The weight is a comparison of the "proposal distribution", named g, which points are sampled from, against the "target distribution" , named f, the true distribution.

It can be seen that the number of lines are denser distributed below the peak of the proposal distribution g, which is a direct outcome of the shape of the proposal distribution, the sampling is done from the proposal distribution. To correct this, particles are relocated below the peaks of the target distribution by use of the particle weights. The weights is as stated above, illustrated in the bottom of 2.5, here it can be seen that the weights are calculated from the difference between the proposal distribution and the target distribution.

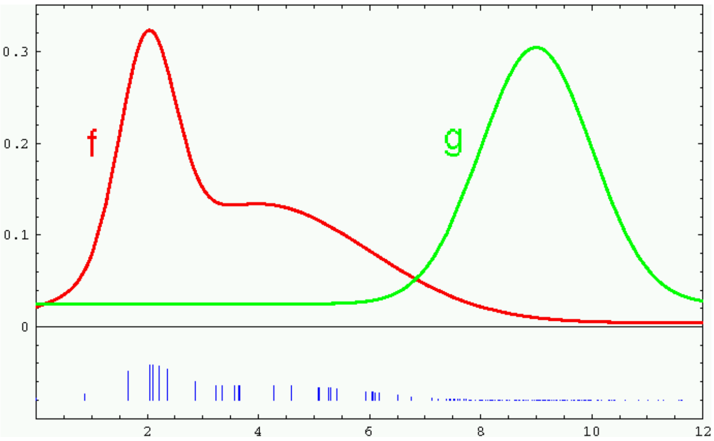


Figure 2.5: Re-sampling figure for particle filtering algorithm, adapted from Thrun et al. [16]

Weights are used to transform the proposal distribution and propagate the existing particles to parts where they make the best performance.
The weights are used to transform the proposal distribution to a better fit against the target distribution. The purpose is to populate the denser part of samples below the proposal distribution g in to below the target distribution f to better represent the target distribution.

To perform the re-sampling, a "Monte-Carlo" algorithm is used to draw particles to the next sampling iteration. The basic methods are described in Figure 2.6 by two different methods. The simplest method is described in the left picture, where a pointer is iterated for the number of particles to select the new particles which are weighted by their representative particle weight. The result is then a new particle set which will be used in the next iteration to make a estimation of the system.
The method implemented with this kind of re-sampling is often recalled as a "Vanilla particle filter". The method is heavily affected by sampling variance [16] which may lead to particle deprivation [16] and eventually to the failure of the filter.

A better way to do the particle selection process is to place a sun wheel with equally spaced pointers, and randomly rotate it. By probabilistic laws, the resulting new sampled particles will represent the particles with highest weight. This method is represented by an illustration in the right picture in Figure 2.6. The method is called "Variance sampling" and performs the re-sampling in a better way in considerations om sampling variance [16]. This effectively reduces particle deprivation of the filter.

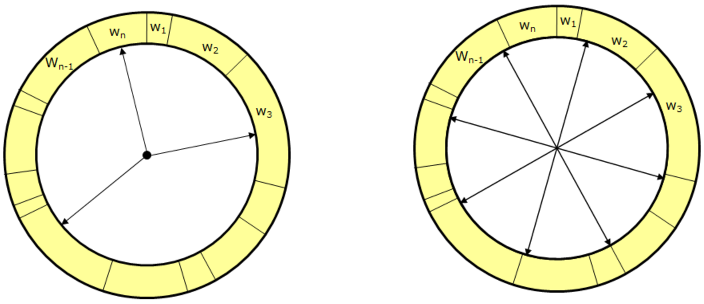


Figure 2.6: Monte-Carlo selection for particle weights. Adapted from Thrun et al. [16]

2.2.4 Mapping Structures

To represent the real world during localisation, computers use sensors to estimate their surrounding environment, and to keep the knowledge saved in a satisfactory way, specific computational structures are built to represent the environment.

It exists a great many ways of representing the environment. The most popular maps used in localisation and mapping are thoroughly analysed and represented in the following parts of this report. In the following sections "feature based maps" and "grid based maps" will be presented as the most commonly used maps. A special usage of range scanning data represented in a line structure is also presented in the end of this section. To simplify the grid map structures, the topological maps are quite widely used to simplify the grid maps by using simplified positions as a map structure. Topological maps will be discussed in the end of this section.

Feature Based Maps

Feature based maps are a structure of maps where each identified feature is referred to as a specific point feature. Features are often used in an outside environment where human architecture and structures are not the only object existing for identification.

Michel Montemerlo [30] used in one of his test runs in Victoria Park in Sidney a featured map to localise the robots position. The robot was running in a two dimensional space, with the tree trunks as the identification features used for localisation. In many outside environments, point features are easily identified. For a navigational purpose in e.g. Paris, The Eifel tower is a perfect landmark for identification due to its observability from most places around the city.

In comparison to inside office environments or more habitated areas with symmetrical environments, point features are more rarely observable. This affects the usage of featured maps to specific areas of usage and has to be selectively chosen for the intended application.

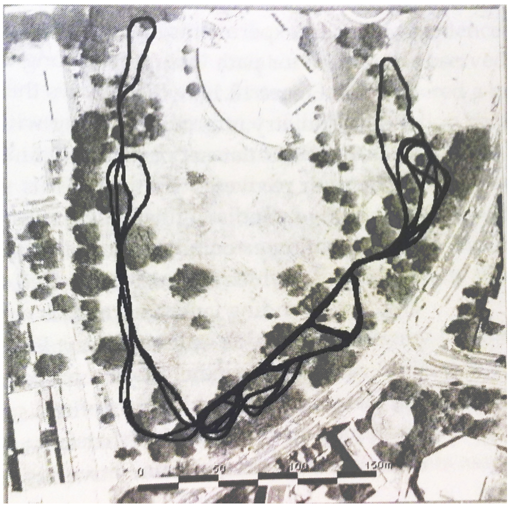


Figure 2.7: Michel Montemerlo used a featured map algorithm in a SEIF SLAM algorithm in Victoria park in Sidney [31] .

Featured maps has been greatly analysed in literature, Thrun and Burgard [16] has concluded that feature map structures is a very efficient mapping structure when feature objects exist and the identification of objects are easily distinguished from other objects. In the case when objects are harder to distinguish from each other, it exist a couple of methods to connect them to specific features.

The simplest algorithm for estimating correspondence, according to Thrun et al. [16], is the "maximum likelihood correspondence" method based on a minimization of the Mahalanobis distance [32] between the existing features and the measured object.
Mahalanobis distance is used because of its greater performance estimating multivariate distribution distances. The Mahalanobis distance includes the corelation of the state into the distance estimation which results in a better estimate as long as the covariance isn't represented by the identity matrix. As Tiku et al. [33] describes in a comparison of The Mahalanobis distance estimator for correspondence against least square estimator, the Mahalanobis method over performs the Most Likelihood estimator(the least square estimator). A great advantage of the featured mapping method is the great reduction of needed memory. In volumetric map mapping, the needed memory is growing quickly and becomes fast a real problem when exploration of bigger map structures is needed.

Volumetric Maps

To represent environments in a more independent way in relation to the geometrics of the environment , a more detailed map structure has been developed called grid maps or volumetric maps. The volumetric map is both representing areas in the map structures for features that exist, but even for free space.
Volumetric maps are based on a binary map system where each pixel is unexplored, occupied or free. This methods accuracy is strongly dependant of the decided grid resolution, among other limiting sensor parameters. The increased resolution has a bad effect on the needed memory with increased resolution. In two dimensional space, the map is constructed by a discrete grid in x and y directions. With increased need for dimensionality the increase of memory is increasing ex-

ponentially. Therefore, the needed memory is constantly growing with exploration independent of identified features. This is a big drawback against the featured map system.

The big advantage is the no need for advanced feature identification algorithms needed in the feature based mapping system. The possibility of identifying structures that are hard to identify by the feature based mapping model makes the grid map structure an often used structure in indoor environments [34].

Line Structure Maps

The effort and computational memory for grid maps confine its boundaries of operation. In the same way the limit for featured maps to represent straight walls and indoor environment are quite limited. To fuse both the property of keeping a low memory usage in the same time as representation of the environment is conducted in a detailed manner, a couple of line extraction algorithms for map creation have been developed during the latest years.
Castellanos and Tardos presented 1996 [35] a method for extracting line features from range scan data which have adaptions from earlier developed image recognition theory. Phister et al. [36] presented 2003 a weighted line fitting algorithm for mobile robot map building which performs initially a Hough transform for initial mapping and then using their proposed method for line fitting to range scanning.

Nguyen et al. [37] presented a big comparison of the most commonly used algorithms for line extraction for range scanners, a short summary of the presented methods can be seen in Table 2.8. Their conclusion is encapsulated to that “Split-and-merge” over performed most of the algorithms due to its lower computational time. They also mentioned that the behaviour is strictly behaving together with the environment its operating in and that the performance of the algorithms are likewise proportional in performance to that specific environment. The performance of the so far developed methods have been performed on LIDAR like laser scanners with a much higher beam accuracy and speed than ultra-sonic or infra-red range scanners and may have to be re-evaluated in those special cases.

Table 2.8: Most used algorithms for Line-Extraction. Adapted from Nguyen et al.[37]

Line extraction Algorithms
Split and Merge: Theodosios and Horwitz [38]
Incremental: Siadat et al. [39], Taylor and Probert [40]
Hough-Transform: Haidekker [41]
Line-Regression: Arras and Siegwart [42]
RANSAC: Forlani et al.[43]
EM: Pfister et al. [44]

Topological Maps

Topological maps are a generalisation of the environment represented by places. The environment is segmented and generalised into specific units. The places are then stored with its connectivity to other units. Graphs are then stored in a map structure in a much more compact form than for metric map representation.

Topological maps are of good use when navigating in not to complex environment like corridors or open spaces with not to much dynamics. Topological maps are often implemented in combination with a metric map allocating space between the topography, to get the best out of two worlds.

In Figure 2.8 an illustration of a typical topological map structure is shown. Specific places are assigned by rings which have in the long term been estimated by range scanning measurements.

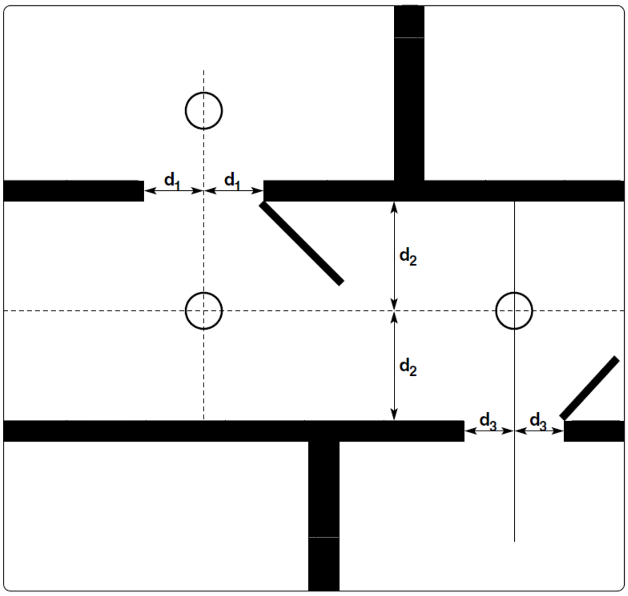


Figure 2.8: Example of placement of nodes, adapted from Althaus (2003) [45]

2.2.5 Process and Measurement Models

The design of models is an important part in the whole filter design, restricting how good the filters will perform, and applies to all of the filters discussed this far. The prediction step is often based on a direct translational and rotational model. In context of robot localisation, the motion models are often constructed from encoder feedback or acceleration from an inertial system, which in all cases are feedback of the control signal.
The construction of range scanner models is often more complex due to physical characteristics of the scanners. Models are though needed for the estimation of measurement range scanning data.

Process Models

Process model or the motion model. The intention of the model used in the prediction phase is to estimate the pose of the robot for the predicted posterior. Until quite recently have robot motion models been solely modeled by static models. Static model reflects the persisting behaviour of the model. In the case of a direct translational model, it wont be compensating the model in the case of a change in pavement changing for an example the slip factor. Static models have been used frequently and in the most cases works fine in a persistent environment. Dissanayake, Newman and Durrant-Whyte [46] used a static encoder model to prove that SLAM was possible from an unknown position, map an environment and localise it self in it.

To get a better performing SLAM algorithm dynamic models have though increased in popularity.

The models have often been constructed by experts and been evaluated and calibrated carefully, which is thoroughly described by Kaboli et al. [47] and Eliazar et al. [48].

For the static case there exist in principal two ways of implementing the models. One by using direct feedback from a control or in the other way using the control signal.

The firstly named and mostly used, the feedback control is popular because of its accuracy and do often sample feedback directly from the wheels. It's named the “odometry motion model” and uses the odometry motion or the wheel-rotation over time to calculate the position. Inertial sensors have increased in accuracy lately and are also a method using an approximation of the direct motion of the robot to construct a motion model.
The second named method is to create a motion model by the use of control signals fed to the vehicle.For an example, the rudder deviation of an airplane give feedback of how much and in what direction the airplane is steering. The odometry method is often preferred because of its greater accuracy, but isn't always possible to implement.

New innovations have lead to the development of more dynamically adaptable models, in a general context that may reflect the algorithms adaptability to change the covariance matrix of the sensor to better reflect the environment or in general to represent an environment where features are not static.
Visatemongkolchai and Zhang [49] describes two machine learning techniques for building dynamically motion models. A “recursive least square method” (RLS) is evaluated to design a static motion model from a state with no motion model. Also a so called “bi-loop recursive least square” (BiRLS) method is also evaluated for a more continuously dynamically changing model. The effort needed for dynamical models compared to statical models is the increased need for computational power and the effort needed implementing it, which has to be evaluated from situation to situation.

Measurement Models

Measurement models are like the process models a vital component of the the whole filter perception and estimation of a convergent posterior. Range scanning sensors have been used in a wide context together with SLAM implementations. In comparison to the laser scanning range finders, the ultrasonic sensor has a much more complex and dynamic behaviour and will because of that be the only sensor model considered in this report. Secondly the models established for the ultrasonic sensor can partwise be used even for laser range scanners, but with other coefficients.

Two distinct technologies do exist today for range scanning, laser/light emitted scanners or ultrasonic scanners. The different scanners has different characteristics considering accuracy and noise levels and will be discussed further in section 2.3.
Without a correct modell of the accuracy and variance in the SLAM algorithm, the filter will have a great potential for divergence. Hans P Moravec [50] was one of the pioneers developing a four step model for the design of a range scanner beam model.
The target model is described as the posterior $p(z_t|x_t, m_t)$, it represents the probabillity of the measurement given the prior map and state. The posterior is composed of the four different models described in Figure 2.9 which then are fusioned together representing the total beam model.

Figure 2.9 is a description of the beam model compounds. The upper left figure, index “a” is describing the Gaussian distribution when a measurement are measured against a feature. Its best estimate is calculated using a Raycasting function which calculates the range to a feature given the prior position and map. The variance of the Gaussian distribution is variable which

could be statically trimmed before operation or as in a more complex model, be tuned dynamically during operation.
The b picture is a decreasing exponential distribution which is representing the probabillity of unexpected features appearing in the environment.
The third picture, picture c is modelling the occasion of the beam attenuating in the environment or don't get reflected. This can be if the sonar is for an instance is mirrored against an angled surface or if a laser beam is hitting a very dark surface absorbing the whole beam.
The last picture is modelling a commonly distributed noise for occasions when phantom measurements occur in the sensor. All of the compounds are then fit together for the resulting beam model.

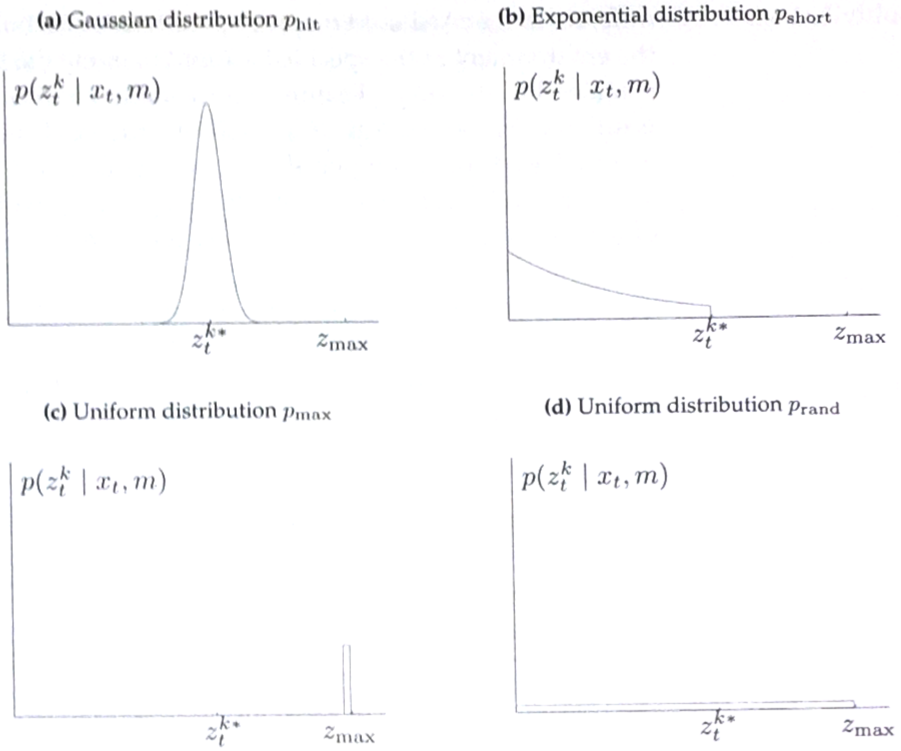


Figure 2.9: Components of distributions creating the beam measurement model. Picture adapted from Thrun et al. [16]

The first distribution, the Gaussian distribution in Figure 2.9 a, is assigned in equation 2.2.12 where the Gaussian distribution is describing the measurement uncertainty around the measured object. The probability function is called p_{hit} and represents the Gaussian probability of a hit given the prior state and map. The Probability is evaluated as a normal Gaussian distribution, unless the maximum beam range is reached which manually sets the probability to zero.
The Gaussian distribution is evaluated as in equation 2.2.13. The best estimate for the Gaussian distribution is evaluated by the use of the map saved in memory so far. By the use of a "Raycasting" technique, the ${z_t}^{k*}$ measurement variable is evaluated by estimating the range for every sensor to the first object in its vicinity.
Because of the if-statement in equation 2.2.12 its possible to get a non-normalised expression.

That means that a normaliser may be needed given the integral over the distribution. It can be seen represented in equation 2.2.14.

The distribution p_{hit} is evaluated as the coditional probabillity of the measurement given the state and the map

$$p_{hit}({z_t}^k|x_t, m) = \begin{cases} \eta \cdot N({z_t}^k; {z_t}^{k*}; {\sigma_{hit}}^k) & if\ 0 < {z_t}^k \leqslant z_{\max} \\ 0 & otherwise \end{cases} \tag{2.2.12}$$

The Gaussian distribution is derived by the use of the variance parameter ${\sigma_{hit}}^k$

$$N({z_t}^k; {z_t}^{k*}; {\sigma_{hit}}^k) = \frac{1}{\sqrt{2\pi{\sigma_{hit}}^2}} e^{-\frac{1}{2}\frac{({z_t}^k - {z_t}^{k*})}{{\sigma_{hit}}^2}} \tag{2.2.13}$$

and the final probability is scaled by use of the normaliser η

$$\eta = \int_0^{z_{\max}} N({z_t}^k; {z_t}^{k*}; {\sigma_{hit}}^k) d{z_t}^k \tag{2.2.14}$$

The second distribution is derived in equation 2.2.15, where unexpected objects are evaluated. The normalizer in equation 2.2.16 can be derived by integrating the distribution over the interval until the “Raycasted” object range.

$$p_{short}({z_t}^k|x_t, m) = \begin{cases} \eta \cdot \lambda_{short} e^{-\lambda_{short}{z_t}^k} & if\ 0 < {z_t}^k \leqslant {z_t}^{k*} \\ 0 & otherwise \end{cases} \tag{2.2.15}$$

$$\eta = \frac{1}{1 - e^{-\lambda_{short}{z_t}^{k*}}} \tag{2.2.16}$$

The third part of the Ultrasonic sensor beam model is describing the case when obstacles are missed during a scan. This can often be the case with Ultrasonic sensors due to specular reflection or for example soft objects like curtains and so on, which might reflect a very weak signal. The mathematical derived behavior for this is described by equation 2.2.17 which is describing a point mass at the maximum beam-range distance.

$$p_{short}({z_t}^k|x_t, m) = \begin{cases} 1 & if\ z = z_{\max} \\ 0 & otherwise \end{cases} \tag{2.2.17}$$

To model behaviours without explanation as described above, a continuous noise distribution is also included in the final model. The mathematical expression can be seen in equation 2.2.18

$$p_{rand}({z_t}^k|x_t, m) = \begin{cases} \frac{1}{z_{\max}} & if\ 0 < {z_t}^k < z_{\max} \\ 0 & otherwise \end{cases} \tag{2.2.18}$$

The final model for the behavior of a Ultrasonic Beam is then propagated by a sum of all of the models as can be seen in equation 2.2.19.

$$p({z_t}^k|x_t,m)=\begin{pmatrix} z_{hit} \\ z_{short} \\ z_{\max} \\ z_{rand} \end{pmatrix}^T \cdot \begin{pmatrix} p_{hit}({z_t}^k|x_t,m) \\ p_{short}({z_t}^k|x_t,m) \\ p_{\max}({z_t}^k|x_t,m) \\ p_{rand}({z_t}^k|x_t,m) \end{pmatrix} \tag{2.2.19}$$

Different beam models exists and are of great importance for the success of the filter and has to be evaluated for every implementation of slam. More detailed models for the beam model can be found in Blahut et al. [51].

2.2.6 Localisation In a Known Environment

Localisation in terms of autonomous robotics is the way a robot perceives the environment and determines itselves position in it. Localisation from a pose in an environment where a map is known from the start is a much easier task than both estimating the environment in the same time as you estimate your own position. The main reason is that reference to nearby features are already given which reduce the estimation to less than half the complexity of the SLAM problem.
This chapter will explain the usage of the previous discussed filtering algorithms and how its used for implementation together with localisation. The following part of the report will focus totally on ramifications of Bayesian inference. Dempster shafer Theory will be neglected because of the existence of a prior distribution and models of the system are either well known or can be derived. By this conditions the best method to use is clearly ramifications of Bayesian Inference.

The implementation of Localisation and as also later discussed Localisation and mapping are divided into two separated groups, much based on the kind of map used for world representation. Feature based map localisation is the first method which is coupled to uni-modal filtering techniques as the EKF or UKF. The representation is single modal which reflects the characteristics of the type of filter.
The second type of localisation are based on the world representation using a grid map structure. This type of localisation algorithm needs the ability of multi-modal distribution and are usually implemented using a particle-filter and called “Monte-Carlo Localisation” .

The case of estimating both the environment and the known position is discussed in chapter 2.2.7 where SLAM (Simultaneous Localisation and Mapping) is discussed in further detail. The parts below are focusing on the task of estimating the position when the mapping is already performed in a preceding context.
This problem is often applied to environments with low dynamics where the robot doesn’t need to adapt for changes in the structural environment, and the mapping is easily performed before initiating the robot. Typical environments where such applications are suitable could be everything from warehouse administrating robots to robots transporting goods in corridors where the environment may be dynamic, but the structure is not changing.

EKF Localisation

The EKF Localisation algorithm is based on the “Extended Kalman filter” [16], which linearizes the system model to cope with nonlinear process and/or sensor-models, derived in table 2.5. The EKF localisation algorithm is limited to unimodal Gaussian distributions which can be a reason for divergence in the estimation when the linearization is unsatisfactory, or linearized around an incorrect state. The models need to be carefully evaluated according to its nonlinearities and

the behaviour of the linearization.
To improve the linearization, an expansion of the EKF exist (IEKF) which performs an iterative procedure to linearize around better and better states. An example is given by Sargantanis and Karim who analyzed a process control by the use of IEKF [52]
The EKF localisation is restricted to applications with feature based mapping. Another restriction of the EKF localisation is the possibility of only estimating unimodal distributions. Adaptions to the EKF exist to cope with multimodal distributions in some way which also allow estimations to adapt with situations when correspondence is ambiguous. This is evaluated by Wilbur et al. [53], but to better cope with multimodal distributions, other filters are recommended.

UKF Localisation

The Unscented Kalman Filter (UKF) is a method very similar to the Extended Kalman filter with the difference that It does not use Taylor expansion for linearization. Burghard et al. explains it in Probabilistic Robotics [16] as a way of mapping points through the nonlinear behaviour and then approximate the mapped points to a Gaussian distribution.
The UKF filtering algorithm is also limited to unimodal Gaussian distributions in its basic design. It is selecting a specific number of points from the previous distribution, called Sigma-points which then are mapped through the process model. The Unscented way of coping with nonlinear behaviour of the robot is often better than the direct linearization of the mean. It usually gets a more spread distribution which adapts better to the model.
The computational complexity is the same for both the EKF and the UKF [16] but the UKF lacks the need of derivations calculating Jacobian matrices.

Monte-Carlo Localisation

Monte-Carlo localisation is based on a particle filtering algorithm as described in table 2.7. The particles are sampled from a proposal distribution as explained by Thrun et al. [16] and then mapped through the process model and weighted according to Particle filtering algorithms. See Doucet or Thrun et al. [29] [16].
The biggest difference of Monte-Carlo localisation algorithms against Kalman filtering algorithms are the natural possibility of estimating multimodal positioning hypothesises. Which is an effect of the particle filter used which can propagate nonlinear models and present the posterior in other forms than Gaussian distributions.

2.2.7 Localisation In a Unknown Environment

The problem of estimating your location and the map in the same time is often referred to as a “chicken or egg problem” because of the need for one source to estimate the other.
Great effort has been spent on solving SLAM in the latest years, considering both the estimation of localisation and mapping in the same time. Truly autonomous navigation is depending on the possibility of perceiving the environment in the right way and to interact with it.
Below will the different algorithms for SLAM be discussed with its pros and cons. The algorithms are built on the general filtering algorithms discussed in the chapters above.

EKF-SLAM

As presented by Thrun et al. [16] EKF SLAM was the earliest and perhaps most influential algorithm for simultaneous localisation and mapping. The algorithm is applying an extended Kalman filter to manage performing online SLAM. This is done by most likelihood estimators which are subjected to a number of limitations and assumptions which reduces the performance of the algorithm and may influence its convergence.
Limitations in the design of the filter is limiting the mapping structure to featured maps, the number of features in the map are also kept to a low number to keep the computational performance at a high level.

Nonlinear behaviour in model structure is handled in the same way for the EKF SLAM algorithm as for the filter alone. First order Taylor expansion is used to linearize the models for better performance. The linearization is far from optimal and will in many cases be subjected to divergence in the filter and the SLAM algorithm will fail.

In an analysis by Lee et al. [54] it's proven also that false landmark registration in the field of landmark association is a very big issue to EKF SLAM algorithms for not converging. When nonlinear models are existing the linearization of the model will implicate that the most likelihood estimator will have a possibility of adding landmarks in the case where should have been subjected to an update, which easily will affect the convergence of the SLAM algorithm.

UKF SLAM

According to former analysis by Gyaquan et al. [55] of the extended Kalman filter and the unscented Kalman filter, the Unscented Kalman filter will make a better approximation to nonlinear conditions than the extended Kalman filter.
The filter is using particle mapping in a better way to approximate the often banana-shaped output of the nonlinear motion model. It approximates the Gaussian output in a more correct way, compared to the true nonlinear function, in relation to the the Taylor expansion approximation. The computational complexity is in the same order as for the extended Kalman filter but may need some extra time for each calculation. The greatest advantage by using sampled points is the better protection against divergence.

Sparse Extended Information Filter - SEIF SLAM

Sparse Extended Information Filters base the filtering algorithm on the same techniques used for the EKF. The big difference is that the EIF is using an information matrix instead of the covariance, which in practical context is the inverse of the covariance matrix. The cause of inverting the calculations used in the EKF is because of the easier computational complexity which may occur for specific systems.
The sparse prefix in the name referees to how the information matrix is used. Big parts of the matrix do assume very low values after the inversion. By making an easy assumption and assume the low values to zero instead, will significantly reduce the computational complexity. The SEIF algorithm was invented by Thrun at al. (2004) [56] and is referred to for further explanation.

The SLAM implementation is done by the use of an online SLAM algorithm. Thrun et al. developed 2004 a robot with a SEIF slam implemented for exploration and mapping a 3D environment of abandoned mines with sucessful result. [57]

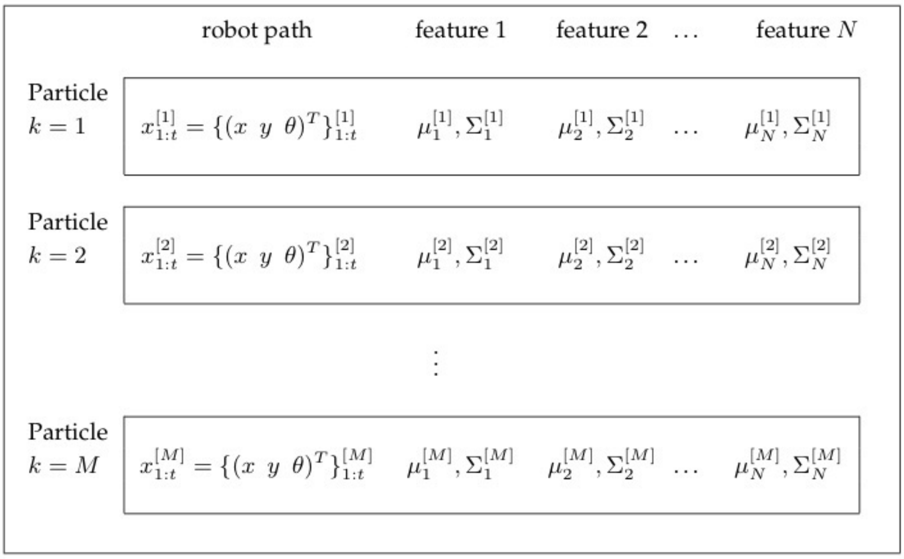
Fast SLAM 1.0 - Rao-BlackWellized SLAM

The Fast SLAM algorithm was invented and tested 2002 by Michael Montemerlo [30] as a great performance improvement of the earlier mostly used EKF SLAM algorithm for autonomous operation.
The Fast Slam algorithm is based on a so called Rao-Blackwellized particle filter[30]. The particle filter is using a method where each particle is representing an individual distributed path of the robot. For each path of the robot, the range scan measurements are used to create an individual map for each particle. This is represented in Figure 2.10 where each particle pose can be seen on the first column on the y-axis and the map features represented as a covariance matrix and a best estimate on the x-axis. This is explained in detail by Thrun et al. in “Probabilistic Robotics” [16].

The big difference from the “Monte-Carlo” localisation algorithm, is the need to continuously construct the map representing the environment. The three previous discussed SLAM algorithms has been implemented using a feature map representation. In the case of Fast SLAM using a particle filter and a grid map, a more thorough implementation constructing the map is needed.

The method plotting the map with probabilistic reasoning is called “Occupancy-Grid Mapping”. By the use of the prior map and the new measured features, represented by range and bearing from the estimated pose, the grid is logically evaluated. Each cell by cell, the occupancy is evaluated and the map is iteratively updated.

The big performance increase in comparison to the previous used SLAM algorithms is due to the smart use of efficient and small sized EKF filters calculating the individual features. The robots pose is initially calculated by a particle filter, instead of calculating a huge sized matrix representing the map features, each feature is calculated by a two by two Kalman filter. This is decreasing the landmark computational complexity from a quadratic order to a logarithmic order.



	robot path	feature 1	feature 2	…	feature N
Particle $k = 1$	$x_{1:t}^{[1]} = \{(x\ \ y\ \ \theta)^T\}_{1:t}^{[1]}$	$\mu_1^{[1]}, \Sigma_1^{[1]}$	$\mu_2^{[1]}, \Sigma_2^{[1]}$	…	$\mu_N^{[1]}, \Sigma_N^{[1]}$
Particle $k = 2$	$x_{1:t}^{[2]} = \{(x\ \ y\ \ \theta)^T\}_{1:t}^{[2]}$	$\mu_1^{[2]}, \Sigma_1^{[2]}$	$\mu_2^{[2]}, \Sigma_2^{[2]}$	…	$\mu_N^{[2]}, \Sigma_N^{[2]}$
	⋮				
Particle $k = M$	$x_{1:t}^{[M]} = \{(x\ \ y\ \ \theta)^T\}_{1:t}^{[M]}$	$\mu_1^{[M]}, \Sigma_1^{[M]}$	$\mu_2^{[M]}, \Sigma_2^{[M]}$	…	$\mu_N^{[M]}, \Sigma_N^{[M]}$

Figure 2.10: Rao-Blackwellized particle filter structure, adapted from Thrun et al. [16]

The key mathematical insight that the Fast SLAM algorithm is holding to is the conditional independency of the landmarks, which can be used to depict a very efficient way of calculating the full SLAM posterior. By drawing the dynamic Bayesian network in Figure 2.11, a direct visualisation of the independency conditioned that all the poses of the robot are known.

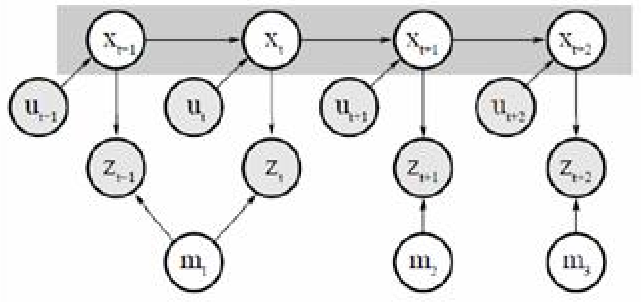


Figure 2.11: Conditional independence visualise by dynamical Bayesian network, adapted from Thrun et al. [16]

By the use of this independence, the full SLAM posterior can easily be calculated as the product of the pose posterior times all the map posterior, see equation 2.2.20

$$p(y_{1:t}|z_{1:t}, u_{1:t}, c_{1:t}) = p(x_{1:t}|z_{1:t}, u_{1:t}, c_{1:t}) \prod_{n=1}^{N} p(m_n|x_{1:t}, z_{1:t}, c_{1:t}) \tag{2.2.20}$$

The Fast SLAM algorithm have been greatly used in a lot of robot applications the latest years [58],[59],[60] and governed a huge popularity because its possibility to observe multimodal distributions in the same way the calculations are performed in a effective way. The problem which still exists is the usage of memory which grows huge when implemented in volumetric maps in the same time as the world is observed in multiple dimensions.

Fast SLAM 2.0

Fast SLAM 2.0 is an extension of the original Fast SLAM algorithm, the extension is also made by Michel Montemerlo [61] and focus on improvements of the prediction step. To get a better estimation of the pose for a robot travelling e.g. in a corridor, the measurements can be implemented into the prediction step. This is exactly what Montemerlo did by integrating the measurement into the calculation of the prediction posterior.
The principal change can be expressed by equation 2.2.21 where the sampled points x_t are sampled according to the probabillity of the posterior state given the prior state, the control signal, the measurement and the prior map.

$$x_t \sim p(x_t|x^{[k]}_{1:t-1}, u_{1:t}, z_{1:t}, c_{1:t}) \tag{2.2.21}$$

The effect of that will for example be that uncertainty decreases when features are identified by measurements. For an example when you travel in a corridor and measures the side-wall distance, the uncertainty in estimation of angle will be much more precise when measurements are implemented. The global effect of this will be that the number of particles needed for convergence can be decreased which increase the performance of the total algorithm.

The backside is that it can be costly and hard to implement in the algorithm. The need for a dynamical control of particles may also be needed for situations when measurements has high variance or are non existing.

2.2.8 Summary of the Section

Localisation in a global environment isn't solved as a problem with one solution as of today. It exists a great many ways of implementing localisation even for a narrowed architecture of sensor applications. To in a good way analyze which method that fits best for a specific application, one have to evaluate the needed performance and precision that is required for the requested task. Also the physical characteristics of the environment in the sense of dynamic behaviour, geometries of features and constructions and also the dimensionality of the space. With these questions answered, a good narrowing of usable methods should be possible to select.
The remaining limitation is of course the very varying computational complexity between algorithms. With increased flexibility in the algorithm, the computational complexity grows. The needed memory allocation is also a subject of concern for the design.
No global solution exist for localisation in an unknown or known environment, but with the above context, a selection of the best methods for individual cases should be able to be selected.

Table 2.9: Comparison of the different SLAM algorithms

	Algorithms used in Simultaneous Localisation and Mapping			
Item	**EKF SLAM**	**UKF SLAM**	**SEIF SLAM**	**FAST SLAM**
Computational Compl.	Medium	Medium	Low	High
Memory/Map Relation	Quadratic	Quadratic	Linear	Linear
Implementation Complexity	Low	Medium	Medium	High
Posterior Distr.	Gaussian	Uni-Modal	Gaussian	Multi-Modal
Mapping	Feature	Feature	Feature	Grid/Feature
Robustness	Low	Medium	Medium	High

2.3 Range scanning hardware - State of the Art

When implementing SLAM on a robot platform the robot some how need to be able to “read” the environment in order to locate it self and build a map. The two more important functions for the robot are to know how it has moved and what is hindering it from moving further. Encoders put on the wheels, if the robot is wheel based, is the easiest way of knowing how it has moved. One can also think of inertial measurement units (IMU) and positioning methods, see section 2.4. Although there are different types of encoder solutions and other possible ways of solving the problem it can still be seen as a fairly easy task compared to the other problem, to know the position of the obstacles in the environment. The solution to this problem is often called range scanning, which will say the robot scans the environment and sense obstacles at different ranges. To achieve this, different range sensors can be used and as always these different sensors have both advantages and disadvantages.

This section will not cover the first problem due to its fairly simplicity, but will focus on the range scanning sensors that are often used. Techniques and sensors that have been used for range scanning in SLAM implementations are laser range scanning[62, 63, 58], ultrasonic scanning[64, 65, 66], vision based scanning[67, 68, 69] and infrared light scanning[70, 71, 72]. This section will evaluate these techniques in the sense of range, cost, beam-width, updating frequency, accuracy and disturbances. One can also think that touch sensors or physical sensors can be used for SLAM, however this require that the robot touches the obstacles which is not of interest for this thesis work and will not be covered.

2.3.1 Laser range scanning

Laser range finders, or often called LIDAR, are the most widely implemented technique on state of the art SLAM implementations[73] and for example used at the DARPA challenge[1]. This is mainly because of the accuracy they provide, distance range and the area scanned versus time, which will say the updating frequency to scan a certain area. However, there are some drawbacks with laser range scanners.

The main principle of range measurements when using laser range scanners is for the scanner to transmit a laser beam. This beam will be reflected on an obstacle and return to the scanner that receive the signal. To measure the distance from the returned beam mainly three different techniques are used, namely time of flight (TOF), triangulation and frequency/phase modulation detection. The most widely used of these three is the TOF technique. The laser range scanner sends a pulse periodically and then measure the time for this laser pulse to return to the receiver[74]. By knowing the speed of light the distance can easily be calculated by $distance = TOF$ x $SpeedOfLight/2$. Because the high speed of light this implementation needs high quality micro-controllers(MCU) to exactly measure the flight time. Which has led to that it is not until recent years the laser range scanner has been deployed on mobile robot system utilizing SLAM. However, as early as 1983 Jarvis[75] developed a system that only used 10 samples/point for scanning a scene and that he said could be deployed on a mobile robot. Todays system uses over 4000 sample points per second which can be achieved with low cost MUCs[76]. That means that more samples can be taken at the same point to get a higher accuracy with the use of a mean value.

To be able to get 2D or 3D scanning, mirror/mirrors or other optic hardware is often used. With this the scanner can control the beam direction and therefore scan up to 360 degrees of

[1] Read more about DARPA challenge here: http://en.wikipedia.org/wiki/DARPA_Grand_Challenge

the environment. This technique is utilized in two of the more common scanners used on mobile robot platforms, the SICK LMS 200 and Hokuyo URG-04LX. To get a good scanning the use of precise encoder to know the scanning angle is important, as well as the need of high quality calibration of the mirror to point where it should. It can be expensive to achieve this and is one of the reasons laser scanners become expensive. Konolige et. al.[76] therefore proposed a low cost system that can be used on consumer products as robot vacuum cleaners. Their proposed solution is currently used by Neato Robotics XV-series vacuum cleaner and is based on a triangulation technique.

A triangulation technique is not that widely used in mobile robot system today but as mentioned above Konolige et al.[76] proposed a system based on that and with a component cost of only $30. The system uses a laser pointer and a CMOS sensor that are calibrated and pointed in certain angles and separated by a small baseline distance. Depending on where the reflected image of the laser beam hit the CMOS sensor, the distance can be calculated with different trigonometric functions. To get a 360 degrees scan a motor turns the whole system and an encoder senses the direction. The system still needs calibration but not as high quality as the mirror and TOF based systems and also uses more cheap and basic components. It still gets an accuracy of three centimeters out to six meters of range. However, because of the use of an fairly low-cost image sensor the reflected laser dot cannot be to small to be detected and this will limit the range of the scanner. The relationship between the image and the reality that is used by the triangulation technique is non linear which will also poses problems for measuring longer distances[76].

The modulated technique also makes use of the flight time to calculate the distance. However, it not uses a timer to time the flight, instead the laser beam power is modulated either by frequency or sinusoidally. By comparing the received beam with the sent beam the difference in frequency or the phase of the sinus wave will tell the flight time[77].

As all measurement methods laser range scanners has its pros and cons. A laser beam is highly focused and contains high energy, which mean it will keep its beam narrow over a longer distance compared to ultrasonic and infrared LED measurements. This yields that the measurements are more accurate or “sharp”.

When scanning in 2D and 3D the scanner need to sweep the laser beam as mentioned. To get good accuracy a good encoder and motor is needed. As Boehler and Marbs[78] stated, the angular accuracy of the laser scanner will depend on the accuracy of detecting the correct beam angle as well as the possible motor increment steps. This however, is not just a problem for laser range scanner, but for both ultrasonic and IR scanners that also can be rotated in order to scan the environment.

Boehler and Marbs[78] has developed some test beds in order to measure accuracy of laser range scanners. Besides angle accuracy, they stated that range accuracy, resolution, edge effects and surface reflectivity are problems for laser range scanners. For a TOF based and frequency/phase modulated range scanner the range accuracy is constant over the whole range. Triangulation based methods on the other side has more problem, the range accuracy decrease with the square of the distance between the obstacle and the scanner[78].

Resolution can mean many different things, Boehler and Marbs[78] state that it is the ability to detect small objects or features. This depend highly on two parameters of the laser range scanner, namely the size of the laser dot and the smallest possible increment of the angle the scanner can do.

In the vicinity of the edges “wrong points” like artifacts or phantom points will affect the range measurements. The wrong points is caused by laser dots hitting the edge but only a part of it will be reflected. The other part may be reflected by objects behind the edge or not at all. This can cause range error of up to several decimeters[78]. A solution would be to have well focused lasers with smaller laser dots to increase the accuracy near edges.

The performance of a laser range scanner will be influenced of the objects surface reflectivity (albedo). A surface with high albedo will reflect more energy of the laser beam than one with low albedo, like a black surface. Depending on the sensitivity of the receiver, different laser range scanners will be affected differently. The color of the surface as well as the glossiness will also affect the performance. Some range scanner had range error of up to 25 centimeters depending on the surface reflectivity, color and glossiness during an empirical test performed by Boehler and Marbs[78].

Surfaces made of glass or a mirror is often not detectable. The laser beam will pass through the glass and measure the distance to an object on the other side of the surface. When in the case of a mirror the laser beam will be totally reflected and the range will be measured in the virtual room “behind” the mirror. This could be a big problem when laser range scanners are used in SLAM implementations. To fuse sensor data is therefore common. If ultrasonic is combined with the laser range finder, transparent surfaces and mirrors can be detected[73, 66].

One should also think of the eye-safety when choosing a laser range finder. Most non-scientific used laser range finders uses laser with a wavelength of 600-1000nm. This wavelength can affect the human eye if the power is too high and the eye is exposed during a to long time. Higher power of the laser will increase the accuracy, especially in ambient light, but will make it non eye-safe. However, if short pulses are used higher power can be allowed[76]. The response of some receivers may be affected with the use of shorter pulses, like the CMOS sensor used in the proposed solution from Konolige et al[76]. It thus becomes a trade off between ambient light rejection and receiver response.

2.3.2 Ultrasonic range scanning

The use of ultrasonic sensors in low-cost autonomous vehicles and industrial applications is usual because of the low cost and simplicity of the sensors. One of the first autonomous vacuum cleaners from Electrolux used ultrasonic sensors to map the environment[79]. However, because of the simplicity the accuracy is also pretty low compared to for example laser range scanners. A panoramic range scanning cannot either be performed as fast as the laser or IR sensors because of the lower speed of sound.

The main principle for range measurements when using ultrasonic sensors is the TOF technique. As for the laser range finders this can be accomplished in mainly two ways. The first technique measures the time difference from transmitted short pulse signal and the received echo signal. This is called pulse-echo techniques[80]. The second technique uses a continuous sonic wave, which is received by a separate receiver. The TOF is then determined by the phase shift between the transmitted and received signals[80]. The second technique often requires more complex hardware than the puls-echo techniques, but are more accurate according to Marioli et al.[80].

A problem that mainly affects the pulse-echo technique is attenuation of the transmitted sonic wave. This is because the receiver often uses a threshold value for the amplitude to register the received signal. The propagation medium and its conditions, like air currents and temperature[81], will affect the attenuation of the sonic wave , which increases with the frequency of the sonic wave[80]. This will limit the maximum range for the sensor in a certain medium. Different surface material, which absorbs the sound energy different, affects the attenuation as well.

Surfaces that have high absorption capacities will be hard to detect for an ultrasonic sensor. Some material and obstacle can even become invisible to the range sensor because of the attenuation from the absorption. This phenomenon is similar to the difficulties the laser scanners has with different surface colors and albedo. Marioli et al.[80] mention that an adjustable amplitude threshold will improve the accuracy and lower the impact of the attenuation.

An ultrasonic sensor often relies that it is directed perpendicular to the surface it measures. This will make it hard for the sensor to actually record the true normal to a surface, that will say it will have problem to find oblique surfaces[81]. As early as 1985 Brown [81] propose an method to obtain the true normal of a plane and also find oblique surfaces. He also provide a novel approach to find curved surfaces with ultrasonic sensors.

In near relation to the above difficulties to find oblique surfaces is the effect of specular reflections[82]. That is, the problem when a sonic beam fails to return directly to the transmitter-receiver unit. Instead the reflected signal is "bouncing" away from the sensor and miss-readings will occur. The sensor will assume that the distance is longer than it truly is because of that the signal has taken a longer path (longer time difference) than the direct path. Specular reflection normally occur because of the incidence angle of the sonic beam is bigger than half the beam angle[83] and because of the surface material[82]. A proposed solution to the uncertanities that will be caused by the specular reflection was presented by Yi et al.[82]. They used a smart algorithm that utilize the Dempster-Shafer method, see section 2.1.2, in a sensor fusion with a sensor model.

The major drawback with the use of ultrasonic range scanners is the wide beam angle, which will cause a low angular accuracy. An ideal sensor will have an angle of up to 30 degrees[84]. If the sensor finds an obstacle it can be anywhere on the "circle segment" created by the beam angle, which in longer ranges become a very high uncertainty. This makes it hard to identify certain feature like edges and corners. There will also be problem if more than one object is located in the beam and these objects are placed at different ranges. The beam angle is dependent on the diameter of the transmitter and on the signal frequency[81]. A higher frequency or bigger diameter will decrease the beam angle, but as mentioned earlier the attenuation created by the propagation medium also depends on the frequency. Thus there is a trade of between the range of the scanner and the beam angle. The threshold level for the received amplitude will also affect the beam angle. However, a higher threshold will cause more sensitivity to attenuations.

A solution to the problem of differencing reflection caused by corners and planes were investigated and proposed by Barshan and Kuc[85]. Instead of a single transmitter-receiver module they proposed the usage of a multi-transducer. Hernández et al.[86] also propose a multi-transducer sensor array to solve problems with finding different features, including edges. By construct two "vector sensors" that are able to measure the reception angle of the echo and put these in a sensor array they could measure the received angle and distance of a feature. Zunino[79] present in his thesis report the possibility to uses multiple transducers to triangulate different features in the environment. The problem when using more than one transducer is the effect off crosstalk. That is, one ultrasonic sensor receives an echo caused by the transmission of another ultrasonic transmitter. One usually says that there is mainly two types for crosstalk, internal and external crosstalk. Internal crosstalk is caused by the own system sensor array, when the external crosstalk is caused by external factors like other sensor arrays. External crosstalk is hard to avoid, but there are strategies to handle the internal crosstalk. These generally fall into two categories namely, firing strategies[84] and unique code modulated signals[87]. Meng et al.[83] has conducted a comparison between different solutions to the internal crosstalk problem.

Other problems that can influence on the quality of the readings when using ultrasonic range sensors are the roughness of the surface and detectable sizes of obstacles. A surface will be acting as an ultrasonic mirror only if the roughness is low and the surface can be considered as smooth. One can think of when a beam hit a rough surface that the echo will be reflected in more directions and the amplitude that is received at the transducer will be attenuated. If the size of the obstacle to be measured is to small the beam will not be reflected and therefore no range will be measured. The closer the obstacle is the more of the beam will hit it and the chance of reflection will therefore be increased. One shall also think of in which environment the sensors shall be used when choosing operating frequencies. Some animals are sensitive to high frequencies and one may therefore chose an even higher frequency then planned. As for the laser range finders the accuracy of a scan will depend on the accuracy of the motor and angle position sensor used. If one choses to have fixed positions and use more sensors, as Jörg[84], the calibration of their exact pointing angles will be important for a correctly built map of the environment.

2.3.3 Infrared light range scanning

Infrared (IR) range sensors is defined by the authors of this master thesis report to be range sensors that utilize an IR LED. This definition is done to differentiate laser range scanners that also uses IR light sources, but with higher energy and that are better focused than LEDs, from sensors that uses a regular low energy IR LED. Because of the similarities with laser range finders the measurement techniques are the same, namely TOF, phase delay detection, and triangulation[88, 89]. One of the more widely used IR range sensors is developed by Sharp[1] and utilizes the triangulation-based technique.

The advantages and disadvantages for IR based range measurements are very similar to those for laser range finders. For example, because of the use of light the sampling rate can be high, but the sensor will have problem with transparent surfaces and surfaces consisting of mirrors[90]. Novotny and Ferrier[91] discuss the use of an illumination model to compensate for some difficulties that occurs because of surface colors and reflectivity.

However, there are some differences between IR range sensors and Laser range finders. A laser range finder have a more accurate range data over longer ranges[90] compared to an IR range sensor. A typical maximum range for an IR sensor is 1.5 meters but can be as long as 3 meters[88]. This is basically because of the lower energy content in the signal created by a LED and that the IR beam is less focused. The lower energy content also makes it more sensitive to ambient light. Even though the energy content is lower and that the IR beam is not as focused as the laser beam the IR beam is still narrower than the ultrasonic beam. The angular accuracy of the IR sensor is therefore higher than for an ultrasonic sensor but lower than for a laser range finder. The resolution, the ability to detect small items, of the IR sensor will also be lower than for the laser range finder because of the beam width but still higher than the resolution for the ultrasonic sensor. A smaller object may reflect light and allow a measurement, but there will be light that travels past the item and may hit the background. Alwan et al.[90] refer to this problem as the mixed pixel problem. The range reading will be somewhere in between the object distance and the background distance. The authors of this thesis therefore doesn't see this as an acceptable reading and item with a size that give rise to this phenomenon is not seen as the smallest detectable item.

Although the IR sensors isn't as good as the laser range finders, they are less expensive, simpler,

[1] One model of the Sharp IR range sensors: http://www.sharpsma.com/webfm_send/1208

not so bulky and heavy as laser range finders. This makes them perfect for a small low-cost robot that shall operate in an indoor environment where obstacles don't need to be detected in longer ranges.

2.3.4 Vision based range scanning

This type of range measurement is the one that has least in common with the earlier mentioned ones. The basic concept of most vision based range measurement techniques is to analyze at least two different images and compare pixels in these. By the spatial difference of pixels associated with the same feature the depth, or range, can be calculated[92]. One can think of the human vision, where the two eyes collect two images of the same features but with some spatial differences. In robotics system the images often is collected by stereo cameras, that will say two cameras directed in the same way but separated with a certain distance, or by using one camera, also called mono-camera based, that takes images at different locations of the robot. Regardless of which collection technique is chosen one will need a camera model in order to map the calculated 3D image position to the real world. Other methods that can be used incorporate some projected references, like laser beam patterns, as a grid. The camera then can use this reference to calculate angle of arrival and further the distance[93]. A well known implementation of this method is the Microsoft Kinect device that uses a technology developed by Prime Sense Ltd[94] for the depth measuring[1].

Stein et al.[95] proposed another method for a specific case that utilize the mapping of image projected height and the real height of the object. Their module had the task to measure range and range rate between the car it was mounted in and the car in front. This in order to control the adaptive cruise control feature in the car. The environment were they used their method is pretty constrained because the camera height is known and object to measure will always lie on the same planar plane and is most probably not suitable for an autonomous robot in indoor environment.

One of the bigger advantages when using vision-based techniques is the low cost and weight of the equipment needed, at least for some systems. Today most of the basic web cameras have high enough resolution to provide good images for the algorithms. The images sequence is also rich of information and there is no need to move the camera as for the laser range scanners to calculate bearings and range to different objects. A vision-based system is also rarely restricted in maximum range[96] as long as it can detect and match features.

This particular task, to match features, is also one of the bigger difficulties for the visual-based systems[96]. Advanced pattern recognition algorithms and image processing algorithms is needed. The system has to take different lightning into account, darker scenes will be more difficult to analyze than lighter and the success highly depend on the processing algorithms. A match between the two images of a scene with similar or repeated pattern or occluded regions, which will say a region that is not visible in one image of the two, will be more difficult. Such a scene will require a more advanced processing and matching algorithm. More advanced algorithms will ultimately require more powerful processors and hardware which can be more complex and expensive. Darabiha et al.[97] proposed the use of a FPGA circuit to handle more information quicker. Goldberg et al.[98] shortly describes their algorithm for image processing and matching that are proposed to be used at NASAs mars rover. A lot of other different approaches for image processing and matching exists, but will not be any further described in this thesis because of

[1] `http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf`

that no vision-based range scanner will be deployed in the implementation part, due to its higher complexity and the limited time that was available.

As for the laser scanner the vision-based system will have problem with detecting mirrors and transparent surfaces. The collected images of a mirror or a window will contain features, reflected or seen through, that will be interpreted at ranges more far away than the mirror or window actually is. This will cause the robot to miss the surface and believe that there is free space there. The easiest way of solving this problem would be to use a sensor that do not use light sources, like an ultrasonic sensor, and fuse them together. One may also think of some algorithm to recognize a mirror that have a frame, but this will add another algorithm to the already needed advanced algorithms.

Further, for the stereo vision systems the higher cost and calibrating difficulties are some problems that arises when compared to a mono-vision system[92]. Advantages when using stereo vision is that the two images to be used is collected at the same time and no motion for the robot is needed. This is one of the bigger disadvantages for the mono-vision system. It highly depends on the accuracy of the robot positioning, like encoders. The mono-vision system also suffers from an initial delay. When a new feature is being added to the map it will just be partially initialized before enough views of the feature is acquired from different locations[68]. Song et al.[92] proposed a method that is able to take different images at one location with only one camera and a mirror. By taking the images “through” the mirror and then rotate the mirror, different images of the same feature will be collected. These images were then matched and ranges could be calculated by the difference in how much the feature had “moved” or as the researchers called it the speed of the pixel movement. Features at shorter ranges will move shorter, have lower speed, than those in longer ranges.

No matter what vision technique that is used interesting features still need to be found in the images. Those algorithms struggles with almost the same difficulties that the matching algorithms, like light in the environment. As for the matching algorithms there exists a lot of different image recognition and feature recognition algorithms and solutions. Because of that no vision-based range scanner will be deployed in the implementation part of this thesis no further description of those algorithms will be made.

2.3.5 Summary of the section

This subsection will summarize the range scanning hardware section. The most important items that are worth thinking about are shortly compared relative to each other in Table 2.10 and explained. A high grade doesn’t mean that for example the accuracy has to be down to millimeters or micrometers, but instead it is much better than a method with low grade. The table shall be seen as an aid when choosing range scanning hardware and not as an answer.

Table 2.10: Comparison between the different range scanning hardware techniques.

	Range scanning techniques			
Item	**Laser**	**Ultrasonic**	**Infrared**	**Vision-based**
Cost	High	Low	Low	Medium
Resolution	High	Low	Medium	High
Angular accuracy	High	Low	Medium	High
Range accuracy	High	Medium	Medium	High
Complexity	Medium	Low	Low	High
Updating frequency	High	Medium	High	Medium
Range	Long	Medium	Short	Medium
Size & weight	Bulky	Small	Small	Medium
Surface problems	Albedo, Mirror and Glass	Absorption Incidence angle and corners	Albedo Mirror and Glass	Repeated pattern, Mirror and Glass
Ambient lighting	Low	N/A	Medium	High

The first thing one may think of when choosing hardware is the cost, especially in a mass produced consumer product. The vision-based system may be considered as a low-cost module if a web-camera is used. However, in most systems better image quality then the cheapest web-cameras can provide is needed to reach the grades presented in the rest of the table. It is also worth to mention that in some implementations, arrays of ultrasonic and/or infrared sensors are used. This will lead to a higher cost, but in this comparison just one or two sensors working together is considered.

Resolution is here defined as the smallest object or feature that can be detected. A high resolution mean that smaller objects then a sensor with low resolution can be detected. Ultrasonic sensors mainly have a low resolution because of the physics of sound, that need a bigger surface to reflect. For the vision-based systems the resolution highly depends on the image quality and the recognition algorithms.

Regarding the two presented accuracies, those displays the minimal increment in range or angle that can be detected. For some of the methods this also depends on the turning motor accuracy and encoder accuracy when scanning. A high accuracy is when the technique can measure small increments. Ultrasonic sensors low angular resolution is mostly due to the wide beam angle. For all except the vision-based technique, the accuracy, mainly the range accuracy, also depends on the environmental conditions as humidity and temperature. A light or sound wave have different propagation speeds in different conditions, which affects the sensor model used to calculate the distance.

Complexity is considered here as how complex the algorithms needs to be in order to handle the data and output a range measurement. It also reflects how complex the hardware that shall handle the signal has to be. For a laser range scanner the hardware need to be able to measure the TOF, which require a high processing speed. This also applies to Infrared scanning, but most Infrared scanners uses triangulation, which do not require as high processing speed and this technique is therefore given a low grade. Most laser range scanners also uses mirrors to direct and read the laser beam, which need to be carefully calibrated and is fragile to large impacts. A larger impact may result in an unaligned mirror, which lower the quality of the readings. The laser range finder still gets a medium grade because the majority of laser range finders comes pre-calibrated. The vision-based algorithms need to handle feature recognition, image comparing and distance estimation algorithms which is much more complex and can be done in different ways. These algorithms often need to be designed by the user himself, when a laser range finder and other sensors usually has all algorithms ready made on the chip and simply output the range.

Updating frequency is the the number of outputted range measurements per second or can also be seen as number of completed scans per second. One scan is seen as all measurements needed to cover a certain area. For the three beam propagation techniques the grade mainly depends on the propagation speed of the beam, where the two light based techniques is getting high grade mainly because of the high speed of light. The vision-based techniques updating frequency depends on the processing algorithms and the processing power in the circuits. Because not all algorithms are that optimized this technique is given a medium grade.

As it sounds, range is the maximum distance that can be measured with each technique. For the vision-based system this mainly depends on the image quality and optics, so that features can be recognized. The attenuation of sound waves is the main reason that ultrasonic sensors has a limited range. For the two light based techniques the beam energy and focusing is the main reasons for the difference. Where Infrared scanners has a low energy and focusing, which means that it can just measure short distances.

The size and weight item refers to the size and weight of the sensor module. For small consumer products as vacuum cleaners, a big laser range scanner will not fit. As mentioned earlier in the section, Konolige et al.[76] proposed a laser range finder that is used by the Neato vacuum cleaners[1]. So it is possible to design a small and also low cost laser range finder for consumer products. However, this one is not an off the shelf product and the majority of the laser range finders are bulky. The smaller laser range finder also has lower quality when it comes to range, range accuracy and angular accuracy.

The surface problem item is about which type of surfaces the technique struggles with. Ultrasonic sensors will have difficulties to measure correct distances if the beam incidence angle, as an angled wall, is to big. Then the surface will act as a mirror. The other techniques will miss mirrors and glass surfaces because the light travel past or is reflected and “ghost” images is collected.

Because of the high intensity in the laser beam it will be affected less than the infrared beam by ambient light. Vision-based systems however, are affected by ambient light when evaluating ranges. In a dark scene, features will be more difficult to identify. This leads to fault identifications and fault measurements. The use of an infrared camera can be applied, but will raise the cost for the system. Because of the use of sound waves ultrasonic sensors is not affected by the ambient light. However, they are affected by other sound sources in the environment (external crosstalk) and by other sensors in the same system (internal crosstalk). The environmental noise is difficult to avoid and has to be considered as a random variable in the senor model, but the internal crosstalk can be solved more easily by beam firing strategies or modulated signals.

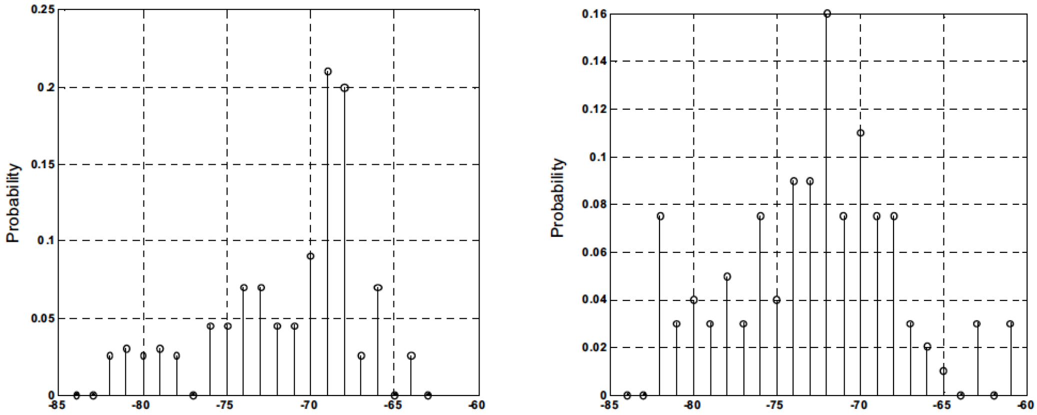
[1] Neato XV series webpage: http://www.neatorobotics.com/series/xv/

2.4 Principles of Positioning

As mentioned in the introduction, chapter 1, indoor positioning system, IPS, is a rapidly growing research field. Outdoor positioning using GPS has already been a big hit. To extend that success to also cover indoor navigation and let you find that store you are looking for, find you friend in a big office or for autonomous forklifts to navigate in a warehouse some other methods needs to be used. That's because the no line of sight the GPS suffers from when trying to positioning indoors.

The winning concept would be to get as accurate positioning as possible with existing infrastructure or at least to lower the need of adding extra hardware. An interesting way to achieve this is to use WLAN, because of the already existing infrastructures in buildings and the already implemented WLAN clients in every smartphone. WLAN as well as other methods usually utilizes three main measurement techniques to calculate the position of a client. These are Angle Of Arrival, AOA, distance related measurements and Received Signal Strength, RSS, profiling techniques[99]. The measurement techniques are later used in different methods like triangulation and trilateration.

The methods are dependent on accurate signal models that will say how the signal propagates in the environment. However, in indoor environments walls, ceilings, furniture and humans are large elements of error. A RF signal will attenuate and reflections will create multipath signals to or from the client. It is therefore hard to make an accurate signal model, especially if the environment is dynamic, like an office environment. In Figure 2.12 Chang, Rashidzadeh and Ahmadi[100] has measured the received signal strength, using WLAN, at a fixed position during different times of the day in an indoor environment. As can be seen, the signal strength for the different frequencies varies at the two different times. To minimize the error, probabilistic functions can be used that take account of the error, by example model the error as a Gaussian distribution.



(a) Signal Strength (dbm) at 9.00 a.m.

(b) Signal Strength (dbm) at 1.00 p.m.

Figure 2.12: The variation in signal strength on a fixed position in an indoor environment measured by Chang, Rashidzadeh and Ahmadi[100].

2.4.1 Angle of Arrival

This measurement technique calculate the angle of the received signal, either at the client side or at the access point, AP, side. The AP is a node in the network where the position is known and the client is the node where the position shall be calculated. The angle can later be used to triangulate the position of the client.

To calculate the angle there are mainly two classes; the first uses the receiver antenna's amplitude response and the second uses the receiver antenna's phase response[99]. Mao et al. [99] talks about how to use the beam forming, the anisotropy in the reception pattern, of an antenna to measure the angle of arrival of a signal. A receiver antenna can be rotated mechanically or electrically to identify the maximal signal strength of the transmitter. Because of the typical beam pattern, seen in Figure 2.13, the maximal signal strength will then correspond to the transmitter direction. Parameters that are good to be aware of are the sensitivity of the receiver and the beam width. Better sensitivity will allow a better detection of the absolute maximum of the beam and a smaller beamwidth will minimize the error of direction.

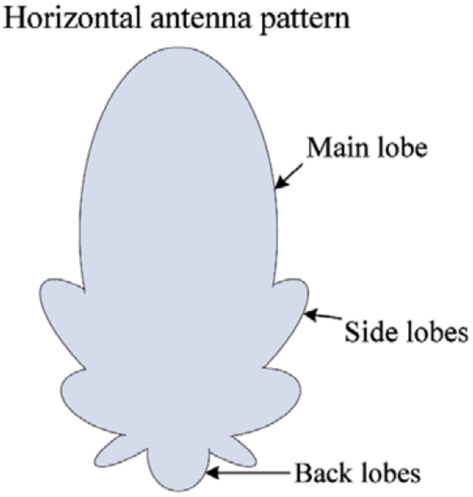


Figure 2.13: An illustration of the horizontal antenna pattern of a typical anisotropic antenna[99].

If the transmitted signal has varying signal strength the receiver will have difficult to differentiate the signal strength caused by the anisotropy beam pattern and the varying strength transmitted by the transmitter. Mao et al. [99] propose a solution that uses another non rotating omnidirectional antenna. The signal strength measured by this antenna is used to normalize the measured signal strength by the rotating antenna. To get rid of the need for an rotating antenna Mao et al.[99] summarize an approach by Koks[101]. This approach uses a minimum of two, but typically at least four, antennas with known anisotropic beam pattern that are stationary, that will say the antennas do not need to rotate. By overlapping the patterns from the different antennas and comparing each antennas received signal strength at the same time, the direction of the transmitter can be calculated. The approach do not suffer from the difficulties that arise when the transmitted signal has a varying signal strength.

Using the receiver antennas phase differences of the arrived transmitted signal, known as phase interferometry, can also derive the AOA. Mao et al.[99] mention that the technique typically requires a large receiver antenna or an antenna array, see Figure 2.14. The bearing of the transmitter, θ, can be obtained by the phase difference,

$$2\pi\frac{d\cos\theta}{\lambda} \tag{2.4.1}$$

between adjacent antenna elements. When one knows the bearing the distance between the transmitter and the ith antenna element can be calculated by,

$$R_i \approx R_0 - id\cos\theta, \tag{2.4.2}$$

where R_0 is the distance between the transmitter and the 0th antenna element[99].

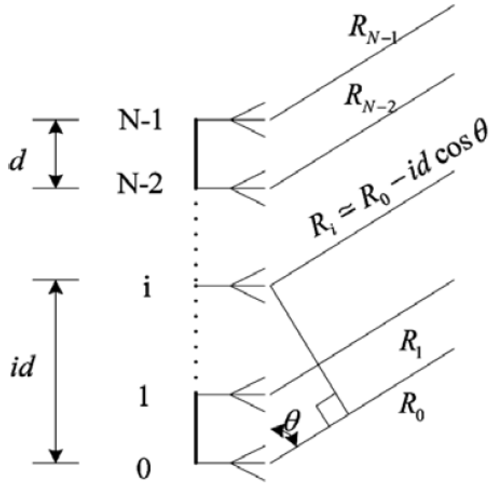


Figure 2.14: An illustration of an antenna array with N numbers of antenna elements separated with the uniform distance d[99].

2.4.2 Distance related measurements

These measurement techniques measure the distance between the client and the AP and use the information to calculate the position of the client by trilateration. Techniques included are for example one-way propagation time measurements, known as time of arrival, TOA, round trip propagation time measurements, time difference of arrival, TDOA, measurements and RSS measurements[99].

Time of arrival calculates the distance with the knowledge about the signal propagation velocity. By measuring the differences between the sending time from the transmitter and the receiving time at the receiver, the distance can easily be calculated. However, this assumes that the transmitter and receiver clocks are exactly synchronized. A small difference can cause big errors because of the high propagation velocity for a RF and light signal. The synchronization requirement may result in high cost sensors and a more complex system[99]. To minimize the need for exact time synchronization the round trip propagation time measurement can be used. This technique calculates the distance by measure the difference from when a signal is sent at a node and the time when the signal is received at the same node. The solution doesn't need the internal clock in a node to be synchronized with an external clock in another node. However, the technique requires that the measurement take account for the delay caused by the handling of the signal at the second node[99].

Time difference of arrival uses one transmitter and a number of receivers with known positions. The position of the transmitter is calculated by taking the time difference from when the same transmitted signal arrives at the different receivers. The TDOA between two receivers i and j is given by,

$$\Delta t_{ij} \triangleq t_i - t_j = \frac{1}{c}(\|r_i - r_t\| - \|r_j - r_t\|), \tag{2.4.3}$$

where t_i and t_j are the time when a signal is received by receiver i and j respectively, c is the propagation velocity and $\|\cdot\|$ denotes the Euclidean norm between the receiver positions $\mathbf{r}_i$, $\mathbf{r}_j$ and the transmitter position $\mathbf{r}_t$[99].
According to Mao et al.[99], the most widely used method to measure the TDOA at two receivers is the generalized cross-correlation method,

$$\rho_{i,j}(\tau) = \frac{1}{T}\int_0^T s_i(t)s_j(t-\tau)\mathrm{d}t, \tag{2.4.4}$$

where s_i and s_j are the signals received at receiver i and j respectively and is used to calculate the cross-correlation function between these signals by integrating the lag product for a sufficiently long time period T. This approach requires that the receivers internal clocks are very accurate synchronized, but do not need the transmitter, the client, to have a synchronized clock. Better measurement accuracy is obtained if the separation between receivers is increased, because of the increased time difference of arrival at the different receivers.

A feature that is found in most wireless devices is the received signal strength indicator, RSSI. By the received signal strength, RSS, measurement obtained from this feature the distance between the transmitter and the receiver can be estimated[99]. This technique is therefore very attractive, because of the no need of extra hardware. However, the most basic function that relates the received power and the distance between two nodes uses a free-space model. This is not the typical case, especially not in indoor environments where walls, ceilings and floor create reflections, diffractions and scattering of the signals. Mao et al.[99] discuss how this problem can be taken care of by model the RSS at any distance as a random and log-normally distributed random variable with a distance-dependent mean value.

2.4.3 RSS profiling measurements

While the aforementioned techniques calculates the position with limited configuration needed, that will say they do not need any pre-defined map just the AP positions, this technique make use of an pre-defined map that consists of sample points taken in an offline phase or in the online phase by sniffing devices[99]. At each sample point a vector of received signal strengths from the different Aps are obtained, called the fingerprint. Together, the fingerprints create an RSS model. This model is highly dependent on the environment and the AP positions. So for every position, for example a floor in an office building, a RSS model needs to be created. The client later uses this model to estimate its position by comparing its measured RSS values with the model, often using probabilistic functions.

2.5 Indoor Positioning Systems - State of the Art

This section will present how researchers have used different technologies to implement indoor positioning and what accuracy they could achieve. The section will present some Radio Frequency implementations more in depth, because these implementations are of high interest for the implementation part in this master thesis. However, other implementations and their accuracy will be presented shortly.

2.5.1 WLAN

Today WLAN networks are widely spread and are found in almost every public and private building. Most mobile devices also contain a WLAN client. To implement a positioning technique in a WLAN network would therefore be very cost effective. The WLAN network uses the unlicensed 2.4 GHz and the 5.0 GHz frequency band, which can cause some problems. Because of that the frequency bands are unlicensed other products uses this band as well, like microwave ovens and Bluetooth devices. Except other issues in indoor environments like multipath signals, this will cause interference problems with these other devices that are commonly used. Golmie and Mouveaux[102] discusses the interference problem in the 2.4 GHz band.

Different researchers propose different solutions to the implementation problem and how the different difficulties can be taken care of. Most of them suggest the use of distance measurements using RSS values or the use of RSS fingerprints. This is because the RSSI function is already built in and no extra hardware is needed.

One of the very first implementation of indoor positioning utilizing the WLAN technique was RADAR, developed by Bahl and Padmanabhan[103] in 2000 at Microsoft Research. The system uses signal strength, RSSI, which was gathered at multiple receiver positions. With this information the system could estimate the position of the client by both RSS profiling and a theoretically computed signal strength propagation model. Their empirical experiment was done on a 980 m^2 floor in an office building using three base stations, APs. During an offline phase they recorded signal strengths at 70 positions and at four different directions. The direction was recorded to see the impact off a human blocking the signals when holding the client. The recorded RSS measurements were saved centrally in a database where the unknown position was estimated by searching in this “look-up table”. Nearest neighbor(s) in signal space, NNSS, was the search method used[103]. That is, they calculated the Euclidean distance[1] between the set of recorded RSS measurements and the RSS at the unknown position. The position was then obtained by minimizing the Euclidean distance.

By using one of the recorded positions as an unknown position, could Bahl and Padmanabhan try their technique by simulations. That means that they estimate the position of a stationed client and not a client in movement. The 50th percentile value of the error distance was 2.94 meters. They further developed their method to, instead of just looking at the nearest neighbor, compare the *k* nearest neighbors. With five nearest neighbors they got an 50th percentile error distance value of 2.75 meters, 9% better than with just one neighbor. An other finding done by the researchers was that if the measurements done in the different directions at a position was merged to one value and used the four nearest neighbors the 50th percentile value of the error distance was 2.13m, 28% better than the first test. They also investigated what impact the number of recorded positions in the database has. In Figure 2.15 the results can be seen and

[1] Any other distance can be used, like the Manhattan distance[103]

it is clear that to many data points will not give any better accuracy, there is a little benefit in collecting data at positions closer than a certain threshold.

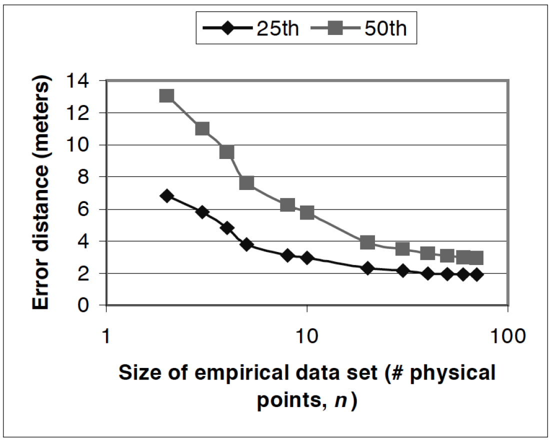


Figure 2.15: The error distance versus the number of recorded positions, on a log scale[103].

Bahl and Padmanabhan also tried to estimate the position using a signal propagation model. This was done in order to compare the empirical profiling method, that consumes a lot of time when the map was configured, with an algebraic profiling method, where the sample points were calculated instead of measured. Because of the indoor environment the researchers needed to take the attenuation in to account. This was done using the Wall Attenuation Factor (WAF) model[103],

$$P(d)[dBm] = P(d_0)[dBm] - 10n\log\left(\frac{d}{d_0}\right) - \begin{cases} nW \cdot WAF & nW < C \\ C \cdot WAF & nW \geq C \end{cases} \tag{2.5.1}$$

where n is the rate at which the path loss increases with distance, $P(d_0)$ is the signal power at some reference distance d_0 and d is the distance between the transmitter and receiver. nW is the number of walls between the transmitter and receiver, C is the maximum number of walls which makes a difference to the attenuation factor and WAF is the wall attenuation factor which was obtained empirically by the researchers. The algebraic method had an accuracy of 4.3 meters at the 50$^{\text{th}}$ percentile which can be compared to the empirical measured data sets accuracy of 2.94 meters.

The findings done by the researchers can be summarized with that empirical acquired data points will give a better resolution. However, the propagation model will be more cost-effective because of the no need of empirical measurements. The researchers also state that the propagation model can be relocated in a different part of the building, which is not the case for the empirical one.

Curran et al.[104] have conducted some experiments in order to compare different existing indoor positioning systems. Two of the tested systems are WLAN based, namely LA-200 and Ekahau.

The LA-200 system makes use of the RSS profiling technique or fingerprinting. A central unit stores all the fingerprints to be able to compare the received RSS at each AP in the online phase. The system software can then be used to map the fingerprints to floor plans in order to get a graphical user interface, GUI, when locating a client in the online phase. The system were tested during a period of nine months and could in average locate clients at room level 70% of the time with an average accuracy of 25m[104].

EPE is the main software used by the Ekahau system to track the position of a client. This software uses probabilistic algorithms to calculate the position, but still need some pre-calibration or fingerprinting stored in the system component called ESS. This component gather calibration information when a person carrying the system around and save the RSSI information from each AP for that area[104]. By combining this data and user defined "paths", that will say paths it's likely that people will walk, the system can estimate a clients position with an accuracy of 1-3m[104] with in 5 seconds. Worth noting is that the Ekahau also need software to run on the client in order to identify this.

An implementation that tries to minimize the effect of the environment, multipath signals and attenuation, was developed by Chang et al.[100]. By using differential APs the RSS variation at the client can be attenuated because noise and interference are rejected as common mode signals.

To explain how this works, take two identical APs with distances d_1 and d_2 to the client. Then according to Friis' free-space model[105], with some modification to take the variation in the indoor environment into account, the signal power received from the two APs are,

$$\begin{aligned} P_{r1} = P_{t1} + G_{t1} + G_r + 20\log\frac{\lambda}{4\pi} - 10n\log d_1 - X_{a1} \\ P_{r2} = P_{t2} + G_{t2} + G_r + 20\log\frac{\lambda}{4\pi} - 10n\log d_2 - X_{a2} \end{aligned}, \tag{2.5.2}$$

where P_{t1} and P_{t2} are the transmitted power from the two APs respectively. G_{t1}, G_{t2} and G_r are the antenna gain at the two APs and at the receiver. λ is the wavelength, n is an environment dependent variable that is two for free space and higher for indoor spaces and X_{a1} and X_{a2} are the normal random variable with zero mean in dB that's representing the shadowing and fading effects. By knowing that the two APs are identically and placed at short distance from each other one can say that $G_{t1} = G_{t2}$, $P_{t1} = P_{t2}$. If the placement of the APs also is so the building blocks, like walls, between them and the client remains equal, the difference between X_{a1} and X_{a2} are minimized. The received differentiated signal can then be estimated by,

$$P_{r1} - P_{r2} = 10n\log\frac{d_2}{d_1} + \varepsilon, \tag{2.5.3}$$

where $\varepsilon = X_{a1} - X_{a2}$. It can be observed that the effect of the antenna orientation is eliminated and the effect of the signal attenuation is reduced. To minimize ε even more, directional antennas have to be used to ensure symmetrical signal paths from the APs to the client[100].

The differentiated method uses RSS profiling measurements where fingerprints are measured on a grid, Chang et al.[100] uses the term Radio-Map for their fingerprints. To adapt the method to the signal strength fluctuation caused by the indoor environment they use the probabilistic search method maximum likelihood (ML). The RSS values are treated as random variables, which is statistically dependent on the position, by the ML algorithm. The probability density function (PDF) at position r_i is given by,

$$\mathrm{PDF}(r_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(r_i - E_i)^2}{2\sigma_i^2}} \tag{2.5.4}$$

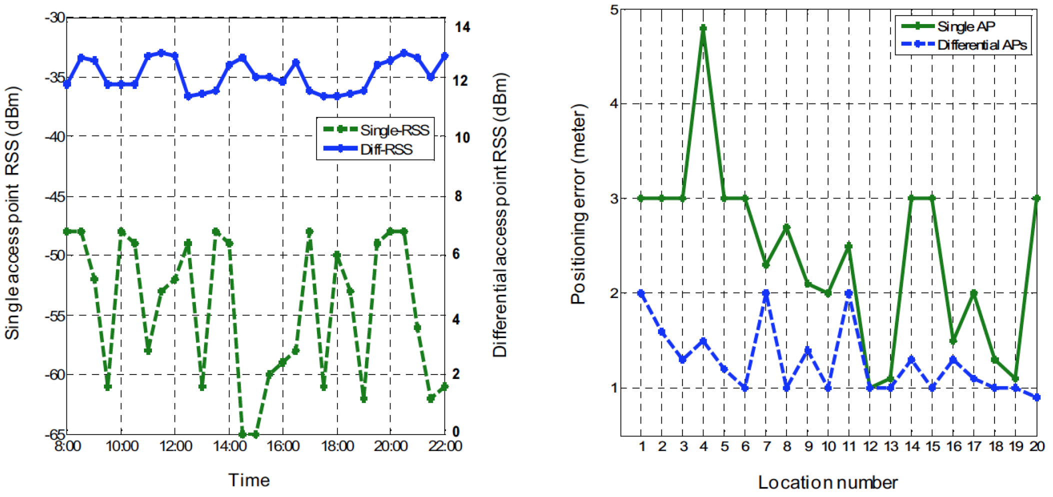
where E_i and σ_i are the mean and standard deviation of the RSS respectively, if the distribution of the RSS is represented by an Gaussian distribution. The PDF variables are calculated for the

APs during the offline training phase and are associated with the grid position in the fingerprint. During the online phase all RSS from the APs and the differentiated RSS are measured. These are then used in the ML algorithm. First the PDF of each measured RSS value is calculated at each specific grid point in the Radio-Map. Then, because the APs can be seen as independent transmitters when they operates at different frequency channels [100], the joint probability can be calculated as,

$$P(S) = P(Diff_{RSS}) \cdot P(S_1) \cdot \ldots \cdot P(S_k) \tag{2.5.5}$$

where $P(S_k)$ is the PDF calculation for the k-th AP at a certain grid point. The position is then estimated by picking the grid point in the Radio-Map where the joint probability is maximized.

Chang et al. [100] used MATLAB and Simulink to perform some simulations on their method as well as some empirical studies. The simulation showed that for a fixed distance the fluctuations with the differentiated AP was less than for a single AP, see Figure 2.16(a). For the empirical studies they used two APs configured as a differential AP and constructed a radio-map with 90 grid points and ten test points. Their test shows that the positioning error of the differentiated method and the single ended method is the same at some times during the day but differs a lot during other times when the environment is crowded with people[100], see Figure 2.16(b). Chang et al.[100] also found that their approach performed better than the existing RADAR[103] method by comparing them, for example at the error distance of 3 meters the accuracy of the differentiated system was 90.6 percent when the RADAR system had an accuracy of 76.4 percent.



(a) Signal strength at a fixed point in indoor environment. The left and right y axis show the strength of received signal from a single access point and the strength of received signal strength from a differential access point[100]

(b) position estimation error for 20 consecutive grids using single-ended and differential access points[100].

Figure 2.16: A comparison between positioning using a single AP and using a differentiated AP.

Drawbacks with RSS profiling methods are that they need a time-consuming offline training phase to create a pre-defined map. This map is also not adaptable to a changing environment. That is, if a new wall is built or a new door is placed the pre-defined map often has to be updated. Lim et al.[106] propose what they call a zero-configuration robust indoor localisation method using WLAN. They base their method on the Lipschitz embedding space, which is a coordinate space where each axis corresponds to a set of reference objects, for example the APs. Coordinates to an object, the client, are the distances between the object and the reference objects[106].

In order to take the difficult and dynamic indoor environment into account Lim et al.[106] propose the use of an signal-distance map (SDM). This is not a map that is pre-measured, but a map that is online calculated and updated. The map defines the relationship between measured RSS values and distance. If there are m APs the Lipschitz embedding space will be m-dimensional and the coordinate for a node i in this space is representing by the signal vector,

$$\mathbf{s}_i = [s_{i1}, \dots, s_{im}]^T$$

where s_{ij} is the RSS at the i-th node from the transmitted j-th AP. As can be seen the distance in the Lipschitz embedding space used by Lim et al. is the signal strength. By putting all Lipschitz coordinates together for all APs an m times m matrix will represent the overall embedding space for the RSS measures between the APs,

$$\mathbf{s}_i = [\mathbf{s}_1, \dots, \mathbf{s}_m]. \tag{2.5.6}$$

The geographic distance between AP i and j are d_{ij} and for one AP the geographic distance vector can be defined as,

$$\mathbf{d}_i = [d_{i1}, \dots, d_{im}]^T.$$

As for the Lipschitz coordinates the geographic distances for every AP can be combined to one m times m matrix $\mathbf{D}$ which has zero diagonal entries.

As the SDM Lim et al.[106] uses the m times m optimal linear transformation matrix $\mathbf{T}$ to transform the measured RSS at the client from the APs to estimated distances. After some algebraic operations, Lim et al.[106] express the SDM as,

$$\mathbf{T} = \log(\mathbf{D})\mathbf{S}^T(\mathbf{S}\mathbf{S}^T)^{-1}. \tag{2.5.7}$$

This yields that the SDM can be estimated online at all times by knowing the geographic distances between all APs and that each AP measures the signal strength to its neighboring APs. The geographical distance between the client n and the neighboring AP can be estimated by,

$$\mathbf{d}_n = e^{\mathbf{T}\mathbf{s}_n} \tag{2.5.8}$$

and thanks to the online calculation of T the estimated distances will take the dynamic environment into account. The SDM can be further improved to the measurement noises, please see [106] how Lim et al. do this. The estimated geographical distances can later be used to estimate a position $\bar{\mathbf{x}}$ for the client by trilateration algorithms. The algorithm used by Lim et al. is an iterative one,

$$\bar{\mathbf{x}}[k+1] = \bar{\mathbf{x}}[k] + \alpha \sum_{i=1}^{m} \left(1 - \frac{\bar{d}_i}{f_d(\bar{\mathbf{x}}, \mathbf{x}_i)}\right)(\bar{\mathbf{x}}[k] - \mathbf{x}_i) \tag{2.5.9}$$

where $\bar{d}_i$ is the estimated distance to the i-th AP computed by SDM and α is a constant step size that Lim et al. has set to 0.1. The initial estimate for the iterative function, $\bar{\mathbf{x}}[0]$ is set to the position of the AP which has the smallest estimated distance to the client.

The empirical experiment was conducted using four to six APs that were connected to a server. Then three different tests were performed, first by using one of the deployed APs as the client to position, second to use a personal digital assistant (PDA) at fixed positions as the client and at last to carry the PDA around a path. The client also needed to be connected to the server so the SDM can be estimated and used to estimate the geographical distances online. All three tests were also performed during different times of the day to see the impact of the dynamic indoor environment, but Lim et al.[106] just present results recorded during the afternoon when most people are in movement. They showed that more APs would give a better accuracy for the SDM estimates, 2.32 meters and 2.76 meters of median error with six APs and five APs respectively. It was also seen that the placement of the APs affected the accuracy of the positioning. At positions surrounded by APs the median accuracy with just five APs was 1.92 meters which was smaller than with the use of six APs and not take the positioning into account. For a moving client the SDM system was able to track and position with an accuracy of ±3 meters.

A RSS profiling positioning method can be computational heavy dependent on the search and matching method used to compare a measured RSS value online to a fingerprinted value. Chiou et al.[107] has therefore proposed a client-based method, the client estimates its own position, that combine a RSS profiling method with a signal propagation method. The method is also develop further to implement a Kalman-Filter (KF), see section 2.1.1, in order to get an even better accuracy. This approach will combine data from different observers to filter variations in the RSS data.

The method developed by Chiou et al.[107] will, as the other mentioned WLAN based methods, make use of the built in RSSI functionality. Then no extra infrastructure or hardware will be needed and the solution is cost effective. First is a fingerprinting done, the offline phase. At this point the signal-to-noise (SNR) ratio and the noise level is associated with different known positions. The easiest way to estimate a position would be to minimize the Euclidean distance in signal space between the measured SNR and the calibrated SNR fingerprints (CSNR),

$$\sum_{i=1}^{n_{AP}} \left(\mathrm{SNR}_i^{on} - \mathrm{CSNR}_i^{off}\right)^2 \tag{2.5.10}$$

The calibration is done in the initialization of the online positioning to take the dynamic environment into account and is given by,

$$\mathrm{CSNR}_i^{off} = \mathrm{SNR}_i^{off} + (N_i^{off}/N_i^{on}) \tag{2.5.11}$$

where N^{off} and N^{on} is the noise measured during fingerprinting and online respectively and i denotes AP i. If the number of fingerprints is high, this will require lot of computation power. Instead a radio propagation model (RPM) is proposed. This model relates the distance between an AP and the client with the measured SNR. Also, the dynamics in the environment is taken into account by an initiating calibration of the model. The calibrated model can be expressed by,

$$\mathrm{SNR}^{on} = f(d) + c + (N^{off}/N^{on}) \tag{2.5.12}$$

where $f(d)$ is the function that relates the distance between the AP and the client and c is the unknown error in the actual relation. Chiou et al.[107] estimate the RPM by a best polynomial fit of degree two to the calibrated SNR fingerprints.

To improve the accuracy an adaptive Kalman filter is implemented. This filter combines a model based on tracking of the previous visited positions and the current measured position. The outcome of the filter should be a more accurate estimate of the client position. In section 2.1.1 a more in-depth description of the Kalman filter can be found and the model used by Chiou et al.[107] can be found in their paper.

In the signal space different positions may be represented similarly. By adding RFID, to the method Chiou et al.[107] tries to minimize this impact to the accuracy. The client will then be equipped with an RFID reader and RFID tags are placed at known positions in the environment. At the initial phase of the online positioning the client will sense which tag is closest and be able to use this as an initial position and therefore chose a cluster of fingerprints that will lower the computational power needed.

Four APs and six RFID tags were used during the empirical study. Each AP was placed in the corners of a rectangular hallway with the dimensions 13 meters times 34 meters. The hallway was located at a floor with offices and with the dimension 21 meters times 50 meters. However, the experiment was only done in the hallway. The RFID tags were deployed in each corner to lowering the cornereffect caused by the KF method[107] and in the middle of the 34 meters parts of the hallway where fingerprints can be close together and hard to pick during the online phase[107]. For the calibrating process 12 PDAs were used that measured the SNR and reported this back to the client. During the offline phase Chiou et al.[107] used a rotary table where they placed the receiver antenna. The SNR value was then measured in four directions and an average of these measures was saved in the database. This technique shall give a more accurate measurement[107]. During the online phase they used some of their fingerprint positions as the test positions, but measured the SNR in real-time to compare the accuracy after the environment has changed its dynamics, furniture has changed, doors are opened and the temperature and humidity is different. The accuracy of the method improved when more APs were deployed. However, a saturation at three APs could be seen, the error distance between using four and three APs was not as great as the difference of using three instead of two. More fingerprint positions will also increase the accuracy, but will lead to a higher computational cost. By using their RFID cluster-selecting algorithm the efficiency can be maintained, more data points can be used and just a few is selected based on the RFID positioning which lowering the computational cost. The 65-percentile error distance by only using fingerprints was 1.67 meters and by using the RPM was 1.89 meters when calibration was used. If no online calibration was used the 65-percentile error distance was 2.14 meter and 2.54 meters for the fingerprint based method and the RPM method respectively. By introducing a Kalman filter the impact on the 65-percentile error distance was not that great, 1.65 meters and 1.59 meters for the fingerprint+Kalman filter method and the RPM+Kalman filter method respectively. However, for the 95-percentile error distance the difference was higher, 3.48 meters versus 5.51 meters for the fingerprint method with and without Kalman filter respectively and 2.5 meters versus 4.74 meters for the RPM method with and without Kalman filter respectively. Chiou et al.[107] explains the phenomenon that the error is higher in the lower percentile for the Kalman filter by the use of the fingerprint positions as the test positions. With the aid of the RFID method the accuracy was further improved by one meter in the 95-percentile error for the fingerbased+Kalman filter method.

Two other examples that combines data from different methods to get a better accuracy was developed by Bejuri et al.[108] and Al Rifai[109]. The first method combines WLAN with an image recognition algorithm. The client takes a picture of the position, maybe a hallway, and also recording the position by a WLAN positioning method. Then a server gets the image and position. The server has pre-trained images to compare the received image from the client. In order to not compare all images the server uses the WLAN position to narrow the images to search from. Bejuri et al.[108] do not present any indepth explanation about their WLAN positioning method or the accuracy they got, because the paper is a pre-study. However, the method is interesting and can have potential to be used in a shopping mall. The user takes a picture of a point of interest and can be located and get information about that point, like a store.

In a master thesis at the Royal Institute of Technology Al Rifai[109] compared the performance of different configurations of an indoor positioning system that should be used at Arlanda airport, Stockholm. In the simulations he tested performance between different amounts of WLAN APs, if adding Bluetooth transmitters would help and the differences between trilateration and fingerprinting in a area of 200 times 200 meters. As expected, the accuracy was better with more WLAN APs, however the 90-percentile error distance was almost the same when using 6, 13 and 17 APs, 33 meters when the trilateration method was used. By adding Bluetooth transmitters and only using four WLAN APs the accuracy could be improved, the 90-percentile accuracy was 8.24 meters when adding 210 Bluetooth transmitters. As others have seen, the number of fingerprints used in the fingerprint method has a big impact on the result. The 90-percentile accuracy was 18.2 meters when using six WLAN APs and 63 fingerprints and 8.8 meters when 325 fingerprints were used. However, the result when using just 63 fingerprints was better than when using the trilateration method. As for the trilateration method the accuracy increased when introducing the Bluetooth transmitters.

2.5.2 RFID

Radio Frequency Identification (RFID) has in recent years been used more and more. Today the technology can be found in areas as Elecronic Article Surveillance (EAS), used in stores to find out if someone hasn't paid for their article, and building access cards. As the name indicates the technology uses radio frequencies for communication. Common operating frequencies are in the range from 60kHz to 5.8 GHz[110]. Basically there exist three different types of tags, which are usually the transmitter in the network, namely passive, semi-active or active tags. The passive tag has no power source and uses the power in the signal from the reader. It exists in both a simpler one where the reader just can sense its presence, like EAS, and more advanced ones that can store information, like public transport cards. The transmitter range for a passive tag is almost always very short. A semi-active tag has some kind of power source, but this is not used to transmit data. It's used to store data during longer time. Semi-active tags can be used in sensor applications that log data and that transmit the stored data to a reader just some times when the reader is in proximity of the semi-active tag. The active tag has a bigger power source and use the power source to transmit over longer distances. An active tag has a transmitter range of over hundred meters[110]. Dubendorf[110] explains RFID more in depth in chapter nine of Wireless Data Technologies.

Since RFID uses radio waves it suffers of the same problem as WLAN in the dynamic indoor environment. That is, diffractions, attenuations, multipaths etc. As for the WLAN, different researchers has tried to solve this in different ways. In this subsection some proposed methods will be presented.

Curran et al.[104] has tested a system called RFID-radar that according to the developers shall have an accuracy of 50 cm in a 100 meter deep area. They mention that this depends on the tags used and probably also the environment (author note). The reader module consists of three antennas, one for powering up the tags and two for receiving transmitted signals from the tag. By comparing the RSS at the two antennas the AOA can be calculated. Curren et al.[104] found that the average error distance for the system was 4.19 meter, not as accurate as the developer stated. They also found that the system needed 10-20s to determine the position, which is not good if an object is moving. The error distance for slowly moving clients were found to be 10 meters. An explanation Curran et al.[104] has to this is that the system need to have LOS to position with good accuracy, which is probably not the case in most indoor office environment.

By placing passive RFID tags in the roof and read more than one at the same time Lim and

Zhang[111] tries to position a client with good accuracy. A reader is carried on the client, like a forklift in a warehouse, and this reader sends information about tags that has been read to a server. This server compares the information with pre-trained fingerprints in the environment. The comparing algorithm is called "Intersection over Union" which calculates the similarities between each sample point and the observation. The sample point with the highest similarity is chosen as the position.

The empirical test was conducted in an area of size 4.2 times 8.4 meters and with 49 grid points. 176 RFID tags were used in 22 columns and 8 rows. With this setup and the proposed method an average error distance was within one meter.

Instead of placing the reader on the client and the tags in the roof, Reza and Geok[112] was placing the tag on the client and then created a grid of reader antennas. The study concluded by them proposed a systematically way to decide on the placement off the antennas as well as a proposed positioning system.

By their placement method an area of 15 times 15 meters should be covered by 64 reader antennas. To estimate the position a statistical average algorithm was used. If k antennas could read the presence of the tag the estimated position can be calculated by,

$$(x_{est}, y_{est}) = \left(\frac{R_{1x} + \ldots + R_{kx}}{k}, \frac{R_{1y} + \ldots + R_{ky}}{k} \right) \tag{2.5.13}$$

where R_{ix} and R_{iy} is the x and y coordinate of the i-th reader antenna.

With this method Reza and Geok[112] could estimate the position of a tag with an average error of 0.4 meters. Which is very good, but also require a lot of hardware, 64 reader antennas for a 15 times 15 meters area.

Another way of positioning with RFID could be to just register if a tag is entering or exiting a certain area. This method is not precise but could for example be used to track if a container has entered a building or not. This is what ABB in Finland used the technique to in 2005[1].

2.5.3 Bluetooth

As Bluetooth can be found in almost every smart phone today it is an interesting technology for indoor positioning. Since the introduction of Bluetooth 4.0 or Bluetooth Low Energy[113] the implementation of Bluetooth in other mobile devices and sensors are probably going to increase, Gomez et al.[113] expect the technology to be incorporated into billions of devices only in the next few years. A BLE module is cheap and the lifetime of a BLE module can be up to 14 years according to theoretical studies[113]. Because of this and as Bluetooth can be found in almost every smartphone today it is an interesting technology for indoor positioning.

Two or more Bluetooth modules are used to create a piconet. To positioning a client at least three modules are often needed, that will say one acting as the client and two as the APs. As with the WLAN, more APs will probably give higher accuracies. The Bluetooth piconet is a smaller local network and does often not have the same range as the WLAN network. However, it still suffers from the dynamic indoor environment because of the use of radio waves.

The Bluetooth protocol defines a list of signal based parameters that could be used as positioning parameters, like Link Quality, Transmitted Power Control and RSSI. Subhan et al.[114] found

[1] `http://www.abb.com/cawp/seitp202/a1c82bd78f74140ec1256ff2002851e6.aspx`

during empirical study that the best parameter for positioning would be RSSI, or as they called it RX-power level which is obtained from the RSSI value.

Subhan et al.[114] discuss how the RSSI and distance can be related with aid of empirical studies. They state that fingerprinting, or RSS profiling, is the most accurate way to position a client, but it's time consuming. Instead a signal propagation model is discussed to estimate the distance. As had been said earlier in this section, propagation models suffer from the inaccuracy that depends on the dynamic indoor environment, NLOS etc. By introducing a gradient filter[114] the average error could be minimized from 5.87 meters to 2.65 meters. As a future work Subhan et al.[114] propose that a trilateration method that uses the estimated distances by the filtered RSSI values and a fingerprinting method is combined to obtain higher accuracy.

When using Bluetooth and the RSSI parameter as positioning system there are two ways of get the RSSI value. Either by establish a connection between a client and AP or by using inquiry-based solution[115]. When a connection is established the Bluetooth standard will make the AP to continually adjust the transmission power, which is not good for a positioning method such as fingerprinting. Instead the inquiry-based solution is better, but slower. The updating frequency is less then one hertz, which can make fingerprinting very time consuming if a lot of samples are needed. Pei et al.[115] therefore propose the use of a Weibull function to estimates the probability distribution function at every reference point with less samples.

An ordinary fingerprint database for a reference point, R_i, in the grid can look like,

$$\mathbf{R}_i = \begin{bmatrix} P(A_1 O_1 \,|R_i\,) & \cdots & P(A_k O_1 \,|R_i\,) \\ \vdots & \ddots & \vdots \\ P(A_1 O_v \,|R_i\,) & \cdots & P(A_k O_v \,|R_i\,) \end{bmatrix} \tag{2.5.14}$$

where A_i is the i-th AP of k APs, O_j is the j-th RSSI measurement from the i-th AP to the references point. Depending on the amount of APs and the variations of the RSSI value, this matrix can be huge. Instead the RSSI values can be divided in different ranges or bins, like values from -55 to -60 dBm is one bin. The matrix at every reference point can therefore be minimized. However, if the measurements shall give a good probability distribution for every reference point a lot of samples needs to be taken. By using the probability distribution function estimated by the Weibull-distribution,

$$f(x) = \begin{cases} \frac{c}{\lambda}\left(\frac{x-\theta}{\lambda}\right)^{c-1} e^{-\left(\frac{x-\theta}{\lambda}\right)^{c}} & x \geqslant \theta \\ 0 & x < \theta \end{cases} \tag{2.5.15}$$

less samples are needed[115]. x is the variable of the function, c is the shape parameter, λ is the scale parameter and θ is the shift parameter. By calculating these variables, see [115] for how, the only thing to save in the radio-map matrix are the reference point coordinates and these values for every AP which reduces the matrix size. At the online positioning phase the algorithm used by Pei et al.[115] was Bayesian Histogram Maximum Likelihood that could be simplified to,

$$\arg\max_p [P(p\,|\mathbf{O}\,)] = \arg\max_p [P(\mathbf{O}\,|p\,)] \tag{2.5.16}$$

where $\mathbf{O} = \{O_1, O_2, \ldots, O_k\}$ is the RSSI measurement vector from the APs to the client and p is the position of the client. The solution is therefore to find the maximum of,

$$P(\mathbf{O}\,|p\,) = \prod_{m=1}^{k} P(A_m B_j \,|R_i\,) = \prod_{m=1}^{k} F_m(x+w) - F_m(x)) \tag{2.5.17}$$

where B_j is the bin that the RSSI measurement from the m-th AP belongs to, x is the left edge of bin B_j and w is the width of the bin. The function $F(x)$ is called the cumulate distribution function[115] and is defined as,

$$F(x) = 1 - e^{-\left(\frac{x-\theta}{\lambda}\right)^c}. \qquad (2.5.18)$$

As an empirical study the authors compared the "real" probability distribution by during 20 hours measure over 11000 samples in a static position, called the benchmark. Then by randomly selecting just 20 of these, they created a Weibull-distribution. They also used this 20 samples to create a probability distribution with equation 2.5.14 to compare with. The result was very promising with a maximum difference of 0.0431 in probability for the 20 samples Weibull-based distribution compared to the benchmark. The ordinary one had a maximum difference of 0.2490 in probability.

Pei et al.[115] also tested an dynamic indoor positioning, that will say position the client in motion. They used their algorithms for both a three Bluetooth AP network and a eight WLAN AP network. The mean error was 5.1 meters for the Bluetooth and 2.2 meters for the WLAN system. However, the Bluetooth just used three APs, which can be one explanation to the difference. It is also promising that this method, using the Weibull-distribution, can be used for other methods like WLAN as well.

Instead of deploy extra infrastructure or use infrastructure in the building, Li et al.[116] propose the use of only mobile phones. The method uses some mobile phones that is positioned with GPS, these are called the beacon phones, and some where the position are unknown, these are called the blind phone. The problem is defined as an optimization problem that shall minimize the estimated distance between a phone and the phone it is linked to. Every link has a maximum value depending on the Bluetooth class that are used; class 1 has a range of 100 meters, class 2 has a range of 10 meters and class 3 has a range of 5 meters. This is used in the optimization problem setup as the "subject to" function as well as the knowledge that some phones are positioned by GPS. The authors propose that the problem is solved by a neural network, see [116] for how.

No average of the accuracy is presented by the authors, instead graphs over the real positions and the estimated positions are presented. With one of their neural network solutions it can be seen that the error distance is around 7 meters [116]. An interesting application of their solution would be to position something in tunnels where the cars in the opening of the tunnel are positioned by GPS and then used to position all other cars in that tunnel.

2.5.4 Other implementations

In this subsection other possible methods that has been used for IPS will be shortly presented together with the accuracy presented by the researchers.

Zhang and Kavehrad[117] present a method that uses visual light communication (VLC). That is, they make use of the light sources in a room. By modulate the light to blink in different frequencies it is possible to communicate with an optical receiver. One can think of fiber optic communication but with the use of for example a ceiling lamp instead. The upside of using VLC is the no need of radio waves in sensitive environments and it doesn't suffer from multipath to the same extent as RF methods[117]. The biggest downside is that a LOS is almost always needed. Zhang and Kavehrad[117] compares the accuracy that can be acquired by using different positioning measurement techniques. For example according to simulations it should be possible to positioning a client with an accuracy of 1.8mm by the use of TDOA or 0.5mm by the use of RSS propagation model. By experiments an accuracy of 4.6cm could be acquired by AOA and an accuracy of room level by the use of proximity, to sense the presence of a client.

Liu et al.[118] also makes use of the VLC. They developed a new receiver that should be able to sense the light in more angles to position with a better accuracy. With their sensor and algorithm they manage to position with an accuracy of less than 0.15 meters for all 20 measurements point when downlight was used.

Zhou et al.[119] developed an algorithm that uses VLC for positioning. They tested with different kind of noises to see how well their algorithm performed in different lightning environment. The study shows that the algorithm can provide an accuracy of at least five centimeters, even when the receiver is in direct sunlight.

Instead of visual light, infra red light (IR) can be used for positioning. Want et al.[120] created a positioning system based on this technique as early as 1992. Their system uses active badges that by IR sends out a unique ID every 15 seconds and is worn by the client. IR receivers placed in the building and connected to a central server then read the ID. Depending on the placement of the receivers the system can have different accuracies. One receiver in every room will give a room level accuracy but can maybe suffer from if the client is turned from the receiver. As for the VLC systems, IR systems also need a LOS from the transmitter to the receiver.

Bischoff et al.[121] propose the use of ultrasound together with an RF link to measure distance. The setup needed for this is a transmitter and a receiver where the transmitter transmits both an RF signal and an ultrasound wave. Because of the different velocities of these to waves, the receiver can use the TDOA between the signals. By this measurement and the knowledge of the propagation of the two signals the distance can be calculated. By an empirical test the maximum mean error was found to be 2.34 cm. This accuracy however assumes LOS for the ultrasound and RF waves.

Schweinzer and Kaniak[122] uses a similar method as Bischoff et al.[121], but instead of an RF signal starting the timing a pseudo-ranging is used. The system uses at least four transmitters and the firstly received signal starts the time measurement. Then the time-intervals of the received signals are used for trilateration. The distance to each transmitter is calculated assuming a well-defined time protocol of the transmissions to be used then an unknown offset can be used. The maximum deviation of the positioning found by the researchers by using this method was around one centimeter. However, this method also assumes a LOS in the evaluation and was conducted in an area of three times four meters.

Another RF technique that can be used for positioning is ultra-wide band. It is not that widely used in consumer products and operates from approximately 3.5 GHz to 9.5 GHz[123]. The technique will therefore not interfere with other wireless consumer devices, like smartphones. Other advantages are that it has fine time resolution and is energy efficient.

Zhou et al.[124] uses the UWB technology in order to position a client. The measurement technique is based on the TOA method. However, instead of the need of an exact synchronization and expensive hardware, they propose a solution based on the difference of the TOA from a trigger signal and the target signal, which is as the TDOA method but without the receiver synchronization[124]. The mean accuracy for the proposed system is approximately 10 cm.

In a master thesis Shiferaw Heyi[123] has used the UWB technology to calculate the distance from an AP to a client. The author uses the time of flight (TOF) to calculate the distance. For positioning a TDOA measurement technique is utilized. In a LOS condition the author could reach a 90-percentile accuracy of 25 cm for the distance measurement. However, this is not often the case so a NLOS test was also conducted with a 50-percentile accuracy of 50 cm.

For the positioning the accuracy was 1.1 meter. The author believes the accuracy can improve with a better synchronization algorithm to measure the TDOA.

By using a camera network and computer vision techniques it's possible to position objects in the environment. Losada et al.[125] mention that there exists mainly two techniques to identify the object to position (the client), either by visual tags/landmarks or by the shape of the object it self. They are using the second to identify the client and could gain an error lower than 300 millimeters when using two or more cameras. During their empirical test they saw that one camera wasn't enough to position the client as good as wanted, but with two cameras the positioning was acceptable and wasn't improved that much by adding a third camera. For a description of their algorithm and setup to identify and calculate the 3D position of the client see [125]

An example of a study that used a landmark on the client was performed by Fernández et al.[126]. To solve problem that can occur in different lights they used infrared LEDs in a T-shape to also be able to determine the orientation of the robot. With the use of four cameras they were able to estimate the position with an error less than 40 millimeters. To read more about the empirical setup and their algorithms see [126].

Both of the multi-camera network methods mentioned need a calibration phase before being able to position a client. This so variables can be estimated that are used to map reference systems in the picture with the camera reference system and also the global reference system. The calibration also calibrates perspective issues that can occur. The different methods have some difference in their calibration process, for example Fernández et al.[126] uses an active calibration pattern with infrared LEDs, which gives better accuracy in the variables than just using a passive one with black dots on a white background.

Instead of a camera network, one camera can also be mounted on the client to identify patterns. Bejuri et al.[108] propose an image database and an algorithm to compare taken pictures to this database in order to locate a person. In order to lower the computational power they propose that the method was combined with WLAN positioning. Bellot[127] propose in a master thesis the use of visual tags to identify a position. In a database ID numbers are stored together with position coordinates and each tag has its own unique ID number that is translated from the picture by an algorithm. Kim and Jun[128] used both the visual tag ID method and an image database to identify a specific image scene. They also developed a system that was able to identify the scene from a video device and not just still pictures in order to implement an augmented reality that in real-time showed turn by turn directions for a user. The accuracy of all these methods depends on the image database or number of tags to identify and is most likely on the room scale and not with in meter accuracy.

The largest drawback of all image based positioning systems are the need of LOS, the system needs to see the object or the tags. If the environment is not square or have an advanced room layout a lot of cameras can be needed in the camera network to cover the room or more tags can be needed, which are maybe not looking good.

2.5.5 Summary of indoor positioning systems

A comparison between different technologies relative to each other that can be used for indoor positioning is found in Table 2.11. The table shall be considered as an aid and not as an answer. The section will also talk about centralized systems versus client-based systems, as well as other items that are worth thinking of when an indoor positioning system shall be decided.

Table 2.11: Comparison of the different indoor positioning systems

	Technologies used in indoor positioning systems					
Item	**WLAN**	**RFID**	**Bluetooth**	**UWB**	**VLC/ IR**	**Vision-based**
Cost	Low	Low/ Medium	Low	Medium	Medium	High
Accuracy	Low	Low/ Medium	Low	Medium	High	High
Calibration/training	Low/ Medium	Medium	Low/ Medium	Medium	Medium	High
Complexity	Low/ Medium	Low/ Medium	Low/ Medium	Medium	High	High
Infrastructure	Low	Medium/ High	Medium	High	High	Medium
LOS needed	No	No	No	No	Yes	Yes
Client size & weight	Small	Small/ Medium	Small	Small/ Medium	Small/ Medium	N/A
Affected by dynamics	High	High	High	Medium	Medium	Low

As for most commercial implemented systems cost is the major consideration when choosing a system. The goal with a company is usually to increase the profit. A WLAN and Bluetooth module is very cheap and easy to get. The same goes for passive RFID tags, but active RFID tags are more costly. Vision-based systems require two or more cameras and in bigger areas and complex environment even more cameras are needed. This increase the cost for such a system. All technologies, except WLAN, are most likely not installed in a building as “default”, which can be seen at the table item “Infrastructure”. However, as mentioned Bluetooth modules are cheap and that give them a low grade.

The accuracy is defined so that high accuracy relates to a low euclidean distance error between estimated point and the true position of the point of interest. This result not only depends on the technology used, but also on the algorithms used. However, the RF based technologies are having a lower accuracy than the light based and vision-based technologies. This mostly depends on the nature of the RF signals, that in greater extent depends on the environmental conditions and is reflected and attenuated by walls, humans, furniture etc. With more receivers or tags RFID technology can however give a better accuracy, but that will lead to a higher cost of the system. UWB is also providing better accuracies then the other RF based technologies. This is mostly because of that it uses frequencies in bands that are not used by other consumer products. The frequency bands used by both Bluetooth and WLAN and some RFID tags are the same as for example microwave ovens, which will increase the noise in the measurements and lower accuracy.

A majority of the WLAN systems uses the RSSI values to estimate position. This is a function that in most WLAN modules is built in and therefore not require any calibration. However, one can choose if a propagation model or offline fingerprinting shall be used. If fingerprinting is used the calibration or training phase will be time consuming and WLAN is therefore given the grade low/medium. The same applies to Bluetooth. For both RFID, UWB and VLC/IR the position of the different tags, receivers and transmitters needs to be calibrated and chosen in a good way for the best result. For VLC/IR it is important to think of where the lights are placed because a LOS is needed for position. The vision-based systems however, need a very careful calibration process in order to be able to relate a captured image to a distance. The cameras also need to be calibrated together so they identify the same feature and work together.

As mentioned earlier both WLAN and Bluetooth modules has built in functions to get the RSSI value and is therefore very easy to get started with. All the technologies needs some kind of algorithms that are more or less complex. However, the vision-based method, as in range measurement techniques see Section 2.3, need to be able to identify features to compare in different images and to be used to calculate distances, this make them more complex than the other technologies that are used. VLC/IR uses light which place them in the same category as the laser range finders and infrared range finders see Section 2.3. To handle light information and read modulated light signals, fast processing speed and more complex hardware and algorithms is require. Regarding RFID and UWB, they do not often, like WLAN and Bluetooth, have built in function for RSSI measurements. Some RFID implementations also uses more advanced hardware, like antenna arrays that can measure angle of arrival, that are more complex than just using the RSSI value.

As has shortly been mentioned, all technologies except WLAN not usually is installed by default in new or old buildings. They therefore require new infrastructure to be installed. Some buildings have cameras for security monitoring, this together with some additional cameras may be used for vision-based positioning and this technology is therefore given a medium grade. The RFID grade is dependent on how many tags or receiver antennas that are needed for the proposed system. RFID tags or antennas can, but are not usually, found in smartphones and one can think that those could also be used for positioning. This also applies for Bluetooth, the grade depends on how many extra AP nodes that is needed. However, unlike the RFID technology, Bluetooth is found in almost every smartphone these days and is therefore just given one grade, medium. For the last two, UWB and VLC/IR, a whole new infrastructure is needed and receivers/transmitters is not usually found in any existing mobile plattform. These are therefore given a high grade.

Line of Sight or LOS will in all technologies give a better measurement. However, the vision-based and light based technologies need a LOS to the client in order to make a position estimation. Without this, they will not find the client as RF based technologies will, no matter if there are obstacles in the way, like a wall.

Client size and weight refers to the need of a specific size of a client used by the technology. This item do not apply to the vision-based, which can identify a person or feature of interest without the need of a special client. For RFID, UWB and VLC/IR the size highly depends on what tags and transceivers that are chosen. WLAN and Bluetooth is given the grade small because they can be found in smartphones and most of the modules available are very small.

With dynamics the authors of this thesis report mean the movement of people, humidity, temperature and other disturbances that change over time. WLAN, RFID and Bluetooth are highly affected by this, due to the attenuation caused by people, temperature and humidity changes. UWB is also RF based and has the same problem as the three mentioned, but is not as sensitive to other radio waves from consumer products as the three. UWB will not be much affected by for example microwave ovens, other WLAN APs, Bluetooth communication between different devices for file transfer and is therefore given a medium grade. The vision-based technology will probably have problem with many people in movement that will make it hard to identify a client. However, it has no other limitations like temperature or humidity and is therefore given the grade low.

Worth to think of when choosing a technology is the energy usage. If the client shall be deployed at some object that is not easy to reach or change battery on, the module used then need to be energy efficient. The new Bluetooth class, Bluetooth 4.0 or also called Bluetooth Low Energy (BLE), is therefore of much interest when it comes to this types of implementations. It is also worth to mention that accuracy and cost higly depends on the size of the area to position in. In a larger area more APs, tags or cameras will be needed and therfore cost more. With the same amount of APs, tags and cameras in a larger area the accuracy will instead decrease.

If the system shall be centralized or client based should also be considered. That is, shall the estimating algorithm be placed in a central place and just get readings from APs and clients or shall the algorithm be placed at the client that also do all the readings. A system that shall keep track on multiple clients in a building and report this to a central should preferably be a centralized system. The centralized system is easier to maintain and can easier make use of more sensor data to estimate a position, which will give a better accuracy. However, it will need a new infrastructure or changes in an already existing infrastructure. It will also be slower then a client based system because of all the data transmissions needed, which is not preferably in a SLAM implementation where data need to be collected as fast as possible.

2.6 Integration of Positioning and Localisation Methods

Two main methods have been discussed considering integration of the two Positioning and Localisation methods including mapping of the environment.
The first method uses the positioning data from the IPS and compares it with the SLAM localisation position. In the case of divergence in the SLAM algorithm, the proposed position ends up in a very narrow probability distribution which does not represent the true localisation. The happening of divergence is mostly hard to identify by the algorithm it self. By the use of the IPS localisation, the Euclidean distance can be monitored between IPS and FAST SLAM position and at a chosen level, the SLAM algorithm can be re-initialised.
Initially before the exploration is initiated, IPS is used to position the robot in the refined grid, which also synchronise both the IPS and the SLAM environment.

A second alternative is to use the positioning data of the IPS to bias the re-sampling step and focus the particles to effective areas, in the case of a particle filter or bias the Kalman Gain in the case of a Bayesian filter.
Both of the methods are at the same order of complexity concerning computational resources, but the second method will have a more complex implementation.

During march 2013 news broke that Apple acquired indoor location start up project called WiFiSLAM. The technology focus effort on estimating position in an indoor environment by fusing accessible data in toady's smart phones. Initially rumors told that the main algorithm uses inertial sensors to represent the process model and by the use of Wifi trilateration estimating distances to fixed points in the environment [129]. This reflects the actuality of the problem and that no solution yet exists.

2.7 Applications

In this section, three different scenarios is presented, where SLAM, IPS or both SLAM and IPS combined can be used. These scenarios will work as examples during the discussion in section 2.8 of how to think when choosing sensors and methods.

2.7.1 RobCab

RobCab[1] is a life science autonomous project developed by RobCab AB. The company today is selling a robotic solution which should replace the work performed by a nurse for a third of the cost. The robot is able to perform transportation tasks like transportation of medicines or medical goods through the corridors of a hospital. It is also capable of helping patients with guidance if they have impaired sight, or if a nurse needs help of transporting a bed. It is also designed to help disabled people in wheelchairs by towing and guiding them around the hospital.

The Project environment is dedicated to the dynamics of a hospital, with a lot of corridors with quite high level of symmetries.

To cope with the dynamical environment of a hospital, the robustness has to be very high. At the same time the precision has to be very exact to dock with beds and for delivering goods at the right place. From the specifications of the operation, the requirement of the sensors are tough. The sensor used is therefore a combination of laser range scanners and a multiple usage of ultrasonic sonar sensors and IR-sensors.

[1] http://robcab.se/index.html

2.7.2 Robotic vacuum cleaner

Robotic vacuum cleaners have grown in popularity among the population. As late as 2012 it was the electronic Christmas present of the year in Sweden[1].

The main task for a robotic vacuum cleaner is to autonomously navigate through the dynamic home environment and vacuum dirt. It is still not good enough to be an replacement of an ordinary vacuum cleaner, but instead a compliment. Most of the available robot vacuum cleaners on the market are able to find their way back to the charging station when the battery runs low. Some models are also capable to return to the exact same position they were on before needing to charge, when other models just know in which direction to go.

How the robotic vacuum cleaners navigate in the environment is different from model to model. Some are just running random, when others are moving in an intelligent pattern, like first go around the room to get the outline and then going in straight lines, back and forth. Different manufacturers has chosen different sensor solutions to the obstacle avoidance and navigation problem. Neato Robotics have developed a laser scanner[2], LG[3] and Samsung[4] are using cameras mapping the ceiling, Electrolux (which is no longer on sale) used ultrasonic sensors[5] while iRobot uses bump sensors[6]. All above solutions are often complemented with other sensors like IR scanners and IMUs.

2.7.3 A Warehouse

In a warehouse different accuracies can be needed, depending on what to monitor or position. High value goods, as for example computers, can be more interesting to know the exact position of, while low value goods maybe only are interesting to know when it has arrived or departed from the warehouse, like the installation at ABB Finland in 2005[7].

There may also be autonomous forklifts in a warehouse. These need to be able to navigate and find goods to move and also how to move the goods. For example, the forklift need to find the holes in the pallet.

A warehouse often has very open areas, but in a high bay warehouse shelves can be hindering otherwise good Line Of Sights. The environment is also highly dynamic and obstacles like high pallets can one day stand in front of an IPS AP and in another day the AP has LOS. This will make it difficult to develop a well calibrated propagation model or creating a good Radio-Map.

2.8 Selecting system

When selecting a system, that is sensors and/or methods, one often is faced with the problem of having the best quality system, with a low cost and taking no time to deploy. It's pretty easy to figure out that this is not possible. In project theory it's usual to talk about the time, cost and quality triangle, see figure 2.17. The triangle is also applicable on the decision making when selecting a system. A higher quality need, or in this thesis a higher accuracy, will make

[1] http://www.elektronikbranschen.se/nyheter/arets-harda-julklapp-robotdammsugaren/

[2] http://www.neatorobotics.com/robot-vacuum/

[3] http://www.lg.com/se/dammsugare/lg-VR62601LV

[4] http://www.samsung.com/se/consumer/appliances-laundry-kitchen/vacuum-cleaners/vacuum-cleaners/VCR8855L3B/XEE

[5] http://en.wikipedia.org/wiki/Electrolux_Trilobite

[6] http://www.irobot.com/us/learn/home/roomba.aspx

[7] http://www.abb.com/cawp/seitp202/a1c82bd78f74140ec1256ff2002851e6.aspx

the system to cost more and be more advanced, which will lead to more time needed. That is, time and cost can not be high rated requirements, because they will in that case conflict with an even higher rated required accuracy. This is fundamental to bear in mind when selecting a system for a specific task.

First one should identify the environment where the system will be deployed, decide where in the triangle to place the system and figure out other important requirements by talking to stakeholders. If the system shall be deployed in a very dynamic environment and is very safety critical, the quality needed has to be raised and one have to let either the system take longer time to deploy or cost more. The system can otherwise not be deployed in a safe way. However, if the environment is dynamic, but not as safety critical, for example the system is allowed to bump into obstacles without harming the environment, a lower accuracy is possible to be allowed for the sensors.

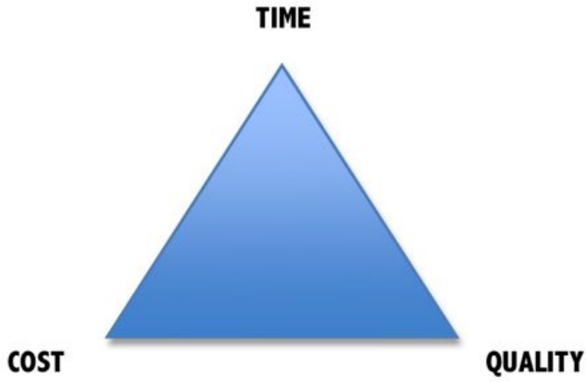


Figure 2.17: Time, cost and quality triangle often used describing projects, but can be applied to when chosing methods and sensors.

It is also important to conclude the needed accuracy for the stakeholders. A consumer product may allow for a more un-reliable output from the sensor, when a safety critical product can not. One shall also think of the possibility to use different type of sensors and then fuse the result together. It is a more time consuming task to tune the sensor fusion algorithms, but will allow for cheaper sensors. That's because two sensors that don't share the same source of error will compliment and correct each other, if the sensor fusion algorithm is well tuned. More fused models and sensors with different source of error will improve the result, there is no theoretical upper limit. However, the improvement will slowly decelerate when reaching a certain amount of fused sensors/models. Even using sensors of the same type fused with each other and models will make an improvement of the result, but not as great as when fusing with sensors with other source of error. That said, if the fusion algorithm is badly calibrated, more sensors and method will probably make the result worse than when only using one sensor.

If accuracy is very important and the system has to be released in a short time, off-the-shelf sensors with for example tuned Kalman-filters is recommended to be used. To tune a sensor fusion algorithm, like a Kalman-filter, is both advanced and time consuming. Some off-the-shelf sensors with fusion algorithms have become cheaper, like IMUs with Kalman-filter, but usually they are still to expensive for a majority of the mass produced consumer product. Consumer products however, mostly not require a well tuned fusion algorithm and would therefore not need off-the-shelf sensor solutions. For high-end consumer products, like smart phones, the company them selves have probably developed the algorithms and are not using off-the-shelf sensors, for example Apples M8 motion coprocessor[1].

If both time and cost is important, as in the case for a mass produced product that has to be released before the competitor product, it is recommended to develop the system with an off-the-shelf solution and in parallel develop an own sensor solution. When the own solution is ready it shall be easily exchanged with the off-the-shelf one.

In this thesis, sensor fusion was used in the SLAM implementation, for instance to fuse output from the ultrasonic sensors with the turning model. There was, however, thoughts to fuse (or combine) SLAM with IPS to get an better localisation result. That's because IPS has a totally different source of error, like attenuation of radio waves, when SLAM has errors like, attenuation of sound waves and a bad tuned turning algorithm. Due to issues with the platform, time was limited and an implementation of the proposed fusion of SLAM and IPS could not be done.

2.8.1 Selecting SLAM

Selection of the best case for SLAM implementation is carried out answering in which environment the operation is supposed to be performed and which capabilities of observing the environment that exists. Generally it's preferred to use a feature based map or a so called topological map when environment grows larger. Concerning complex environments with a lot of structure it may though be necessary to use a grid map structure with fine grids. The result will be a more complex mapping structure and will require more time consuming calculations .

Reasoning about the type of SLAM algorithm to use is a question of complexity of the implemented algorithm. The development of the FastSLAM algorithms by Michael Montemerlo[61] has improved the performance of SLAM tremendously and today the algorithm outperform the more simplified EKF SLAM algorithms. Although dependent of implementation, if linearization is enough for the environment, it may be sufficient to use an EKF SLAM algorithm for mapping and navigation which then requires a much easier implementation, see section 2.2.7.

2.8.2 Selecting IPS

For indoor positioning its very important to first determine the environment to employ the IPS. This because different type of solutions struggles with different types of environment. For example a VLC solution, requires a line of sight, when a WiFi solution is disturbed by microwaves. All of the solutions are somewhat disturbed by the dynamics, but some more than others and it's important to determine the dynamics in the environment as well. The cost and complexity is also some what dependent on the environment. In an home environment there is probably just one WiFi AP, when in an office there can be hundreds. That means in an office no extra APs are needed, the infrastructure is already there, if a WiFi solution is used and this solution will therefore be more cost effective. In a home environment, however, extra APs (no existing infrastructure) will be needed. It is therefore not as obvious in a cost point of view to chose WiFi in front of for example Bluetooth.

When the environment have been identified, the absolute needed accuracy shall be decided, with aid of the intended environment. Because there will be possibilities to do limitations in the needed accuracy depending on the environment. If the desire is to position an object only in the vicinity of an other object or to just position a person to a room level map, one can allow less

[1] https://www.apple.com/iphone-6/technology/

accuracy, less complexity and less time tuning algorithms. However, if the desire is to position the object in the 3D space with an absolute position, the system will be more complex, leading to a higher cost and more time needed.

Depending on the method to use for position, more or less advanced hardware will be needed and it is back to the cost, time and quality triangle. An AOA method will require a more advanced and costly hardware, but can provide good estimations if used in the right way. Distance related measurements will as well require advanced synchronization or a good propagation model. A good propagation model can be time consuming to calibrate, which the building of a radio map also is. The different methods can also be fused together to get an even better result as was mentioned before about sensor fusion, but will need more tuning time and cost more. An example is the TLS system developed by Atlas Copco[1], which locates a tool with the use of both AOA and TDOA of an UWB signal. One may also think of improving this positioning by fusing other sensor data like an IMU for the orientation. This will, however, lead to an even more costly and complex system.

2.8.3 Selecting for different applications

In section 2.7 above, different application areas or scenarios were described. Those will be used to give example of a way of thinking when choosing sensors and methods for different applications. The approximated accuracy that can be needed in these scenarios will also be discussed.

For the RobCab the accuracy (quality) should be the prioritized parameter compared to cost and time because of the hospital environment where people are moving around almost all the hours of the day, which leads to high quality sensors with advanced and complex methods. Just one type of sensors will not be enough, different type of sensors has to be fused together to reduce the error. The size of the robot is not a big problem, bulky sensors can therefore be allowed, like high accuracy laser range scanners. The laser range scanners have to be fused with ultrasonic sensors to detect glass obstacles. If an IPS method shall be used to improve the result, one has to think of the hospital environment and sensitive equipment. Not all IPS infrastructures are good to use in this environment, like WLAN or older Bluetooth protocols (Bluetooth version four, BLE, may be used because of its short range and low energy waves).

A robotic vacuum cleaner is a consumer product and will be mass produced. Therefore both time and cost is more important than accuracy. The robot it self is usually pretty small and is not driven in high speeds. The environment is not to sensitive to impacts and a lower accuracy is therefore allowed. One shall bear in mind that a robotic vacuum cleaner today is not a replacement to a regular vacuum cleaner, but a complement. Therefore, a robot that just drives around randomly and turns around when bumping in to things can be allowed, the impact can not be too high though. This means that no complex SLAM algorithm or advanced sensors has to be used. Advanced robotic vacuum cleaners today is more a fancy object to be able to talk about at the dinner table. However, if one in the future want the robot to actually replace the regular vacuum cleaner, a better navigation technique and sensors has to be used. It still don't need to be as advanced and expensive as in the RobCab example.

For the warehouse one can think of two different implementations, to track goods and autonomous forklifts or other robots to carry goods. To implement a way of tracking goods, the wanted accuracy shall first be decided. As have been mentioned, two good examples of different accuracies are high and low value goods. For high value goods, better accuracy will be needed, positioning

[1] Tool Location System by Atlas Copco, http://www.atlascopco.com/images/toollocationsystem_tcm19-1433190.pdf

in 3D will, however, probably not be needed for either of the two. For low value goods, it could be enough if all boxes were tagged with an RFID tag and scanned when they are coming in and out of the warehouse. If the warehouse is big, one could think of using different gates to drive through in order to locate the goods in different sectors. To get a better accuracy than that for the high value goods, an IPS together with SLAM in the forklifts can be used. If the movement of the forklifts is tracked by a SLAM algorithm, with the driver in the loop, the trace could be fused with position data from the IPS. When choosing IPS infrastructure, one has to think of the large distances in a warehouse and the dynamics, that will be lower than for an hospital environment. For the second implementation, autonomous forklifts, high accuracy is needed. A forklift with goods can have a weight of up to five tons or more. A collision would be devastating. Forklifts are not consumer products and are allowed to cost more, a better accuracy can therefore be guaranteed by using high value sensors as laser scanners. For the forklift to dock with a pallet, one can think of two possible ways. The simplest thing would be to let a human take over the docking and the forklift just doing the long runs. The second solution would be to have a good laser scanner or a vision based system with complex algorithms to identify the feature, the holes, and to dock. For a somewhat cheaper system that will sell better, the first solution would be preferred. Because in bigger warehouses it is usually the long runs that take time and is important to get rid of for more effectiveness.

Chapter 3

Selection of the IPS and SLAM systems

3.1 Requirements

In order to design the system and choose methods and sensors for the implementation it is important to know the requirements they should meet. A short requirement meeting was therefore performed to formulate the requirements that can be found under the headings below.

When formulating these requirements wishes from the stakeholder ÅF, delimitations as well as knowledge from the background study of what is possible have been taken in consideration. Most important wishes from ÅF was low cost, limited amount of time, as good accuracy as possible and to get the Digital Lobster, ÅF:s six wheel autonomous robot to run autonomously. Some of these wishes compete with each other, for example to get as good accuracy as possible the system needs more advanced sensors and will cost more. In discussions with ÅF it was decided that a low-cost is most important and a lower accuracy can be allowed for the use of low cost sensors to reduce the cost.

Measure Range

a: The Digital Lobster shall measure range distances to an object that is at least 3 metres away.
b: The Digital Lobster should at least measure range distances down to 10 centimetres.
c: The Digital lobster shall measure the range distance with a range accuracy of at least ± 3 centimetres.
d: The Digital lobster shall measure ranges with a frequency of at least 10 Hz.

Detect Obstacle

e: The Digital lobster shall detect obstacles with an angular accuracy of at least ± 15 degrees.
f: The Digital lobster shall detect obstacles with a diameter of at least 10 cm.

Communicate

g: The Digital lobster shall communicate its sensor data wireless to a host computer for SLAM and IPS calculations.
h: The Digital lobster should be able to communicate its position to an external host computer wirelessly.
i: The Digital lobster should be able to receive steering commands from a host computer.

Measure Distance Travelled

j: The Digital lobster shall measure distance travelled with an accuracy of at least ± 1 cm.

Measure Absolute Heading

k: The Digital lobster shall measure absolute heading with an accuracy of at least ± 10 degrees.
l: The Digital lobster should measure absolute heading with an accuracy of at least ± 5 degrees.

Measure Received Signal Strength Indicator Values

m: The Digital lobster shall measure Received Signal Strength Indicator values with a frequency of at least 1/3 Hz.

Data Storage

n: The Digital Lobster should be able to store map data of at least a size of 100Mb.

Calculate

o: The Digital Lobster should be able to calculate SLAM and IPS locally.

Map

p: The Digital Lobster shall build a map using grids of a maximum grid cell size of 20x20 cm.
q: The Digital Lobster shall build a map using SLAM.
r: The Digital Lobster shall build a map combining SLAM and IPS.

Position Itself

s: The Digital Lobster shall position itself with an accuracy of at least the area of a 30 square metres circle, when only IPS methods are used.
t: The Digital Lobster should position itself with a combination of SLAM and IPS.

Navigate

u: The Digital Lobster should be able to navigate with the purpose of exploration.
v: The Digital Lobster shall be able to explore the environment in a random way.
w: The Digital Lobster should be able to navigate to a given coordinate with an accuracy of at least ±15 cm.

3.2 Proposed Methods

In this section proposed methods for SLAM, IPS and the integration of these two will be presented. Knowledge after the background study as well as the requirements has been taken in consideration when these methods have been chosen. The methods will also be explained. For how they are implemented and which sensors that have been chosen see chapter 4.

3.2.1 SLAM

The selection of the SLAM algorithm is all based on the requirements specified in chapter 3.1 which reflects the environment of operation, the computational power necessary and affordable, the sensor cost factor and the precision of localisation and mapping.

A big question to start with was the major issue to determine if a solely localising algorithm using a predetermined map of the environment or if a full SLAM algorithm should be implement. Localisation in a predetermined map is much simpler to implement both in sense of computational cost and also of the precision of the localisation. Although a wish from AF was to implement a full SLAM algorithm due to the lack of necessary maps describing the environment and also due to the possibility of operating in different environments with unknown mapping, in a simple way. This has also resulted in the requirement of mapping in the previous section.

The only possible solution was to implement a full SLAM algorithm including both mapping the environment in the same time as it determines the position.

By the decision taken to implement SLAM, different types of filters exists and are working good in different kinds of implementations. See chapter 2.2.7 for further explanation. Due to non-linear characteristics of the robot movement, a particle filter was the best choice and has also been proven as a working concept from the "State of the Art" research.

The selection to use a grid map was based on the operation in an indoor environment requirement, which requests a diversity of details that exceeds the featured maps representation possibilities.

The remaining decisions was to select sensors perceiving the environment, and sensors reading the control feedback. As discussed in chapter 2.3.5, the only possible sensor measuring range, in the cost category specified in the requirements, are ultrasonic sensors. The sensor has also been used in a couple of working platforms (2.7.1). To get the necessary precision in distance traveled, wheel hub encoders was among "Inertial Measurement Units" composed of Gyro:s and accelerometers the only appropriate alternatives for control sensing.

Due to the great increase of complexity implementing an IMU and also our affordable workload, the possibility of implementing an IMU was neglected. Thereby wheel hub encoders was selected as the feed back of the control signal to still have a quite accurate process model feedback.The odometry feedback model has also been used successfully in other projects as stated by the “State of the art” study, see section 2.2.5.

It has though, despite research in the subject not been found any previous report implementing SLAM on a six wheeled robot. The numbers of driving wheels will contribute to a complex non linear model describing the turning behaviour, which has to modelled carefully.

Used Algorithm

The demanded performance of the SLAM algorithm and its operational environment has limited the usable algorithms to a grid based fast slam particle filter algorithm. Particle filter SLAM algorithm has shown a great performance of precision compared to computational complexity. Secondly it's the only algorithm able to calculate multimodal posterior distributions which may be the difference between divergence or not in a complex indoor environment.

The algorithm requires an initial set of particles, control signals and a set of range measurements which are used in the best way to calculate the vehicles position and mapping. The algorithm is very similar to a occupancy-grid-mapping algorithm using a particle filter which was discussed in chapter 2.2.7.

In table 3.1 the main principle of the grid-SLAM algorithm is presented in pseudo-code. This will be the major framework or main-function for the SLAM algorithm implemented. The compounds of it will stepwise be presented in the rest of this chapter

The first step of the main function, "the prediction" is using the encoder data to by dead reckoning move the robot to its next iterative step. This is performed for every particle and is performed by the first "for" loop, where the *predict*() function is iterated. A disturbance is added by the use of a covariance matrix to comply with the uncertainty in encoder measurement concerning translational and rotational movement.

For all the particles in the particle filter, an iterative loop is evaluating each particle and its position and compliance with the real world. The function *beam_range_finder_model()* on line 10 in table 3.1 is evaluating the correctness of each particles perception of the environment to derive each particles weight, which is used in the re-sample step. The function calculates the probability of the present particle, using a specific probability density function. It calculates the probability of each particle by a comparison of a raycasted range to the previous observed features and the actual sensor range measurements. For a detailed derivation of the model, see chapter 2.2.5.

At line 11 in the same table the algorithm executes the *"occupancy_grid_mapping"* function which is updating the map using the range measurements. It steps through all particles and each individual cell in the map and compares a calculated range and bearing, for each individual grid cell, against the sensor range and bearing. The comparison is then evaluated and reports if the cell is occupied or not. The comparison is made by the use of a so called "inverse sensor model" which will be further detailed in the following pages.

Table 3.1: The Gridslam 2.0 main function

```
GRIDSLAM 2.0 ($x_{t-1}^k, u_t, Q$)
    if [encoder data recieved]
        for [all particles]
            ${x_t}^k = predict({x_{t-1}}^k, u, Q)$
        end
    end

    if [range data recieved & enc data recieved]
        for [all particles]
            ${x_t}^k = beam\_range\_finder\_\text{model}(z, {x_t}^k, m_{t-1})$
            ${m_{t-1}}^k = Occupancy\_grid\_mapping(z_t, {x_t}^k, {m_{t-1}}^k)$
        end
        $Y_t = resample\_particles(w_t, {x_t}^k, {m_t}^k)$
    end
```

The prediction step is identical to the theoretical prediction step described in chapter 2.2.3. The sampling is made from the previous pose with the translational or rotational movement added. To apply the characteristic diffusion to the sampling, a Gaussian noise Q is added, representing the translational and rotational uncertainty of the robot.
The algorithm is presented in table 3.2. The algorithm is also divided in an if-statement due to the different geometrical models that apply depending of a turn is performed or if a regular drive is performed.

Table 3.2: Explanatory algorithm for the predict function

```
PREDICT(${x_{t-1}}^k, u_t, Q$)
    if [TURN]
            $sample\ x_t \sim p(x_t|u_t, x_{t-1})$
    elseif [DRIVE]
            $sample\ x_t \sim p(x_t|u_t, x_{t-1})$
    end
```

The $Beam\_Range\_Finder\_Model()$ function is derived in equation 2.2.19 and described in the same section. The big changes applied is that the coefficients have been tuned for the specific sensor used in the sense of the correct Gaussian distribution, the correct maximum beam range and a random noise level. See section 4.2.3 for sensor data.

Table 3.3: The Beam Range Finder model function in a principal view

```
**BEAM_RANGE_FINDER_MODEL**$(z, x_t{}^k, m_{t-1})$
    $\hat{z} = RayCast(z\_meas_t, x_t{}^k, m_{t-1}, BeamMax)$
    **for** [All RayCasted ranges]
        **if** [feature detected]
            $p_{rand}(z_t{}^k|x_t, m) = ...$
            $p_{\max}(z_t{}^k|x_t, m) = ...$
            $p_{short}(z_t{}^k|x_t, m) = ...$
            $p_{hit}(z_t{}^k|x_t, m) = ...$
            $p(z_t{}^k|x_t, m) = \begin{pmatrix} z_{hit} \\ z_{short} \\ z_{\max} \\ z_{rand} \end{pmatrix}^T \cdot \begin{pmatrix} p_{hit}(z_t{}^k, x_t, m) \\ p_{short}(z_t{}^k, x_t, m) \\ p_{\max}(z_t{}^k, x_t, m) \\ p_{rand}(z_t{}^k, x_t, m) \end{pmatrix}.$
        **else**
            $p(z_t{}^k|x_t, m) = 1$
        **end**
    **end**
```

As discussed in the introduction to the process and measurement models chapter 2.2.3, Raycast is used to calculate ranges and bearing to previously observed features which are used in the *beam_range_finder_model*() to calculate the probability of each particle. The algorithm is composed of trigonometric functions and logic cases to give the correct ranges and bearing.

Table 3.4: The Principle of the Raycasting function

```
**RAYCAST**
    **for** [all measurements]
        Evaluate if a feature has been detected in this
        direction earlier then save range and bearing.
    **end**
```

The methodology of the Raycasting algorithm operated in a grid map construction of the world is illustrated in Figure 3.1. The vehicle is illustrated in the down left corner and measure the range by use of the 45 degree front right sensor against an object.The object in this case has to be stored in the previous grid map to be possible for recognition. The algorithm is then geometrically evaluating each grid cell around the beam in a specific pattern until it identifies an occupied cell. If an occupied cell is identified, it uses the range to the middle of the cell and the absolute heading as the bearing of the vehicle together with the relative bearing of the active sensor.

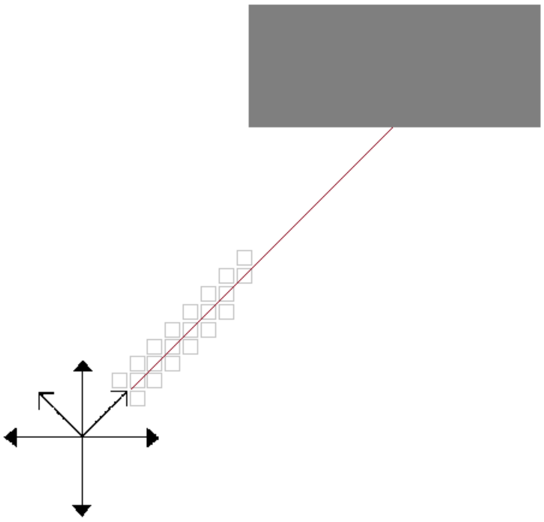


Figure 3.1: Illustration of the Raycast methodology

The evaluation process to select if a grid-cell is occupied or not is following the golden standard as described by the state of the art report, section 2.2.7. The bearing and range of all the sensor are evaluating if the specific grid cell is the one the range scan has identified in the real world, “Occupancy-Grid mapping”. In an iterative context, this is performed for every grid cell and saved as the updated map and perception of the environment.

The algorithm is presented in table 3.5 where all cells are iterated, but evaluated cells are limited to cells in the vicinity of the range scanning sensors.

Table 3.5: The principal operation of the grid occupancy function

```
OCCUPANCY_GRID_MAPPING($z_t, {x_t}^k, m_{t-1}$)
    for [ all cells $m_i$ do]
        if [$m_i$ in perceptual field of $z_t$ then]
            $m_{t,i} = m_{t,i-1} + \text{inverse\_sensor\_model}(m_i, x_t, z_t)$
        else
            $m_{t,i} = m_{t,i-1}$
        end
    end
```

The last and most important step of the main algorithm is the re-sampling step which focus all of the particles to areas where they perform the best duty. The re-sampling method is composed of a "stratified resampling" technique where the particles are grouped in separate groups to avoid particle deprivation. This is used because of the possibility of biased re-sampling which will affect the convergence.
The separate groups are then resampled using a low-variance resampling method as described in section 2.2.3.

The main algorithm is in a simplified form presented in table 3.6. The function is first iterated over all particles, at line 2-4 and performs a normalisation. At line five it executes the function *stratified_resample* which is presented below in table 3.7.
The stratified re-sample function is executing the low-variance re-sampling technique described in chapter 2.2.3. The actual sampling is made by the cumulative sum function, which sums the different weights into a array. The array is then used for sampling the particles proportional to $p(x_t|z_t, u_t, x_{t-1})$.
The remaining part of the "Resample Particles" function is writing the re-sampled particles to the memory and resetting the weight to equal values.

Table 3.6: The principal operation of the re-sampling function

```
1 : RESAMPLE_PARTICLES($particles$)
2 :     for [ all $particles$ do]
3 :         normalise $particles.weight$
4 :     end
5 :     $[keep, Neff] = stratified\_resample(particles.weight)$
6 :     if [$Neff <$ "Resampling Factor"]
7 :         $particles = particles(keep)$
8 :         "reset particle weights"
9 :     end
```

Table 3.7: The principal operation of the stratified-resample function

```
STRATIFIED_RESAMPLE($particles.weight$)
    $select = Stratified\_Random$("number of particles")
    $w = cumsum$("particle weights")
    for [ all $particles$ do]
        while [ $counter <$ "range of particles" & $select < w$]
            $keep(ctr)$ ="particle index"
            ctr++
        end
    end
```

3.2.2 IPS

A wish from the stakeholder ÅF was that the Digital Lobster should position it self as good as possible. However, it “shall position itself with an accuracy of at least the area of a 30 square metres circle, when only IPS methods are used”, see section 3.1 Requirements. In reality the total mean euclidean error shall be smaller than three metres. No extra infrastructure should be needed. That is, the implementation shall be possible with the infrastructure all ready existing in the ÅF head office building. i.e. is WiFi access points for the wireless network hidden in the roof and no access to the servers. However, some small extra beacons and receiver was allowed to be used for the positioning system. The system shall not have a too high complexity level, because of the tight time frame.

The infrastructure criteria and cost were more important than the accuracy. ÅF could allow a result with lower accuracy that not needed any extra infrastructure. This made both the UWB and VLC/IR positioning not interesting to use. A vision based solution was also quickly decided to not be a good solution, because of a high complexity level and the need of a server based solution.

The Digital Lobster shall also be able to communicate wireless with a remote host computer, see section 3.1. This is achievable with the three remaining technologies, WLAN, RFID, Bluetooth. However, to do this with an RFID link, more complex and advanced hardware is needed than one can find in a low cost solution made for positioning.

It was not easily decided if a WLAN or Bluetooth solution should be implemented. They are both similar in the complexity and a cheap WLAN or Bluetooth module can easily transmit the data needed by the requirements. Even if the Bluetooth solution need a new infrastructure in a bigger extent than the WLAN, it is still cheap enough. However, the WLAN was all ready well employed in the office building and two extra WLAN routers existed. Because of this and that the performance that can be expected from both of them, see section 2.5, is similar a WLAN solution was decided to be used.

Next that had to be decided was which type of positioning method to use. An AOA method will require very advanced and sophisticated hardware in order to measure the angle of arrival. This was to lead to an more complex and expensive solution.

A distance related measurement method was considered. A time based method should require a high level of precision and synchronization, which would had risen the complexity level. The RSS measurement method was therefore considered. However, to use the distance related measurement method the locations of the AP's had to be known. This was not the case in the head office building, all AP's are built in the ceilings and are therefore hard to locate.

The RSS profiling measurement method is therefore proposed to be used. It will require a more time consuming pre-learning phase, when the fingerprint map, the radio-map, needs to be built. However, this will allow the use of all the existing AP's in the building, without knowing the position of them. The extra time needed for building the radio-map was seen to be fair in comparison to the extra complexity needed by the hardware and software for the AOA and distance related methods.

In order to reduce the time needed to build the radio-map, a Weibull distribution was proposed. This distribution will reduce the amount of samples needed [115] and also the time needed, to get a radio-map with high accuracy. Pei et. al. [115] presented that they could achieve an accuracy of 2.2 meters when they used their method in a WLAN positioning system. This accuracy was well under the required accuracy.

To sum up, a WLAN based IPS was proposed to be use in the Digital Lobster implementation. The positioning algorithm should be based on the Weibull distribution method proposed by Pei et. al. [115] and is described under the next heading, "Used positioning method".

Used positioning method

The method is divided in two different phases, the offline pre-learning phase and the online position estimation phase. In each phase RSSI measurements are collected from different APs.

At the offline phase around 20-30 RSSI samples is taken from each AP. With the RSSI measurements, Weibull parameters was calculated and stored in the Radio-Map, together with the position information. The Weibull parameters was calculated as follows [115]:

$$\delta = \sqrt{\frac{1}{n}\sum_{i=0}^{n}\left(O_i - \overline{O}\right)^2} \tag{3.2.1}$$

where δ is the standard deviation, n is the number of RSSI samples taken, O_i is the i-th RSSI measurement, $\overline{O}$ is the mean value of the RSSI samples,

$$k = \delta / \ln(2) \tag{3.2.2}$$

where k is the shape parameter,

$$\lambda = \begin{cases} 2 \times 1/\sqrt{\Gamma(1+2/k) - \Gamma(1-1/k)^2} & \delta < 2 \\ \delta \times 1/\sqrt{\Gamma(1+2/k) - \Gamma(1-1/k)^2} & 2 \leqslant \delta \leqslant 3.5 \\ 3.5 \times 1/\sqrt{\Gamma(1+2/k) - \Gamma(1-1/k)^2} & \delta > 3.5 \end{cases} \tag{3.2.3}$$

where λ is the scale parameter, Γ is the gamma function,

$$\theta = \overline{O} - \lambda \times \Gamma(1+1/k) \tag{3.2.4}$$

where θ is the shift parameter. For the calculation of the scale parameter, λ, no simplification has been done in the equation by substituting $1/\sqrt{\Gamma(1+2/k) - \Gamma(1-1/k)^2}$ to $(k+0.15)$ as was done by Pei et. al.[115].

At the online phase only one RSSI value from each AP is sampled. First the bin to which the value belong to is decided, the bin is a band of RSSI values. The probability for the actual bin is then calculated by

$$P\left(A_m B_n \left| R_i \right.\right) = F(x+w) - F(x) \tag{3.2.5}$$

where A_m is the AP from which the signal come, B_n is the n-th bin, R_i is the i-th reference point in the Radio-Map where the client may be located, x is the left edge of the bin, w is the width of the bin which is chosen to be $2dBm$ and

$$F(x) = \begin{cases} 1 - e^{-\left(\frac{x-\theta}{\lambda}\right)^k}, & x \geqslant \theta \\ 0, & x < \theta \end{cases} \tag{3.2.6}$$

is the Cumulate Distribution Function, CDF, for the Weibull distribution. The probability for each RSSI at a position is multiplied with each other. The position in the Radio Map that corresponds to the maximum probability,

$$P(\mathbf{O} \left| p \right.) = \prod_{m=1}^{k} P(A_m B_n \left| R_i \right.) \tag{3.2.7}$$

where $P(\mathbf{O} \left| p \right.)$ is the probability of a position p given the RSSI measurement vector $\mathbf{O}$ and k is the number of APs at the specific location in the Radio Map, is then considered to be the estimated position for the client.

A drawback when using the type of position algorithm as has been described above is that the position will only be estimated with the positions stored in the Radio Map. If a client is located in between two stored positions, lets say 1.5 in x- and y-direction on the relative grid map it can only be positioned to 1 or 2 in x- and y-direction respectively if these are the only positions that was stored. An approach to a solution is therefore proposed and later tested during simulation and implementation. When the algorithm have estimated a position that has the maximum probability it also checks the surrounding positions in the Radio Map. By then weighting the probability for those with the estimated positions probability, a position between two positions in the Radio Map can be obtained.

3.2.3 Integration

To fuse the positioning and localisation with intention to improve performance and make a better performing SLAM algorithm, it has been chosen to use a pragmatic context where the accumulated difference between the SLAM algorithm itself and the Wifi Positioning estimation tells if the SLAM algorithm is about to diverge and needs to be re-initiated. The method has been named Wifi SLAM in further context of this report.

The SLAM algorithm is then initiated at the best estimate of the Positioning Wifi-algorithm with a spread of particles according to the standard deviation of the Wifi RSSI index. The previous map stored in memory is used part-wise dependent of the the quality, which is derived from the path travelled and the comparison between Wifi Positioning and SLAM.

3.3 Simulation

Before implementing any software on the platform, the proposed algorithms were tested in a simulation environment. All simulations were done in MATLAB[1], where the final implementation

[1] Copyright, Mathworks Inc.

of the algorithms also will be. The decision of not deploying the algorithms locally on the robot platform was mainly based on the limited time in the project to use and learn a complex embedded platform with enough computational power, see chapter 4 for the used hardware and software. The simulated algorithms were therefore written as close as possible to its final structure to minimize the needed changes.

3.3.1 SLAM Simulation and Visualisation

The complexity of the "Particle SLAM" algorithm is quite nested and to simplify the development process an extended simulation environment was created in MATLAB, both for verification and validation purposes, but also for the process tuning the algorithm.

All of the parts in the algorithm, described in chapter 3.2.1 is developed in individual subroutines and the correct execution of the part-program is verified in the simulation environment.

The simulation environment constructs an imaginary room of operation where simulated sensor readings such as encoder feedback and beam range measurements are created as a result of the robot moving in it. The sensor data is then used to evaluate different configurations of the particle filter and the SLAM operation.

The only difference between the final executable SLAM algorithm and the simulation is the sensor readings which have to be sampled from satisfactory models and to be included into the simulation. Empirically ,the needed sensor inputs are the range scanning readings, which explores the environment in the proximity of the robot.
Also readings for the distance traveled and turn indication are needed which are fed in real execution from the wheel hub encoders.
To in the best way simulate the distance traveled and turning deviation, a model was created with the parameters of the wheel which transforms the input motor control signal to an output distance traveled or an angle for turning. Because of the velocity independence of the vehicle to determine distance traveled, when using wheel hub encoders, the only input to the model was parameters describing the vehicle and encoders.

The Range scanning readings to the simulation was a more computational complex algorithm. At first the simulation has to know the true vehicle position and also the environmental objects. The algorithm is working in the manner of a ray-casting algorithm specialised for a grid map environment as described in the previous chapter. To determine the distance from the vehicle to any feature, a progressing evaluation is performed in the direction of the sensor. If the grid cell is unoccupied the algorithm proceeds to the next grid until it reaches any feature or its maximum beam range. The range is then returned to the simulation as a distance to the centre of the identified grid cell and if no feature has been detected, maximum beam range is returned.

The Simulation is created around a specified grid of operation. Within the environment a selective amount of two dimensional static features can be placed as examples of corridor, rooms and so on. The navigation is performed by an use of a proportional controller navigating around a predetermined set of rally-points.

An example of the environment can be seen in Figure 3.2 where a simple environment composed of walls all around the grid and also three closed areas where the robot can't move.

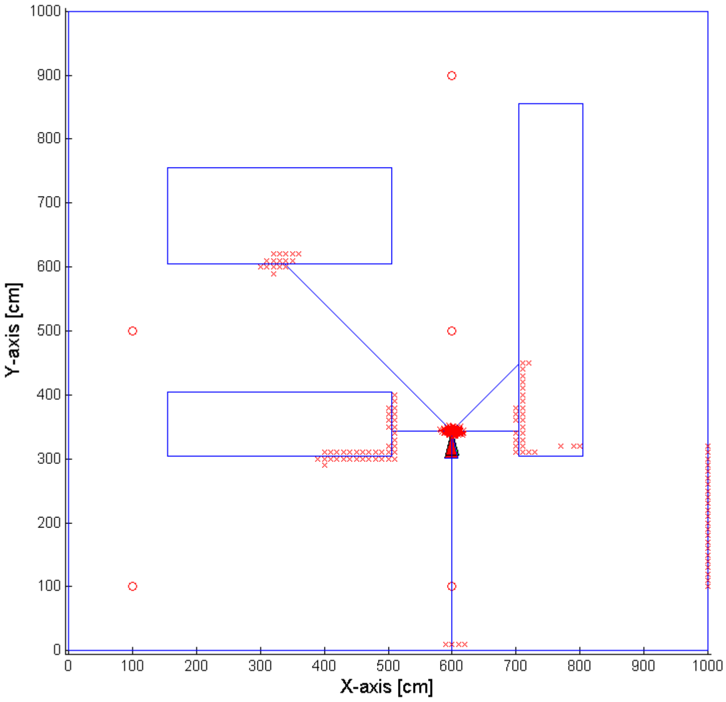


Figure 3.2: Illustration of the SLAM simulation environment

3.3.2 IPS Simulation

The purpose of simulating the proposed IPS was to get an indication if the sensor network based on WLAN and the Weibull-distribution based RSS profiling algorithm could fulfill the requirements stated in section 3.1. Disturbance factors, as attenuation, was altered to see how big impact these had on the system performance. The simulation also should give an indication of the minimum "set-up" needed, that is for example APs needed, to fulfill the requirements. The weighting method proposed to be able to estimate a position in between fingerprints, as was described in section 3.2.2, was also to be tested during simulation to see if it would increase the accuracy. This gave a hint if it would be worth using the weighting method in the implemented system.

First a model of the propagation of the radio signal used by the WLAN system had to be created. By using Friis' free-space model[105],

$$P_r(dBm) = P_t(dBm) + G_t(dBi) + G_r(dBi) + 20\log(\lambda/4\pi) - 20\log d, \qquad (3.3.1)$$

a good estimation of the propagation could be achieved. P_r is the received power by the client, P_t is the transmitted power, G_t and G_r are the antenna gain at the transmitter and the receiver respectively, $\lambda = c/f(Hz)$ is the wavelength and d is the Euclidean distance between the transmitter, the AP, and the receiver, the client. However to take the more difficult dynamic indoor environment into account the model was modified [100],

$$P_r(dBm) = P_t(dBm) + G_t(dBi) + G_r(dBi) + 20\log(\lambda/4\pi) - 10n\log d - X_a, \qquad (3.3.2)$$

to include a random Gaussian distributed noise factor, X_a, as well as an attenuation factor, n. By just simulate a distance a varied RSSI value at the client location could be calculated. The model was not calibrated against the real environment where the Digital Lobster should be positioned due to the limited amount of time available. Instead different factors as the attenuation factor, variance and number of access points was varied to see how these factors affect the accuracy of the IPS.

During the simulated offline phase the user was allowed to type the number of AP's that should be used, the position of these, the room size, the grid size and the number of samples taken at every position from every AP. With this information the simulation could generate randomized RSSI measurements for every position by using equation 3.3.2. The generated RSSI values was then used to calculate the Weibull parameters and stored in the Radio-Map, as described in section 3.2.2

For simulating the positioning phase of the IPS, the online phase, the user was allowed to chose if just one position or a grid of positions should be used. When the grid functionality was used it was also possible to simulate multiple positionings at each position in the grid in order to more efficient build a mean error diagram to generate the accuracy for the algorithm.

At each position one RSSI measurement from each AP was generated by equation 3.3.2. The RSSI values was then used in the online phase part of the algorithm described in section 3.2.2. In the simulation it was also tested if the proposed weighting of estimated positions will improve the result of the algorithm and if it should be used during implementation.

The standard setup to which the variations in variables was compared against can be seen in table 3.8. With this setup a total estimated Euclidean mean error of *0.49 meters* could be achieved. It is also worth to mention that it has not been tested what, for example, an increased attenuation factor of 0.5 or an increased variance of 2 is represented by in reality. This was due to time limitations in the project as well as no embedded hardware platform ready to be used during the simulation phase. However the simulated results can still give a hint what to think of when designing and choosing the setup in the real environment.

By changing the attenuation factor, n, from *4* to *4.5* the mean Euclidean error increased to *1.53 meters*. This meant that if the environment changed from the offline phase to the online phase, for example a new wall has been built, the result will be affected. It's important to have this in mind when changing the environment, a to large difference in the attenuation factor will lead to that a new time consuming offline phased has to be performed.

Table 3.8: Standard setup for comparing variations in different variables during simulation of IPS.

Room width	10 meters
Room depth	10 meters
Number of access points	4
Placement of access points	Corners
Number of samples	20
Attenuation factor (n)	4
Variance (Random variable)	1 dBm
Using weighted positions	Yes
Estimated Euclidean mean error	**0.49 meters**

As was expected, if the dynamics increases or changing during the day this will lead to an increased error. By changing the variance from *1* to *3*, for example from morning to before lunch more people will be in movement, the error increased to *1.16 meters*. This was also mentioned as a problem by Chang et al. [100].

In the real implementation only two access points, apart from the hidden ones in the ÅF head office building, were proposed to be used. A simulation with only two access points was therefore performed. As expected from the background study the mean error increased. With the two access points in the opposite corners, position [0,0] and [10,10], the mean error increased to *3.16 meters*. By rearranging them to standing next to each other the result was a mean error of *2.76 meters*. So the placement of the access points was also important to have in mind when setting up the real test cases.

The result was of course improved when adding more access points. By just placing a third access point in the middle the result was *2.42 meters* and by adding a fifth in the middle to the originally four the result was improved to 0.23 meters.

The own developed weighting of position method was proven to improve the result. In the originally setup only test samples at each radio-map position was taken, step of 1 meter. To better test this method, samples in step of 0.5 meter was conducted. The resulted mean Euclidean error with the use of weighting position was *0.58 meters*. Without the method the error increased to *0.67 meters*. It was therefore decided to keep the weighting function for the real implementation.

A conclusion of the simulations was that the proposed system will be able to fulfill the stated requirements in section 3.1 and that the two available access points that was proposed to be used would be enough. It was also shown that the system is sensitive to disturbances as attenuation and dynamics in the environment. However, no test was done in the simulations to try and solve this issue due to time limitations. The proposed weighting method did improve the result. Even if it was not by much it was still such improvement that it was decided to be implemented in the final system.

Chapter 4

Implementation and testing

The following chapter will explain the implementation phase where the algorithms and functionality based on the stated requirements are explained in more detail. At first, the basics of the platform is introduced explaining the existing hardware and its limitations. Further on, different models and characteristics will be explained, like models describing movement of the Digital Lobster, different communication protocols and also how the robot is controlled. Finally the concept of the implemented SLAM algorithm and IPS positioning algorithm will be presented in detail.

4.1 The Platform Digital Lobster

To evaluate and answer the question ”if SLAM could be improved using positioning systems based on Wifi trilateration” a platform for implementing was needed for real-world evaluation. The platform is used to implement the SLAM based on encoder data and ultrasonic range sensors together with a wireless Wifi node scanning for signal strength from access-points in the vicinity.

The Digital Lobster platform is built around a chassis from ”Polulu Robotics and Electronics” called ”Wild Thumper”. Wild Thumper is a six wheeled platform with pairwise suspension of the six wheels. The motors driving the platform is six 34:1 DC gear motors with encoders installed on all wheels. However, only encoder signals from the middle wheel-pair was used.

By considerations of the requirements and the necessary resolution of range scanning sensors, the platform have been equipped with six range scanning ultrasonic sensors selectively mounted around the platform. The sensors are intended to measure the distance from the robot to proximity objects for SLAM algorithm calculations and obstacle avoidance.

Computational resources are distributed between two embedded platforms and one external server(PC) for calculations. A Xilinx zynq 7000 SoC[1] platform was used to monitor and store the encoder data and pass it to the Hercules MCU[2] using a SPI communication. The selection using the Zynq SoC was mainly based on the possibility of the FPGA to count for six parallel encoders at the same time which was not possible to do with a regular MCU, although only two encoders was used simplifying the implementation. Beside the encoder calculations, all the real-time applications run on the Hercules MCU. The MCU is used to receive control and set PWM

[1] `http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/`

[2] `http://www.ti.com/tool/LAUNCHXL-RM42`

signals for the motor driver, it also receives and store the ultrasonic signals and the encoder signals which is communicated to the computational computer.

To communicate with the external PC performing more extensive calculations, a wireless TCP/IP protocol was set up. Through the link, both steering commands and sensor data was passed between the Digital Lobster and the PC-Server. A Wifi module from Roving Networks (RN-171) has been used to send information between the Hercules MCU and the PC. The module is a powerful tool which is built around a 32-bit processor together with a 802.11 b/g radio transceiver. The module is communicating with a UART protocol to other modules which has been used as a link between the Hercules MCU and the Wifi module.

In Figure 4.1, the overall infrastructure of the set-up is illustrated. The Digital Lobster platform itself is delimited by a patched line and the surrounding hardware is outside.

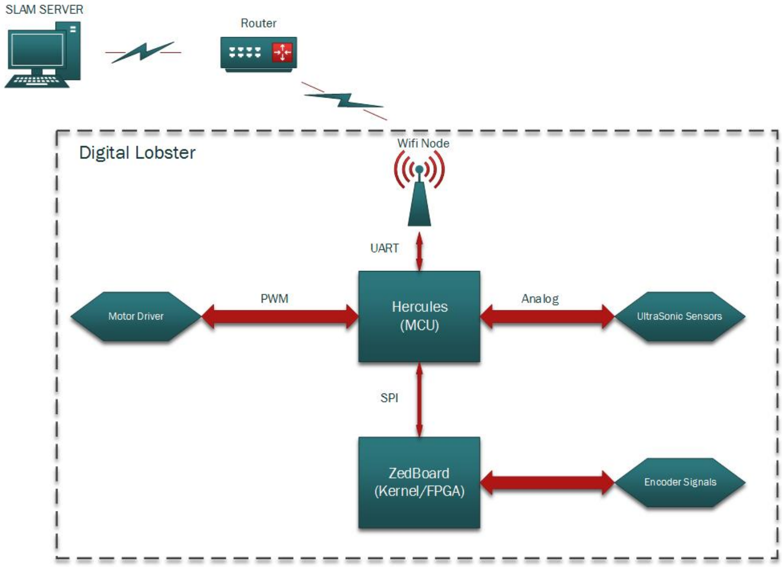


Figure 4.1: Infrastructure of Digital Lobster

4.2 Implementation on Digital Lobster

This section will, as have been mentioned, explain the implemented SLAM and IPS algorithms, as well as implemented models, control, sensors, communication protocols and real-time operating schedule. Everything has been done as part of the thesis if nothing else is mentioned, because of the need to change most of the infrastructure on the Digital Lobster.

Only a two dimensional map was used in the implementations of the algorithms, as was mentioned in section 1.3. That is, a plane in $x-$ and $y-$coordinate. The decision was made to lower the complexity and based on the limited amount of time available for the thesis.

If more than one floor was to be used in the implementation and testing, one could think of using a semi-3D map where the different floors was the $z-$coordinate. However, this was not the case due to delimitation to only test at floor seven at the ÅF head office.

4.2.1 Mathematical models

Driving Model

The driving model is used in the state where the Digital Lobster is running straight forward. It is using the encoder feed-back to determine and to control the distance traveled. The transformation of encoder counts to distance traveled is generalised to be dependent of the wheel radius only.
By the use of an average count from the two different encoders, a model was tuned to determine the relation of distance traveled against the encoder counts. This model has then been used controlling the distance traveled in the Hercules MCU.

Turning Model

The basic Idea of SLAM is composed on the fuse of multiple sensor data, where one of the sensor data is range scanning data to objects and the other sensor source is some kind of control correspondence from the vehicles maneuvers.

In more detail, the Digital Lobster is controlled by PWM signals to six individual wheels which gives a great choice of operation. During straight forward/backward drift, all of the wheels have the same PWM set with a small compensation for unbalanced friction. In turning mode each side is having the opposite sides reciprocal turning direction and is the operated three by three.

The Digital Lobster is also equipped with wheel hub encoders on every wheel which will be a good way to monitor vehicle movement in both translational and rotational direction. To construct a satisfactory model of this turning robot, we have chosen to simplify the robot to a two wheeled robot with a rotational centre right in the middle between the middle wheel pair. On this firm ground we add a slip factor for the remaining wheels which tend to oppose the movement of the modeled behaviour.

Equation 4.2.1 is describing the relation between angle deviation in robot heading θ against wheel rotation. In Figure 4.2, the wheel model and the total six wheeled platform are depicted to clarify the parameter and model parameters. This relation is then used to factorize the needed encoder count for a specific angle of deviation which are controlled by a feedback loop in the Hercules MCU.

$$\theta \cdot \frac{WB}{2} = \frac{counts}{counts/rev} \cdot 2\pi r \rightarrow counts = \frac{\theta \cdot WB \cdot counts/rev}{4\pi r} \tag{4.2.1}$$

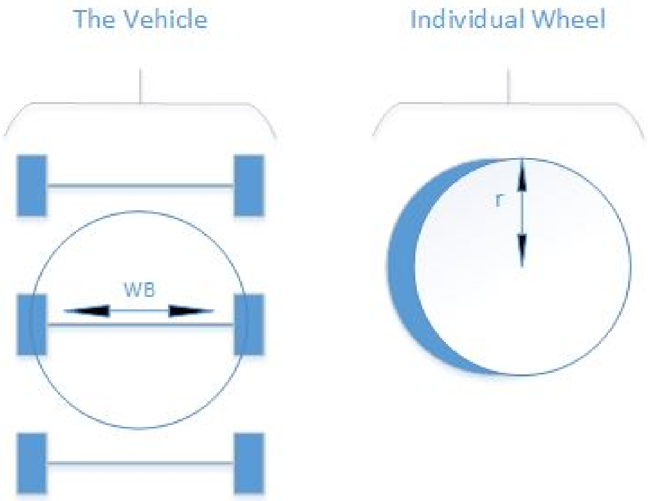


Figure 4.2: Illustration of turning model. Left picture describes rotational centre and the right explains the wheel parameters.

4.2.2 Control

Navigation

The total performance of the Digital Lobster is evaluated in two different steps. The ramifications of the evaluation lead also to two different means of navigation. At first a test navigating through a specified set of checkpoints are programmed to make a simple data set of how good the robot can follow a predetermined track. The coordinates are specified in the physical grid of the Gridmap which then are used for calculating the needed steering headings.

A second approach is carried out in a so called “explorational” context. The robot is performing navigation with the intent of exploring bigger areas of the map in the same time as it performs SLAM, also called active SLAM in literature [16]. This is made by a pragmatic algorithm which sets the heading in the direction which maximize the Raycast algorithm.

This second approach has though only been tested in simulation due to lack of time and not satisfactory behaviour in the first test operating following specified way points.

Motordrivers - Translational/Rotational Motion

As was presented in the introduction to this section, 4.1, six 25D mm DC gear motors with a 34:1 gearbox from Polulu was used[1]. The motor drivers for these motors, Pololu Dual VNH5019 Motor Driver Shield[2], were pre-defined before this master thesis by the engineers at ÅF. Each motor driver could control up to two separate motors, which led to that three motor drivers was needed.

Each motor driver needed four inputs per motor, one to enable the motor output, two to set the direction of rotation and one PWM input for rotational speed, 0-100 percent of maximum

[1] http://www.pololu.com/product/1573

[2] http://www.pololu.com/docs/pdf/0J49/dual_vnh5019_motor_driver_shield.pdf

speed. This meant that the TI Hercules needed 24 free pins just to control the motors. To reduce the number of needed pins it was decided that all motors will be enabled at the same time and that the three motors on the same side of the platform will rotate in the same direction. The simplification reduced the needed number of pins to just 11 and allowed the TI Hercules to also input sensor data without any extra expansion board.

4.2.3 Sensing

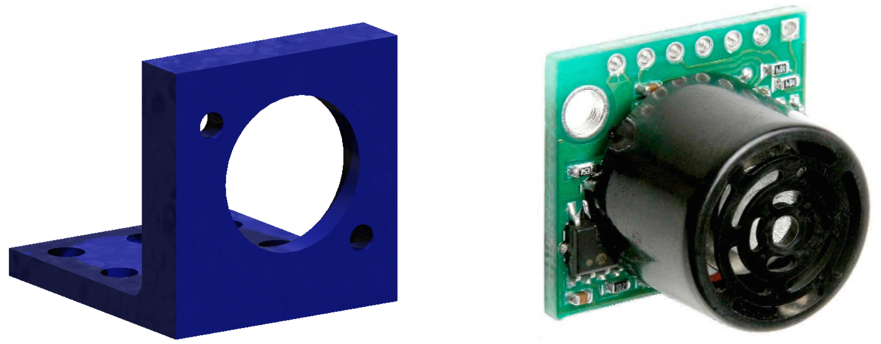
In order for the Digital Lobster to fully know its surroundings, it both has to be able to detect obstacles in the environment as well as measure the distance to the obstacle. In addition to this, it should also be able to know how it self has been moving. Ultrasonic sensors was chosen for range and obstacle detection and hall effect encoders was chosen to measure distance and rotation of the Digital Lobster. Rotation was calculated by the turning model presented in section 4.2.1. Justification for the choice of sensors and technical data of the selected sensors are presented under the headings below.

Ultrasonic Sensors

As for the choice of positioning method cost was a hard requirement that had to be considered when the range measurement hardware was chosen. The hardware was also limited by its size and weight because of the size of the Digital Lobster. A laser scanner was therefore not considered. Even though there exist solutions for a small cost efficient laser scanner [76] the time for building this was considered to long. A vision-based system was not considered because of the higher complexity it requires and the limited time that was available.

The office environment that the Digital Lobster was to navigate in had some glass walls which infrared sensors struggle with. At the same time the environment also consisted of surfaces that ultrasonic sensors struggle with, like sound damping material. Low-cost ultrasonic sensors however, often have longer range distances and wider beam angles than low-cost infrared sensors. A wider beam angle was to lead to a worse angular accuracy, but also allowed a detection of obstacles around the Digital Lobster with fewer sensors. With a good senor model this was considered good enough for the SLAM algorithm and the wide beam angle also complied with the requirements. An ultrasonic sensor was therefore considered to be able to comply with the requirements and at the same time have a low total cost.

The sensors can be mounted on the Digital Lobster by different strategies. Either one or more sensors are mounted on a motor that can sweep 360 degrees in order to scan the environment or more sensors can be mounted fixed to the frame of the platform. By adding a motor another source of error will be added, that will say to fully know the angle of the motor. This could be achieved with a step motor or by a regular DC motor with an encoder. It was however considered that there already was to many uncertainties on the Digital Lobster. It was therefore decided to mount the ultrasonic sensors fixed to the frame with known relative angles. Own developed mounts was developed and printed by the ÅF 3D-printer, see Figure 4.3(a). This allowed very precise angles of the mounts.



(a) The special designed mount for the ultrasonic sensors.

(b) The ultrasonic sensor used, LV-EZ3 from Maxbotix.[1]

Figure 4.3: The special designed mount for the ultrasonic sensors and the chosen sensor from Maxbotix. Two versions of the mount designed, one pointing zero degrees from the base and one pointing 45 degrees from the base, the hole structure is the difference.

A total of six sensors was decided to be used. The most important direction to scan is the driving direction of the Digital Lobster. To cover this area three sensors was decided to be enough, because of their wide beam angle and to avoid crosstalk. In order for the Digital Lobster to scan 360 degrees around it self one sensor was also mounted on each side as well as one sensor in the back, see Figure 4.4. In section 5.2.4 it is discussed if this solution was good enough complying with the requirements.

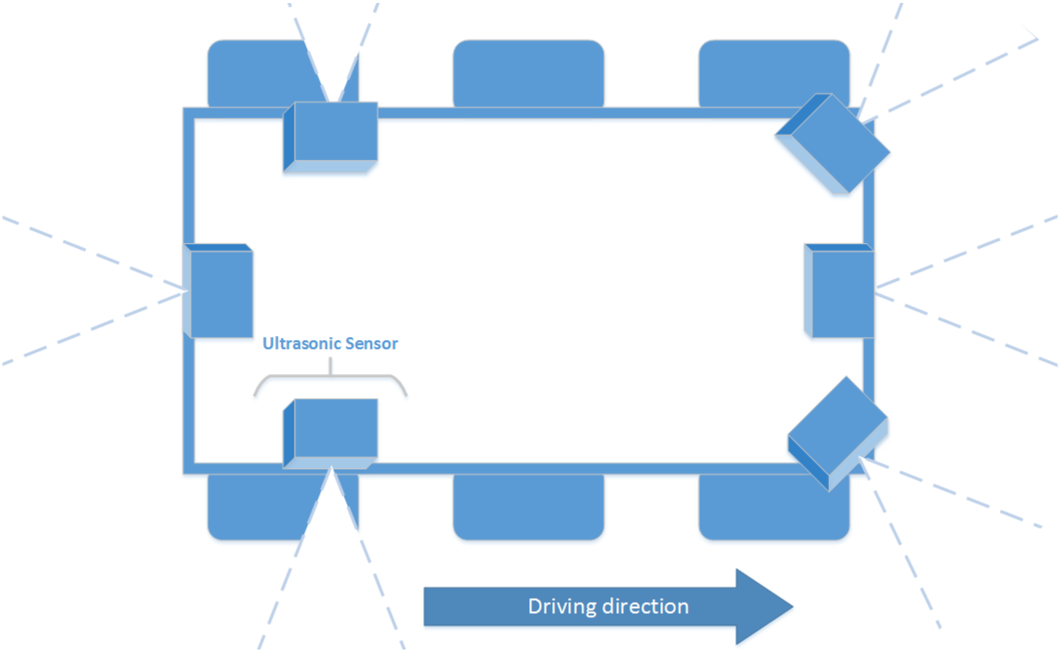


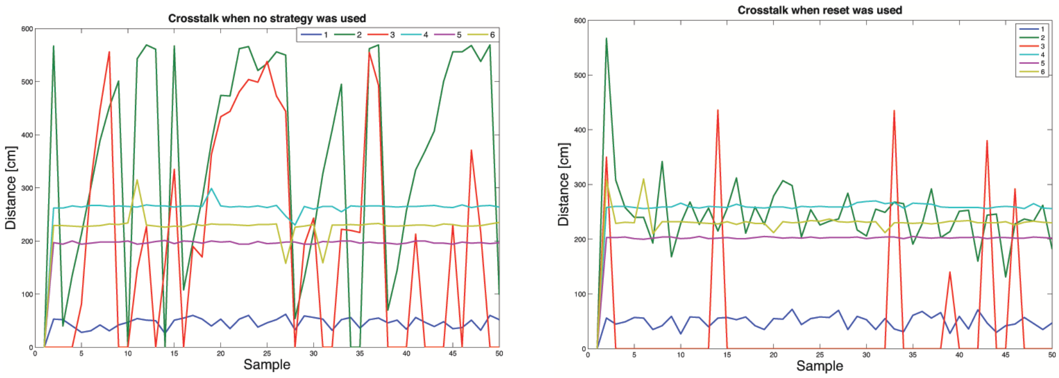
Figure 4.4: Placement of the six ultrasonic sensors.

[1] Source: http://maxbotix.com/documents/MB1030_Datasheet.pdf

There exist different kind of low-cost ultrasonic sensors on the market. Some are built with the microphone and speaker separated, when some have both integrated in the same unit. A sensor with just one unit will require a smaller space. The **LV-EZ3** ultrasonic range finder, Figure 4.3(b), from **Maxbotix**[1] was therefore chosen. It is a small sensor that fulfills the requirements, see section 3.1. Because of its self calibrating feature, its many different outputs and its well documented beam pattern it was chosen before other small low-cost sensors that also fulfilled the requirements.In this thesis the analog output was chosen to be used, because of its simplicity and the number of ADC ports available on the TI Hercules platform.

As was explained in section 2.3.2 crosstalk will most likely be a problem when more than one ultrasonic sensor is used. The crosstalk will be caused by that some sound waves will leak to the other sensors in the network, but also by that the sensors is not synced. That is, the sensors will not perfectly have the same timing and transmit a signal and start listening on the reflected signal at the same time. This will lead to that a sensor can start listening when the reflected signal from another sensor is arriving. The time from when the sensor was transmitting its own signal until it receives another sensors signal will be shorter than when it should had received its own signal and the distance will be miss interpreted as a shorter distance.

Because of that the beam width was pretty narrow and the sensors was placed in such a way that no beam will interfere with another beam it was believed that no or a very small amount of cross talk will occur. However, that was not the case. As can be seen in Figure 4.5(a) the raw measurements from the sensors with no firing strategy was very fluctuating and not good.



(a) No strategy used.

(b) Re-synchronization used.

Figure 4.5: Raw measurements from the ultrasonic sensors when no strategy was used and when all sensors was re-synchronized before every measurement.

An attempt was done to re-synchronize the sensors at each measurement. That is, the sensors was restarted at each time and re-calibrated the timing. This lowered the measurement frequency but it was still possible to achieve a frequency above 10 Hz as was specified in the requirements. As can be seen in Figure 4.5(b) the raw measurements is still fluctuating, but not as much as when no strategy was used. However it was considered not to be good enough for the implementation.

[1] Datasheet for Maxbotix LV-EZ3: `http://maxbotix.com/documents/MB1030_Datasheet.pdf`

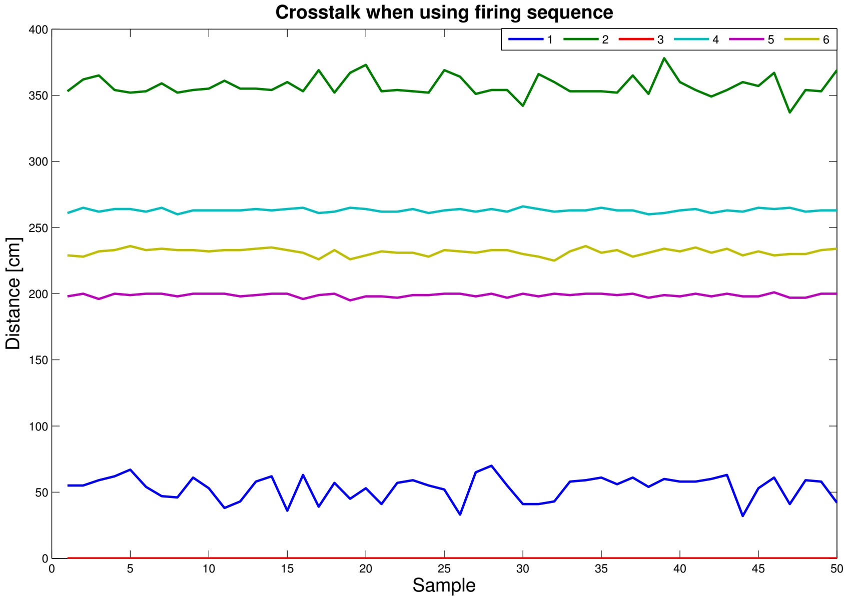


Figure 4.6: The raw measurement from the ultrasonic sensors when each and every sensor was fired separately.

A beam firing strategy where each and every sensor was fired on its own was therefore chosen. All other sensors was shutdown except the one that was currently measuring. Because of the calibrating period of approximately 50 ms the sensors was doing on start-up a measurement frequency of at least 10 Hz was no longer possible. It was however considered that a lower measurement frequency was allowed, in favor of a better measurement result. In Figure 4.6 it can be seen that the raw measurements when firing every sensor by it self was hugely improved from when just running the sensors freely without any firing strategy.

Wheel Encoders

The Wild Thumper[1] platform from Polulu, which is the platform the Digital Lobster is built on, was originally delivered without any wheel encoders. As has been mentioned, encoders are very important and one of the easiest ways to measure the distance traveled. Among the different possible modifications of the platform that was studied was to put an optic encoder with a dashed plate on the wheels. This was however considered to be to bulky and sensitive against ambient light. A hall effect sensor solution was therefore considered. It was found that Polulu had made modifications to the motor and put a small hall effect sensor on. To buy new motors with hall effect sensors and then just moving the gearbox to the new motors was decided to be the best and quickest solution.

The package used was a **high-power motor with 48 CPR encoder**[2]. The encoder was a quadrature hall effect encoder with 48 counts per revolution. Counts per revolution on the output shaft from the added gearbox, 34:1, that was connected to the wheels was therefore 1632. The

[1] `http://www.pololu.com/product/1563`

[2] `http://www.pololu.com/product/2269`

stakeholder ÅF wanted to only use the Xilinx ZeedBoard, without the added TI Hercules board in the future. A deficit of pins will then occur and it was therefore decided together with ÅF to only use one channel from the encoders and only signals from two wheels. Signals from the middle wheel encoders was decided to be used, because they will provide best information for the turning model, see section 4.2.1.

A counter was implemented on the FPGA part of the Xilinx ZedBoard in collaboration with one engineer at ÅF. It was considered to be the easiest solution to count higher numbers of ticks and send the information to the TI Hercules by a small number of pins. Regular counters usually sends the information by a parallel port which require a pin number corresponding to the resolution, for examples 8 pins for 8-bit resolution. However, by using the FPGA an own designed counter with 16-bit resolution, that sends the information by a serial port could be used. The SPI protocol was decided to be the best suited for the task. The protocol is robust, have a medium speed and was easy to implement on the TI Hercules platform, which already had dedicate peripheral ports and drivers for SPI communication.

In a discussion with the engineer it was decided that the easiest and least time consuming implementation for the counter was to only count the rising edge from the encoders. This led to that only 408 counts per wheel revolution could be used. It was however, considered to be enough to “measure distance travelled with an accuracy of at least ± 1 cm”, which was one of the requirements, section 3.1.

4.2.4 Communication protocol layers

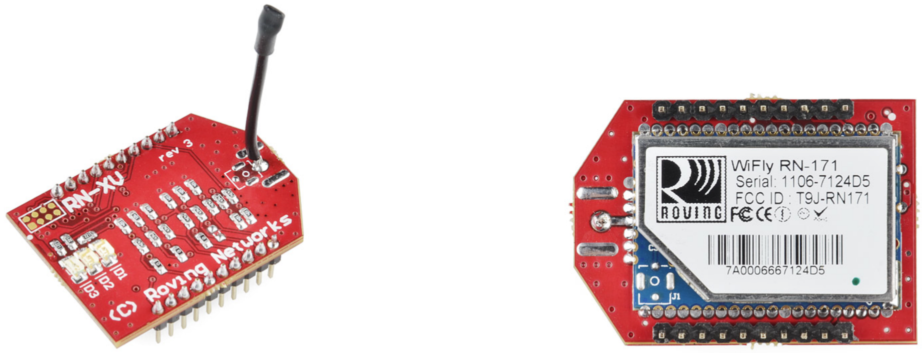
Two main communication protocols were used, as have been mentioned earlier, namely WiFi and SPI. However, for those protocol, own developed layers have been designed and implemented. This had to be done in order for the nodes to know what message to send and what to respond in the respective protocol.

WiFi

For the IPS implementation as well as the communication to the host computer a WiFi protocol was considered to be the best solution. The protocol is well established and worked with MATLAB, which was running the calculations on the host computer.

A breakout board from Sparkfun was chosen, RN-XV WiFly, see Figure 4.7(a), for the implementation on the Digital Lobster. The breakout board was based on the RN-171 WiFi module from Roving Networks, see Figure 4.7(b). It was chosen based on its low price, simplicity, small all in one package and possibility to send serial data through the TCP/IP protocol. The wire antenna the breakout board used was considered to be a potential problem for the IPS due to its presumed issue with different gain in different incident angles[1]. It was however, still chosen in favor for an external more bulky antenna.

[1] This assumption was drawn together with an antenna expert at ÅF



(a) The breakout board from Sparkfun with a wire antenna.

(b) The WiFi module from Roving Networks the breakout board used.

Figure 4.7: Two images of the breakout board from Sparkfun.[1]

On top of the TCP/IP protocol a protocol that defined a wireless serial master and slave relationship as well as messages was designed and implemented. Even though the WiFly module was decided to be the server side in the TCP/IP protocol, it was considered that the host PC was best suited for the master role. That's because all calculations and decisions was made by MATLAB algorithms running on the host PC. The WiFly module together with the TI Hercules therefore had a slave role and waited for messages, before sending anything back to the host PC. Messages sent from the host PC and responses from the WiFly and TI Hercules can be found in table 4.1. Each message was terminated by a carriage return, <CR>, if nothing else is stated.

Table 4.1: Communication protocol host PC as master and WiFly together with TI Hercules as slave. In the data columns each [ ] is considered as one byte

Master message		*Slave response*	
Identifier	**Data**	**Identifier**	**Data**
SendData	N/A	S	[MS][LS] per sensor, left then right wheel followed by ultrasonic sensors one to six
Ctrl	[speed], [MS], [LS] turn ticks	N/A	N/A
Dist	[MS] [LS] distance in ticks	N/A	N/A
N/A	N/A	Ready[2]	N/A
ScanNow	N/A	N/A	17 byte MAC address + one byte per RSSI sample.[3][4]
ScanFind	N/A	N/A	17 byte MAC address + [RSSI value][3]
N/A	N/A	ERR[5]	N/A

[1] Source: `https://www.sparkfun.com/products/10822`

[2] Sends "Ready" when reached the distance defined by the host PC commando "Dist" or when turned the defined ticks by the "Ctrl" commando, but only if a "Ctrl" commando was sent by the host PC.

[3] Number off RSSI samples pre-defined in the source code.

[4] One row for every unique AP found, each row terminated by a <CR>. The end of the transmission is terminated by "END<CR>".

[5] Sent when the slave didn't recognize the message identifier.

SPI

In order to send the counted values from each encoder the serial protocol SPI was considered the best, as have been mention in section 4.2.3 the SPI protocol is robust and easy to use. The TI Hercules will act as a connection between the algorithms in MATLAB and the encoders, it will also handle the scheduling, section 4.2.5, and collect all other necessary data for the SLAM and IPS. It was therefore early decided that the TI Hercules should act as the master and the FPGA counter as the slave, according to the SPI protocol.

Each of the encoders will do 408 counts per wheel revolution. In order to reduce the needed read frequency for the TI Hercules a 8-bit counter and a 8-bit message length will not be enough. The TI Hercules SPI peripheral handles a message length up to 16-bit. It was therefore decided to use this message length for the application in order to reduce the needed read frequency as much as possible. The longer transmission time needed to send 16-bit was considered neglectable compared to the time for the SLAM algorithm to execute.

On top of the standard SPI protocol, a layer for the message structure was designed in collaboration with an engineer at ÅF. Because of the fully duplex nature of SPI the master has to send data in order for the slave to send data back. In the application the master sent three dummy messages, the first to trigger the slave to store the counted values in the SPI buffer and also clear the counters. The following two kept the SPI clock running for the slave to send the data.

Data sent by the slave also consisted of three messages. The first was the dummy data sent by the first message from the master. This was done as a fault check, to know that the received data was not corrupt. The following two messages contained the counted values for the left and right wheel respectively.

4.2.5 Scheduling

A Real-Time Operating System, RTOS, was considered to be a good solution to control timing of different tasks the TI Hercules was to perform. With a RTOS some tasks can be given a lower priority in order to be sure more important tasks are performed when they shall. However, it is still important to plan the scheduling thoroughly to know what is going to happen and to not get any tasks blocking the whole execution.

Free-RTOS[1] is a prioritized preemptive scheduler and was chosen based on that it was free, supported and integrated in the TI Hercules software, HALCoGen[2]. Because of its scheduling nature it has no concept of a deadline and is used in soft real-time systems. That means, it will not know if a task has overrun its deadline and blocking other tasks. This makes it more important for the user to plan the schedule and task priorities in order to not get a high priority task run for ever and block other tasks as have been mentioned.

The developed system was considered a soft real-time system, because of the low level of harm it can cause the surrounding environment if it is missing the deadlines of its tasks. A missed deadline will not lead to any total system failures, but more that the system will maybe not act as intended. More important for the system is to be able to prioritize tasks so for example communication always works. The Free-RTOS was considered to solve this demand in a good way and at the same time be easy to incorporate in the source code for the Ti Hercules.

[1] http://www.freertos.org

[2] http://www.ti.com/tool/halcogen

Main tasks for the TI Hercules to perform was to read the sensor data, both all six ultrasonic sensors and the two wheel encoders, control the six motor drivers and to send as well as receive data through WiFi. These tasks were therefore chosen to be the different tasks in the scheduler. A scheduling task for the wheel encoder reading was however not used. Instead the task was incorporated in the motor controlling scheduling task. Because this task maintained the control of how far the platform had traveled in order to know when to stop and needed to read the encoder values either way.

Table 4.2: Tasks used for scheduling together with priorities, periodicity, task execution time and the risk for a deadline overrun.

Task	**Priority**	**Periodicity**	**Execution time**	**Risk for**
		[ms]	**[ms]**	**deadline overrun**
WiFi Send Data	4	N/A[1]	2	N
WiFi Receive Data	3	40	<1/355[2]	Y
Motor Control	4	30	1	N
Read US[3] sensors	2	350	350	Y
Drive around[4]	1	200	<1	N

In Table 4.2 all the different tasks with their respective priorities, periodicity, execution time and risk for a deadline overrun is displayed. As can be seen two of the tasks were having a potential overrun risk, the “Read Ultrasonic Sensor” task almost all time and the “WiFi Receive Data” task just some times. The task for reading the ultrasonic sensors was however just used for emergency stops if the platform was about to hit an obstacle that had appeared between the SLAM calculations. It was therefore not a critical task and it was considered to be enough if it was just running all the time and interrupted by the more important tasks for the SLAM algorithm, therefore the lower priority.

The WiFi Receive task had different execution times depending on the received message. The longest time was dependent on the ultrasonic sensor read, each sensor took 50 ms to read plus an extra 49 ms restart time. That was done when the SLAM algorithm in MATLAB requested new measured values. The algorithm therefore known the time delay and could take account for it. Other commandos sent to the platform took just around one millisecond to execute. The periodicity of the WiFi Receive task was still set to 40 millisecond because of the need to check for new messages. If no new messages had been received through WiFi, the task was halted and put to blocked mode, that’s why the shortest execution time could be seen as smaller than one millisecond.

Motor Controller task was chosen to have the same high priority as the WiFi Send task. It was considered as important to send data through WiFi as for the platform to know when to stop in order to have a good performance of the SLAM algorithm. The WiFi Send task was also not executing if any other task hadn’t put something in its queue, which was mainly done by the WiFi Receive task and the Motor Control task. Execution time for the motor control task was about one millisecond and the WiFi send task was therefore not being delayed by more time than the SLAM algorithm could take account for.

[1] Used a queue, so was only executed if another task put something in that queue.

[2] Depends on the received message.

[3] Ultrasonic.

[4] Only used to test some functions, not used in the finalized algorithms.

One may also see that there is no task for indoor positioning readings. That's because a decision of not integrate SLAM and IPS due to limited amount of time and bad performance of the both algorithms. See section 3.2.3 and 5.2 for a deeper explanation. IPS readings were instead only done by a state machine explained in section 4.2.7, that will say the state machine was not executed by the RTOS. However, if IPS and SLAM should have been integrated the used state machine should have been executed in its own task in the RTOS.

4.2.6 SLAM

The detailed operation of the SLAM compounds are described in the previous chapter 3.2.1. Here will the operation of the empirically implemented algorithm be presented. In Figure 4.8 the operation and execution is presented in a flowchart. Details are missing as they have been discussed in the previous chapters.

The execution is started on a call from MATLAB triggering the Digital Lobster to navigate for the next way-point in its memory. A decided amount of distance to travel is initiated at start which will be the distance traveled during all loops. In the case of turning, the turn is executed, but no distance is traveled before the filter re-calculates the pose. This is a little more time consuming, but improves the precision.

After either a turn or a drive has been performed the data is sent wireless to MATLAB executing the prediction step, the update step, maps the environment for every particle and finally resamples the particle set. The operation has to be executed in real time and the robot can't be moved again before sensor data has been sent to MATLAB. Also because of the steering is calculated during every step in the loop, the steering can't be calculated again before the SLAM algorithm has been fully executed. The best position is then represented by the particle with the highest weight.

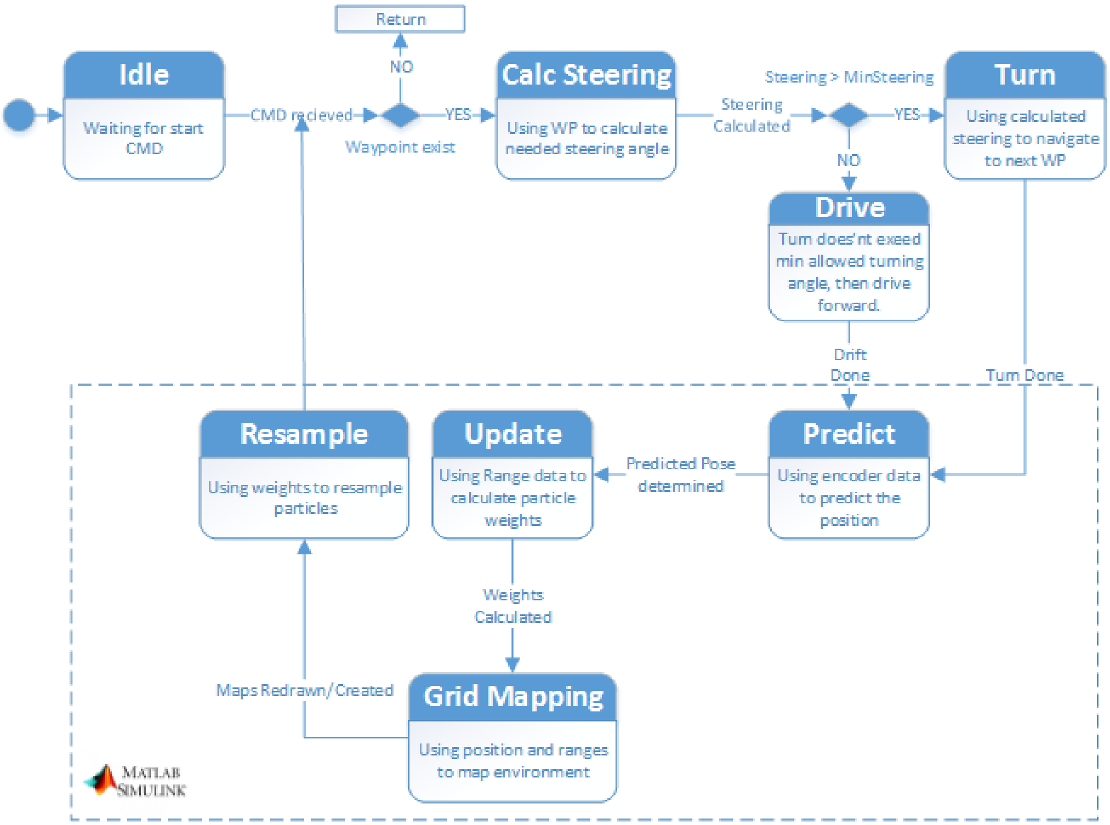

Return
NO
Idle
Waiting for start CMD
CMD recieved
YES
Waypoint exist
Calc Steering
Using WP to calculate needed steering angle
Steering Calculated
Steering > MinSteering
YES
Turn
Using calculated steering to navigate to next WP
NO
Drive
Turn does'nt exeed min allowed turning angle, then drive forward.
Drift Done
Turn Done
Resample
Using weights to resample particles
Update
Using Range data to calculate particle weights
Predicted Pose determined
Predict
Using encoder data to predict the position
Weights Calculated
Maps Redrawn/Created
Grid Mapping
Using position and ranges to map environment
MATLAB SIMULINK

Figure 4.8: The SLAM algorithm flowchart.

4.2.7 IPS

In section 3.2.2 an explanation of the proposed IPS method can be found, any deeper explanation of the different equations will not be done in this section. This part will instead cover the implementation of the method with pseudo code and state machine charts.

The method was divided in two modules, first the major algorithm running in MATLAB and second a RSSI scanning state machine employed on the TI Hercules. The decision was based on the calculation power in MATLAB as well as the possibility to store a large Radio-Map thanks to the memory capacity in a personal computer. However, the RSSI measurements still needs to be performed at the platforms location and was therefore needed to be executed by the TI Hercules.

Implemented algorithm

In MATLAB the two major phase algorithms were employed, namely the offline Radio-Mapping phase and the online positioning phase, as have been explained earlier in section 3.2.2. The main function for the offline phase was called “CreateRadioMap”, and a pseudo code can be found in Table 4.3.

This function handle everything needed to build the Radio-Map, from connecting to the TI

Hercules through WiFi to sort and store all calculated variables connected to each AP and location. As have been mentioned, the only thing it didn't do was to take the RSSI measurements.

An object, *TcpIpObj*, to store all settings was needed for the TCP/IP communications had to be used in all communications between MATLAB and the TI Hercules. This object also maintained the TCP/IP stack. So before any algorithm could proceed with its main task it needed to create this object and connect to the TI Hercules and WiFly module, section 4.2.4.

The Radio-Mapping algorithm then was running until the user told it to stop. This was basically only done when the mapping was finished and all positioned had been visited. When a new position had been reach the scanning commando was sent to the TI Hercules, which then sent the result back to MATLAB for processing. If there had been any error, nothing was saved in the Radio-Map and the user was asked to redo the last scan. Otherwise the user was asked to input the positions *x*- and *y*-coordinate to be stored in the Radio-Map.

The scanning result was processed by a function and passed as a row vector to the main algorithm, where each row represented a unique AP with mac address and samples. For each row, or unique AP, Weibull parameters were calculated and stored in the Radio-Map, as well as the MAC address for that AP.

When the user called for a "STOP", the algorithm was terminated and the current Radio-Map was saved to a file name set by the user. The Radio-Map file was later used by the positioning algorithm, see Table 4.4.

Table 4.3: The principal operation of the offline Radio-Maping function.

```
CreateRadioMap()
    $TcpIpObj = connectTcpIp()$
    while [ $userStop == FALSE$]
      $scanResult = scanAndSort(TcpIpObj)$
      if [ $noError(scanResult)$ ]
        $radioMap(row).position$ ="User Input [x,y]"
        for [ all rows in $scanResult$ do]
          $radioMap(row).ap.parameters = calcWeibullParameters(scanResult)$
          $radioMap(row).ap.mac = getMacAddress(scanResult)$
        end
      end
      $row++$
      if [ "User input $STOP$" ]
        $userStop = TRUE$
      end
    end
    save $radioMap$
```

Unlike the offline phase function that by it self connected to the TI Hercules, the online phase function, “wifi_IPS”, needed a TCP/IP object passed to it. The decision to do like that was based on that the function should be incorporated in a bigger algorithm performing both SLAM and IPS in the future. In that case, the bigger algorithm will handle the connection and just call for the positioning function when needed.

After the Radio-Map had been loaded, the positioning function sent a scanning commando to the TI Hercules. The received data was processed and returned as a row vector to the positioning function. Unlike the vector returned to the Radio-Mapping function, the returned vector only consisted of one sampled value per unique AP.

Each row in the Radio-Map was associated with a position. So the row with maximum probability calculated by the Weibull Cumulate Distribution Function, CDF, was considered to be the estimated position. For each row, each found APs MAC address was then compared to the stored MAC addresses in order to find the correct Weibull-parameters to use. The parameters together with the scanned RSSI value for that AP was used in the Weibull CDF. All probabilities together with the positions were also stored to be used in the weighting function described in section 4.2.4.

Table 4.4: The principal operation of the online positioning function.

```
**wifi_IPS**($TcpIpObj$)
    **load** $radioMap$
    $scanResult = scanAndSort(TcpIpObj)$
    **for** [ all rows in $radioMap$ do]
        **for** [ all rows in $scanResult$ do ]
            **for** [ all APs in $radioMap(row)$ do ]
                **if** [ $macIsEqual(radioMap(row).ap.mac, scanResult)$ ]
                    $currentRowProb$ *= $weibullCDF(radioMap(row).ap.parameters, scanResult)$
                **end**
            **end**
        **end**
        $allRowProb(row).position = radioMap(row).position$
        $allRowProb(row).probability = currentRowProb$
        **if** [ $currentRowProb > maxProb$ ]
            $maxProb = currentRowProb$
            $estimatedPosition = radioMap(row).position$
        **end**
    **end**
    **return** $weightPosition(estimatedPostion, allRowProb)$
```

RSSI sampling

For the implementation and RSSI sampling on the TI Hercules, a different approach was taken. No "regular" function or algorithm was used. The decision was based on the handling of the WiFly module, that needed certain steps in order to enter a command mode, meassure RSSI values and then exit the command mode. For this a state machine was considered the best solution.

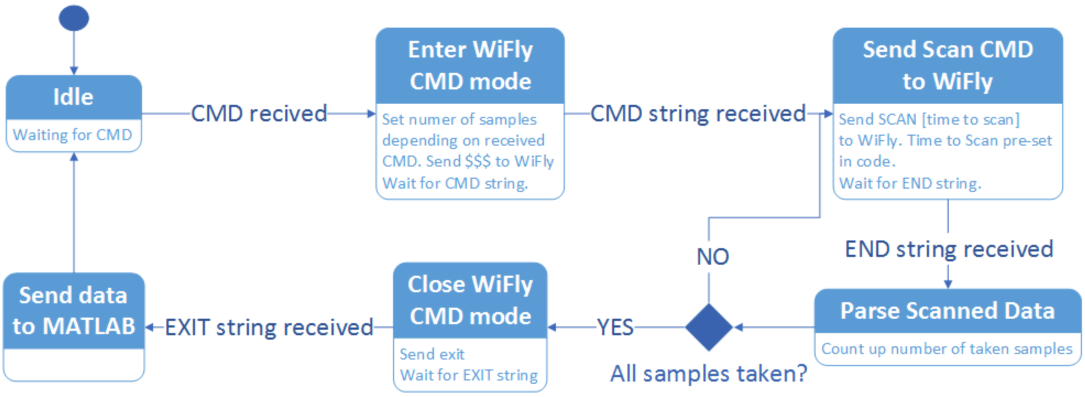

Idle
Waiting for CMD
CMD recived
Enter WiFly CMD mode
Set numer of samples depending on received CMD. Send $$$ to WiFly Wait for CMD string.
CMD string received
Send Scan CMD to WiFly
Send SCAN [time to scan] to WiFly. Time to Scan pre-set in code. Wait for END string.
END string received
Parse Scanned Data
Count up number of taken samples
NO
YES
All samples taken?
Close WiFly CMD mode
Send exit Wait for EXIT string
EXIT string received
Send data to MATLAB

Figure 4.9: State machine used by TI Hercules to take RSSI samples.

In Figure 4.9 the designed state machine can be seen. First it was in an idle mode, waiting for a commando from MATLAB. When the commando was received it evaluate the commando and set the amount of samples to be taken from each AP. There was basically two different cases, if the Radio-Mapping function sent the scanning commando the number of samples was set to a pre-set value between 20 to 30 and if it was the positioning function that sent the commando the sample value was set to one.

Next the WiFly module had to be switched to commando mode. When this had been done the module sent an acknowledge string telling the state machine to proceed. The TI Hercules sent the scanning commando in order to initiate the scan function in the module. Each finalized scan was ended by the string "END" and told the state machine to process the data.

The scanning commando and data processing was done the set amount of samples. When all sample was taken the state machine closed the WiFly command mode. While in command mode the WiFly module couldn't communicate through TCP/IP so it was highly important to safely exit this mode. The data was sent back to MATLAB for use in the IPS algorithms and the TI Hercules was put to idle.

4.3 Testing procedures

All testing was conducted at floor seven in the ÅF head office building. This was a good office environment with people in movement as well as a good number of obstacles, like chairs and tables. The floor was divided in four different rectangles, called maps, for the testing of SLAM, IPS and the models. These maps and part of floor seven can be found in Figure 4.10. Approximate sizes of the four different maps can be found in Table 4.5.

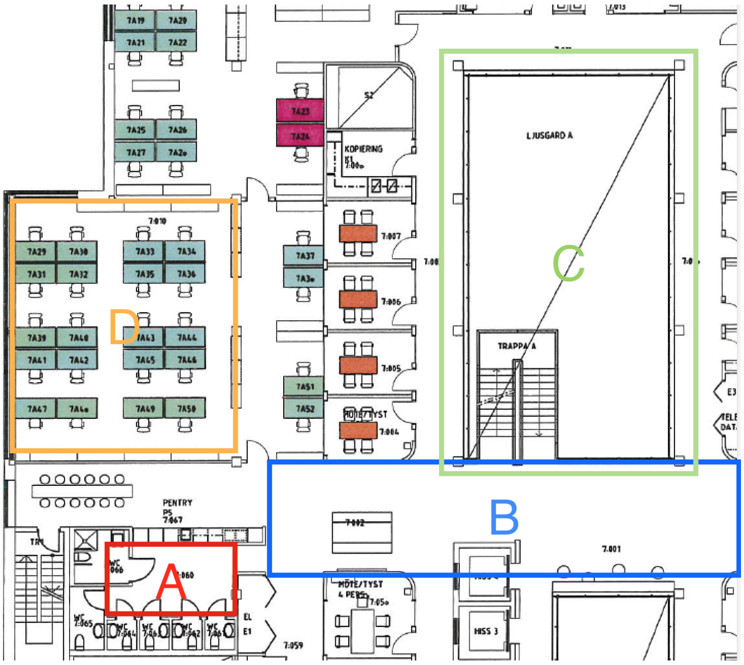


Figure 4.10: Drawing of floor 7 at ÅF head office with the four different maps marked.

When testing models the platform was placed on different surfaces to see how the slip affected the performance. In map *D* a carpet covered the whole floor and in the rest of the maps the floor was made of oiled wood. The models were therefore tested in a small part of map *D* and *B*.

SLAM testing was performed in map *A* because of the small numbers of obstacles and plain surface walls. Because of the limited amount of time in the project, no testing with more obstacles and more advanced maps were performed, due to that a more time consuming tuning would have been needed.

For IPS testing, maps *B* and *C* were used. Map *C* had a atrium in the middle, so testing could only be performed around the edges of the map. This map was therefore only used to see how the IPS performed without Radio-Map points in the middle.

Table 4.5: Approximate sizes of the different maps used to test SLAM, IPS and the models.

	A	B	C	D
Length [m]	4.6	16	8.5	7.5
Width [m]	2.7	4	12.5	8.5

4.3.1 Testing Models

To get a good approximation of the uncertainties in translational and rotational movement, both of the maneuvers have to be thoroughly tested. The factors incorporated into the system like wheel slip, wheel suspension characteristics, applied torque to the wheels and encoder errors make it hard to calculate a model depending on the encoder counts. The model is determined to be very non linear and would have been to complex to analytically calculate. System identification using MATLAB to calculate a continuous black-box model of the transfer function was also considered, but decided to be very time consuming.
Iterative testing constructing a correction, non-linear table for allowed turns (also seen as a black-box model) was chosen to be the only practical method performable.

Driving Model

By the use of the driving model presented in chapter 4.2.1, the drift was measured by sending different control signals to the robot. The models were at first tuned to in the best way represent the input control signal. Secondly the variance of the distance traveled was measured using a set of measurements. An approximation was made and the variance was approximated to ±0.05 m with a traveled distance up to two meters.

Turning model

The turning model was also used as described in chapter 4.2.1. The turning model does though rely on the slip between wheels and the floor. The operational environment has luckily approximately the same friction factor which the models have been tuned and tested against. After the tuning, different factors as battery charge, applied torque and timing between control and feedback resulted in a big variation of the turn.
In the end a variation was estimated to ±13 degrees at most, using good battery status.

A distinct bias was also identified. The battery charge level was strongly affecting the torque applied to the wheels by the motors which affected the slip of the wheel. This made the model very dependent of the battery which haven't been analysed further. The operation was thereby very limited to good battery operation.

4.3.2 Testing of SLAM

SLAM was initially tested in simulation environment where sensor process and measurement models were validated and also the correct execution and behaviour of the geometric characteristics of the SLAM algorithm. When the algorithm behaviour was satisfactory in simulation the testing on the real platform was initiated. The testing of the SLAM algorithm in real world operation was based a lot of imitating the simulated operation, but with the difference of using real sensor feedback. The operation was then tested in area A in Figure 4.10 with all doors closed constructing an asymmetric room with few obstacles other than walls.

The algorithm for SLAM did run on a hosting server where every calculations were performed and commands of operation was sent wireless to the Hercules MCU controlling the Digital lobster.

The major task of the real operation differing from the simulation was to verify that the derived model for driving/turning and sensing corresponded to the analytically derived sensor model and the tested driving/turning model.

The sensor model described in section 2.2.5 did also needed tuning to get the best probability distribution corresponding the measurements. Repeated firing testing of the ultrasonic sensor and sensor data was used to tune the sensor model.

4.3.3 Testing of IPS

A good testing ground for the IPS was, as have been mentioned, the seventh floor in the head office bulding for ÅF. The office building had many APs already deployed to be used in the positioning testing. People and obstacles that usually can be found in an regular office environment could also be found on the seventh floor.

All testing in all maps were done in the same way. A laptop was connected to the TI Hercules for power supply. MATLAB algorithms were however still comunicating through WiFi to the TI Hercules. The setup was placed on a bar stool with the WiFlys antenna always pointed towards the known APs, whenever these was used.

When a measurement was started by the user, who typed the actual relative location, the user moved away so only natural disturbances were affecting the result. All measurement was taken with a space of one meter in the respective map during the Radio-Mapping phase. For the estimation results not all mapped positions were used due to limited amount of time. However, 15 estimations were done in order to build a mean euclidean error at each estimated position.

Map C was mainly chosen for its open areas without objects and less people was moving in the area. A drawback was that no measurements could be taken in the middle of the area, because of the big atrium. This however was seen in a positive way for testing how the method will perform with only a frame of reference points, because that is how different parts of a regular office environment can look like. It could be difficult to take measurements in every spot due to obstacles or, like in this case, an atrium.

A concern was that some measurements could be wrongly mirrored due to the lack of reference points. To lower the risk of mirroring a decision was taken to use a mean method instead of the product method. That is, instead of multiplying the probabilities for each position during estimation, a mean value of the probabilities was calculated. The position in the Radio-Map with the highest mean probability was considered the estimated position.

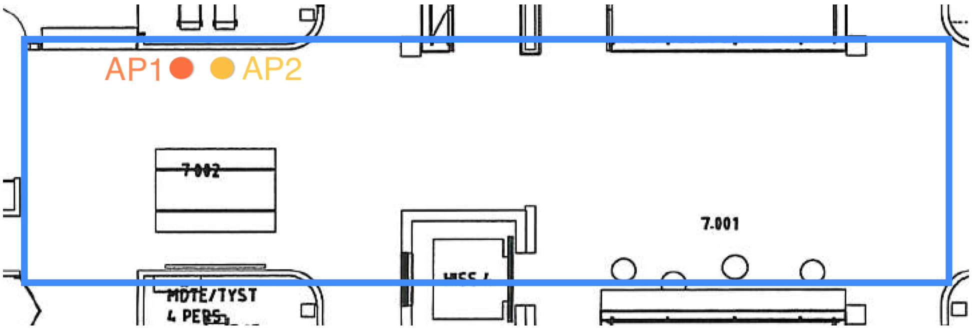


Figure 4.11: The placement of the two “known” APs.

For the rest of the testing, map B was considered the perfect area. In this area the main elevators as well as entries to the toilets were located. One could also find the coffee makers for the whole floor, which made the area to a natural meeting point for the employees. The elevators and some walls together with the peoples in movement created a perfect office environment to test the IPS. However, due to the time delay in the project, testing was conducted near midsummer eve. This meant that fewer people was working, but was considered to still be enough for a good result.

With the availability to measure reference points in almost the entire area the proposed method could be tested. However, due to some furniture and the elevators, no reference points were decided to be taken along the bottom wall, from the far left to the elevator. This meant that the area for testing was only three meters in width, during a contraction existing along nine meters of corridor.

The same method used in map C, mean value estimation, was also tested in map B to be able to compare the result. Not only to the test conducted in map C, but also how well it performed compared to the proposed method.

Two different setups were used to test the proposed method. All available APs in the building were used in the first, as was done in the mean estimation tests. This allowed a comparison between the different methods. Secondly, only two “known” APs were used, as had been done in earlier tests by other authors, to be able to compare the result against the simulations. The placement of these two can be seen in Figure 4.11. No more “known” APs could be used during testing due to limited amount of available APs.

The same Radio-Map with records of all APs was however used to reduce the time needed for the test. At estimation, the algorithm only took account for measurements from the two APs. If one or both of these two APs were not found during Radio-Mapping it could result in unwanted errors. This was considered a permissible risk due to the limited amount of time available for testing. High level of difference in a small number of positions in the result could also be neglected as a known source of error.

Chapter 5

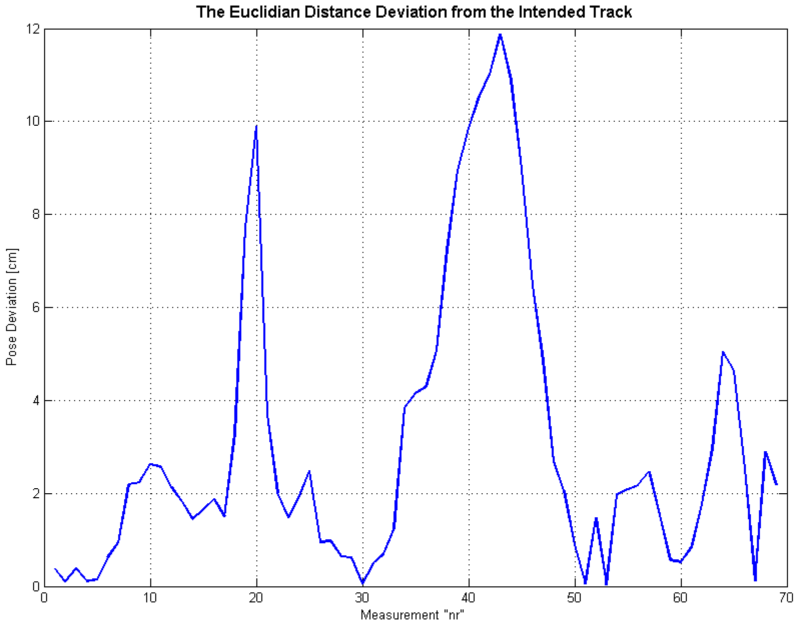
Results and Discussion

5.1 Results

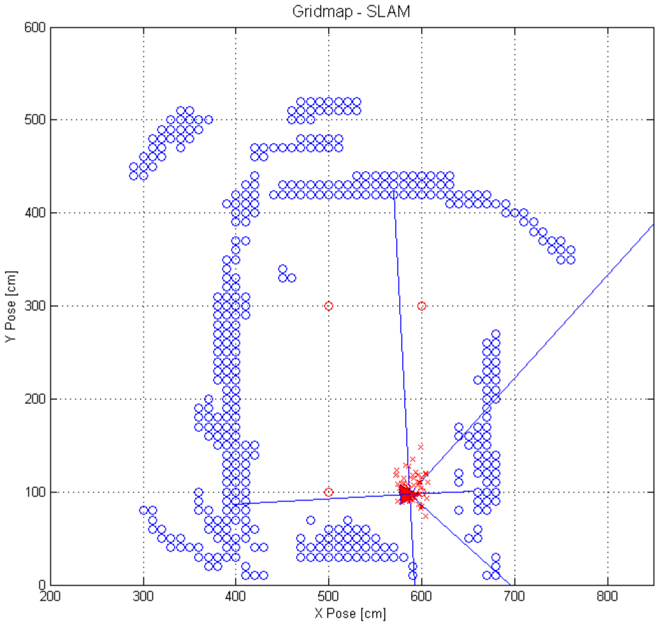
The section is divided in two subsections, one for each algorithm. It will present the result from the performed tests described in section 4.3. No test or implementations was done for the integration, called WiFi SLAM, due to time limitation as have been mentioned earlier. Therefore, no result exists for WiFi SLAM. For discussion about the result see the following section 5.2.

5.1.1 SLAM

The testing of the SLAM algorithm has for good operation been limited to operation with a fully charged battery and only for turning angles carefully modelled in advance to get the correct angle deviation measured corresponding to the actual movement. During such conditions and in a simple environment, test runs have been performed with converging algorithms. The result of the best run can be seen in Figure 5.1 where it diverges at most $12cm$ from the perfect track. Although some errors can be seen in the upper right corner of the map corresponding to the big uncertainty in the turning feedback and crosstalking sonar beams.



(a) The Euclidian distance from the track intended, represented for every measurement taken.



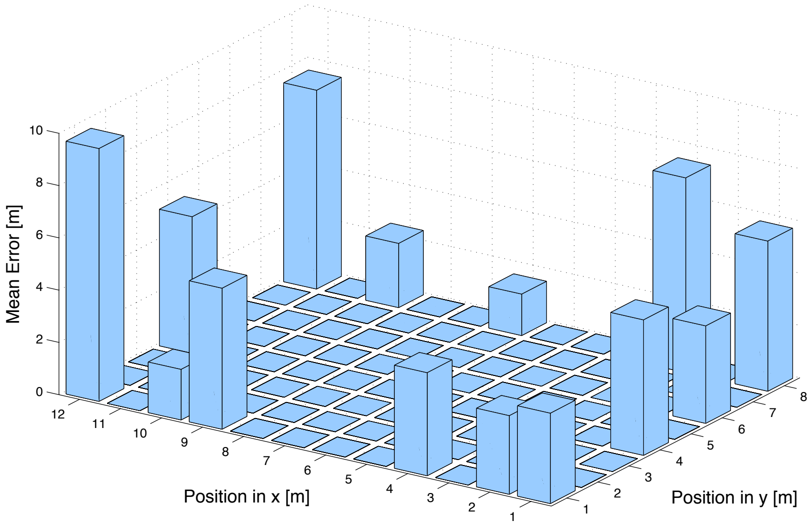
(b) The Gridmap plot corresponding to the Euclidian Distance

Figure 5.1: SLAM maped up in a symetric environment

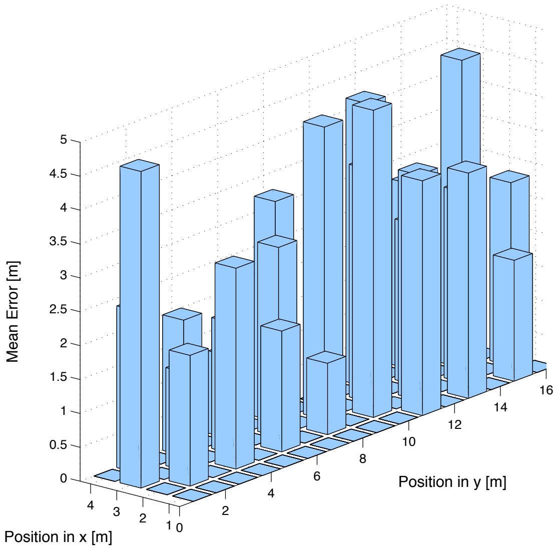
5.1.2 IPS

Four different test cases were performed to test the performance of the IPS method used. In section 4.3.3 these cases are described.

A total mean euclidean error of 4.74 meters could be achieved in case one where map C was used and the mean probability estimation method. When using the same method but in map B, with measurements points in the middle of the map, a mean euclidean error of 2.80 meters was achieved. The mean euclidean error at each measured point can be found in Figure 5.2.



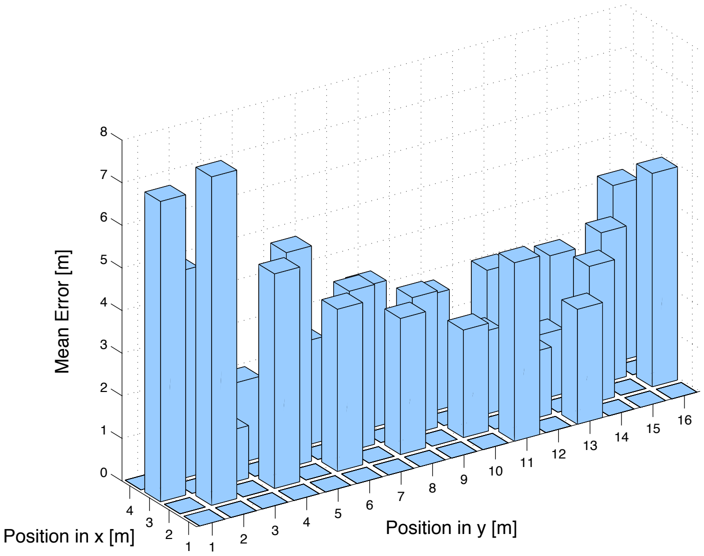
(a) Map *C*.



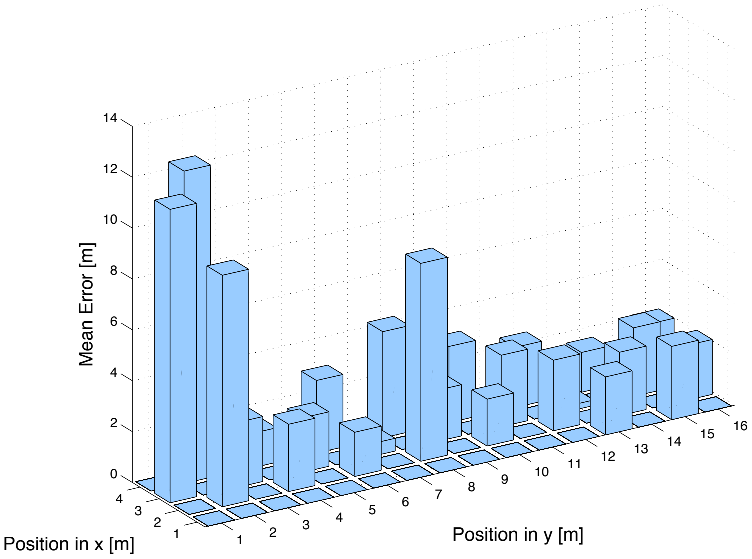
(b) Map *B*.

Figure 5.2: Resulted mean errors on different positions when using mean estimation.

The proposed method built on a product estimation resulted in a total mean euclidean error of 3.52 meters. By testing to only use two APs that was known to be found at every position the total euclidean error was reduced to 3.27 meters. However, as was explained in section 5.2 result at a small number of positions that had a high level of difference from the rest of the result could be neglected. If the result at position (3,1), (2,2) and (4,2) was neglected, the total euclidean mean error was reduced to 2.47 meters. Measured mean euclidean error at each position using product estimation can be found in Figure 5.3.



(a) Map *B*, using all available APs.



(b) Map *B*, using only two known APs.

Figure 5.3: Resulted mean errors on different positions using product estimation.

5.2 Discussion

This section will discuss the result presented in the above results section 5.1. Even though no integration of SLAM and IPS was done, the section will discuss different advantages and disadvantages with a WiFi SLAM solution.

5.2.1 SLAM

The selection of algorithm and type of mapping technique has been deterministic and resulted in the particle SLAM algorithm using grid map representation. This is although the big requirement of memory the best and most precise algorithm for the operation specified in the requirements. The decided grid size of ten centimeters has been selected according to the precision of the range scanning sensors and Digital Lobsters size. To reduce the required memory, this grid size could have been increased with very small effects on performance.

The dependencies to have a good converging SLAM algorithm are many. To make a really cheap good SLAM implementation is even harder. The decided configuration of a “not sweeping” range scanning sensor together with the big variation in heading prediction have been the big design alternative affecting the SLAM algorithm performance. This resulted in a high valued time criteria with precision as the claim.

To get a faster, more robust and converging algorithm, the perception (range scanning measurements) or the prediction has to be more precise.
To get a better perception, a sweeping range scanning sensor discovering the whole surrounding environment gives a better basis for the particle weight calculations, which will create a better converging algorithm. The sweeping sensor was early in the project sorted out because of the timely procedure to sweep over the environment with ultrasonic sensor. To still fulfill the requirement of low cost, the loop time has to be allowed to increased to get a better performing algorithm.

A second way of increasing the performance, is to better model the turns performed by the Digital Lobster. Today it is depending on to many varying factors which makes the turns very hard to control and predict. A specific controller using the encoder values to control turning has to be constructed to reduce the uncertainty. Reducing the variation of the prediction step has been thoroughly tested in simulation and resulted in a good converging algorithm even with locked sensor bearings.
To cope with a good turning model for the Digital Lobster, a current control loop may be necessary to have a consistency in the slip. By having a symmetric suspension of the robot and controlling each wheel both by velocity and torque should give a much easier behaviour to model. Even considering implementing an IMU may be necessary to get a better predictable turning of the Digital Lobster lowering the turning variance. Considering the workload, this might even be a easier solution than designing a totally new control algorithm.

The Digital Lobster do also have a problems drifting in straight forward operation. This has been analysed and concluded to be originating from the suspension, motor internal friction and difference in wheel diameters. The problem will be taken care of with a working SLAM algorithm, but as a quick fix, a compensating factor has been introduced. To get a better performing operation, this may be better tuned and be controlled in a local control algorithm.

5.2.2 IPS

From the background study it was expected that areas with a higher No Line Of Sight, NLOS, were to give worse result. That is, areas with more objects between the transmitter and receiver or positions close to hindering objects. This was shown during testing. Positions in the result plots in section 5.1.2 with higher mean Euclidean error was in a bigger extent placed in NLOS, than positions with lower mean error. For examples position [12,1] and [12,8] in Map C was placed next to a concrete pillar.

It was shown that when using all available APs in the building, a lower accuracy was achieved. The cause was believed to be that at one position in the Radio-Map five APs had been stored when in another position only three of these APs had been stored during the offline phase. At the online estimation phase, when trying to estimate the position with five stored APs in the Radio-Map, it was shown that the probability for a higher CDF result at the position with only three of those APs in the Radio-Map was higher. Probably because of that the lower amount of multiplications will give a larger number. This problem was not considered before testing, because in other research reports that had been studied only "known" APs have been used. That is, APs with known locations, known MAC-addresses and known to be found at every position.

The above problem with inconsistency in number of APs was proven to be worst when only using the frame of reference points in Map C. If one position on the opposite side had fewer APs in the Radio-Map, the result could be "mirrored". There was no points in the middle to help and "catch" the miss estimation. As was explained in section 4.3.3 another method that utilized a mean value was tested instead. This method took better account for the different amount of APs in the Radio-Map and a lower rate of "mirroring" could be achieved. However, if the result in Map C is compared to the result in Map B with points in the middle of the map, it is easily seen that there is problem with areas without points in the whole Radio-Map. This should be kept in mind and a distance related measurement method may be better to implement.

The inconsistency in found APs was probably because of a to low scanning time. The problem of missing APs when scanning have been discussed by Pei et al. [115]. They discussed the problem when using bluetooth, but the same discussion can be used for WiFi that is similar. When the amount of APs and clients in the environment increases, it is more likely that some of them is occupied by a transmission, like to a PC or mobile phone, when the client is scanning and if the scanning time is to short this AP will be missed. In the head office building up to 30 APs could be in use at the same time, which will lead to a higher risk of missed APs. By increasing the scanning time the result can probably be improved, but this will also give a longer offline phase. So it is a trade off between accuracy and time available for the offline phase. Because of the short scanning time, only 15 samples per AP was used at some positions. A problem that maybe affected the result negatively.

In section 4.2.4 it was mentioned that the chosen wire antenna could be a potential problem. During testing it was shown to be a bigger problem than expected. The variation in RSSI value, as can be seen in Figure 5.4, could be as much as 10 dBm, just depending on the orientation of the antenna. This variation can lead to an error of up to five meters or more in estimated position, which can be a disaster. The test should probably been conducted earlier in the project so this problem was found in an earlier state. Then the antenna would immediately have been replaced for an omnidirectional antenna.

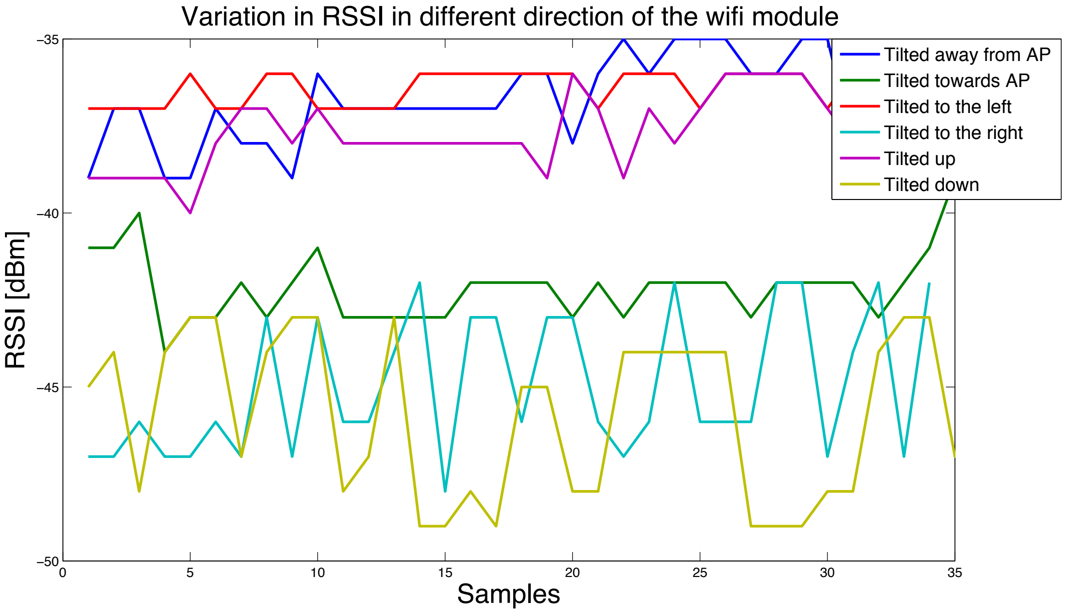


Figure 5.4: Variations in RSSI values for the used wire antenna in different poses at on position.

In the last test that was conducted, only two “known” APs were used. The three highest bars were neglected, at these position one or both of the two “known” APs were missed during the offline phase, which lead to high errors during the online estimation phase. After the result at three positions was neglected, the result was highly improved to when all available APs in the building were used, the third test. It is therefore highly recommended to to only use “known” APs. As was shown by the simulations, more than two “known” APs will increase the accuracy.

The result from the last test, 2.47 meter in mean Euclidean error, can be compared to the result of the similar set-up used in the simulation, section 3.3.2, which was 2.76 meters. This give an indication of that the model used in simulation of IPS was close to the real environment. The model can therefore be used with good reasons, even if it still might need tuning, when developing the system further.

5.2.3 WiFi SLAM

Wifi SLAM was never implemented in real application on the Digital Lobster, but the individual results intended for the composition of wifi SLAM gives a grasp of the possible robustness gain the combined solution would imply. The results of the IPS itself gives a variance of a couple of meters which could reduce the tendency of a SLAM algorithm to diverge, or in the second case give information about the happening.

Drawbacks do off course exist integrating both SLAM and IPS on a robot platform. The need of parallel processes computing SLAM and IPS simultaneously is a necessity which will acquire greater computational resources. The risk of increasing the loop-time is also a potential drawback.

5.2.4 Achieving the Requirements

Measure Range

The range measuring requirements have been achived according to measuring update frequency and measurment precision.

Detect Obstacle

The detecting obstacles requirement was based on a better dynamical model of the vehicle than the real world scenario. Due to the very complicated and dependent dynamical model of the vehicle, the angle accuracy of detecting an object has not been achieved.

Communicate

The communicating ”shall” requirements have all been achieved. The position reporting should requirement has not been achieved due to not implementing local calculations of SLAM and IPS.

Measure Distance Travelled

Measuring longitudinal traveled distance has all been fulfilled due the requirements specification.

Measure Absolute Heading

The absolute heading requirement was not fully achieved either because of the complicated dynamical model of the vehicle.

Measure RSSI Values

The Digital Lobster was able to measure RSSI values with a frequency above 1/3 Hz.

Data Storage

The data storage requirement was never a delimitation due to that calculation never were implemented locally.

Calculate

The requirement of calculating SLAM and IPS locally was never implemented due to lack of time and resources.

Map

The only requirement not achieved in the mapping specification was the one combining SLAM and IPS which also was out sorted due to lack of time in the project.

Position Itself

The Digital Lobster was able to position itself within an area of 30 square meters using only IPS. No implementation of positioning using IPS and SLAM together was implemented.

Navigate

The Digital Lobster algorithm was able to navigate with the purpose of exploration.

Chapter 6

Conclusion and Further Work

6.1 Conclusion

From the results and discussions, some conclusions have been drawn, both for SLAM and IPS, but also for the WiFi SLAM. It was decided to give some short conclusion for the WiFi SLAM, even though no implementation and testing has been made, because there is probable potential in an IPS and SLAM integration.

6.1.1 SLAM

To in the best way select an algorithm endorsing the criteria of precision, cost and timing, a grid SLAM algorithm with adaptions from the the fast SLAM algorithm was chosen. Also a grid map representation was necessary to represent the existing diversity of the environment. Further improvement may be reached using loop closure algorithms fitting the map.

The SLAM design implemented does only work in perfect conditions as it exists today. Due to too big variance in process model and the few range measurements per loop execution, an excess of particles would be needed to have a better converging algorithm. The choice of implementing a sweeping sensor discovering at least 180 degrees in the direction the Digital lobster is moving would improve the robustness of the algorithm many times. The choice would be time consuming but will lead to a better converging algorithm.

The Result operating in a simple environment with quite good precision shows, despite the uncertainties a very functional particle SLAM algorithm operating over a grid fence.

6.1.2 IPS

A WiFi implementation was seen to be a good method for indoor positioning. There is no, or little, need for extra infrastructure. The protocol can handle communication, so one module can be used for two things, and can position even if there is NLOS. It is therefore very cost effective, which was one of the most important criteria from the stakeholder ÅF.

The resulting solution with both the mean estimation method and the product estimation method using only two “known” APs fulfils the requirements in section 3.1, a total mean error of ±3 meters. However, the accuracy can probably be improved by adding even more “known” APs or by develop the algorithm to better take account for inconsistency in found APs.

To integrate indoor positioning and SLAM in smaller areas, as the tested Map A, the requirement is probably not good enough and has to be improved. With more time for fine tuning and testing, a result closer to 1.5 meter should be achievable.

The chosen antenna was too bad. It should have been tested earlier in the project, but due to platform issues and time limitations it was not doable until it was to late. For a good indoor positioning system an omnidirectional antenna is probably the best choice.

If the simulation model was calibrated against the real environment, more tests and fine tuning of the algorithm could have been conducted. This would have saved time and probably improved the result. With a good simulation model distance related methods could easily be implemented and improved the result when there is no possibility to take measurements at certain positions, as in Map C.

Distance related methods are also easier to adapt to new environments, because they do not need a Radio-Map, just the positions of the APs. This will save needed storage memory in an embedded application as well as needed time to build a new Radio-Map.

6.1.3 WiFi SLAM

WiFi Slam as written in the introduction, has not been implemented during this master thesis. Methods have although been discussed in the previous parts of this report and the individual results of both the SLAM algorithm and the IPS precision will give a great potential implementing the fuse of both the algorithms. The combination will enhance the performance of the SLAM algorithm and make it more robust. It will also be possible to identify divergence which elsewhere may be very difficult.

6.2 Further Work

It is believed that the proposed methods have a good potential for the Digital Lobster. However, due to the need of re-designing large parts of the platform, less time was given to calibrate, test different methods, find error and bugs and to integrate a WiFi SLAM solution. A large set of potential improvements and tests to be done have therefore been collected as further work under this section.

6.2.1 SLAM

As discussed about the problem with a bad converging algorithm in the discussion chapter above, one can either...

- Re-arrange the Range Scanning sensors and complement with one sweeping the environment.
- Create a control system managing torque and velocity control of every wheel.

By performing one of these alternatives, a much more robust SLAM algorithm would probably be able to operate in an office environment.

Even work concerning moving the algorithm to a local computer will reduce time sending information to a server. This will be a perfect opportunity to use the full potential of Zync-7000 SoC on the Zedboard development platform.

Alternatively solutions to the bad performing turning can be analysing...

- How will a Gyro IMU affect the feed-back control of the turning phase

6.2.2 IPS

The proposed further work for the indoor positioning application highly relates to the conclusions in section 6.1.2:

- More “known” APs should be used and placed in a good way by testing different arrangements with the simulation model.
- Make sure more than 15 samples can be taken from each AP in the Radio-Map at each position, by for example increasing the scan time. When using only “known” APs, the scanning can be limited to the channels of those APs to lower needed scanning time.
- Change the antenna to a better solution, probably an omnidirectional antenna.
- Develop the algorithm further to take account for inconsistency in found APs if the use of all available APs is wanted.

As for the SLAM application, the algorithm can be moved to the client. This require the client to have a large memory available for the Radio-Map. It will however, reduce the time and also risks concerning that the communication between client and server interfere with the measurements.

A drawback to move to a fully client based solution is that it is more difficult to take account for dynamics. If the system still is semi-centralized or fully centralized one can think of using the APs to scan each other or a client with known position. By these measurements a weighting variable can be calculated and used, both in the proposed method, but also in a distance related method.

In this master thesis there was no time to test the differentiated access points. Based on the good results presented by Chang et al. [100] it is highly recommended to see if the result can be improved.

To put time into the simulation model will probably give a better result and is recommended. Then a change to distance related measurements with trilateration can be done. It will be easier to adapt this method to new environments, as a new building or room. This method needs at least three "known" APs, but at least four is recommended, to position, which can be good to bear in mind.

6.2.3 WiFi SLAM

The implementation of fusing SLAM and IPS needs to be implemented to be tested in real world applications. Individual results show that IPS and SLAM integrated would get more robust and counteract divergence.

Bibliography

[1] Department of Defence. *Global Positioning System Standard Positioning Service Performance Standard.* Tech. rep. Department of Defence,USA, Sept. 2008. URL: `http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf`.

[2] V. Maximov and O. Tabarovsky. "Survey of Accuracy Improvement Approaches for Tightly Coupled ToA/IMU Personal Indoor Navigation System". In: *Indoor Positioning and Indoor Navigation, 2013 International Conference on.* Oct. 2013.

[3] Aspray W. *The Intel 4004 microprocessor:what constituted ivention?* Tech. rep. 1997.

[4] James L Crowley and Yves Demazeau. *Principles and Techniques for Sensor Data Fusion.* Tech. rep. Grenoble Cédex, FRANCE.

[5] Kalman R. E. *A New Approach to Linear Filtering and Prediction Problems.* Tech. rep. 1960.

[6] Ronald Yager et al. *Classic Works of the Dempster-Shafer Theory of Belief Functions.* Poland: Springer, 2008.

[7] Paul P. Wang, Da Ruan, and Eitenne E. Kerre. *Studies in Fuzziness and soft Computing.* Berlin Heidelberg: Professor Dr. Da Ruan, 2007.

[8] David L. Hall and James Llinas. *Handbook of Multisensor Data Fusion.* CRC Press, 2001.

[9] Huadong Wu. *Sensor Data Fusion for Context-Aware Computing Using Dempster-Shafer Theory.* Tech. rep. Pittsburgh, 2003.

[10] Alan N. Steinberg, Christopher L. Bowman, and Franklin E. White. *Revisions to the JDL data fusion model.* 1999. DOI: `10.1117/12.341367`. URL: `http://dx.doi.org/10.1117/12.341367`.

[11] S. Blackman. *Multiple-Target Tracking with Radar Applications.* Tech. rep. Norwood;MA, 1986.

[12] D. Smith and S. Singh. "Approaches to Multisensor Data Fusion in Target Tracking: A Survey". In: *Knowledge and Data Engineering, IEEE Transactions on* 18.12 (Dec. 2006), pp. 1696–1710. ISSN: 1041-4347. DOI: `10.1109/TKDE.2006.183`.

[13] E. Waltz and J. Llinas. *Multisensor Data Fusion.* Tech. rep. Norwood, MA, 1990.

[14] N. Xiong and P. Svensson. "Multi-sensor management for information fusion: issues and approaches". In: *Information Fusion* 3.2 (2002), pp. 163–186. ISSN: 1566-2535. DOI: `http://dx.doi.org/10.1016/S1566-2535(02)00055-6`. URL: `http://www.sciencedirect.com/science/article/pii/S1566253502000556`.

[15] Kalman R. E. *Contributions to the theory of optimal control.* Tech. rep. Mexicana, 1960.

[16] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics.* Massachusets-MIT: Sebastian Thrun, 2006.

[17] Lawrence A. Klein. *Sensor and Data Fusion.* Bellingham, Washington: SPIE-The International Society for Optical Engineering, 2004.

[18] J.W. Betz, J.L. Prince, and M. Bello. “Representation and transformation of uncertainty in an evidence theory framework”. In: *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR ’89., IEEE Computer Society Conference on.* June 1989, pp. 646–652. DOI: 10.1109/CVPR.1989.37914.

[19] USAF Bethany G. Foley Captain. *A Dempster-Shafer Method for Multi-Sensor Fusion.* Tech. rep. Dayton, 2012.

[20] Uwe Kay Rakowsky. “Fundamentals Of The Dempster-Shafer Theory And Its Applications To Reliability Modeling”. In: *International Journal of Reliability, Quality and Safety Engineering* 14.06 (2007), pp. 579–601. DOI: 10.1142/S0218539307002817. eprint: http://www.worldscientific.com/doi/pdf/10.1142/S0218539307002817. URL: http://www.worldscientific.com/doi/abs/10.1142/S0218539307002817.

[21] Steven S. Muchnick and Peter Schnupp. *Building Expert Systems in Prolog.* New York: Dennis Meritt, 1989.

[22] J. Z. Sasadek and Q. Wang. *Sensor Fusion Based on Fuzzy Kalman Filtering for Autonomus Robot Vehicle.* Tech. rep. Detroit,Michigan, 1999.

[23] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo: Morgan Kaufmann, 1988.

[24] G. Shafer. *A Mathematical Theory of Evidence.* Tech. rep. Princeton,NJ, 1976.

[25] H. Leung and J. Wu. *Bayesian and Dempster-Shafer target identification for radar surveillance.* Tech. rep. 2000.

[26] O. Cohen and Y. Edan. *Adaptive Fuzzy logic Algorithms for Sensor Fusion Mapping.* Tech. rep. Beer Sheva,Israel, 2004.

[27] JOAKIM HUGMARK. “Inertial-aided EKF-based Structure from Motionfor Robust Realtime Augmented Reality”. MA thesis. KTH, School of Computer Science and Communication (CSC), 2013.

[28] Rui Xiong et al. “A robust state-of-charge estimator for multiple types of lithium-ion batteries using adaptive extended Kalman filter”. In: *Journal of Power Sources* 243 (2013), pp. 805–816. ISSN: 0378-7753. DOI: http://dx.doi.org/10.1016/j.jpowsour.2013.06.076. URL: http://www.sciencedirect.com/science/article/pii/S0378775313010835.

[29] Arnaud Doucet and Adam M. Johansen. *A Tutorial on Particle Filtering and Smoothing:* tech. rep. Tokyo,Coventry, 2008.

[30] Sebastian Thrun Michael Monternerlo. *Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM.* Tech. rep. Taipei,Taiwan, 2003.

[31] Lina M. Paz, J.D. Tardos, and J. Neira. “Divide and Conquer: EKF SLAM in O(n)”. In: *Robotics, IEEE Transactions on* 24.5 (Oct. 2008), pp. 1107–1120. ISSN: 1552-3098. DOI: 10.1109/TRO.2008.2004639.

[32] Geoffrey John McLachlan. *Mahalanobis Distance.* Tech. rep. Brisbane, 1999.

[33] Moti L. Tiku, M. Qamarul Islam, and B. Sahar. *Mahalanobis distance under non-normaility.* Tech. rep. Ankara, 2009.

[34] A Nuchter et al. “6D SLAM with an application in autonomous mine mapping”. In: *Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004 IEEE International Conference on.* Vol. 2. Apr. 2004, 1998–2003 Vol.2. DOI: 10.1109/ROBOT.2004.1308117.

[35] J. A. Castellanosf J. D. Tarddsf G. Schmidt3. *Building a Global Map of the Environment of a Mobile Robot: The Importance of Correlations.* Tech. rep. Albequerque, 1996.

[36] Joel W. Burdick Samuel T. Pfister Stergios I. Roumeliotis. *Weighted Line Fitting Algorithms for.* Tech. rep. Taipei, 2003.

[37] Viet Nguyen Siegwart et al. *A comparison of line extraction algorithms using 2D range.* Tech. rep. Zürich, 2006.

[38] Theodosios Pavlidis and S.L. Horowitz. "Segmentation of Plane Curves". In: *Computers, IEEE Transactions on* C-23.8 (Aug. 1974), pp. 860–870. ISSN: 0018-9340. DOI: 10.1109/T-C.1974.224041.

[39] Ali Siadat et al. *An Optimized Segmentation Method For A 2D Laser-Scanner Applied To Mobile Robot Navigation.* 1997.

[40] R. M. Taylor and P.J. Probert. "Range finding and feature extraction by segmentation of images for mobile robot navigation". In: *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on.* Vol. 1. Apr. 1996, 95–100 vol.1. DOI: 10.1109/ROBOT.1996.503579.

[41] M Haidekker, ed. *Hough Transform.* Oxford: Wiley-IEEE Press, 2011, pp. 211–235.

[42] Kai O. Arras and Roland Y. Siegwart. *Feature extraction and scene interpretation for map-based navigation and map building.* 1998. DOI: 10.1117/12.299565. URL: http://dx.doi.org/10.1117/12.299565.

[43] C. Nardinocchi, M. Scaioni, and G. Forlani. "Building extraction from LIDAR data". In: *Remote Sensing and Data Fusion over Urban Areas, IEEE/ISPRS Joint Workshop 2001.* 2001, pp. 79–84. DOI: 10.1109/DFUA.2001.985731.

[44] S.T. Pfister, S.I. Roumeliotis, and J.W. Burdick. "Weighted line fitting algorithms for mobile robot map building and efficient data representation". In: *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on.* Vol. 1. Sept. 2003, 1304–1311 vol.1. DOI: 10.1109/ROBOT.2003.1241772.

[45] Philipp Althaus. "Indoor Navigation for Mobile Robots : Control and Representations". PhD thesis. KTH, Numerical Analysis and Computer Science, NADA, 2003, pp. x, 109.

[46] M. W M G Dissanayake et al. "A solution to the simultaneous localization and map building (SLAM) problem". In: *Robotics and Automation, IEEE Transactions on* 17.3 (June 2001), pp. 229–241. ISSN: 1042-296X. DOI: 10.1109/70.938381.

[47] Armita Kaboli, Michael Bowling, and Petr Musilek. *Bayesian Calibration for Monte Carlo Localization.* Tech. rep. 2006. URL: http://www.aaai.org/Papers/AAAI/2006/AAAI06-151.pdf.

[48] Austin I. Eliazar and Ronald Parr. *Learning Probabilistic Motion Models for Mobile Robots.* English. Tech. rep. 2004.

[49] Artit. Visatemongkolchai and Hong Zhang. "Building Probabilistic Motion Models for SLAM". In: *Building Probabilistic Motion Models for SLAM, 2008. Proceedings 2007 IEEE International Conference on Robotics and Biomimetics.* 2008.

[50] Hans P. Moravec. *Sensor Fusion in Certainty Grids for Mobile Robots.* English. Tech. rep. 1988.

[51] Calvin H Wilcox Richard E Blahut Willard Miller, ed. New York: : Springer-Verlag New York Inc., 1991, p. 271. ISBN: 9781468471021.

[52] John G. Sargantanis and M. Nazmul Karim. "Multivariable Iterative Extended Kalman Filter Based Adaptive Control: Case Study of Solid Substrate Fermentation". In: *Industrial & Engineering Chemistry Research* 33.4 (1994), pp. 878–888. DOI: 10.1021/ie00028a014. eprint: http://pubs.acs.org/doi/pdf/10.1021/ie00028a014. URL: http://pubs.acs.org/doi/abs/10.1021/ie00028a014.

[53] J. Wilbur et al. *A Multi-hypothesis Filter for Passive Tracking of Surface and Sub-Surface Target Localization.* Tech. rep. Panama City Beach,FL.

[54] Young Hoon Lee et al. *The analysis of effect of observation models and data associations on the consistency of EKF SLAM.* Tech. rep. Los Angles, 2011.

[55] Antastasios I. Mourikis Gyaquan P. Huang and Stergios I. Roumeliotis. "On the Complexity and Consistency of UKF-Based SLAM". In: *IEEE International Conference on Robotics and Automation* (2009), p. 8.

[56] Sebastian Thrun et al. “Simultaneous Localization and Mapping with Sparse Extended Information Filters”. In: *The International Journal of Robotics Research* 23.7-8 (2004), pp. 693–716. DOI: 10.1177/0278364904045479. eprint: http://ijr.sagepub.com/content/23/7-8/693.full.pdf+html. URL: http://ijr.sagepub.com/content/23/7-8/693.abstract.

[57] S. Thrun et al. “Autonomous exploration and mapping of abandoned mines”. In: *Robotics Automation Magazine, IEEE* 11.4 (Dec. 2004), pp. 79–91. ISSN: 1070-9932. DOI: 10.1109/MRA.2004.1371614.

[58] D.M. Cole and P.M. Newman. “Using laser range data for 3D SLAM in outdoor environments”. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on.* May 2006, pp. 1556–1563. DOI: 10.1109/ROBOT.2006.1641929.

[59] Shi Jianli et al. “Unscented FastSLAM for UAV”. In: *Computer Science and Network Technology (ICCSNT), 2011 International Conference on.* Vol. 4. Dec. 2011, pp. 2529–2532. DOI: 10.1109/ICCSNT.2011.6182484.

[60] Masashi Yokozuka and Osamu Matsumoto. “Sub-Map Dividing and Realignment FastSLAM by Blocking Gibbs MCEM for LargeScale 3D Grid Mapping”. In: *Advanced Robotics* 26.14 (2012), pp. 1649–1675. DOI: 10.1080/01691864.2012.695892. eprint: http://www.tandfonline.com/doi/pdf/10.1080/01691864.2012.695892. URL: http://www.tandfonline.com/doi/abs/10.1080/01691864.2012.695892.

[61] M. Montemerlo et al. “FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges”. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI).* IJCAI. Acapulco, Mexico, 2003.

[62] Hartmut Surmann, Andreas Nüchter, and Joachim Hertzberg. “An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments”. In: *Robotics and Autonomous Systems* 45 (2003), pp. 181–198. ISSN: 0921-8890. DOI: http://dx.doi.org/10.1016/j.robot.2003.09.004. URL: http://www.sciencedirect.com/science/article/pii/S0921889003001556.

[63] K. Lingemann et al. “High-Speed Laser Localization For Mobile Robots”. In: *Robotics and Autonomous Systems* 51 (2005), pp. 275–296. ISSN: 0921-8890. DOI: 10.1016/j.robot.2005.02.004.

[64] M. Ocana et al. “Indoor robot navigation using a POMDP based on WiFi and ultrasound observations”. In: *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on.* Aug. 2005, pp. 2592–2597. DOI: 10.1109/IROS.2005.1545031.

[65] JinWoo Choi, Sunghwan Ahn, and Wan Kyun Chung. “Robust sonar feature detection for the SLAM of mobile robot”. In: *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on.* Aug. 2005, pp. 3415–3420. DOI: 10.1109/IROS.2005.1545284.

[66] A. Diosi and L. Kleeman. “Advanced sonar and laser range finder fusion for simultaneous localization and mapping”. In: *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on.* Vol. 2. Sept. 2004, 1854–1859 vol.2. DOI: 10.1109/IROS.2004.1389667.

[67] A.J. Davison et al. “MonoSLAM: Real-Time Single Camera SLAM”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29.6 (June 2007), pp. 1052–1067. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.1049.

[68] E. Eade and Tom Drummond. “Scalable Monocular SLAM”. In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on.* Vol. 1. June 2006, pp. 469–476. DOI: 10.1109/CVPR.2006.263.

[69] SungHwan Ahn et al. “A practical approach for EKF-SLAM in an indoor environment: fusing ultrasonic sensors and stereo camera”. English. In: *Autonomous Robots* 24.3 (2008), pp. 315–335. ISSN: 0929-5593. DOI: 10.1007/s10514-007-9083-2. URL: http://dx.doi.org/10.1007/s10514-007-9083-2.

[70] K. R. Beevers and W. H. Huang. “An Embedded Implementation of SLAM with Sparse Sensing”. In: *IEEE International Conference on Robotics & Automation (ICRA 2008).* 2008.

[71] Young-Ho Choi, Tae-Kyeong Lee, and Se-Young Oh. “A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot”. English. In: *Autonomous Robots* 24.1 (2008), pp. 13–27. ISSN: 0929-5593. DOI: 10.1007/s10514-007-9050-y. URL: http://dx.doi.org/10.1007/s10514-007-9050-y.

[72] S. Magnenat et al. “Affordable SLAM through the co-design of hardware and methodology”. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on.* May 2010, pp. 5395–5401. DOI: 10.1109/ROBOT.2010.5509196.

[73] Shao-Wen Yang and Chieh-Chih Wang. “Dealing with laser scanner failure: Mirrors and windows”. In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on.* May 2008, pp. 3009–3015. DOI: 10.1109/ROBOT.2008.4543667.

[74] L. Qingquan, L. Bijun, and C. Jing. “Research on laser range scanning and its application”. English. In: *Geo-spatial Information Science* 4.1 (2001), pp. 37–42. DOI: 10.1007/BF02826635. URL: http://dx.doi.org/10.1007/BF02826635.

[75] R. A. Jarvis. “A Laser Time-of-Flight Range Scanner for Robotic Vision”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* PAMI-5.5 (Sept. 1983), pp. 505–512. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1983.4767429.

[76] K. Konolige et al. “A low-cost laser distance sensor”. In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on.* May 2008, pp. 3002–3008. DOI: 10.1109/ROBOT.2008.4543666.

[77] S. M. Nejad and S. Olyaee. “Comparison of TOF, FMCW and Phase-Shift Laser Range-Finding Methods by Simulation and Measurement”. English. In: *Quartarly Journal of Technology & Education* 1.1 (2009), pp. 11–18.

[78] Wolfgang Boehler, M Bordas Vicent, and Andreas Marbs. “Investigating laser scanner accuracy”. In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 34.Part 5 (2003), pp. 696–701.

[79] G. Zunino. “Simultaneous Localization and Mapping for Navigation in Realistic Environments”. MA thesis. KTH, Royal Institute of Technology Sweden, 2002.

[80] D. Marioli et al. “Digital time-of-flight measurement for ultrasonic sensors”. In: *Instrumentation and Measurement, IEEE Transactions on* 41.1 (Feb. 1992), pp. 93–97. ISSN: 0018-9456. DOI: 10.1109/19.126639.

[81] M.K. Brown. “Feature extraction techniques for recognizing solid objects with an ultrasonic range sensor”. In: *Robotics and Automation, IEEE Journal of* 1.4 (Dec. 1985), pp. 191–205. ISSN: 0882-4967. DOI: 10.1109/JRA.1985.1087022.

[82] Zou Yi et al. “Multi-ultrasonic sensor fusion for mobile robots”. In: *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE.* 2000, pp. 387–391. DOI: 10.1109/IVS.2000.898374.

[83] Qinghao Meng, Fengjuan Yao, and Yuehua Wu. “Review of Crosstalk Elimination Methods for Ultrasonic Range Systems in Mobile Robots”. In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on.* Oct. 2006, pp. 1164–1169. DOI: 10.1109/IROS.2006.281848.

[84] Klaus-Werner Jörg. "World modeling for an autonomous mobile robot using heterogenous sensor information". In: *Robotics and Autonomous Systems* 14.2-3 (1995). Research on Autonomous Mobile Robots, pp. 159–170. ISSN: 0921-8890. DOI: http://dx.doi.org/10.1016/0921-8890(94)00027-Y. URL: http://www.sciencedirect.com/science/article/pii/092188909400027Y.

[85] B. Barshan and R. Kuc. "Differentiating sonar reflections from corners and planes by employing an intelligent sensor". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 12.6 (June 1990), pp. 560–569. ISSN: 0162-8828. DOI: 10.1109/34.56192.

[86] A. Hernandez et al. "Ultrasonic ranging sensor using simultaneous emissions from different transducers". In: *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on* 51.12 (Dec. 2004), pp. 1660–1670. ISSN: 0885-3010. DOI: 10.1109/TUFFC.2004.1386683.

[87] Zhen-Jing Yao et al. "Non-crosstalk real-time ultrasonic range system with optimized chaotic pulse position-width modulation excitation". In: *Ultrasonics Symposium, 2008. IUS 2008. IEEE.* Nov. 2008, pp. 729–732. DOI: 10.1109/ULTSYM.2008.0174.

[88] L. Korba, S. Elgazzar, and T. Welch. "Active infrared sensors for mobile robots". In: *Instrumentation and Measurement, IEEE Transactions on* 43.2 (Apr. 1994), pp. 283–287. ISSN: 0018-9456. DOI: 10.1109/19.293434.

[89] Heon-Hui Kim, Yun-Su Ha, and Gang-Gyoo Jin. "A study on the environmental map building for a mobile robot using infrared range-finder sensors". In: *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on.* Vol. 1. Oct. 2003, 711–716 vol.1. DOI: 10.1109/IROS.2003.1250713.

[90] M. Alwan et al. "Characterization of Infrared Range-Finder PBS-03JN for 2-D Mapping". In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on.* Apr. 2005, pp. 3936–3941. DOI: 10.1109/ROBOT.2005.1570722.

[91] P. M. Novotny and N.J. Ferrier. "Using infrared sensors and the Phong illumination model to measure distances". In: *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on.* Vol. 2. 1999, 1644–1649 vol.2. DOI: 10.1109/ROBOT.1999.772595.

[92] Jaehong Song et al. "A Depth Measurement System Associated with a Mono-camera and a Rotating Mirror". English. In: *Advances in Multimedia Information Processing - PCM 2002.* Ed. by Yung-Chang Chen, Long-Wen Chang, and Chiou-Ting Hsu. Vol. 2532. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, pp. 1145–1152. ISBN: 978-3-540-00262-8. DOI: 10.1007/3-540-36228-2_142. URL: http://dx.doi.org/10.1007/3-540-36228-2_142.

[93] R. Mautz and S. Tilch. "Survey of optical indoor positioning systems". In: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on.* Sept. 2011, pp. 1–7. DOI: 10.1109/IPIN.2011.6071925.

[94] B. Freedman et al. *Depth mapping using projected patterns.* US Patent App. 12/522,171. May 13, 2010. URL: http://www.google.com/patents/US20100118123.

[95] G.P. Stein, O. Mano, and A. Shashua. "Vision-based ACC with a single camera: bounds on range and range rate accuracy". In: *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE.* June 2003, pp. 120–125. DOI: 10.1109/IVS.2003.1212895.

[96] Jae-Hean Kim and Myung-Jin Chung. "SLAM with omni-directional stereo vision sensor". In: *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on.* Vol. 1. Oct. 2003, 442–447 vol.1. DOI: 10.1109/IROS.2003.1250669.

[97] A. Darabiha, J. Rose, and J.W. Maclean. "Video-rate stereo depth measurement on programmable hardware". In: *Computer Vision and Pattern Recognition, 2003. Proceedings.*

2003 IEEE Computer Society Conference on. Vol. 1. June 2003, pp. 203–210. DOI: 10.1109/CVPR.2003.1211355.

[98] S.B. Goldberg, M.W. Maimone, and L. Matthies. "Stereo vision and rover navigation software for planetary exploration". In: *Aerospace Conference Proceedings, 2002. IEEE.* Vol. 5. 2002, pp. 2025–2036. DOI: 10.1109/AERO.2002.1035370.

[99] G. Mao, B. Fidan, and B. D. O. Anderson. "Wireless sensor network localization techniques". In: *Computer Networks* 51 (Jan. 2007), pp. 2529–2553.

[100] N. Chang, R. Rashidzadeh, and M. Ahmadi. "Robust Indoor Positioning using Differential Wi-Fi Access Points". In: *IEEE Transactions on Consumer Electronics* 56.3 (Aug. 2010), pp. 1860–1867.

[101] D. Koks. "Numerical Calculations for Passive Geolocation Scenarios". In: Defence Science and Technology Organisation, Electronic Warfare and Radar Division, 2007.

[102] N. Golmie and F. Mouveaux. "Interference in the 2.4 GHz ISM band: impact on the Bluetooth access control performance". In: *Communications, 2001. ICC 2001. IEEE International Conference on.* Vol. 8. 2001, 2540–2545 vol.8. DOI: 10.1109/ICC.2001.936608.

[103] P. Bahl and V. N. Padmanabhan. "RADAR: An In-Building RF-based User Location and Tracking System". In: *Proc. INFOCOM.* 2000, pp. 775–784.

[104] K. Curran et al. "An evaluation of indoor location determination technologies". In: *Journal of Location Based Services* 5.2 (2011), pp. 61–78. DOI: 10.1080/17489725.2011.562927.

[105] C. A. Balanis. "Friis Transmission Equation and Radar Range Equation". In: Antenna Theory, Analysis and Design, 1982. Chap. 2.17.

[106] H Lim et al. "Zero-Configuration, Robust Indoor Localization: Theory and Experimentation". In: *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings.* Apr. 2006, pp. 1–12. DOI: 10.1109/INFOCOM.2006.223.

[107] Y. S. Chiou, C. L. Wang, and S. C. Yeh. "An adaptive location estimator using tracking algorithms for indoor WLANs". In: *Wireless Network* 16.2 (2010), pp. 1987–2012. DOI: 10.1007/s11276-010-0240-8.

[108] W. M. Y. W. Bejuri et al. "Wireless sensor network localization techniques". In: (2012).

[109] A. Al Rifai. "Indoor Positioning at Arlanda Airport". MA thesis. KTH, Royal Institute of Technology Sweden, June 2009.

[110] V. A. Dubendorf. "RFID". In: Wireless Data Technologies, 2003. Chap. 9, pp. 161–181.

[111] A. Lim and K. Zhang. "A Robust RFID-Based Method for Precise Indoor Positioning". In: *Advances in Applied Artificial Intelligence.* Ed. by M. Ali and R. Dapoigny. Vol. 4031. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 1189–1199. ISBN: 978-3-540-35453-6. DOI: 10.1007/11779568_126. URL: http://dx.doi.org/10.1007/11779568_126.

[112] A. W. Reza and T. K. Geok. "Investigation of Indoor Location Sensing via RFID Reader Network Utilizing Grid Covering Algorithm". English. In: *Wireless Personal Communications* 49.1 (2009), pp. 67–80. ISSN: 0929-6212. DOI: 10.1007/s11277-008-9556-4. URL: http://dx.doi.org/10.1007/s11277-008-9556-4.

[113] C. Gomez, J. Oller, and J. Paradells. "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology". In: *Sensors* 12.9 (2012), pp. 11734–11753. DOI: 10.3390/s120911734.

[114] F. Subhan et al. "Indoor positioning in Bluetooth networks using fingerprinting and lateration approach". In: *Information Science and Applications (ICISA), 2011 International Conference on.* Apr. 2011, pp. 1–9. DOI: 10.1109/ICISA.2011.5772436.

[115] L. Pei et al. "Using Inquiry-based Bluetooth RSSI Probability Distributions for Indoor Positioning". In: *Journal of Global Positioning Systems* 9.2 (2010), pp. 11734–11753. DOI: 10.5081/jgps.9.2.122.

[116] S. Li, B. Liu, and B. Chen. "Neural network based mobile phone localization using Bluetooth connectivity". In: *Neural Computing & Applications* 23 (2013), pp. 667–675. DOI: 10.1007/s00521-012-0950-1.

[117] W. Zhang and M. Kavehrad. "Comparison of VLC-based indoor positioning techniques". In: *Broadband Access Communication Technologies VII. Proceedings of the SPIE.* Vol. 8645. 2013. DOI: 10.1117/12.2001569.

[118] X. Liu, H. Makino, and K. Mase. "Improved Indoor Location Estimation Using Fluorescent Light Communication System with Nine-Channel Receiver". In: *IEICE Trans. Commun.* E93-B.11 (Nov. 2010), pp. 2936–2944. DOI: 10.1587/transcom.E93.B.2936.

[119] Z. Zhou, M. Kavehrad, and P. Deng. "Indoor positioning algorithm using light-emitting diode visible light communications". In: *Optical Engineering* 51.8, 085009 (Aug. 2012). DOI: 10.1117/1.OE.51.8.085009.

[120] R. Want et al. "The Active Badge Location System". In: *ACM Trans. Inf. Syst.* 10.1 (Jan. 1992), pp. 91–102. ISSN: 1046-8188. DOI: 10.1145/128756.128759. URL: http://doi.acm.org/10.1145/128756.128759.

[121] O. Bischoff et al. "Implementation of an ultrasonic distance measuring system with kalman filtering in wireless sensor networks for transport logistics". In: *Procedia Engineering* 5 (2010). Eurosensor {XXIV} Conference Eurosensor {XXIV} Conference, pp. 196–199. ISSN: 1877-7058. DOI: http://dx.doi.org/10.1016/j.proeng.2010.09.081. URL: http://www.sciencedirect.com/science/article/pii/S1877705810006284.

[122] H Schweinzer and G Kaniak. "Ultrasonic device localization and its potential for wireless sensor network security". In: *Control Engineering Practice* 18.8 (2010), pp. 852–862. ISSN: 0967-0661. DOI: http://dx.doi.org/10.1016/j.conengprac.2008.12.007. URL: http://www.sciencedirect.com/science/article/pii/S0967066109000021.

[123] B. S. Heyi. "Implementation of Indoor Positioning using IEEE802.15.4a (UWB)". MA thesis. KTH, Royal Institute of Technology Sweden, 2013.

[124] Y. Zhou et al. "Indoor Elliptical Localization Based on Asynchronous UWB Range Measurement". In: *Instrumentation and Measurement, IEEE Transactions on* 60.1 (Jan. 2011), pp. 248–257. ISSN: 0018-9456. DOI: 10.1109/TIM.2010.2049185.

[125] C. Losada et al. "Multi-Camera Sensor System for 3D Segmentation and Localization of Multiple Mobile Robots". In: *Sensors* 10.4 (2010), pp. 3261–3279. DOI: 10.3390/s100403261.

[126] I. Fernández et al. "Guidance of a mobile robot using an array of static cameras located in the environment". In: *Auton. Robot.* 23 (2007), pp. 305–324. DOI: 10.1007/s10514-007-9049-4.

[127] S. A. Bellot. "Visual tag recognition for indoor positioning". MA thesis. KTH, Royal Institute of Technology Sweden, 2011.

[128] J. Kim and H. Jun. "Vision-based location positioning using augmented reality for indoor navigation". In: *Consumer Electronics, IEEE Transactions on* 54.3 (Aug. 2008), pp. 954–962. ISSN: 0098-3063. DOI: 10.1109/TCE.2008.4637573.

[129] Panzarino Mathew. *What exactly WiFiSLAM is, and why Apple acquired it.* 2013. URL: http://thenextweb.com/apple/2013/03/26/what-exactly-wifislam-is-and-why-apple-acquired-it/.

TRITA MMK 2014:101 MDA 475