Improved Situational Awareness of Heavy Machinery Operators in Low Visibility Conditions

JOAKIM TROBECK

KTH Industrial Engineering
and Management

Master of Science Thesis
Stockholm, Sweden 2014

Improved Situational Awareness of Heavy Machinery Operators in Low Visibility Conditions

Joakim Trobeck

Master of Science Thesis MMK 2014:93 MDA 480
KTH Industrial Engineering and Management
Machine Design
SE-100 44 STOCKHOLM

Förbättrad Omvärldsuppfattning för Operatörer av Tunga Maskiner vid Dålig Sikt

av

Joakim Trobeck

Examensarbete MMK 2014:93 MDA 480
KTH Industriell teknik och management
Maskinkonstruktion
SE-100 44 STOCKHOLM

KTH Industrial Engineering and Management

Master of Science Thesis MMK 2014:93 MDA 480

Improved Situational Awareness of Heavy Machinery Operators in Low Visibility Conditions

Joakim Trobeck

Approved	Examiner	Supervisor
2015-month-day	De-Jiu Chen	Björn Möller
	Commissioner	Contact person
	Combitech AB	Magnus Gens

Abstract

This thesis studies potential solutions to improve the situational awareness of operators of heavy machinery in low visibility conditions caused by snow, heavy rain or fog. Sensors and techniques intended to provide situational awareness for vehicles have been studied in previous research, but these studies have not covered the robustness of the sensor platform to fog, heavy rain or snow. Furthermore, the information from the sensors has not been presented to a human operator. Four main functionalities are identified: localization, obstacle detection, data fusion and user interface. A number of potential concepts are identified within each functionality using a literature study. The identified concepts are evaluated in terms of feasibility and performance by analysis and in some cases an implemented prototype. Using the concepts evaluated for each function, a system prototype is developed and evaluated in an urban environment. The final proposition for the system involves a high precision GPS for localization, an automotive radar for obstacle detection, an unscented Kalman filter for data fusion and a user interface based upon a 3D map in which the localized obstacles are visualized. The suggested prototype is judged to be suitable for production in small series, aimed at applications where down time caused by bad visibility is present and costly.

KTH Industriell teknik och management

Examensarbete MMK 2014:93 MDA 480

Förbättrad Omvärldsuppfattning för Operatörer av Tunga Maskiner vid Dålig Sikt

Joakim Trobeck

Godkänt	Examinator	Handledare
2015-mån-dag	De-Jiu Chen	Björn Möller
	Uppdragsgivare	Kontaktperson
	Combitech AB	Magnus Gens

Sammanfattning

Detta examensarbete behandlar möjliga lösningar för att förbättra omvärldsuppfattningen för operatörer av tunga fordon vid dålig sikt uppkommen på grund av tät dimma, snöfall eller kraftigt regn. Sensorer och tekniker för att ge en förbättrad omvärldsuppfattning i fordonstillämpningar har studerats tidigare, men dessa studier har inte beaktat robustheten mot snö, regn och dimma eller presenterat resultatet för en operatör. Fyra huvudfunktionaliteter identifieras: lokalisering, hinderdetektering, databehandling och användargränssnitt. Genom en litteraturstudie identifieras möjliga koncept för varje funktionalitet. De identifierade koncepten utvärderas med avseende på genomförbarhet och prestanda genom att studera och analysera litteratur och i vissa fall implementera en prototyp. Med hjälp av koncepten för varje delfunktion utvecklas en systemprototyp som utvärderas i stadsmiljö. Det slutgiltiga förslaget för ett system består av en högprecisions-GPS för lokalisering, en fordonsradar för hinderdetektering, ett UKF-filter för databehandling och användargränssnitt baserat på en 3D-karta, i vilken detekterade hinder visualiseras. Den föreslagna prototypen bedöms vara lämplig för småskaliga applikationer där den aktuella typen av dålig sikt orsakar kostsamma avbrott i verksamheten.

List of Figures

Contents

Chapter 1

Introduction

This chapter provides background information and the purpose description along with the hypothesis and a description of the methodology used and the delimitations of the thesis.

1.1 Background and problem description

Many off-highway machinery operators operate machines in low visibility conditions, such as fog, heavy rain, snow or darkness. These machines are often part of a production system that relies on their operation. If they can not operate as intended, the production is seriously affected or even halted. Typical applications are military applications, forestry, mining, construction equipment or agricultural machines. In some of these applications, downtime is extremely costly and large investments are profitable if they slightly reduce this down-time.

The system owner wants to evaluate a system to support the machinery operators in these situations, to evaluate if the safety can be improved and the down-time reduced. The aim of the system owner is to develop and sell the evaluated system if it is judged feasible and profitable.

1.2 Purpose

The purpose of the project is to develop, evaluate and compare different configurations of a system designed to improve the situational awareness of operators of heavy machinery in low visibility conditions.

1.3 Hypothesis

The hypothesis of the thesis is that it is possible to develop a system that can provide an increased situational awareness for heavy machinery operators.

1.3.1 Questions to be answered by the thesis

In order to answer the hypothesis, a number of questions must be answered. The sensor technology must be determined and the robustness of different sensor technologies must be assessed. The desired coverage of sensors and hence the number of sensors needed must be determined. The thesis must also answer how the sensor data should be interpreted and point out suitable algorithms for these tasks. Furthermore, the thesis must answer which technology that is needed to visualize the static surroundings of the heavy machinery.

1.4 Methodology

The methodology used in the thesis is a combination of literature study and physical prototyping. Initially a literature study is used to derive concepts for each functionality and analyze the concepts. If possible, the concept selection is based upon literature. However, if the literature study can not give a clear answer, the concepts are evaluated by physical prototyping and the choice of concept is based upon this evaluation. Once all functionalities have an identified concept, a system prototype is assembled and evaluated. Based upon this system prototype analysis, conclusions are drawn regarding the feasibility and performance potential of the system.

1.5 Delimitations

To limit the scope of the thesis, no tests are performed in an authentic environment or using authentic vehicles. The system prototype tests are carried out using a passenger car in an urban environment. Furthermore, the robustness to snow, fog and rain conditions are not evaluated by a physical prototype, but the sensor choise is based upon previous research.

Chapter 2

Literature study

This literature study covers different solutions for positioning of a vehicle and perception of its surroundings. Different concepts and solutions are reviewed and their strengths and weaknesses are highlighted.

2.1 Research Scope

This literature study focuses on vehicles that need a good perception of their surroundings, as well as specific technologies to obtain this perception and algorithms to fuse information from different sources to one belief of the vehicle surroundings. Throughout this thesis, the term "belief" will be used to describe the most likely state of a system, given all available information.

A special interest of this literature study is the robustness of different sensors to disturbances, as robustness against one or more of snow, fog and rain conditions is important in the intended applications.

The aim of this literature study is to form a background for concept generation as well as to identify technologies that are suitable for the application.

Despite that the intended application is in an off-road environment, sources considering on-road applications are considered relevant in this literature study. The reasons for this are the many similarities between vehicle perception in on-road and off-road environments. In terms of applications, the regulations and requirements may differ between the two environments, but the differences are considered to be sufficiently small to use on-road references in this literature study.

The following sections describe several areas where technology of interest for this thesis is implemented. Sensor platforms of autonomous vehicles are studied as a starting point for the study, since these sensor platforms have been developed to provide a perception of the environment which is sufficiently accurate to control the vehicle. Furthermore, these platforms are thoroughly tested in real-world conditions and proven to give rich and accurate information of the vehicle surroundings.

Based upon the sensor platform of autonomous vehicles, a few key technologies

and aspects are studied in-depth. These include localization in section 2.3, obstacle detection in subsection 2.4.1, and sensor fusion algorithms in section 2.5.

2.2 Sensor platforms of autonomous vehicles

The sensor platforms of autonomous vehicles have to provide a very accurate estimate of the vehicle surroundings. An early implementation of autonomous vehicles is demonstrated by [Gray, 2000], who has investigated how to perform autonomous operation of a farm tractor. The sensors used are a high-end GPS, an advanced high precision gyroscope and a 2D LIDAR. A LIDAR is a sensor measuring the range to surrounding objects by laser beams, providing a large number of distance measurements at different angles around the sensor. The author prefers a radar sensor, but the price of such a radar at the time of the thesis is too high. The LIDAR used instead suffers from false positive readings caused by dust from the tires of the tractor, which is worked around by ignoring all readings within seven feet, disabling the possibility to avoid any new obstacle within seven feet. Detected objects are stored in a map and used by the tractor itself, such that the tractor can avoid obstacles despite not detecting them by the LIDAR when they are close to the vehicle. This map is a so called certainty map, where objects that have been detected several times have large certainty, while objects present in the map but not detected in a scan are decreased in certainty [Gray, 2000].

A major driver of technological advancements for autonomous vehicle is the DARPA Grand Challenge organized by the Defense Advanced Research Projects Agency (DARPA) within the American Department of Defense. The challenge include to race an autonomous vehicle through a track in the Mojave desert completely without human intervention. The track was unknown to the competitors until a few hours before the race, and the vehicles hence had to be able to drive through unknown terrain. The vehicles need a robust and accurate sensor platform in order to manage the demanding task. Note that algorithms specifying the autonomous behavior of the vehicles are outside the scope of this thesis and not studied in further detail.

The winning vehicle is the contender from Stanford University, named "Stanley". Stanley featured a sophisticated GPS receiver, which included a GPS compass using two antennae, a high-end IMU, two radar sensors using 24 GHz technology with a range of 200 meters and a 20 degree field of view, five 2D laser scanners, here after called LIDARs and a color camera, which is used for road detection. The five LIDARs, visible in Figure 2.1, are facing the road ahead of the vehicle at different angles, and hence measuring the ground profile at different distances in front of the vehicle. The maximum range of the LIDARs used is 25 meters. Beyond this range, the color camera is used to detect drivable areas. The area concluded to be drivable by the LIDARs is used as a training set, and the vision algorithm searches for similar surface further ahead. The radars are used to detect obstacles far in front of the vehicle. The LIDARs return a 3D point cloud based upon the range

Figure 2.1. Stanley with the visible sensor platform, showing the roof mounted LIDARs [Woodward]

measurement and the known angle of each laser beam. These 3D points are mapped to the global coordinate system, where their height coordinates are compared to neighboring points. If their height differ more than a preset threshold, the area is classified as an obstacle. This process is subject to false positive obstacles, i.e. areas erroneously marked as obstacles, due to errors in the vehicle pose estimate, i.e which roll and pitch angle the vehicle has. This inaccuracy is tackled using a probabilistic test, which identifies outliers, i.e. erroneous measurements far from the likely values. The vehicle must also be very robust against sensor failures, and be able to handle a GPS outage, i.e. a period during which GPS position information is unavailable. This is handled using odometry and the IMU. Odometry is measurements of wheel speeds and distances traveled by one or more wheels. This information is used, together with the accelerometers and gyroscopes of the IMU to form an estimate of the vehicle position. To prevent the vehicle position from drifting due to inaccuracies in the accelerometers and gyroscopes, the vehicle reduces its speed to 10 mph and uses a model of how it might move to constrain the drift. As GPS is available, a less restrictive model is used, which allows the vehicle position estimate to move in all directions. A model in this context is a mathematical description of the expected behavior, used as an extra source of information [Thrun et al., 2006b].

The contenders from Carnegie Mellon University used a similar sensor platform, extended by a long range LIDAR suspended using a mechanical gimbal that enables the long range LIDAR to be angled at different areas. The sensor fusion is however different: instead of mapping all sensor measurement into one map and assess the terrain in this map, each sensor creates its own cost map, with high cost corresponding to rough terrain. All these local cost maps are added, and the sum of all costs are used as an estimate of the terrain. The authors from Carnegie Mellon University points out the compromise while using radar. A radar is less accurate than a LIDAR, but is a lot more robust to disturbances such as dust [Urmson et al., 2006].

The next large event within the development of autonomous cars was the DARPA Urban Challenge, organized by DARPA in 2007. As the name implies, the autonomous vehicles were this time driven in an urban environment. A large difference

compared to the Grand Challenge was that the vehicles should this time compete in the same area, and had to handle other autonomous vehicles as well as other traffic, which forced the use of more advanced sensor platforms. This is demonstrated for instance by the winning team from Carnegie Mellon University and their vehicle "Boss". Boss uses a number of 2D LIDARs, including two long range LIDARS, with a sensing distance of up to 300 meters. The 2D LIDARS are complemented by a 3D LIDAR, which can be seen in Figure 2.2, that uses 64 vertically distributed laser beams that spins 360 degrees around the vehicle and measures the range to nearby objects. The data is used to estimate the road shape around the vehicle, together

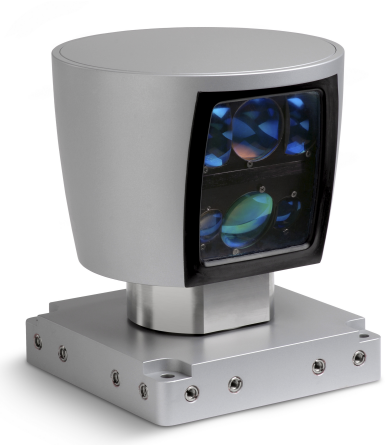

Figure 2.2. The Velodyne HDL-64E sensor used by Carnegie Mellon University [Velodyne, 2011]

with the 2D LIDARs. Using these estimates, the vehicle creates a map of its surroundings and tracks moving objects. In addition to the LIDARs, the vehicle uses a high-end GPS/IMU unit and a number of automotive radars using an operating frequency of 77 GHz. Moving obstacles are tracked, i.e. they are followed during multiple scans and their motion is estimated, using one of two models, the box model and the point model. The box model specifies the length and width of each object and models the movement of each object by the bicycle model, modeling each object by a speed and a steering direction. The point model assumes the object to move with constant acceleration, but has no information of size or shape of the object. The vehicle switches between the models and tries to adapt the best model at any given time. Interestingly, the vehicle generates multiple hypotheses at any point where a tracked object could make a choice, for instance at intersections.

The entry from Stanford University, called "Junior", finished second using a similar sensor platform. Junior used several LIDARs at different angles to the ground to map the height of the ground at each point in the map, this was compared to neighboring points, and marked as an obstacle if the relative height between two points exceeded a threshold. This algorithm did also compensate for roll and pitch

movement of the vehicle. Static obstacles detected by the vehicle are stored in a map that is used to avoid obstacles that are no longer observed by the sensors. Despite using a high-end positioning solution, Junior did not rely on GPS to determine its lateral position, since GPS was considered to be too inaccurate. Instead, the vehicle used LIDAR data to form an estimate of the curb location, and used this to output an offset to the position received from the GPS. Another challenge brought up by the Stanford team was the difficulty to map sensor data to a global map as the GPS might give discontinuities in its position. The challenge was to map two measurements of the same spot to the correct spot in the map, despite the fact that the position offset might be different between the two measurements. This is solved by using local and global coordinates, and use a filter to smooth the offset between the two [Montemerlo et al., 2008].

A rather different approach to the DARPA Urban Challenge is represented by Team Victor Tango and their vehicle Odin. The sensor platform used for Odin was less complex than those of Junior and Boss, it did for instance not incorporate a 3D LIDAR, but a number of 2D LIDARs, including two coordinated four plane 2D LIDARs, essentially being four 2D LIDARs each. Apart from these four plane 2D LIDARs, the vehicle used five planar 2D LIDARs, cameras, odometry and a GPS/INS positioning unit. The team performed well and finished third in the Urban Challenge, showing that much information can be obtained using a comparably simple sensor platform [Bacha et al., 2008].

The dependence on an accurate global position was somewhat questioned by MIT and their entry Talos. The vehicle used a local reference frame and all sensor data was referenced to this system. External information was transferred to the local frame, where it was interpreted. The motivation for this approach was that if a human can safely drive a car without having a clue of its global position, an autonomous vehicle should be able to do the same. The transformation between the global reference system and the local one used data from GPS, INS and odometry. The data fusion algorithms used are a hybrid particle filter and an extended Kalman filter (EKF). The vehicle used a similar sensor platform as Junior and Boss[Leonard et al., 2008] [MIT, 2007].

An Italian research team set up and completed their own VisLab Intercontinental Autonomous Challenge (VIAC), where they used 16 1D laser scanners, called laser beams, which provided 16 distance measurements around the vehicle. The sensor platform relied heavily on cameras and featured a total of 10 cameras and was apart from this similar to the platforms used in the DARPA Grand Challenge. The research team claimed good performance in harsh environments, but did not provide any formal test results [Bertozzi et al., 2012] [Ber, 2010].

While the solutions described earlier in this section have featured high-end sensors, Goncalves et al. have investigated low cost perception platforms for autonomous vehicles in laboratory scale. The implemented sensor platform consists of a low resolution camera and ultrasound distance sensors. The camera is used to detect lanes, while the ultrasound sensors are used to detect objects in the vicinity of the vehicle. The lane detection faces several difficulties, mainly limited visibility

due to low light conditions or other vehicles blocking the camera field of view. The authors furthermore assume that the lane width of the road is known, which might not be true on unknown roads. The camera module is evaluated in a full-scale test on a highway, but the ultrasound sensors are only tested in a small scale environment, and the authors do not consider how these sensors could be scaled up to full-scale vehicles. The robustness of the sensor platform is not evaluated [Goncalves et al., 2007].

2.3 Localization

An important aspect in a system designed to aid a heavy machinery operator is accurate positioning, which is thoroughly covered in the study by Skog and Handel. The authors point out five main sources of position information:

- Global navigation satellite system (GNSS) and other radio-frequency based methods
- Sensors observing vehicle dynamics
- Road maps
- Vehicle models
- Sensors measuring relative to their surroundings

The fifth category was not thoroughly covered by the authors, as it is rarely used by in-car navigation systems, mainly due to the cost of these sensors. The authors did specify four criteria to which the performance of a positioning system is validated:

- Accuracy
- Integrity
- Availability
- Continuity of service

While accuracy, availability and continuity of service is self explanatory, integrity might need further explanation: integrity is the concept of knowing when the positioning solution is inaccurate. To achieve this, redundant information is needed, as the minimum amount of information will not include any contradicting information, which might be used to detect errors. For instance, a GNSS receiver, such as a GPS receiver, calculates the distance to several satellites, and can hence use the amount of contradiction as an estimate of its uncertainty, this is known as receiver autonomous integrity monitoring (RAIM). There are however some disturbances that are hard to manage. For instance, GNSS receivers suffer from multi-path errors, which might arise due to reflections of the signal from the environment, these errors are generally

not discovered by the receiver unit. To further improve the integrity, more sensors, such as odometry, accelerometers and gyros are often used in car navigation systems. Another technique is to use maps of the road network as further information, as a car is most likely to remain on the road.

To further improve the accuracy of GNSS positioning, common mode errors such as atmospheric disturbances can be compensated for by a stationary GNSS receiver at a known position. Since the common mode errors are the same for GNSS receivers close to each other, the relative position between the stationary receiver and the mobile one can be determined. As the absolute position of the stationary receiver and the relative position of the mobile receiver is known, the absolute position of the mobile receiver can be determined. This solution can however not compensate for non-common mode errors such as multi-path errors and sensor noise [Skog and Handel, 2009].

Different techniques for compensated GNSS positioning have been discussed by Mekik and Arslanoglu. The two major techniques discussed are differential GPS (D-GPS) and Real Time Kinematics - GPS (RTK-GPS). Both technologies estimates the time-delay caused by external disturbances, but they differ in how accurately these time-delay measurements are made. While DGPS measures the periods of the GPS carrier wave, RTK-GPS systems are able to measure the phase of the GPS carrier wave, enabling more accurate measurements of the time delay, and hence enabling a more accurate compensation for common mode errors. The result is that while DGPS-systems typically can reach accuracies of 1-5 m, the RTK-GPS system evaluated by the authors reach an accuracy of 2.5 cm, compared to the accuracy of an unassisted GPS: 10-100 m [Mekik and Arslanoglu, 2009].

Another method to localize is to use bearing only information, as implemented with infrared (IR) beacons and receivers by Krejsa and Vechet. By knowing the identity of each beacon and roughly estimating the angle to these receivers, a position can be obtained without knowing the distance to beacons. Although bearing only information is less accurate than distance only information, bearing only implementations might be cheaper and hence preferable in some implementations [Krejsa and Vechet, 2012].

IR technology might also be used to set up reference position markers with patterns recognizable by a camera. To distinguish different markers transmitting different frequencies, a separate IR receiver might be used, as demonstrated by Brassart et al., where each marker is built up using three IR LEDs in an asymmetrical pattern, allowing position as well as heading information from a single measurement, as the reference markers are mounted at a known height above the camera. If a camera is used to measure position, an accurate model of the camera optics is important to achieve good performance [Brassart et al., 2000].

Laser scanners, commonly known as LIDARs, can be used to localize relative to a given map. Baldwin and Newman have implemented localization using two orthogonal 2D LIDARs based upon a 3D map, known in advance. The implementation uses no external sensors, but measures the movement by comparing the offset in distance and angle between two consecutive laser scans, a technique known as

scan-matching. A difficulty using this technique is to avoid moving objects from influencing the output of the estimation. To avoid this influence, a position dependent sensor model can be used and trained to avoid areas typically influenced by moving objects [Baldwin and Newman, 2012].

Another way to incorporate LIDARs is to limit INS drift during GPS outages by fusing LIDAR measurements of known landmarks into the position estimate. This solution has been evaluated in simulation and shows good performance in terms of mean error during the simulation if updates from the LIDAR is sufficiently frequent. In the simulation, an update frequency of 0.2 Hz from the LIDAR shows performance equivalent to continuous GPS/INS localization [Klein and Filin, 2011].

In order to provide localization using sensors that measure relative to their surroundings a map that describes the surroundings is needed. This can either be done in advance, as during the three examples discussed in section 2.3, or by the vehicle as it localizes in an environment. This is known as simultaneous localization and mapping (SLAM). In its simplest form, this is done off-line using data that has been collected by a vehicle moving in the environment. This is implemented by Yang et al., using a LIDAR and odometry to map an area. This map is later used to localize the vehicle within this map during GPS outages. The mapping process can take place without GPS, but the estimate is improved by using the entrance and exit coordinates of the mapping area to compensate for any drift. Once the sensor data has been obtained and processed off-line, the vehicle can later use it to localize without GPS, using an unscented Kalman filter (UKF) to fuse odometry and LIDAR data. [Yang et al., 2011].

As an extension of off-line SLAM, SLAM can also be performed on-line, i.e. in real-time, as the vehicle drives through an environment [Thrun et al., 2006a].

2.3.1 Localization Concepts

The studied techniques addressing localization are:

- D-GPS
- RTK-GPS
- GNSS/INS fusion
- Local beacons
- Sensor measuring relative to their surroundings
- Vehicle models
- Road maps

GNSS-based systems provide a direct measurement of the absolute position, while sensors measuring relative to their surroundings, such as LIDARs, can only

provide an estimate of the relative position to the surrounding. This relative position can be translated to a global position using knowledge of the absolute position of the surroundings. The same technique can be used to obtain a global position from local beacons. As described by Skog and Handel, many different sources of information are generally fused to one belief of the vehicle position. Vehicle models can provide yet more information, but more advanced models need to be vehicle specific. Road maps might be used, but require an assumption that the vehicle stays on the road, and is hence unsuitable for safety systems, where such an assumption would reduce functionality.

The main difference between RTK-GPS and D-GPS is cost and precision, where the choice is a trade-off between the two. Both D-GPS and RTK-GPS can be implemented together with an INS, to improve the update rate, as an INS typically has a higher update frequency than a GPS-system[Zhang et al., 2005].

Another advantage of using several sensors is that it enables fault monitoring, as contradicting information from different sensors will indicate faulty measurements. For instance, a sudden change in GPS-position without a large acceleration measured by an INS, is an unlikely situation, and suggests that either the GPS or the INS has provided faulty measurement. This situation could occur as a multi-path error is induced to the GPS measurements, caused by a GPS-signal being reflected before reaching the GPS antenna. If a vehicle model is used together with the GPS and INS units, the sudden change in position from the GPS is contradicted by both the vehicle model and the INS. It is hence possible to determine that the GPS-measurement is the most likely to be the faulty one.

2.4 Obstacle detection

Related to mapping is obstacle detection, which involves finding objects in the vehicle path that should be avoided. Obstacle avoidance involves detecting objects and finding some path that prevents a collision with the detected object. This is implemented by H.Schafer et al. in an off-road environment using three 2D LIDARs, out of which one is paned to provide a 3D LIDAR scan. In this implementation, data is recorded in a scanner local reference frame, which is fixed to the sensor, and later transformed into the global reference frame using the sensor pan and vehicle pitch and roll angles. As objects are detected, they are stored in a local map in order for the vehicle to know their position as they are outside the field of view of all sensors [H.Schafer et al., 2008].

Obstacle avoidance can be performed using a variety of sensors. An implementation based on LIDAR is described above and an implementation using marine radar has been performed by Larson et al.. The authors also implement tracking, which is the ability to follow moving objects through consecutive radar readings. The tracking is used to predict the path of moving obstacles and based on this information estimate the future positions of the moving objects. This information is used to predict the risk of collisions for different future paths of the USV [Larson et al.,

2007].

In its simplest form, a tracking algorithm uses a predefined model of the objects to be tracked and uses a sensor which is stationary. Dominguez et al. implements a tracking algorithm which does not rely on any prior assumptions of the size or movement of the tracked objects, the scope is however limited to a stationary sensor. Using LIDAR data, data points close to each other are joined in segments, which can be joined with nearby segments into clusters if they are assumed to belong to the same object. These clusters are later associated to previously detected objects and the velocity including the direction of each object is estimated. The authors thoroughly study the implication of varying the threshold for the case when two clusters of measurement data are joined into one object. Ideally, two separate objects, no matter how close they are, should be identified as two separate objects and one object should never be identified as two separate objects. This is however not possible due to occlusion by other objects and the resolution of the sensor. This leads to a compromise between joining close objects together or splitting single objects into several objects. The issue arising from occlusion of objects by other objects in the foreground are addressed by an algorithm which joins segments if they are sufficiently close [Dominguez et al., 2011].

An extension to the solution presented above is implemented by Wang et al. using a 2D LIDAR mounted on a moving vehicle, releasing the assumption of a stationary sensor, without any prior assumptions regarding the size or shape of the objects to be tracked. Each object is identified in a similar fashion to the solution presented above, and tracked using a constant velocity model. Meaning that each object is predicted to be where it was last seen, plus the distance traveled since the last measurement assuming it had the same velocity. Objects are tracked using an extended Kalman filter [Wang et al., 2013].

A more sophisticated version is implemented by Thuy and Leon, based on a similar sensor platform. The authors have implemented a real time particle filter tracking algorithm operating in global coordinates. To predict the motion of the tracked objects, a non-linear process model is used, assuming that the tracked objects can not move sideways without moving forward as well, modeling the typical behavior of a road vehicle [Thuy and Leon, 2009].

Moving object tracking is usually considered as a challenge separated from localization and mapping, but they are all three simultaneously addressed by Wang et al.. The implementation is not dependent on GPS or any other global localization. The main advantage is while mapping and localizing in dynamic environments, as many of the detected objects are dynamic, and must be distinguished from the static background [Wang et al., 2004].

A further extended version is SLAM simultaneous with detection, classification and tracking of moving objects (DTCMO) implemented by Thorpe et al.. The extension is classification, which means that each tracked object is determined to be a certain type of object. To further improve the accuracy, a so called light stripe range finder is used. This system emits a pulsating laser stripe on the side of the vehicle, which is observed by a camera that is coordinated with the laser pulses. If

the ground is not flat, the laser stripe will distort in shape, which can be detected and analyzed by the camera and used to accurately detect the curb. A probabilistic collision detection algorithm is implemented, which tracks potential trajectories of the tracked objects and the ego-vehicle. The number of coincidences between these trajectories is used as a measure of the probability of collision [Thorpe et al., 2003].

Classification can be performed using different sensors. Cassiola has implemented a classification system designed to separate cars from other objects by using a computer vision algorithms to identify typical car attributes such as tail lights and wheels [Cassiola, 2007].

2.4.1 Robustness to disturbances

The low visibility considered by this thesis is caused by snow, dust, fog or rain. It is hence critical that a system designed to aid a machine operator in low-visibility conditions is robust to these conditions.

Foessel et al. investigates the robustness of a short range millimeter wave (MMW) radar in polar environments, where blowing snow poses a large challenge to sensors. As described by the authors, LIDARs and camera systems are unable to provide accurate measurements in these harsh environments, but a millimeter wave radar is tested and shows to be only slightly affected by heavy blowing snow compared to clear conditions [Foessel et al., 1999].

Similar results are shown by Yamauchi as an ultra wide band (UWB) radar is compared to a LIDAR in a foggy environment. The accuracy of the LIDAR is superior to the accuracy of the radar, mainly due to the poor angular resolution of the radar, which shows for instance by walls at a long distance being represented as arcs rather than straight lines. This could be improved by more advanced signal processing. The radar measures the signal strength in 256 segments ranged between 0 and 78 meters and the radar data is interpreted by subtracting an estimate of the background noise and consider the distance region with the highest intensity as the range measurement. While tested in a fog-filled room, the LIDAR produces too short range measurements, as the laser beams fails to penetrate the fog. Meanwhile, the UWB radar is mostly unaffected by the fog. [Yamauchi, 2010].

Another test of the robustness of different sensors is performed by Reina et al., as a MMW radar mounted on an unmanned outdoor vehicle is used to study the radar background, which in this case is the ground returns from the radar, in order to classify the ground. A key feature of the radar in this context is its ability to detect multiple objects in one scan, along with its robustness to disturbances. The authors implement a model to predict the radar return given a pose of the vehicle, this prediction is compared to the MMW radar return, and deviations from the prediction are classified as an obstacle. A comparative test is performed between LIDAR and MMW radar in clear and dusty environments. The LIDAR is heavily influenced by the dust and unable to separate obstacles from dust, while the MMW radar detects all obstacles in both clear and dusty environments [Reina et al., 2011].

2.4.2 Radar

Tutusaus evaluates a Bosch long range radar (LRR) in the context of human detection. A frequency modulated continuous wave (FMCW) radar like the LRR uses Doppler shift to measure distance to and velocities of objects. A FMCW radar provides direct measurements of the velocity of objects in the radial direction seen from the radar sensor, without differentiating position measurements, which has to be done with most other sensors, such as an odometer or a LIDAR. The radar does not provide a direct measurement of the size of the objects, but returns the radar cross section (RCS) as a measurement of how intense a radar echo is. This measurement depends on many factors such as surface material, shape and size. A pedestrian may have varying RCS depending on for instance orientation, body shape, size and clothing. The ability of a radar to detect an object is dependent on its RCS, meaning that targets with smaller RCS, such as humans, are harder to detect than objects with a large RCS, such as cars. The performance of the radar is found to be reduced significantly when used for human detection compared to when used for detecting cars. One large difficulty using a radar is to separate an object having a small RCS from a nearby object having a large RCS. This leads to difficulties to detect a human close to a vehicle, which is one of the main reasons why the author discourages the use of an automotive radar for human detection [Tutusaus, 2008].

Apart from automotive radars, radars using two antennae only providing distance information might be used to localize objects in the plane. Using three distances as shown in Figure 2.3: from the first antenna to the object and back again, denoted r_{11}, from the second antenna to the object and back again, denoted r_{22} and the distance from one antenna, to the object and back to the other antenna, denoted t_{21}, the position of the object can be determined. The two former distances span two circles in the plane and the latter distance spans an ellipsoid in the plane. The object position is calculated using trilateration, assuming that the radar antennae do not receive echoes from objects behind the antennae, as shown by the shaded area in the lower half of the figure. This technique does however suffer from false positive readings, called ghosts. An algorithm to suppress these ghosts can be implemented, but tuning the parameters of that algorithm is a trade-off between suppressing ghosts and detecting all present objects [Rabe et al., 2009].

2.4.3 Obstacle Detection Concepts

The methods studied that can provide obstacle detection are:

- Automotive radar
- Marine radar
- Distance only radar
- LIDAR

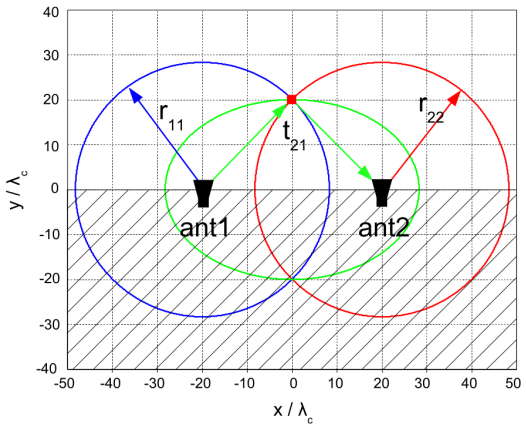

Figure 2.3. Showing how the three distances can be used to determine the position of an object using only distance information [Rabe et al., 2009]

- Laser beams (1D)
- Ultrasonic sensors
- Mono camera
- Stereo camera

All the above technologies may or may not be implemented together with a tracking algorithm in the data fusion.

These sensors are in four main categories, radar, laser scanners, ultrasonic and cameras. Cameras can provide a lot of information, but they are not a good complement to a human driver, since they use the same frequency spectrum as the human eye. If the human operator can not see the surroundings, the cameras are likely to be blind to. They are hence not suitable for this application.

Ultrasonic sensors suffer from poor accuracy and range, which makes them unsuitable for this application.

Laser scanners and radars have complementary strengths, as the laser scanners are more accurate than radars, but the radars are more robust to disturbances. They might hence be suitable for the application.

2.5 Data Fusion

Once a sensor platform is implemented and measurements collected, the data from all sensors must be filtered and joined to one estimate of the vehicle surroundings. This process is called sensor fusion, and is described in this section.

Most modern filters are some variant of Bayesian filters, i.e. they are based upon Bayes' theorem. These include the Kalman filter and the particle filter. The original

Kalman filter, introduced by Kalman in 1960 is the foundation for techniques such as the extended Kalman filter and unscented Kalman filter described later. The idea of the Kalman filter is to use a process model and the control signal of a linear system as an extra source of information, which is used together with the sensor readings. The Kalman filter is designed for linear processes with white Gaussian noise, i.e. the noise should be distributed according to a normal distribution with zero mean. Both inaccuracies in the process model and in the measurements should be accurately described by white Gaussian noise. If these three criteria are fulfilled, the Kalman filter is the optimum estimator of the process states. The filter works in two main steps, the first, called prediction, uses the process model and the control signals to forecast the system states at the next time instant. Based upon the estimated noise levels in the measurement and process respectively, a Kalman gain is calculated. This gain is used in the second main step, called measurement update, where the prediction and the measurement is weighted together. The Kalman gain works as a weighting factor, and increases the influence of the most certain source of information, either the measurement or the prediction [Kalman, 1960].

The main drawback of the original Kalman filter is its limitation to linear systems, as most real-life systems are non-linear. This has been addressed by the extended Kalman filter (EKF), which linearizes the process model around a working point and evaluates the system behavior using Jacobian matrices. This linearized system is then processed as in the original Kalman filter [Thrun et al., 2006a].

However, the EKF relies on an approximation of the linear system and is hence not the ideal estimator, as the Kalman filter is for linear systems. Furthermore it requires the derivation of Jacobian matrices, which might be non-trivial in some cases. As an alternative to the EKF, the unscented Kalman filter (UKF) is introduced by Julier and Uhlmann. The main concept of the UKF is to generate $2n + 1$ sigma points to estimate an n-dimensional variable. These sigma points are transformed through the non-linear system equations and the mean and variance of the system is evaluated through a weighted mean value and the spread of the processed sigma points respectively. The sigma points and their weights are not generated at random, but according to a deterministic algorithm. The UKF is argued to be a more accurate estimation algorithm than EKF when applied to non-linearities [Julier and Uhlmann, 1997].

The UKF is one type of sigma-point Kalman filters (SPKF), others are the central-difference Kalman filter (CDKF) and square root implementations. What differs these filters is how their sigma-points and their weights are selected. It is possible to perform data fusion with delayed measurements, such as GPS, by using a latency compensated SPKF (LC-SPKF). The LC-SPKF is implemented by maintaining the cross-correlation between time steps and hence calculate the correlation between a GPS measurement a number of time steps ago and the current state. van der Merwe et al. has implemented a LC-SPKF and tested it compared to SPKF and EKF. The LC-SPKF is shown to outperform the standard SPKF, which in turn outperforms EKF when applied to navigation of a unmanned aerial vehicle (UAV). The error in the EKF estimate is especially large during aggressive

maneuvers that increase the non-linear behavior. [van der Merwe et al., 2004].

A similar comparison between EKF and UKF was implemented by El-Sheimy et al. with the filters applied to fusion of GPS and INS measurements. EKF and UKF shows similar performance while the behavior is fairly linear. As a large disturbance is induced, and the filters have to operate far from their initial belief, the EKF diverges and is not able to recover, while the UKF converges [El-Sheimy et al., 2006].

While implementing UKF it is less critical to have an accurate process model, since there is no need to differentiate the model to form a Jacobian matrix. It is hence possible to use an approximative representation of the dynamic motion model, such as a Runge Kutta integration, to propagate the sigma points [Zhang et al., 2005].

In addition to Kalman filtering, minima controlled recursive averaging (MCRA)-filtering can be adopted in order to subtract the noise power frequency spectrum from the measured noisy power spectra. This has shown to increase the accuracy of a localization solution using vehicle constraints, an IMU and odometry [Liut et al., 2005].

Another challenge when fusing GPS and IMU is that the two data sources generally have different update rates, typically several IMU measurements are available between each GPS measurement. In general, the GPS measurements are not synchronized with any IMU measurement. To solve this, the prediction can take place once for every IMU measurement, and the latest prediction can be propagated by extrapolation to the time of the GPS measurement [Zhang et al., 2005].

As multiple sensors are fused, their significance will vary over time, depending on the context of the system. A typical scenario where this might occur is when GPS and INS are fused to a position estimate and the GPS signal is interrupted. This problem can be addressed by using a context based filter, which considers the context of the system and trusts readings from sensors with high accuracy in the present context. As one sensor is detected to be inaccurate, it might then be given a low or zero weight in the estimation process. This algorithm has been evaluated and can provide results that are as good as current industrial standards for GPS/IMU fusion, while it outperforms the reference solution during a GPS bias [Caron et al., 2006].

If the system that should be estimated is significantly non-linear or the system noise is significantly non-Gaussian or biased, the Kalman filters is not able to handle the estimation, and more advanced filters are needed. This is usually handled by a particle filter, which propagates a large number of random so called particles, each representing a hypothesis of the system states, through the process model, and evaluates the system behavior based on these particles. These filters are able to handle any distribution as long as the number of particles are sufficient. Implementations of particle filters are usually a trade-off between robustness and accuracy on one hand and computational complexity on the other, depending on the number of particles used. The computational complexity of a particle filter is generally much larger than the complexity of a Kalman filter, while the accuracy of the particle filter is

generally better when applied to non-linear processes [Thrun et al., 2006a].

2.5.1 Data Fusion Concepts

The studied technologies for data fusion are:

- Linear Kalman filter (KF)
- Extended Kalman filter (EKF)
- Unscented Kalman filter (UKF)
- Iterative extended Kalman filter (IEKF)
- Latency compensated - sigma point Kalman filter (LC-SPKF)
- Particle filter (PF)

These filters can be used together with:

- Extrapolation of data to fuse different update rates
- Context based filters

Since the localization and obstacle tracking processes in question does not have significant non-linearities and the processes can be assumed to have fairly Gaussian noise, the benefits of a particle filter would probably be small compared to Kalman filters, and would hence not justify the computational complexity associated with particle filters.

The processes do however contain non-linearities, as the geometrical relationships involved in the localization and obstacle tracking processes are non-linear, and will not be well-described by small angle approximation as the angles increase. This makes the linear Kalman filter an unsuitable solution, as it would not handle these non-linearities well.

The literature study shows several examples of situations where the unscented and extended Kalman filters have been successfully applied in applications similar to the application that is intended in this thesis. While the IEKF and LC-SPKF could be used, the UKF and EKF are likely to work, are less complex to implement and have lower computational complexity.

The choice between UKF and EKF have been covered in the literature study, where the UKF is argued to be the best solution. Due to this, the UKF is chosen as the preferred filter to be used in the prototype.

Chapter 3

Requirements

This chapter identifies the key stakeholders of the system and derives requirements of the system performance and limitations set by the system environment.

3.1 Stakeholder analysis

In order to derive top level requirements for the system, the stakeholders of the system must be identified and their priorities evaluated. The stakeholder analysis in this thesis will take part in three major steps. Stakeholders will be brainstormed, the key stakeholders will be identified among all identified stakeholders and the main interests of these key stakeholders will be summarized.

3.1.1 Stakeholder identification

In this identification, a stakeholder will be defined as someone potentially affecting or being affected by the system in any way. A number of stakeholders are identified and presented below with no particular order:

- The system owner
- Component suppliers
- Machine operators
- Machine owners
- Vehicle manufacturers
- National governments
- Customer of machine services

Most stakeholders presented above could be further separated into for instance employees and owners of each involved stakeholder, but that separation is considered outside the scope of this thesis.

3.1.2 Stakeholder prioritization

The stakeholders identified in subsection 3.1.1 must be evaluated and prioritized. To limit the scope of this thesis and avoid an intractable complexity, a simplified stakeholder analysis is used. In order to better prioritize the stakeholders and their relevance to the system, a stakeholder network overview is reviewed.

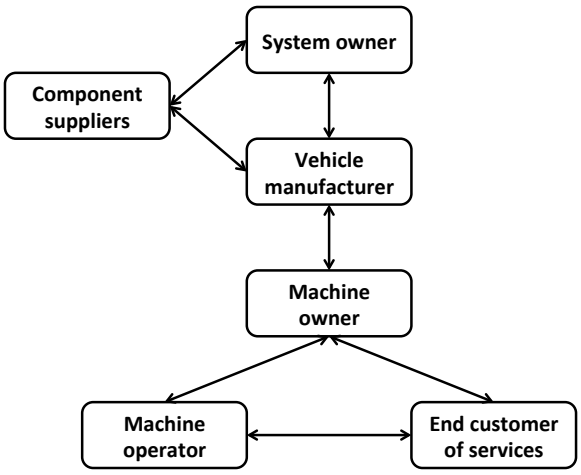

Figure 3.1. The stakeholder network, showing the stakeholder relations.

Figure 3.1 shows the relationships between the different stakeholders. The system owner intends to sell the system to vehicle manufacturers, who in turn sell machines using the system to machine owners. These machine owners use the system together with a machine operator (who may or may not be the same person as the machine owner) and perform some sort of service using the machine to an end customer. This service may be for instance transportation or excavation work. Governmental regulations may affect all other stakeholders, but are omitted in Figure 3.1 to give a better overview.

The simplified stakeholder analysis should be as efficient as possible, i.e. the maximum amount of information should be gathered given the available time frame of the thesis. In order to achieve this, the stakeholders to consider must be carefully

selected. The key aspect in this choice is to use knowledge of other stakeholders gathered by stakeholders themselves. As a vehicle manufacturer has large resources to gather information of what the machine owners and end users demand and prefer, they can provide valuable information about these stakeholders. This implies that the stakeholder analysis can be made faster, while the loss of information as a consequence of the simplification is minimized.

The ideal method for a stakeholder analysis would be to perform the process for each individual stakeholder, but the resources needed for this are not available. In particular, the time frame of this thesis could not allow such an analysis.

The system owner will be considered as a main stakeholder given its position as product owner and project initializer, while governmental regulations will be considered outside the scope of this thesis. The main reason for not considering governmental regulations is that the aim of the thesis is to develop a concept and the governmental regulations can be met during later product development stages. Component suppliers does not determine the requirements of the system, but gives limitations in terms of which products they can deliver, to what cost and in which quantities. They are hence reviewed in some cases, but are not used in chapter 3 to derive system requirements.

3.1.3 The system owner

The system owner wants to evaluate a prototype for a system that would enhance the situational awareness of heavy machinery operated by human operators. The main interest is a fair evaluation of this type of system in terms of performance potential and what type of components that would be needed. Based upon the cost and performance of such a system, the company can take further strategic decisions.

The main technical interest of the system owner is that the system should be modular, such that components from this system can be reused in other systems or in alternative configurations of this system. The number of components used by the system should ideally be as few as possible, to enable retrofitting of the system to existing vehicles.

3.1.4 Vehicle manufacturers

The main technical request of the vehicle manufacturer in question is that existing sensors should be used to the largest possible extent. Furthermore, sensors must be robust, not only in terms of sensor data, but also physically robust to vibrations and harsh conditions such as cold weather and heavy wind. Typically, sensors using moving parts have been found insufficiently robust by the manufacturer.

Sensors mounted to the external parts of the vehicle should be small in order to allow robust installations. The manufacturer distinguishes two separate types of installations: small-scale high-end installations and large-scale mass-production installations. In the former category, rather large internal hardware and high costs would be acceptable. In the latter, all hardware must be fully integrated to the

interior of the vehicle and background information such as maps must be widely available, at the same time, the cost of each unit must be low.

A system used in products (in small or large scale) must be fault tolerant and have large integrity (i.e. identify inaccuracies). It is however not required that the system can compensate for errors that occur, the safety criteria is fulfilled as the system can detect errors and notify the driver.

The manufacturer also mentions an alternative application, where multiple vehicles driving close to each other and blocking the field of view of vehicles behind could gather information and send this information to drivers in following vehicles. This is an example of poor visibility not caused by extreme weather.

3.2 System requirements

The requirements derived based upon the stakeholder interests presented in subsection 3.1.3 and subsection 3.1.4 are categorized into three categories: *compulsory requirements*, *optional requirements* and *preferences*. The compulsory requirements have to be fulfilled if the system is to be considered working, these requirements are presented in subsection 3.2.1. The optional requirements are not strictly binding, but considered as system objectives. These are presented in subsection 3.2.2. The last category, preferences, are requirements that the system aims to fulfill, if it is considered possible and does not compromise any stronger requirement. These requirements are presented in subsection 3.2.3.

3.2.1 Compulsory requirements

The *preliminary* compulsory requirements derived from the stakeholder interests are:

1. The system shall be fault tolerant and have a sufficiently high integrity to inaccuracies.

2. The system shall have a sufficiently high accuracy for the operator to navigate safely.

3. The system shall detect a sufficiently large proportion of potentially harmful obstacles in harsh conditions.

The completion of the project is verified against these requirements and they must hence be specific in order to provide clear success criterions. These requirements are, as stated above, too vague and must be further specified. However, this specification can not be made based upon the information provided by the stakeholders. Further information is obtained by evaluation of the system prototype in chapter 5.

3.2.2 Optional requirements

The optional requirements are, as stated in section 3.2, less strict than the compulsory requirements. The optional requirements derived from stakeholder interests are:

1. The system should be compatible with existing software.
2. The system should be robust to the harsh conditions that the vehicle can be exposed to.

The above requirements are general for all system implementations. In a large scale mass production case, the following requirements are added:

3. The system should be fully integrated into the interior of the vehicle.
4. The system should only use data sources that are widely available.
5. The system should not add a significant cost to the overall vehicle price.

Despite the fact that some of these requirements are somewhat vague, they concretize the stakeholder interests into criterion more easily overviewed.

3.2.3 Preferences

The requirements in this section are considered as desirable, but may be compromised by requirements with higher priority. The preferences derived from the stakeholder interests are:

1. The system may preferably consist of as few units as possible.
2. The system may preferably use existing sensors and sensor technology.
3. The system may preferably not use any large units mounted to the exterior of the vehicle.
4. The system may preferably be as inexpensive as possible.

Chapter 4

Implementation

This chapter describes the implemented prototype and its subsystems, including technical descriptions.

4.1 System architecture

The implemented prototype must provide the following main functionalities:

- Localization
- Obstacle detection
- Data fusion
- User interface

The aim of the prototype is to provide a platform to evaluate the concepts found relevant in the literature study. Some functionalities and aspects will be evaluated using the literature study and others using the prototype.

4.2 Implementation scope

The scope of the thesis in terms of the presented prototype are described in this section.

The mobile prototyping platform shown in Figure 4.1 is not constructed and built within the thesis. However, it is by the thesis expanded with an automotive radar and a physical CAN-network. The test vehicle used in chapter 5 is equipped with hardware within the scope of the thesis.

The development environment and implementation framework is set up by the system owner and made available to the thesis project.

The LIDAR localization system implemented with the mobile prototyping platform is developed by the system owner and made available to the thesis project.

The data produced by the localization system is used in the thesis and interfaced through the ROS-framework described in section 4.3.

The obstacle detection system is developed and implemented within the scope of the thesis and so is the obstacle tracker. This includes hard- and software for interfacing and encoding data from the sensor used. The tracker uses existing algorithms such as UKF and the constant velocity model, but the algorithm selection and implementation is made within the thesis. The main tracker structure shown in Figure 4.3 is developed within the scope of the thesis. Furthermore, the prioritization algorithm and the tracker data structure are developed as part of the thesis.

The user interface is based upon an open source visualization tool, which is configured by and used in the thesis project. The maps used are provided and adapted to the visualization tool by the system owner.

4.3 Implementation framework

The system owner does have an existing mobile prototyping platform suitable for this application, which features a LIDAR, and an on-board computer. The on-board computer features a quad core processor and SSD storage, enabling substantial computational capacity. This platform may be extended with further sensors if needed. The existing platform have an implemented LIDAR localization system, which uses a particle filter for positioning using the LIDAR scans and maps recorded by the platform itself. The existing platform is shown in Figure 4.1, when fitted with an automotive radar. The software of the existing platform is implemented in an Ubuntu environment using the Robot Operating System (ROS). ROS is a framework that among other capabilities provides communication between different software processes, the ability to record, playback and visualize data. Each software process is implemented in a separate C++ file, which can either interface hardware, the user or data from other software processes, called nodes. The communication between different software processes is realized using so called topics, to which one or more nodes can publish data. The data published to a topic is received by one or more nodes if the nodes have subscribed to that topic. The ROS framework provides message handling with configurable buffers, such that short bursts of messages can be handled by slower nodes.

The existing platform is interfaced through wireless network from a stationary computer, allowing all topics to be monitored together with the user interface. The sensor data from the system can be recorded into a data file and later played back, to compare the difference between different parameters or software algorithms applied to the same sensor data.

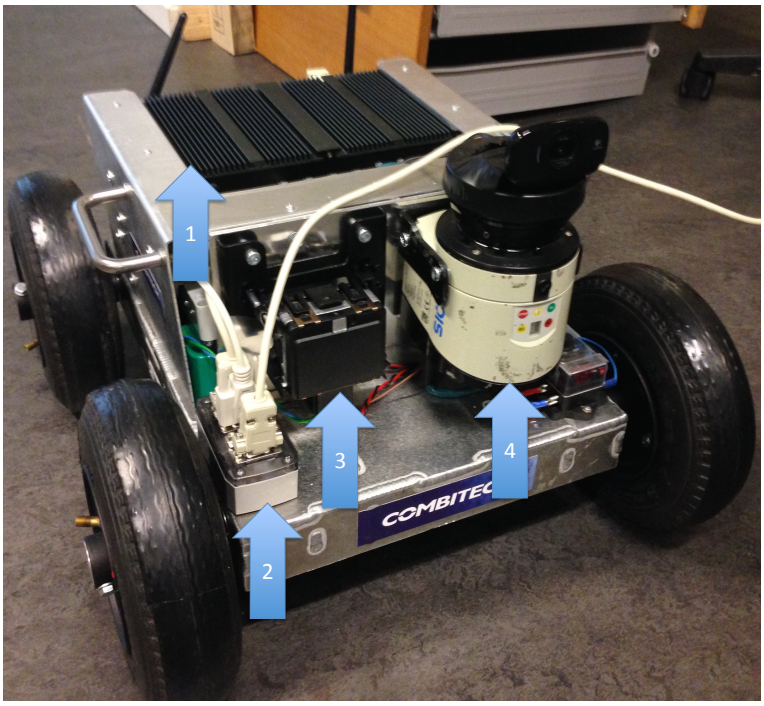

Figure 4.1. The mobile prototyping platform, fitted with an on-board computer (1), a CAN-network (2), an automotive radar (3) and a LIDAR scanner (4).

4.4 Localization

The existing platform does, as stated above, have an implemented localization system, which can be used indoors as well as outdoors. The localization system can record maps and localize within these maps. The mapping and localization uses a LIDAR sensor, which through scan-matching provides a measurement of position change, which is used in the prediction step of a particle filter. The LIDAR sensor is used to determine the most likely position in the measurement update step of the particle filter used. The implemented system is well tested and provides a high precision localization, which can be used during the prototyping.

The interesting aspect in terms of positioning is to determine how accurate the positioning needs to be, in order to provide the intended functionality. This determines which type of localization solution that is suitable in the final system. There are a wide variety of positioning solutions on the market, offering varying accuracies and price levels, as discussed by Mekik and Arslanoglu. This means that if the accuracy requirement can be determined, the localization can be solved by using a system on the market with sufficient performance. In order to determine how less accurate positioning solutions perform, a positioning bias can be added in simulation, and the results reviewed.

4.5 Obstacle detection

The obstacle detection sensor is selected in subsection 4.5.1 and the implementation is described in subsection 4.5.2.

4.5.1 Sensor selection

In the literature study, two obstacle detection sensor types are proposed, LIDAR and automotive radar.

LIDAR sensors provide a very wide field of view and high resolution, but might suffer from insufficient range and poor robustness. A number of manufacturers provide various models that are commercially available. The LIDAR sensors contains moving parts, which make them less robust to vibrations. Furthermore, subsection 2.4.1 shows that the LIDAR sensors are not robust to disturbances caused by dust and fog, while radar sensors provide robust measurements in the same conditions.

Automotive radars are commonly used in safety and comfort features such as adaptive cruise control and emergency braking systems, where they are used to sense the distance to surrounding vehicles. As described in subsection 2.4.1 and subsection 2.4.2, these radars are very robust to disturbances and have a rather large range, but might suffer from insufficient accuracy and a narrow field of view. As a consequence of the rather narrow field of view, several sensors are needed to cover the full surroundings of a vehicle, if that coverage is necessary. The typical usages of these radars have made them commercially available and they are produced in large quantities and several different models.

Due to the lacking robustness of LIDAR sensors shown in subsection 2.4.1, the LIDAR sensors are judged to be unsuitable for the intended application, leaving the automotive radars as the preferred sensor, to be evaluated using a prototype.

4.5.2 Sensor implementation

The automotive radar used communicates using CAN-bus, and is interfaced using several different software nodes. Each CAN message consists of an identifier, a message length specifier and a number of data bytes. Each message can transfer a total of 64 bits of data. The radar interface overview can be seen below, in Figure 4.2.

The CAN-messages transferred from the radar are received by a CAN-adapter, which is a hardware chip considered a black box in this implementation. The automotive radar is connected to the CAN-adapter by a CAN-network that is implemented on the mobile platform. The CAN-adapter is controlled by the serial interface, which can both receive and write CAN-messages through the CAN-adapter. The serial interface interfaces the hardware chip and sends each received sign to the serial decoder. The communication between the interface and the decoder uses a buffer, since the received CAN-messages often arrive in short bursts, too fast to be

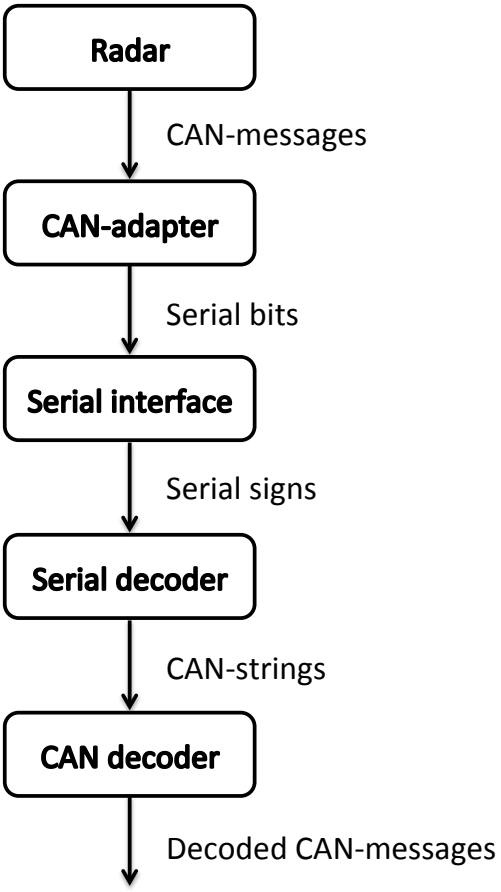

Figure 4.2. The radar interface overview, showing the different hardware and software nodes

decoded as they arrive. The decoder can however handle all data traffic during a measurement cycle, by using this buffer.

The serial signs are processed by the serial decoder, which identifies the start and ender bits of each CAN-message and identifies the received CAN-ID, determines the length of the message and reads the content of the CAN data bytes, containing all transferred data.

This information is transferred to the CAN decoder, which is able to decode the CAN messages into relevant data, by finding the right bits in the CAN data bytes, often using data from several data bytes. The data is later converted to physical quantities and sent to the rest of the system, which can read data such as obstacle velocity or position.

Thanks to ROS, described in section 4.3, data published by one software node can be read by all other nodes, and hence used throughout the system. To maintain

maximum modularity, as desired in section 3.2, the general CAN implementation has been separated from the hardware specific one. The general functionality, which interfaces hardware and publishes CAN-messages to ROS is contained in CAN-adapter, the serial interface and the serial decoder. All implementation specific to the radar is contained in the CAN-decoder software node. If more than one CAN unit is needed in the system, the former three software nodes can be re-used, and the latter can be used in parallel with other nodes, specific to the CAN units in question, to achieve parallel functionality of multiple CAN units.

4.6 Data fusion

Once obstacles have been identified by a sensor, the position of each of these obstacles must be determined by the system. If only one sensor is used, the position of each obstacle could be calculated based upon the orientation of the sensor, the position of the vehicle and the reading from the sensor. As described in section 4.5, several sensors might be needed to provide sufficient coverage of the vehicle surroundings. Since obstacles might be moving and the ego-vehicle will be moving at most times during the system usage, an obstacle is likely to move between more than one sensor and might be observed by more than one sensor at the same time. To handle this situation, and to achieve a more robust positioning of obstacles surrounding the vehicle, the sensor readings must be processed. As described in chapter 2, this task is often performed using some sort of Bayesian filters. In section 2.5, UKF is determined to be the best solution for this implementation.

In order to process the data from sensors into valuable information of the vehicle surroundings, a tracker algorithm is developed. The main steps of this tracker algorithm are shown in Figure 4.3. The obstacle tracker estimates the states of a number of objects, called tracker objects. The number of tracker objects is limited, to restrict the computational complexity of the tracker algorithm, which is necessary to ensure real time performance.

The first step is to receive measurements of the surrounding objects and the ego-vehicle position. The next step is to transform each measurement into global coordinates, where the tracking is performed. Once all measurements are transformed to global coordinates, the existing tracker objects are propagated using the tracker model, to predict the state of all tracker objects at the current time instant, during which the measurement is performed. The selection of which model to use in the prediction step is described further in subsection 4.6.1. The tracker then seeks to associate the measurements to the propagated tracker objects. This association is further described in subsection 4.6.2. If a measurement is successfully associated to an existing tracker object, the tracker performs the measurement update steps of the UKF directly. If however, the measurement is not successfully associated with a tracker object, a new object is created. If the current number of tracker objects in the tracker is lower than the maximum amount of tracker objects, the new object is added to tracker and the prioritize objects step is skipped. If however the amount

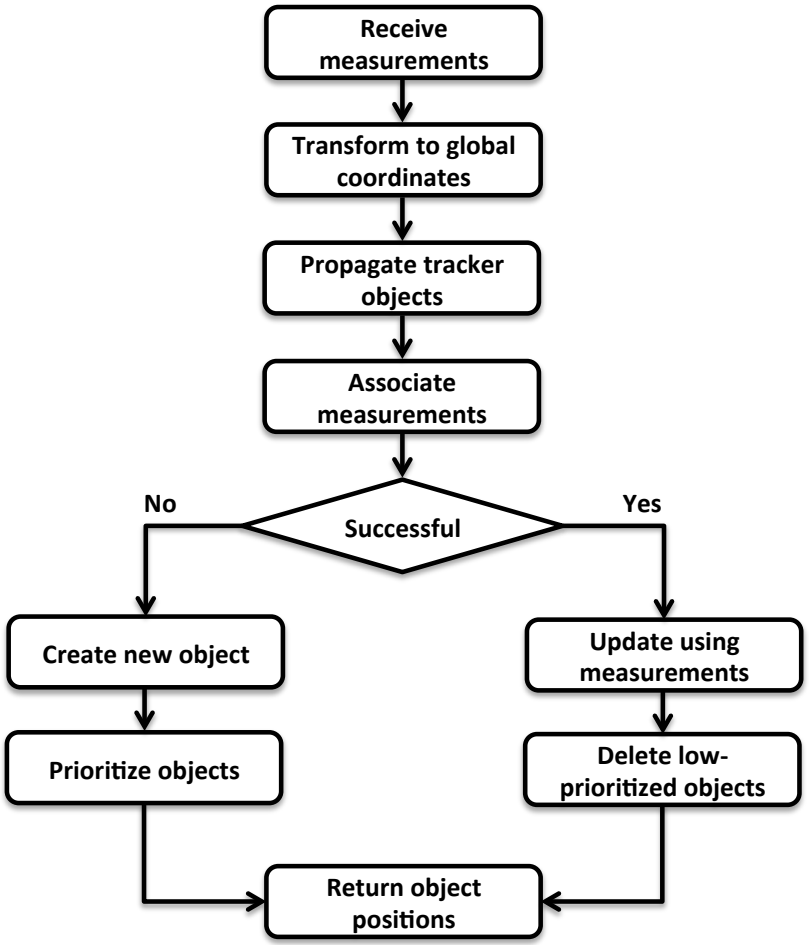

Figure 4.3. An overview of the tracker algorithm, showing the main steps involved in the tracking process during one time instant.

of tracker objects is at the maximum amount, the object with the lowest priority is deleted from the tracker, and the new object is added, unless the new object has the lowest priority. The algorithm performing the prioritization of objects is described in subsection 4.6.4. If the new object is rejected as being the object with the lowest priority, the tracker ignores the measurement associated with this object. If a new object is created, the tracker returns the object state according to the object initialization.

If the measurement is correctly associated and the measurement update is performed, it is followed by a step which deletes objects having a too low priority. The aim with this step is to reduce the amount of objects in the tracker, if many of them have very low priority. After this step, the object state is returned.

4.6.1 Tracker model selection

The tracker model is used in the prediction step of the Kalman filter algorithms. The aim of the model is to use knowledge of how the tracked object is most likely to move together with the measurements. It follows from this that the model has to predict the behavior of the object well, if it should improve the accuracy of the estimate. A poor model could make the filter output less accurate than the measurement itself.

Since the obstacle tracker should track objects varying from vehicles of various sizes to humans and stationary obstacles, the model needs to be very versatile. Alternatively, several models can be used, and the tracker can switch between these. This solution is however more complex to implement and requires the objects to be classified, such that the correct model can be used.

At the same time, the model must not be too computationally expensive, since the prediction step must take place in real time, for each tracked object at each time instant. If a model is too computationally expensive, this real-time criteria can not be met, or the hardware required to perform the prediction in real time might be too costly.

The tracking algorithm should be easily adaptable to different environments. Ideally, the algorithm should be versatile enough to handle all environments without modification.

Some models studied in chapter 2 are:

- Stationary object
- Constant velocity
- Constant acceleration
- Bicycle model
- Non-linear vehicle models

Throughout the report, the coordinate definition in Figure 4.4 is used.

A stationary object model simply assumes that the obstacle has remained in the same position as where it was last estimated to be at.

$$\begin{aligned} x_{t+1} &= x_t \\ y_{t+1} &= y_t \\ \gamma_{t+1} &= \gamma_t \end{aligned} \tag{4.1}$$

The mathematical description of the stationary model is found in Equation 4.1, where x and y denotes the position in the x- and y-direction respectively and γ denotes the planar orientation. The subscripts denotes the discrete time instants. It is obviously a very simple model, with no calculations to be made in the prediction step, the prediction is simply the estimated position at the last time instant. On the other hand, if the obstacle has moved, which is likely in the intended application, the model will not be able to predict that behavior.

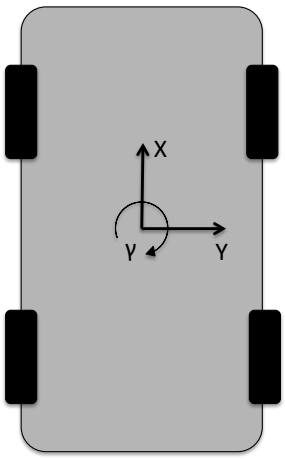

Figure 4.4. The coordinate definition. The direction of travel is the positive x-direction.

A constant velocity model assumes that the tracked object moves with constant velocity and direction. The mathematical description of this model is found in Equation 4.2, where x, y, γ and the subscripts have the same definition as in Equation 4.1. v^x and v^y denotes the translational velocities in the x- and y-direction respectively and ω denotes the planar rotational velocity.

$$\begin{aligned} x_{t+1} &= x_t + v_t^x * \Delta t \\ y_{t+1} &= y_t + v_t^y * \Delta t \\ \gamma_{t+1} &= \gamma_t + \omega * \Delta t \\ v_{t+1}^x &= v_t^x \\ v_{t+1}^y &= v_t^y \\ \omega_{t+1} &= \omega_t \end{aligned} \tag{4.2}$$

This model is still relatively simple in terms of computational requirements, as a linear prediction using the estimated velocity and direction requires few calculations. The constant velocity model has no drawbacks in terms of tracking performance compared to the constant position model, as it can track stationary objects as having the constant velocity zero, but is more general and can handle non-stationary objects better. It does on the other hand require more calculations at each time instant and more data to be transferred between time instants.

The constant acceleration model assumes, as the name implies, that the tracked object moves with constant acceleration. The mathematical description of the con-

stant acceleration model is found in Equation 4.3.

$$\begin{aligned} x_{t+1} &= x_t + v_t^x * \Delta t + \frac{a_t^x * \Delta t^2}{2} \\ y_{t+1} &= y_t + v_t^y * \Delta t + \frac{a_t^y * \Delta t^2}{2} \\ \gamma_{t+1} &= \gamma_t + \omega * \Delta t + \frac{\dot{\omega}_t * \Delta t^2}{2} \\ v_{t+1}^x &= v_t^x + a_x^t * \Delta t \\ v_{t+1}^y &= v_t^y + a_y^t * \Delta t \\ \omega_{t+1} &= \omega_t + \dot{\omega} * \Delta t \\ a_{t+1}^x &= a_t^x \\ a_{t+1}^y &= a_t^y \\ \dot{\omega}_{t+1} &= \dot{\omega}_t \end{aligned} \tag{4.3}$$

In Equation 4.3, x, y, γ, v^x, v^y, ω, Δt and the subscripts have the same definition as in Equation 4.2. Furthermore, a^x, a^y and $\dot{\omega}$ denotes the translational and rotational accelerations.This is an extension to the constant velocity model, which allows more accurate tracking at the cost of increased computational complexity. The increased tracking performance is only beneficial if the tracked objects have varying velocity, but it can also handle objects with constant velocity equally well as the constant velocity model. The main drawback of the constant acceleration model compared to the constant velocity model is increased computational complexity.

The bicycle model is yet more advanced than the constant acceleration model and estimates the steering angle of the vehicle [Milliken and Milliken, 1995]. It has been used for instance by Carneigie Mellon University, the winners of the DARPA Urban Challenge, as described in section 2.2. The bicycle model assumes that the tires do not slip. A figure of a bicycle model can be seen in Figure 4.5. It follows

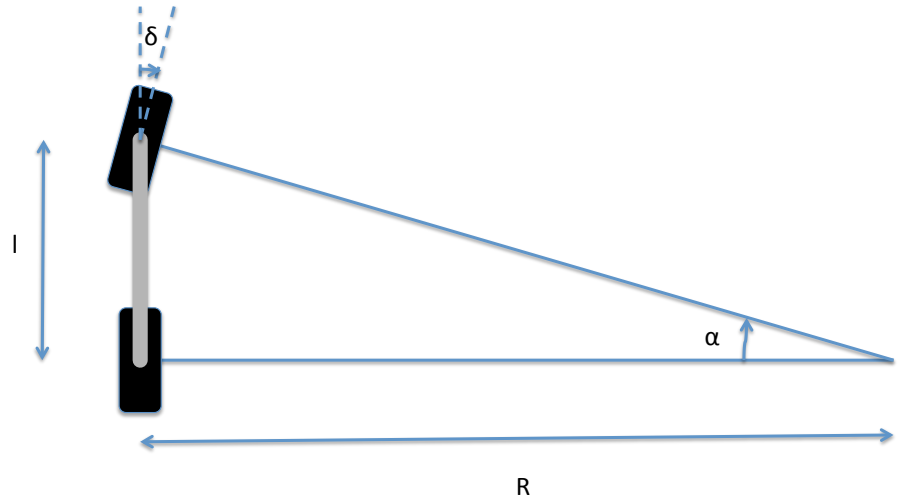

Figure 4.5. The bicycle model, with turning radius R, steering angle δ, wheelbase l and wheelbase angle α.

from the geometrical relations that:

$$\delta = \alpha \tag{4.4}$$

Since l and R are perpendicular, it follows from trigonometrical relations that:

$$tan(\alpha) = \frac{l}{R} \tag{4.5}$$

Combining Equation 4.4 and Equation 4.5 defines the turning radius as a function of the vehicle wheelbase and the steering angle:

$$R = \frac{l}{tan(\delta)} \tag{4.6}$$

The movement between two time instances in the tracker is shown in Figure 4.6. The relation between the turning radius, the angle β, and the traveled distance s is

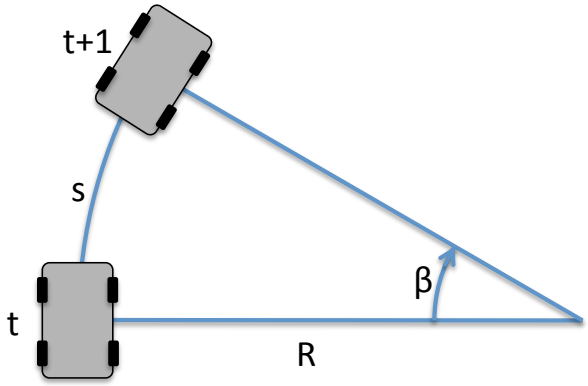

Figure 4.6. The movement between two time instants, showing the vehicle at time t and $t+1$ having moved a displacement s along a turn with turning radius R.

given by Equation 4.7.

$$s = R * \beta \tag{4.7}$$

By defining Δt as the time between the two time instants t and $t+1$, Equation 4.7 can be expressed in terms of the translational velocity v, as stated in Equation 4.8.

$$\beta = \frac{v * \Delta t}{R} \tag{4.8}$$

By using the coordinate definition from Figure 4.4, the distance traveled in the x-, y- and γ-direction can be expressed in terms of the angle β and the turning radius R. This is shown in Equation 4.9, where Δ denotes the difference between the two time instants.

$$\begin{aligned} \Delta x &= R * sin(\beta) \\ \Delta y &= R * (1 - cos(\beta)) \\ \Delta \gamma &= \beta \end{aligned} \tag{4.9}$$

Using Equation 4.6, Equation 4.8 and Equation 4.9, the movement equations of the bicycle model can be stated in Equation 4.10.

$$\begin{aligned} x_{t+1} &= x_t + \frac{l}{tan(\delta)} * sin(\frac{v*\Delta t*tan(\delta)}{l}) \\ y_{t+1} &= y_t + \frac{l}{tan(\delta)} * (1 - cos(\frac{v*\Delta t*tan(\delta)}{l})) \\ \gamma_{t+1} &= \gamma_t + \frac{v*\Delta t*tan(\delta)}{l} \end{aligned} \tag{4.10}$$

Equation 4.10 contains several non-linear calculations, which if Δt and the steering angle δ are sufficiently low, can be simplified. A low Δt corresponds to a small angle β, which in combination with a small δ allows small angle approximation. This simplification is however omitted in this thesis.

The bicycle model uses, as seen in Equation 4.10, the wheelbase of the vehicle, and estimates the translational velocity v and the steering angle δ. If the wheelbase is unknown, which it is in an obstacle tracking application with varying obstacle types, the wheelbase must be estimated for each obstacle as well. This is implemented by adding the wheelbase to the tracker state. The main drawback of the bicycle model compared to the constant acceleration or velocity models is its increased implementation complexity. It requires properties of the tracked vehicles to be estimated by the tracker, and it is not designed to track other types of obstacles, such as pedestrians.

A non-linear vehicle model is a the most advanced model category. It uses a more or less simplified vehicle model, and can if used with the correct parameters track the vehicle behavior with large accuracy. These models are however dependent on accurate vehicle parameters, which will vary substantially between different vehicles, and must hence be estimated by the tracker, similar to how the wheelbase must be estimated if a bicycle model is used. The amount of calculations to be made at each time step and the amount of data to be saved between time steps increases with the complexity of the model. There are many different versions of non-linear vehicle models, but the generic form of such a model is shown in Equation 4.11

$$X_{t+1} = f(X_t) \tag{4.11}$$

The variable X denotes the obstacle states, containing all the quantities that are to be estimated, such as velocity, size and position. The prediction of the states at the next time instant is a non-linear function of some or all of the states at the present time instant.

As the complexity of the model increases, it is not only the number of calculations that increases, but also the amount of data that needs to be saved and processed. This can be seen when comparing for instance Equation 4.1 with Equation 4.3. The amount of variables from time instant t that are needed when predicting the time instant $t + 1$ is significantly higher when using the more advanced model, and so is the amount of calculation to be made.

In this prototype implementation, a simple yet efficient model is preferred. The bicycle model and the non-linear vehicle models are considered too advanced and

complex for the scope of the thesis, and do require many calculations to be made and a lot of data to be saved between time instants. They are hence not selected. The stationary object model is however considered too simple, as it can not predict any motion, which is not acceptable in an environment where many obstacles are moving. Many of the obstacles, such as pedestrians and vehicles move with a constant velocity and changes this velocity rather slowly. This behavior is captured by the constant velocity model, which is judged to be sufficiently accurate and is successfully used by Wang et al., while being relatively simple to implement. The constant velocity model is chosen because of it being the least complex model that has been successfully implemented in a similar application, and is thus judged likely to work in the prototype.

4.6.2 Association of measurements

Each measurement from a sensor observing the vehicle surroundings does likely contain multiple objects, such as buildings, vehicles or pedestrians. Each of these obstacles is tracked by an obstacle tracker using the model described in subsection 4.6.1, where measurements are fused together by using the model together with sensor readings. In order to perform the tracking, each object in the measurement must be associated with the corresponding object in the obstacle tracker. A typical measurement scenario is shown in Figure 4.7. Typically, all the vehicles, including

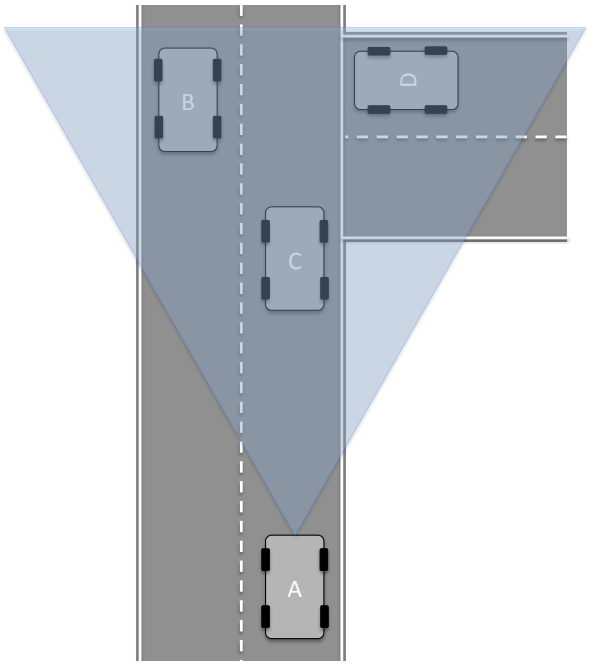

Figure 4.7. A typical vehicle surrounding measurement situation, where the ego-vehicle A observes the surrounding through a sensor with the field of view indicated by the blue triangle. B, C and D are other vehicles.

the ego-vehicle, may be moving during the tracking, which further complicates the association. The association algorithm can use the available information from the sensor and the tracker to determine the most likely association between tracker and sensor objects. In addition to the position of an object the sensor can usually provide additional information such as size and shape.

The tracker will at each time instant use the model described in subsection 4.6.1 to predict the position and velocity of each object. These predictions represents the most likely state of each obstacle at the time instant. Due to this, the model used is important to ensure proper association performance. A likely measurement situation one time instant after the situation shown in Figure 4.7 is shown in Figure 4.8. If the

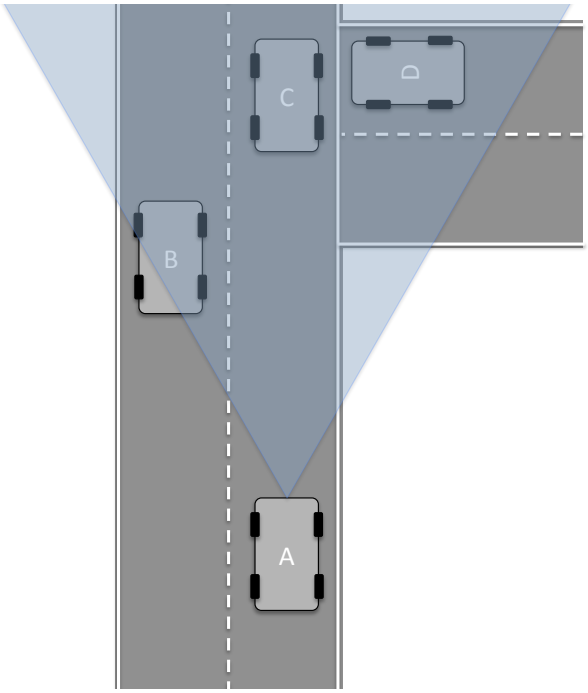

Figure 4.8. A typical vehicle surrounding measurement situation, shown one time instant later than in Figure 4.7, where the ego-vehicle A observes the surrounding through a sensor with the field of view indicated by the blue triangle. B, C and D are other vehicles, and all vehicles except for D has moved.

model would not have predicted the movements of the vehicles shown in Figure 4.7 and Figure 4.8, the association would likely have been faulty, since B and C have moved. Note that the ego vehicle motion influences the relative measurements of surrounding objects.

It is intuitive that the measurement closest to each prediction should be associated with the predicted object. However, this method would not make use of additional information such as size and shape. In order to weigh this additional information into the association process, a more advanced algorithm is needed.

A commonly used algorithm to make use of more data is the Mahalanobis dis-

tance [Thrun et al., 2006a]. The Mahalanobis distance is calculated as a weighed average of the differences between the expected state values and the measurements. The weighting is made based upon the uncertainties of each state. By using the uncertainties, two issues are resolved, first of all, the uncertainties are usually roughly proportional to the state size, meaning that states of different size can be weighted together without the larger state being dominant. In addition, weighting by the uncertainties means that the most certain properties of the measurement have a larger influence in the association process.

The basic concept is to use the Mahalanobis distance to estimate how likely each measurement is to belong to each tracker object. Note that one Mahalanobis distance is created for each combination of measurement and tracker object. The lower the Mahalanobis distance between an object and a measurement, the more likely it is that the measurement belongs to that object. If the smallest Mahalanobis distance between a measurement and all tracker objects is sufficiently large, a new tracker object is created, since it is unlikely to belong to any existing tracker object.

For each tracker object, a prediction of the vehicle state at the next time instant is determined based upon the selected tracker model. Using the predicted vehicle state and a model of the measurement, the expected measurement for each tracker object is determined. These predicted measurements, called $\bar{z}$, are compared to the real measurements from the sensors, called z, the variables $\bar{z}$ and z denote vectors containing all measurements. Given the two measurement vectors and their uncertainty matrix, denoted S, the Mahalanobis distance, denoted D_m, is given according to Equation 4.12

$$D_m = \sqrt{(z - \bar{z})^T S^{-1} (z - \bar{z})} \tag{4.12}$$

The larger the differences between the prediction and the actual measurement, the larger the Mahalanobis distance for the combination of sensor measurement and measurement prediction, hence indicating that the measurement is more unlikely to belong to the obstacle from which the measurement prediction is derived. The multiplication with the inverse of the uncertainty matrix S implements the weighting described earlier, where measurements having a lower uncertainty are weighted more heavily.

4.6.3 The unscented Kalman filter

The Unscented Kalman Filter (UKF) is a version of the Kalman Filter that is designed to handle non-linear processes without using derivatives. As the original Kalman filter described in section 2.5, the UKF uses the process model together with the measurements to estimate the vehicle surroundings. The process model used is the tracker model described in subsection 4.6.1. The fundamental idea behind the UKF is to use the unscented transform to handle the non-linearities. This involves determining a number of so called sigma points, which are propagated through the process model. An overview of the UKF algorithm is found in Figure 4.9. The

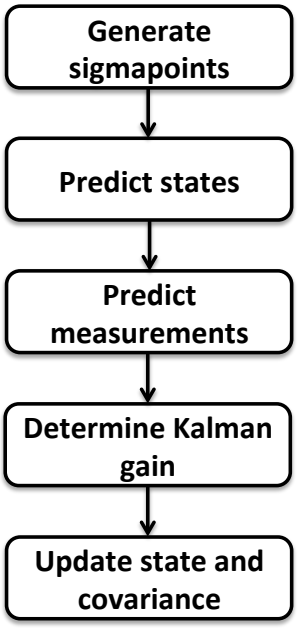

Figure 4.9. An overview of the unscented Kalman filter, showing the main steps involved in the estimation process for one time instant.

number of sigma points ,n_Σ, is related to the number of variables in the tracker state, n_X, according to Equation 4.13.

$$n_\Sigma = 2n_x + 1 \tag{4.13}$$

The sigma points are determined based upon the filter belief of mean value μ_{t-1} and covariance Σ_{t-1} at the previous time instant. The first sigma point χ^0 is equal to the belief, while the other sigma points are determined as the mean value shifted in one state. This is implemented according to Equation 4.14 for $i = 1, ..., n_X$.

$$\chi^i = \mu_{t-1} + (\sqrt{(n_X + \lambda)\Sigma_{t-1}})_i \tag{4.14}$$

Similarly, the sigma points corresponding to $i = n_X + 1, ..., 2n_X$ is determined according to Equation 4.15

$$\chi^i = \mu_{t-1} - (\sqrt{(n_X + \lambda)\Sigma_{t-1}})_{i-n_X} \tag{4.15}$$

In Equation 4.14 and Equation 4.15, λ determines the spread of the sigma points, and is calculated according to Equation 4.16. In Equation 4.14 and Equation 4.15, the right hand index denotes the column to be used from the matrix.

$$\lambda = \alpha^2(n_X + \kappa) - n_X \tag{4.16}$$

In Equation 4.16, α and κ are parameters determining the distribution of sigma points around the mean value.

As the prediction is determined in the UKF, each sigma point is propagated through the process model and the result from each sigma point is weighted together to form the filter belief of the mean value. The spread of the result from the different sigma points is used to estimate the covariance of the prediction. The weights used to calculate the predicted mean w_m are determined according to Equation 4.17.

$$w_m^i = \begin{cases} \dfrac{\lambda}{n_X + \lambda} & \text{for } i = 0 \\ \dfrac{1}{2(n_X + \lambda)} & \text{for } i = 1, ..., 2n_X \end{cases} \tag{4.17}$$

The weights used to calculate the predicted covariance w_c are determined according to Equation 4.18

$$w_c^i = \begin{cases} \dfrac{\lambda}{n_X + \lambda} + 1 - \alpha^2 + \beta & \text{for } i = 0 \\ \dfrac{1}{2(n_X + \lambda)} & \text{for } i = 1, ..., 2n_X \end{cases} \tag{4.18}$$

Given a propagation function $g(X)$, which describes the model used to predict the motion of the obstacle being tracked, the propagated sigma points Υ are determined according to Equation 4.19

$$\Upsilon^i = g(\chi^i) \tag{4.19}$$

Once the n_Σ sigma points have been propagated, the mean $\bar{\mu}$ is determined using a weighted average of the propagated sigma points determined in Equation 4.19 and the weight determined by Equation 4.17.

$$\bar{\mu} = \sum_{i=0}^{n_\Sigma} w_m^i \Upsilon^i \tag{4.20}$$

Similar to Equation 4.20, the covariance of the prediction $\bar{\Sigma}$ is determined according to Equation 4.21

$$\bar{\Sigma} = \sum_{i=0}^{n_\Sigma} w_c^i (\Upsilon^i - \bar{\mu})(\Upsilon^i - \bar{\mu})^T + R_t \tag{4.21}$$

The matrix R_t is the model noise matrix, which increases the predicted uncertainty, reflecting the fact that there is uncertainty in the model.

Based upon the predicted mean $\bar{\mu}$ and covariance $\bar{\Sigma}$, new sigma points are extracted analog to Equation 4.14 and Equation 4.15, substituting μ_{t-1} with $\bar{\mu}$ and Σ_{t-1} with $\bar{\Sigma}$. The new sigma points are denoted $\bar{\chi}_t$.

The UKF uses a measurement model, which predicts the measurements for each sigma point. The measurement model is denoted h, and relates the sigma points and the predicted measurements sigma points $\bar{Z}_t$ according to Equation 4.22

$$\bar{Z}_t = h(\bar{\chi}_t) \tag{4.22}$$

Given the measurement sigma points, the predicted measurement is calculated as a weighted sum according to Equation 4.23

$$\bar{z}_t = \sum_{i=0}^{n_\Sigma} w_m^i \bar{Z}_t^i \tag{4.23}$$

The measurement prediction uncertainty, denoted S_t, is calculated according to Equation 4.24

$$S_t = \sum_{i=0}^{n_\Sigma} w_c^i (\bar{Z}_t^i - \bar{z}_t)(\bar{Z}_t^i - \bar{z}_t)^T + Q_t \tag{4.24}$$

The matrix Q_t is the covariance of the additive measurement uncertainty, which reflects the uncertainty associated with the measurement.

The cross-covariance $\bar{\Sigma}_t^{x,z}$ between the tracker states and the measurements is calculated according to Equation 4.25

$$\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{n_\Sigma} w_c^i (\bar{\chi}_t^i - \bar{\mu}_t)(\bar{Z}_t^i - \bar{z}_t)^T \tag{4.25}$$

The measurement prediction uncertainty from Equation 4.24 is used together with the cross covariance from Equation 4.25 to calculate the Kalman gain K_t, which is used to determine the filter weighting between the prediction by the process model and the measurements. The Kalman gain is calculated according to Equation 4.26

$$K_t = \bar{\Sigma}_t^{x,z} S_t^{-1} \tag{4.26}$$

The Kalman gain K_t is used to calculate the estimate of the states and their uncertainty according to Equation 4.27 and Equation 4.28.

$$\mu_t = \bar{\mu}_t + K_t(z_t - \bar{z}_t) \tag{4.27}$$

$$\Sigma_t = \bar{\Sigma}_t + K_t S_t K_t^T \tag{4.28}$$

The final estimate μ_t is returned as the estimate from the filter given both the process model and the measurement. This estimate is returned together with its covariance matrix Σ_t. In Equation 4.27, the importance of the Kalman gain is clear. The size of the Kalman gain determines the influence of the measurement to the prediction from the model. If the Kalman gain is small, the influence of the measurement is small, while a large Kalman gain will allow the measurement to have a large influence on the returned estimate μ_t.

4.6.4 Prioritization algorithm

Since the obstacle tracker is run in real time, the amount of objects in the tracker must be limited. As this limitation is implemented, it is preferable if the most

relevant objects can be separated from the least important ones. To implement this prioritization of objects, a simple prioritization algorithm is implemented. The main objective of the implemented algorithm is to provide a flexible framework that can easily be adjusted to different environments and types of objects. The implemented algorithm is a linear combination of the state vector and a weight vector. By using negative weights, a smaller state value gives increased priority. The weight vector is shown in Equation 4.29.

$$w_{prio} = \begin{pmatrix} w_p^x & w_p^y & w_p^\alpha & w_p^{v_{trans}} & w_p^\omega & w_p^P & w_p^{A_{rcs}} \end{pmatrix} \tag{4.29}$$

Using this weight vector, the prioritization number of the i:th object, denoted p_i is determined according to Equation 4.30, where X_i denotes the state vector of the i:th tracker object.

$$p_i = w_p * X_i \tag{4.30}$$

The resulting scalar p_i can easily be compared between different objects and the most relevant objects can be separated from less relevant ones.

4.6.5 Tracker data structure

The obstacle tracker is implemented using the process model described in subsection 4.6.1, the association algorithm described in subsection 4.6.2 and the UKF described in subsection 4.6.3.

The tracker relies on the assumption that the ego vehicle position provided by the localization algorithm is known and accurate. Given this assumption, the absolute position of each measured object is determined from the absolute position of the ego vehicle and the relative measurement between the ego vehicle and the object. These absolute positions are used in the tracker, as the tracker model used predicts the movement of surrounding obstacles in absolute positions, rather than in the relative position between the object and the ego vehicle.

Furthermore, the tracker runs one separate obstacle tracker for each surrounding object, where the measurement is paired with a tracker object using the association algorithm. As described in subsection 4.6.2, if a measurement is sufficiently unlikely to belong to any existing tracker object, a new tracker object is created.

The state vector used in the tracker is presented in Equation 4.31

$$X = \begin{pmatrix} x \\ y \\ \alpha \\ v_{trans} \\ \omega \\ P \\ A_{rcs} \end{pmatrix} \tag{4.31}$$

The variables x, y, α and v_{trans} denote the x- and y- position, the rotational orientation and the translational velocity respectively. They are illustrated in Figure 4.10

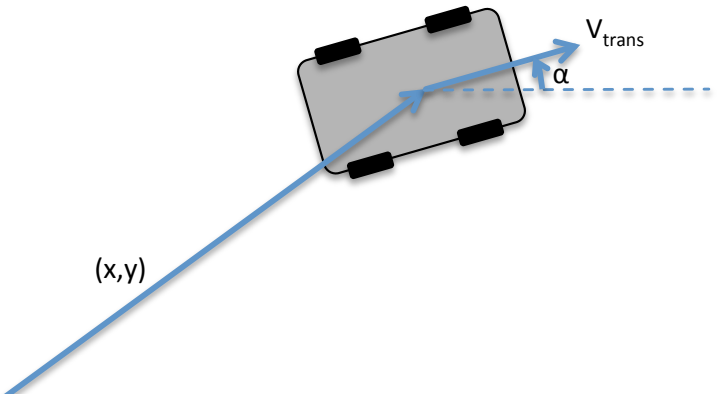

Figure 4.10. The object coordinates defined. x and y denotes the absolute position, α denotes the absolute orientation and v_{trans} denotes the translational velocity.

The state variable ω denotes the rotational velocity of the object, being the derivate of α. P is the existence probability, returned from the internal signal processing of the radar sensor and used to prioritize objects. Finally, A_{rcs} is the radar cross section area, returned by the radar and used to make the association of measurements to tracker objects more robust. Note that this radar cross section area is not a direct measurement of size or geometrical cross section area, but depends on several factors such as orientation and shape of the measured object.

The first five states in Equation 4.31 parallels the constant velocity model described in Equation 4.2, but the velocity perpendicular to the direction of travel has been assumed to be zero and omitted from the state vector, reflecting the assumption that the surrounding objects are not sliding.

4.7 User interface

The user interface consists of two major parts, the visualization of map and sensor data, described in subsection 4.7.1 and the parameter set up, described in subsection 4.7.2.

4.7.1 Visualization

The user interface presents the information gathered by the system to the machine operator, such that the operator can use the information to control the vehicle. Within the scope of this thesis a prototype user interface is implemented using the robot operating system (ROS), which is introduced in section 4.3. The tool within ROS that is used is called rviz and provides a framework where data available in the ROS system can be visualized. The ego-vehicle position is exported to rviz and visualized along with the surrounding objects identified by the data fusion algorithms.

The maps used for visualization within this thesis are static, but must be brought in to the ROS framework in order to be visualized by rviz. The system owner has provided a 3D mapping node in ROS, which uses maps consisting of a geo-referenced aerial photograph together with height data for each pixel. Each pixel is placed at the height specified by the map and visualized as a so called point-cloud, where each pixel has a color determined by the aerial photograph and the position in the 3D space is determined by the height data and the geo-referencing. The pixel density is adjustable, such that the resolution can be lowered below the full resolution of the original map. It is furthermore possible to only display a subset of the original map, which can lower the computational demand.

4.7.2 Parameter adjustment

The implemented system has a number of parameters that must be tuned to obtain satisfactory performance. In order to significantly speed up the tuning process, and allow the prototype to be more flexible, a dynamic parameter system is implemented using a ROS library called dynamic reconfigure. The dynamic parameter system allows parameters to be adjusted during run time, enabling instant feedback to parameter changes. This is preferable and allows more configurations and parameters to be tested during the same time span, compared to an approach where re-compiling the software is necessary to change the parameters.

Chapter 5

Evaluation

This chapter presents the evaluation of the different subsystems and their performance as a system prototype.

5.1 Evaluation strategy

5.1.1 Evaluation areas

The areas to be evaluated are positioning, maps and obstacle detection.

5.1.2 Evaluation methods

In terms of positioning, the accuracy requirement must be determined. The strategy to determine this requirement is to add noise to the precise position data obtained from the LIDAR positioning in order to simulate less accurate position data.

Different maps are tested to evaluate to what extent they provide relevant information to the operator of the vehicle. The available maps are an abstract 2D map and a photo-realistic 3D map, which can also be viewed straight from above, to simulate a photo-realistic 2D map.

The obstacle detection is evaluated to several criteria: sensor range, field of view, accuracy and ability to detect obstacles. As a reference to the obstacle detection, video is captured together with the sensor data, and can be used to indicate the true positions of surrounding obstacles. The robustness of the sensor platform is not evaluated by these tests, but the results found in subsection 2.4.1 show that the sensor type used is very robust to disturbances.

5.2 Test design

5.2.1 Evaluation vehicle and setup

The sensors used by the prototype are an automotive radar, a LIDAR sensor and a camera. The test vehicle is a passenger car, with the LIDAR sensor fitted to the

roof, the automotive radar fitted to the front of the car and the camera mounted on the dashboard, facing forward. The test vehicle can be seen in Figure 5.1.

Figure 5.1. The test vehicle, showing the front-mounted automotive radar (1) and the roof-mounted LIDAR (2).

To evaluate the prototype, the LIDAR is used to localize the vehicle in a 2D grid map created by the vehicle itself. This grip map is geo-referenced and overlayed by a photo-realistic 3D map, into which obstacle data is plotted. The basic test process can be summarized in the following steps:

1. Create an abstract 2D grid map of the area to be tested.

2. Geo-reference this map manually by localizing key points in the grid map.

3. Localize the vehicle in the 2D grid map using the existing LIDAR localization system.

4. The local position in the 2D grid map together with the geo-referencing of the map gives the global position.

5. The photo-realistic 3D map is geo-referenced and can be overlayed onto the grid map.

6. Enable the overlay of obstacle data into the 3D map.

7. Evaluate system performance by using captured video as reference.

5.2.2 Test cases

Localization

In order to evaluate requirement 2 in subsection 3.2.1 the desired accuracy of the positioning solution to be used must be determined. This is determined by simulating typical inaccuracies of different positioning systems. These inaccuracies can be found in section 2.3 and are presented in Table 5.1.

Table 5.1. Typical inaccuracies of different positioning solutions

Localization method	Typical accuracy [m]
Unassisted GPS	10-100
D-GPS	1-5
RTK-GPS	0.025

The inaccuracy of a RTK-GPS system is very small compared to the size of a vehicle, and the position of such a system is assumed to be the ground truth position. To verify the performance of other available systems, position biases of 1 m, 5 m, 10 m and 100 m respectively are simulated and the data reviewed. The minimum and maximum values of the inaccuracies give an estimate of how the system can be expected to perform during ideal and worst case situations, respectively.

Map

To determine the requirements for the underlying map, three types of maps are available. Apart from map type, the resolution must be determined, such that the accuracy of the map fulfills requirement 2 in subsection 3.2.1.

The simplest map is the grid map created by the vehicle itself. The details behind the SLAM (Simultaneous Localization And Mapping) algorithm used to create this abstract map is outside the scope of this thesis, but briefly the map is constructed in 2D by the measurements from the LIDAR. The 2D space is divided into grid cells, and each grid cell is labeled by the system as either occupied, free or unknown.

A 3D map is generated by using a high resolution geo-referenced aerial photograph, where each pixel in the photograph is provided together with height information. Each pixel is later plotted in 3D space by using the longitude and latitude of each pixel to determine the position in the plane, and the height information to determine the height of each pixel in 3D space. The pixel color is maintained from the aerial photograph, creating a photo-realistic 3D view. The resolution of the map can be scaled, such that only a fraction of the pixels are plotted, to simulate a low resolution map.

The 3D map can be viewed straight from above, neutralizing the affect of the height information, and effectively being a photo-realistic 2D-map. As with the full 3D-map, the resolution can be lowered.

By evaluating the three maps, and varying the resolution, the desired map type and the resolution requirement can be specified.

Obstacle detection

In order to fulfill requirement 3 in subsection 3.2.1, the obstacle detection performance must be evaluated. This is evaluated using measurements of radar data, which are compared to simultaneously captured video and LIDAR measurements. As reference data the video and LIDAR data have complementing strengths: the video has long range, but a fairly narrow field of view, while the LIDAR has short range (20 m), but a wide field of view (270 °). Hence, the LIDAR can be used to identify objects approaching from the sides and from behind, while the camera can be used to identify objects in front of the vehicle. The requirements to be specified in terms of obstacle detection is the desired coverage around the vehicle, the desired range, the desired accuracy and how false negative measurements affect the result.

The test will be performed by evaluating through video which obstacles that should be detected, and verify if the prototype's range and field of view is sufficient to achieve this and if the obstacle has been observed. If not, the required range and field of view will be estimated through the obstacle position and the ego-vehicle position.

5.3 Test results

5.3.1 Subsystem testing

Localization

As described in subsection 5.2.2, several different accuracy levels are simulated. These accuracy levels are all evaluated using the same real position, to simplify assessment. As a reference, the on board video is shown in Figure 5.2

Figure 5.2. The on board camera of the test vehicle showing the reference position, i.e. that the test vehicle is positioned in the middle of the right lane.

Note that the vehicle is positioned in the middle of the lane, a position which is accurately tracked by the LIDAR positioning, as shown in Figure 5.3

Figure 5.3. The graphical representation of the test vehicle showing the reference position. The blue cuboid represents the test vehicle position.

This accuracy is at least what can be expected from an RTK-GPS system. To simulate the accuracy from a D-GPS system, a bias of 1 m is added. This is, according to Table 5.1 what can be expected from a D-GPS in ideal conditions. The resulting graphical representation is shown in Figure 5.4

It can be seen that the position estimate has now moved to intersect with parking cars. Please note that the cars being parked in the map are not the cars being viewed in Figure 5.2, since the cars in the map are the cars parked at the spot when the map was made.

The inaccuracy seen in Figure 5.4 can be acceptable in some cases, but in an environment similar to the evaluated one, the inaccuracy would potentially cause collisions.

The upper tolerance of a D-GPS localization is given as 5 m by Table 5.1. This bias is simulated in Figure 5.5

Figure 5.4. The graphical representation of the test vehicle showing the position with a 1 m bias, the lower interval of a D-GPS localization. The blue cuboid represents the ego-vehicle position.

Figure 5.5. The graphical representation of the test vehicle showing the position with a 5 m bias, the upper interval of a D-GPS localization. The blue cuboid represents the ego-vehicle position.

The 5 m inaccuracy simulated in Figure 5.5 is clearly too inaccurate to be used in an environment similar to the evaluated one. It would be impossible for a driver to determine the road position of the vehicle during these inaccuracies.

An unassisted GPS system has even larger inaccuracies, but these simulations are not included in this thesis, as such an inaccuracy is obviously too large to provide any useful information in this application.

Maps

There are two main factors in terms of map selection. These two are resolution and map type. There are three map types available in this evaluation, these are an abstract 2D map, a 3D photo-realistic map and the same photo-realistic map viewed from above to simulate a 2D photo-realistic map. The resolution of this photo-realistic map can be changed, and three different levels are evaluated. The lowest resolution evaluated is 1 m, which is shown in Figure 5.6.

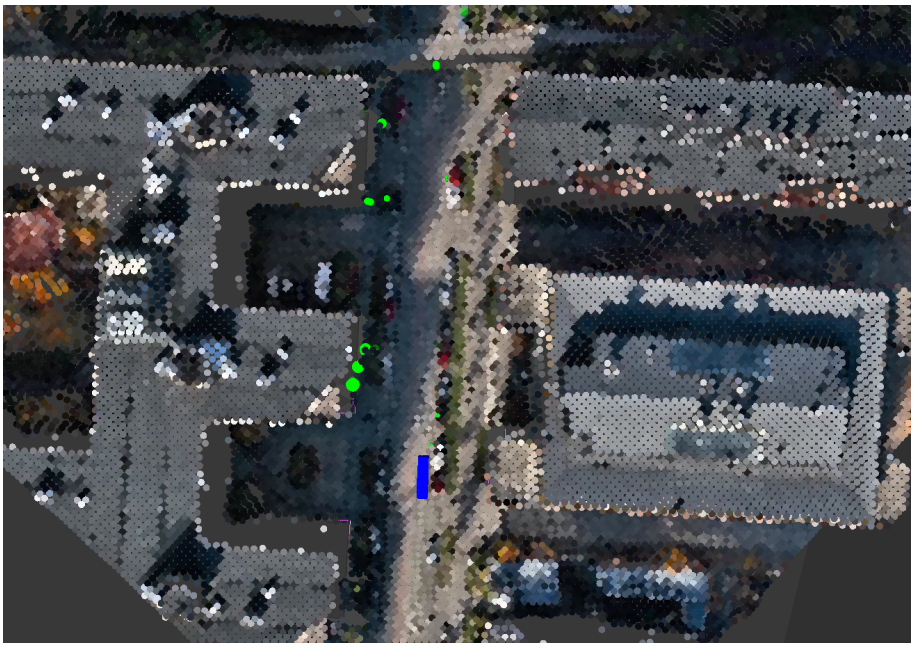

Figure 5.6. A 2D photo-realistic map with 1 m resolution. The blue rectangle represents the ego-vehicle position and the green circles represent identified obstacles.

This resolution is too poor to provide valuable information to the machine operator. The next resolution level evaluated is 0.5 m, which is shown in Figure 5.7.

The difference between Figure 5.6 and Figure 5.7 is significant, and the 0.5 m resolution provide a lot more detail to the operator. The best available resolution is 0.25 m, which is shown in Figure 5.8. The resolution showed in Figure 5.8 allows significantly closer zoom levels, which gives larger flexibility to find the ideal zoom level for a specific operator. A resolution of 0.5 m is judged to be insufficient, since the zoom-level possible with this resolution does not reveal information that could be of importance to the operator. Comparing Figure 5.7 and Figure 5.8 shows that while the latter give an indication of the distance to the region where cars are normally parked, the former can not give that information.

In addition to the different resolutions, different map types are available. The basic abstract 2D map used by the LIDAR localization can be viewed in Figure 5.9

The abstract 2D map does not provide any information about the terrain and is very hard to interpret for a human operator. Note that no difference is made in the map between road and vegetation. It is judged that this map does not give a sufficient amount of information to the machine operator. This type of map can be enriched by further information, such as a more clear definition of drivable surfaces, but is yet judged too unintuitive.

The most advanced map evaluated is the photo-realistic 3D map seen in Figure 5.10. Since the map is created based upon aerial photographs and height infor-

Figure 5.7. A 2D photo-realistic map with 0.5 m resolution. The blue rectangle represents the ego-vehicle position and the green circles represent identified obstacles.

mation, no information is available regarding vertical surfaces. This is clearly seen as building walls are not shown. The performance of the map is however satisfactory at surfaces that are horizontal or near horizontal, such as road and terrain surfaces, which is where the system is intended to be used. For instance, the vegetation at the left side of the road that the ego-vehicle is driving at is shown in an intuitive way, and all drivable surfaces are shown accurately. The map can be further cropped, as shown with the 2D-map in Figure 5.8.

Obstacle detection

As described in subsection 5.2.2, the range and field of view of the obstacle detection are important requirements that must be determined from tests. The needed range is dependent on the application. In this case, the intended application is heavy machinery, typically driving at low speed, which means that the required range is lower than in a high speed application. During the situation shown in Figure 5.10, the automotive radar provides measurements of objects more than 100 meters ahead of the vehicle. At typical operating speeds of heavy machinery, this gives a reaction time of more than 10 seconds.

To assess the field of view of the sensor used, a situation in an intersection is reviewed. The test vehicle is stationary throughout the measurement series presented in this section. In Figure 5.11, another vehicle enters the field of view of the on board camera.

Figure 5.8. A 2D photo-realistic map with 0.25 m resolution, further cropped. The blue rectangle represents the ego-vehicle position and the green circles represent identified obstacles.

As the passing vehicle is located to the left of the ego vehicle (as shown in Figure 5.11), the system provides the output shown in Figure 5.12, where the crossing vehicle is shown by the LIDAR measurements, but not by the automotive radar. In Figure 5.13, the system output is shown as the passing vehicle is straight in front of the ego vehicle. The passing vehicle is now detected by the automotive radar. The automotive radar returns the position of the closest corner of the detected vehicle, and a comparison to video shows that the position of the passing vehicle is accurate. The LIDAR measurements are to the right of the radar measurement, which is explained by the LIDAR being roof mounted and returning measurements of the upper part of the passing vehicle, while the automotive radar is mounted at a low position, returning a measurement of the lower parts of the passing vehicle, which extends further than the upper parts.

In Figure 5.14 the position of the passing vehicle at the time of the last radar measurement is shown. As the passing vehicle moves further, no measurements of the vehicle is received from the automotive radar. This sequence demonstrates the narrow field of view of the automotive radar, which is especially evident at short ranges, since the width of the field of view is smaller closer to the ego vehicle.

It is evident that in order to detect vehicles approaching from the sides of the vehicle, a wider sensor coverage than the one provided by a single automotive radar is needed.

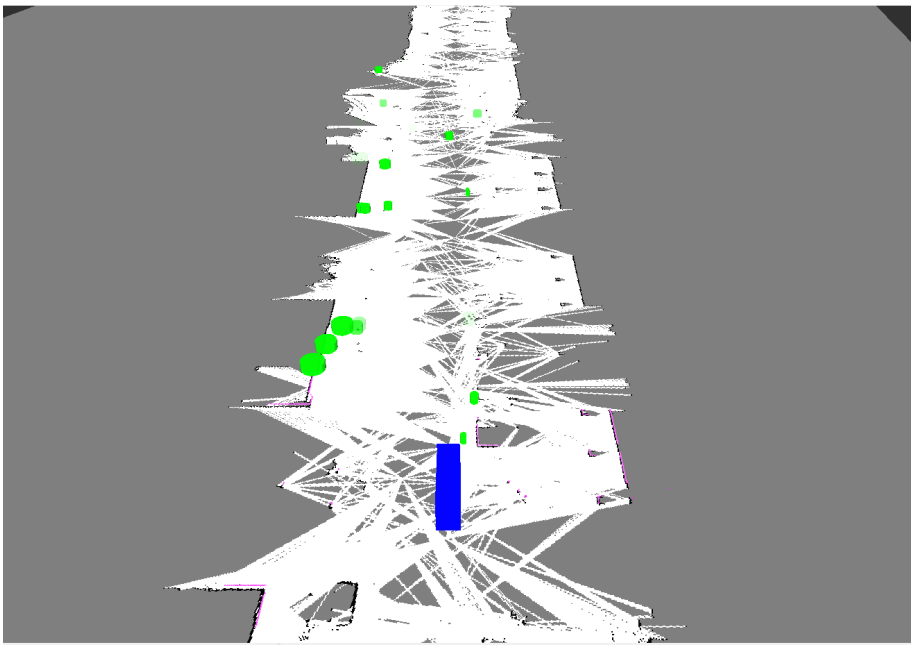

Figure 5.9. The abstract 2D map used by the LIDAR localization. The blue cuboid represents the ego-vehicle and the green cylinders indicate identified obstacles. The white areas of the map are marked as unoccupied.

5.3.2 System prototype evaluation

This section summarizes the information received by the system in a typical use case. This use case is illustrated by the image reference in Figure 5.15. The corresponding system output is shown in Figure 5.16.

To evaluate the system performance, the information available in Figure 5.16 is assessed. Note that the cars parked in the map are not present during the test. The cars present during the test can be identified and positioned by the reference image in Figure 5.15.

The two vehicles parked in the edges of the image are not detected by the automotive radar, due to the limited field of view. The two cars parked to the right are however detected and displayed in the system output. The two cars parked at the right side of the road further ahead are marked and displayed as well as the road signs marking the pedestrian crossing under the bridge. On the left side of the ego vehicle, the buildings give rise to measurements of a number of objects at the building walls. These measurements are not present at the right side of the image, where most of the buildings are hidden behind the parked cars. The prototype evaluated during this test does not classify the different objects.

To conclude, the obstacles present in Figure 5.15 are shown in Figure 5.16, showing that the system can provide valuable information to the machine operator, despite only using sensors robust to rain, snow and fog.

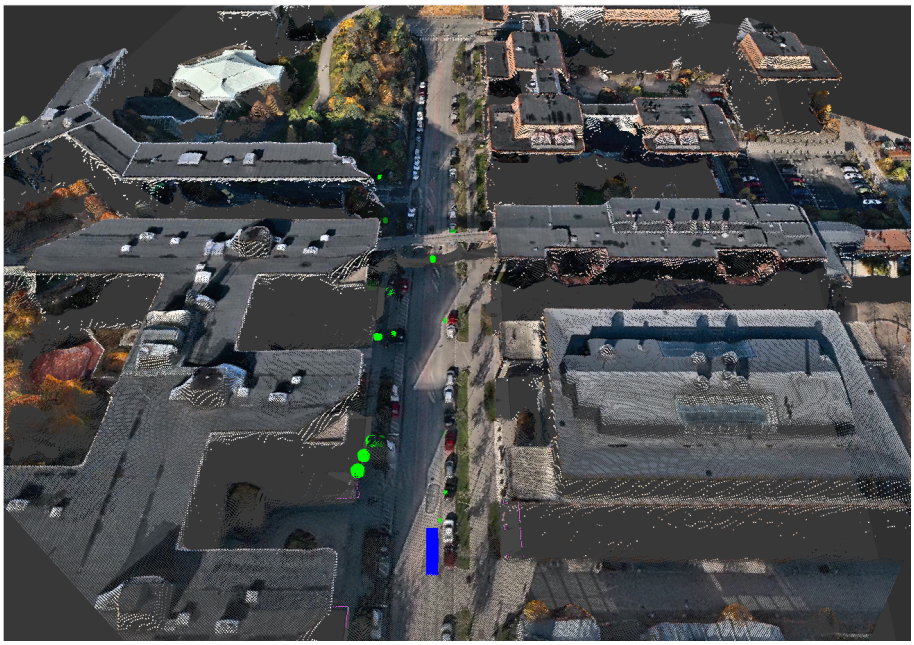

Figure 5.10. A photo-realistic 3D map with 0.25 m resolution. The blue cuboid represents the ego-vehicle and the green cylinders indicate identified obstacles

Figure 5.11. A vehicle enters the field of view of the on board camera, used as a reference.

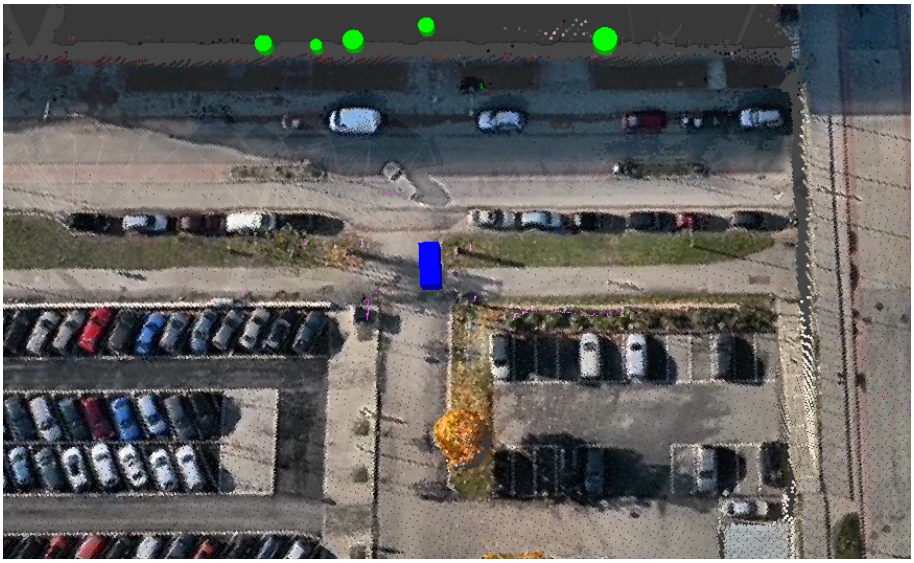

Figure 5.12. The system output as a vehicle is crossing ahead of the ego vehicle, slightly to the left. The blue cuboid shows the ego vehicle position, the green cylinders indicate radar obstacles and the purple dots indicate LIDAR measurements.

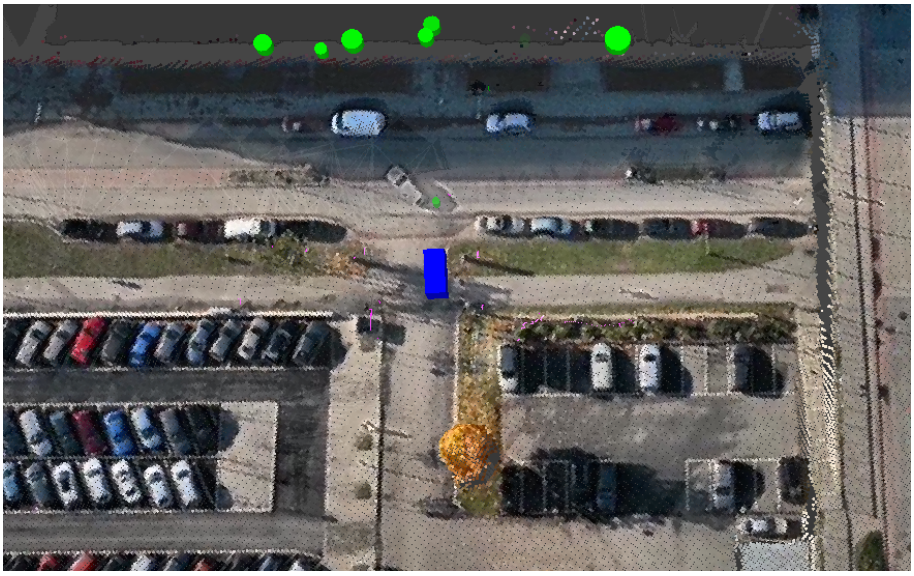

Figure 5.13. The system output as a vehicle is crossing ahead of the ego vehicle, straight in front of the ego vehicle. The blue cuboid shows the ego vehicle position, the green cylinders indicate radar obstacles and the purple dots indicate LIDAR measurements.

Figure 5.14. Video showing the position of the passing vehicle as it leaves the field of view of the automotive radar.

Figure 5.15. The image reference of a typical use case.

Figure 5.16. The system output of a typical use case. The blue cuboid shows the ego vehicle position, the green cylinders indicate radar obstacles and the purple dots indicate LIDAR measurements.

Chapter 6

Future work

This chapter outlines areas of interest for further study and potential improvements that must be made to turn this prototype into a product.

6.1 Areas of interest for further study

A number of questions must be addressed before a system like the one suggested in this thesis can be manufactured and used.

6.1.1 Full scale evaluation

To further investigate the system performance, a full scale evaluation should be performed. This full scale evaluation should take place with heavy machinery test vehicles in a realistic operating environment. The system output should be displayed to the operator through an in-vehicle display and the operator feedback should be assessed.

6.1.2 Refinement of requirements

The requirements presented in section 3.2 need to be further specified. The tests performed within this thesis can give an clear indication which technologies that are suitable, but a more precise specification of the technical requirements are needed in order to asses the process of turning the prototype into a product.

6.1.3 Compliance with governmental regulations

The compliance with governmental regulations is not considered in this thesis, but must be addressed before further development is done. It is possible that the system must be expanded or changed in order to comply with existing regulations.

6.2 Potential system expansions

The prototype evaluated in this thesis could be further expanded, this section presents some suggestions within a few areas.

6.2.1 Obstacle classification

The current prototype does not classify the obstacles detected by the system. It lacks information regarding the type and expected behavior of obstacles, except for the expected obstacle movement. The information gathered by the system can however be used to classify the obstacles into different categories. For instance, the obstacle velocity, the radar cross section, and the movement pattern of each obstacle is gathered by the system. The observed characteristics of each obstacle can be compared to typical characteristics of each category.

This categorization can be made more or less advanced. In a simple case, velocity can be used to separate stationary objects from moving ones. This classification can be refined, such that an obstacle moving at very slow speed can be assumed to be a pedestrian, while objects moving fast are assumed to be vehicles. By saving the maximum speed of each obstacle, a car that has slowed down should still be classified as a car. To make this classification more robust, further information can be used. The radar cross section is returned and can be used to further improve classification.

This intuitive classification could be implemented by the Mahalanobis distance technique described in subsection 4.6.2, where each obstacle and its characteristics can be compared to objects that has typical characteristics for each category. By varying the uncertainties, different deviations from the typical values can be allowed for different types of information.

6.2.2 Full 3D-map

The 3D-map used in the prototype does not have any information of vertical surfaces such as walls and steep terrain. This information could be helpful to the operator in order to make the system output more similar to the real surroundings, which in turn can help the operator to localize and drive more intuitively. Technologies to show a full 3D-map with the desired resolution exist, but has not been available to this thesis.

6.2.3 Map updates

Heavy machinery often operate in changing environments, where new obstacles may appear. To provide a good support to the operator, these new obstacles should preferably be added to the map. The simplest implementation of this is to re-map the area frequently. This can however be a costly process that should be avoided if possible.

A more affordable option is to use information gathered by the vehicles operating in an environment to update the map. The map must be certain, and large caution

is needed if observed obstacles are to be added to the map. However, if an obstacle is observed repeatedly at several different times, the obstacle can certainly be assumed to be a static object and can hence added to the map. This technique can be used to share information between vehicles, to further improve the available information and certainty in the information that is being presented to the operators.

The two techniques above could be used together, where the updates performed by the vehicles themselves allows longer time between re-mappings.

A yet more advanced option is if the vehicles can create their own map. This would allow the vehicles to have updated maps, and share new obstacles with each other. The visual mapping of the surroundings can take place in full visibility conditions, and later used by the vehicles during low-visibility conditions. A more precise design of such a system is outside the scope of this thesis, but the implementation would essentially be similar to the aerial mapping techniques used to create the 3D-map used.

6.2.4 Improved localization integrity

A precise localization is critical to the success of the system. The system output should support the operators to navigate heavy machinery, and must hence be both accurate and have large integrity. To provide large integrity, multiple sources of information are needed, such that contradicting information can be used to detect inaccuracies.

Global localization techniques such as GPS can be supported by alternative localization techniques such as inertial measurement units (IMU:s) and vehicle models that describe how the vehicle is likely to behave. Further information is available from sensors measuring wheel movement, known as odometry information.

By estimating the position from different sensors and comparing the results, contradicting results could be identified and used to improve the integrity. For instance, if the GNSS system inaccurately indicates a sudden move sideways, this is contradicted by the vehicle model stating that the machine can not perform the reported movement and the IMU measuring low accelerations. Using this information, the GNSS-update can be identified as likely to be inaccurate.

6.2.5 Augmented reality

To expand the functionality of the system, the system output can be enhanced with information that improve productivity and facilitate the vehicle operation. This could for instance be navigation information, such as which route the operator should use, if some areas should be avoided or are preferred. To include this type of information would require a minimum extra effort.

Information can also be shared between vehicles. In the simplest case, this can be the position of other vehicles, showing if traffic from other vehicles equipped with the same system is expected. In a more advanced application, more information can be shared. If the traction control system of one vehicle reports that the road is

slippery, this information can be shared among vehicles to warn operators of other machines.

Chapter 7

Conclusion

This chapter summarizes the outcome of this thesis and seeks to answer the problem description.

7.1 Suggested system architecture

The suggested system architecture should provide localization, obstacle detection and a user interface.

The test performed in subsection 5.3.1 shows that the accuracy provided by a D-GPS solution is insufficient to support the operator, while the position provided by a RTK-GPS system is sufficiently accurate. The system would hence need an RTK-GPS to operate properly.

Based upon subsection 2.4.1, a LIDAR does not provide sufficient robustness to detect obstacles in the harsh conditions demanded in subsection 3.2.2, while the radar sensors evaluated the same studies show satisfactory robustness. It follows from this that automotive radars are the one evaluated solution that could operate in these conditions and satisfy the requirements. However, as presented in subsection 5.3.1, the field of view of a single automotive radar is too narrow to be used as the only sensor. To compensate for the narrow field of view, multiple automotive radars are needed. Note that the obstacle tracker described in subsection 4.6.5 provides support for multiple sensors.

The map evaluation indicates that a 3D-map would enhance the operator experience and that such a map would help the operator to navigate more intuitively. Preferably, a system should implement an even more advanced 3D-map than what is evaluated in this thesis, such that building walls and and steep terrain would be displayed as well. However, an early version of the system could use the map presented and evaluated in this thesis.

To satisfy the integrity requirement presented in subsection 3.2.1, the RTK-GPS must be complemented by other navigation sensors. As described in chapter 6, a combination of RTK-GPS, an IMU and vehicle models would provide good integrity, and can be implemented using existing standard components.

7.2 Feasibility

Requirement 1 in subsection 3.2.1 is not satisfied by the implemented prototype, but the system is judged to be able to satisfy this requirement if the proposed integrity algorithms are implemented.

Requirement 2 in subsection 3.2.1 is satisfied by the proposed RTK-GPS localization system and the 0.25 m map resolution proposed.

Requirement 3 in subsection 3.2.1 forces the use of multiple sensors, as proposed in section 7.1. If multiple sensors are used, this requirement is judged to be satisfied, as the proposed sensor type show good performance within the intended field of view.

Requirement 1 in subsection 3.2.2 is satisfied by the implemented prototype, since the prototype uses the same framework as the existing software and communication between the existing software and the software implemented in the prototype is established.

Requirement 2 in subsection 3.2.2 is judged o be satisfied by the proposed system, as the radar sensors proposed are tested and concluded to perform well in harsh conditions by several studies.

Due to this, the system presented in section 7.1 is judged to be implementable without violating any of the binding requirements, while using available standard components. A full scale implementation of the proposed system is hence judged be feasible.

By implementing the suggested system, several improvements can be made. By enabling machine operators to have a better perception of their surroundings, the safety of these machine operators can be improved, as well as the safety of pedestrians and other vehicles operating in the vicinity of heavy machinery. The cost of the proposed system would be high, but it is likely that the system in some applications would save money by decreasing the down time, in addition to the social benefit of increased safety. Furthermore, the proposed system expansion in subsection 6.2.5 can improve the efficiency of the machines used, for instance by helping the machine operators to chose the most fuel efficient routes or avoid queuing in areas where other machines are already operating. This can, in addition to increasing safety, also reduce both costs and emissions, thus enabling a more sustainable operation of heavy machinery.

7.3 Potential market segments

The use of RTK-GPS adds a significant cost to a vehicle, and makes a large scale production of a system like this unlikely. The high-end maps used in this implementation, and required by the system, would further increase the cost and decrease the availability of the system. These facts can be compared to the requirements specified in subsection 3.2.2, where they clearly contradict the requirements for mass production. Due to this, the system proposed in section 7.1 must initially be aimed at small series production focused on extreme applications, where down time caused

by bad visibility is present and very costly.

However, the obstacle detection sensors proposed are standard production units, available at reasonable costs. To visualize the data, once maps are available, can be made inexpensively and well integrated into the machines. This means that if sufficiently advanced maps can be made available and accurate localization can be implemented at a reasonable prize, an expansion of the system could be suitable for production in larger series.

Bibliography

The VisLab Intercontinental Autonomous Challenge: 13,000 km, 3 months, ...no driver, 2010.

Andrew Bacha, Cheryl Bauman, Ruel Faruque, Michael Fleming, Chris Terwelp, Charles Reinholtz, Dennis Hong, Al Wicks, Thomas Alberi, David Anderson, Stephen Cacciola, Patrick Currier, Aaron Dalton, Jesse Farmer, Jesse Hurdus, Shawn Kimmel, Peter King, Andrew Taylor, David Van Covern, and Mike Webster. Odin: Team victortango's entry in the darpa urban challenge. *Journal of Field Robotics*, 25(8):468–492, 2008.

Ian Baldwin and Paul Newman. Laser-only road-vehicle localization with dual 2d push-broom lidars and 3d priors. *IEEE/RSJ International Conference on Intelligent Robot Systems*, pages 2490–2497, 2012.

Massimo Bertozzi, Alberto Broggi, Elena Cardarelli, Stefano Cattani, and Maria Chiara Laghi. Equipment and capabilities of the vehicles for the vislab intercontinental autonomous challenge. Technical report, Universita degli Studi di Parma, 2012.

Eric Brassart, Claude Pegard, and Mustapha Mouaddib. Localization using infrared beacons. *Robotica*, 18(2):153–161, Mars 2000. ISSN 0263-5747. URL `http://dx.doi.org/10.1017/S0263574799001927`.

Francois Caron, Emmanuel Duflos, Denis Pomorski, and Philippe Vanheeghe. Gps/imu data fusion using multisensor kalman filtering: Introduction of contextual aspects. *Inf. Fusion*, 7(2):221–230, June 2006.

Stephen James Cassiola. Fusion of laser range-finding and computer vision data for traffic detection by autonomous vehicles. Master's thesis, Virginia Polytechnic Institute and State University, 2007.

R. Dominguez, E. Onieva, J. Alonso, J.Villagra, and C.Gonzalez. Lidar based perception solution for autonomous vehicles. In *11:th International Conference on Intelligent Systems Sedign and Applications*, 2011.

N. El-Sheimy, E. H. Shin, and X. Niu. Kalman filter face-off extended vs. unscented kalman filters for integrated gps and mems inertial. *Inside GNSS*, pages 48–54, 2006.

Alex Foessel, , Dimitrios Apostolopoulos, and Sachin Chheda. Short-range millimeter-wave radar perception in a polar environment. In *Proceedings of the Field and Service Robotics Conference*, August 1999.

Andre Goncalves, Andre Godinho, and Joao Sequeira. Low cost sensing for autonomous car driving in highways. In Janan Zaytoon, Jean-Louis Ferrier, Juan Andrade-Cetto, and Joaquim Filipe, editors, *International Conference on Informatics in Control, Automation and Robotics and Automation 2*, pages 370–377. INSTICC Press, 2007. ISBN 978-972-8865-83-2. URL `http://dblp.uni-trier.de/db/conf/icinco/icinco2007-ra2.html#GoncalvesGS07`.

Keith W. Gray. Obstacle detection and avoidance for an autonomous farm tractor. Master's thesis, Utah State University, 2000.

H.Schafer, A.Hach, M.Proetzsch, and K.Berns. 3d obstacle detection and avoidance in vegetated off-road terrain. *IEEE International Conference on Robotics and Automation*, pages 923–928, May 2008.

Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls*, 1997.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45, 1960.

Itzik Klein and Sagi Filin. Lidar and ins fusion in periods of gps outages for mobile laser scanning mapping systems. In *ISPRS Workshop, Laser scanning 2011, ISPRS, Volume XXXVIII, Calgary,Canada*, 2011.

J. Krejsa and S. Vechet. Infrared beacons based localization of mobile robot. *ELECTRONICS AND ELECTRICAL ENGINEERING*, 117(1):17–22, 2012.

Jacoby Larson, Michael Bruch, Ryan Halterman andJohn Rogers, and Robert Webster. Advances in autonomous obstacle avoidance for unmanned surface vehicles. Technical report, Space and Naval Warfare Systems Center, San Diego, 2007.

John Leonard, Jonathan How, Seth Teller, Mitch Berger, Stefan Campbell, Gaston Fiore, Luke Fletcher, Emilio Frazzoli, Albert Huang, Sertac Karaman, Olivier Koch, Yoshiaki Kuwata, David Moore, Edwin Olson, Steve Peters, Justin Teo, Robert Truax, Matthew Walter, David Barrett, Alexander Epstein, Keoni Maheloni, Katy Moyer, Troy Jones, Ryan Buckley, Matthew Antone, Robert Galejs, Siddhartha Krishnamurthy, and Jonathan Williams. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25:727–774, 2008.

Bingbing Liut, Martin Adamst, and Javier Ibahez-Guzmdnu. Minima controlled recursive averaging noise reduction for multi-aided inertial navigation of ground vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

Cetin Mekik and Murat Arslanoglu. Investigation on accuracies of real time kinematic gps for gis applications. *Remote Sensing*, (1):22–35, 2009.

William F Milliken and Douglas L. Milliken. *Race Car Vehicle Dynamics*. Society of Automotive Engineers Inc, 1995.

Team MIT. Technical report - darpa urban challenge. Technical report, MIT, 2007.

M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun. Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 2008.

H. Rabe, E. Denicke, G. Armbrecht, T. Musch, and L. Rolfes. Considerations on radar localization in multi-target environments. *Advances in Radio Science*, 7: 5–10, 2009.

Giulio Reina, James Underwood, Graham Brooker, and Hugh Durrant-Whyte. Radar-based perception for autonomous outdoor vehicles. *Journal of Field Robotics*, 28:894–913, 2011.

Isaac Skog and Peter Handel. In-car positioning and navigation technologies - a survey. *IEEE Transactions an Intelligent Transportation Systems*, 10:4–21, March 2009.

Chuck Thorpe, Justin David Carlson, David Duggins, Jay Gowdy, Robert MacLachlan, Christoph Mertz, Arne Suppe, and Chieh-Chih Wang. Safe robot driving in cluttered environments. In *Proceedings of the 11th International Symposium of Robotics Research*, October 2003.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2006a.

Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the darpa grand challenge. 23:662–692, 2006b.

Michael Thuy and Fernando Puente Leon. Universitat karlsruhe. In *IEEE Intelligent Vehicles Symposium*, pages 532–537, June 2009.

Marc Mir Tutusaus. Evaluation of automotive commercial radar for human detection. Master's thesis, Helsinki University of Technology, 2008.

Chris Urmson, Charlie Ragusa, David Ray, Joshua Anhalt, Daniel Bartz, Tugrul Galatali, Alexander Gutierrez, Josh Johnston, Sam Harbaugh, Hiroki Yu Kato, William Messner, Nick Miller, Kevin Peterson, Bryon Smith, Jarrod Snider, Spencer Spiker, Jason Ziglar, William Red Whittaker, Michael Clark, Phillip Koon, Aaron Mosher, and Josh Stryble. A robust approach to high-speed navigation for unrehearsed desert terrain. *Journal of Field Robotics*, 23(8):468–508, 2006.

Rudolph van der Merwe, Eric A. Wan, and Simon I. Julier. Sigma-point kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation. In *In Proceedings of the AIAA Guidance, Navigation & Control Conference*, 2004.

Velodyne. Velodyne hdl-64e, 2011. URL `http://velodynelidar.com/lidar/hdlimages/HDL-64E.jpg`. 2014-03-25.

Chieh-Chih Wang, Charles Thorpe, Martial Hebert, Sebastian Thrun, and Hugh Durrant-whyte. Simultaneous localization, mapping and moving object tracking. Technical report, International Journal of Robotics Research, 2004.

Dominic Zeng Wang, Ingmar Posner, and Paul Newman. A new approach to model-free tracking with 2d lidar. In *Proceedings of the International Symposium on Robotics Research (ISRR)*, 2013.

Mark Phillip Woodward. Stanley. URL `http://cs.stanford.edu/~woodward/images/stanley.jpg`. 2014-03-25.

Brian Yamauchi. Ieee international conference on robotics and automation. In *All-Weather Perception for Man-Portable Robots Using Ultra-Wideband Radar*, May 2010.

Ming Yang, Chunxiang Wang, Hui Fang, and Bing Wang. Laser radar based vehicle localization in gps signal blocked areas. *International Journal of Computational Intelligence Systems*, 4(6):1100–1109, 2011.

Pifu Zhang, Jason Gu, Evangelos E. Milios, and Peter Huynh. Navigation with imu/gps/digital compass with unscented kalman filter. In *Proceedings of the IEEE International Conference on Mechatronics & Automation*, pages 1497–1502, July 2005.