EE

Localization of Autonomous Vehicle using 1D Automotive Radar Sensor

Car localization based on Radar and LIDAR Fusion

Delft University of Technology

Robin van Gaalen, 4286049

Saturday 29th February, 2020

Robin van Gaalen 4286049
Saturday 29th February, 2020

Contents

1 Abstract **5**

2 Introduction **7**
2.1 Research Questions 7
2.2 Report Structure 8

3 Fundamentals of radar SLAM **9**
3.1 Frequency Modulated Continuous Wave (FMCW) radar 9
3.2 MIMO (Minimum Input Minimum Output) 11
3.3 Super-resolution spectral estimation 13
3.3.1 Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) 13
3.3.2 MUltiple SIgnal Classification (MUSIC). 14
3.3.3 Space-Time Adaptive Processing (STAP) 14
3.4 Simultaneous Localization And Mapping (SLAM) 14
3.4.1 Kalman filter 14
3.4.2 Particle filter 16
3.4.3 Registration and the Genetic Algorithm. 17
3.5 Height Estimation 19
3.5.1 Vertical Doppler Beam Sharpening 19
3.5.2 Multipath Height Finding 20

4 Design choices for radar SLAM with 1D array **23**
4.1 Registration 23
4.2 Height extraction 23
4.3 Super resolution target acquisition 25
4.4 Conclusion 26

5 Simulation Results **27**
5.1 Conclusion 30

6 Experimental verification **31**
6.1 Adaptations 31
6.1.1 Adapted Genetic Algorithm 31
6.1.2 Ego Motion 33
6.1.3 Proposed mapping method 36
6.1.4 Current System 37
6.2 Results. 38

7 Incorporation of Lidar **43**
7.1 Lidar vs. Radar 43
7.2 Proposed Method 44
7.3 Implementation 47

8 Conclusion and Recommendations **55**
8.1 Conclusions. 55
8.2 Recommendations 57

Acknowledgments **59**

Bibliography **61**

1

Abstract

The overall purpose of this study is to establish novel methods of vehicle localization and mapping using a 1D linear automotive radar array in conjuncture with pre-existing lidar maps, and to test if the generated radar map can be made to be 3 dimensional. The reason for creating a novel localization method was to try and alleviate some of the weaknesses of existing localization techniques, such as GPS'(Global Positioning System) inability to localize in tunnels and other large structures, or lidar based positionings' low quality of service in certain weather conditions. The basic design of this study was to implement a SLAM(Simultaneous Localization And Mapping) system that co-registers radar data to radar data, and to then attempt to register radar data to lidar data, all taken from experiments with cars with these sensors driving around the TU Delft campus. After the execution of experiments, both on simulations and on real world data, it was established that it is in fact possible to localize the car by relating observed radar data to premade lidar maps, and to continually add to a cumulative map made with the radar data that can further aid the localization process. Furthermore, it can be observed that the radar map created using the 1D linear automotive array can be made to be 3D, though more experiments to establish the full potential of this capability are recommended.

2

Introduction

In a world where autonomous driving is becoming ever more prevalent, the need for secure/reliable localization methods has never been higher. Autonomous vehicles often have various means of localizing themselves, ranging from GPS to using visual sensors. While these methods are sufficient in many circumstances, there are certain scenarios where they tend to fail. The GPS sensor has a hard time providing accurate localization inside structures, such as tunnels or parking garages, and visual sensors, such as cameras and lidar, stop to function properly in thick mist and rain. To fix these problems we will try to find a new method to provide localization estimates, a radar based method. In this report we propose a novel method of radar based localization using a 1D array, to create a 3D map of the world, incorporating a previously created lidar map.
Not only does this approach provide a possibility for radar based localization, but it also provides a continuously updated road map everywhere these cars drive (a result providing many applications on its own right, from building continuously updated roadmaps for navigation systems, to detecting road blockage and even aiding environmental/ecological studies[1]).

2.1. Research Questions

To more clearly delineate the direction of this thesis project, a set of research questions was created, to help guide the project in a productive direction.

1. Can localization be done by relating radar measurements to a lidar point cloud?
   (a) How does the radar point cloud relate to the lidar point cloud, and how can they be made more relatable?
   (b) Can a 3D radar point cloud be created using DBS(Doppler Beam Sharpening), and can it be used in localization?
   (c) Is it possible to use only the radar based point cloud in the long term for localization, if so: how?
2. Can this be implemented using a cost effective 1D antenna array?
3. Can pure-radar (without a priori maps) 3D SLAM(Simultaneous Localization And Mapping) be implemented using a cost effective 1D antenna array?

The gist of question 1 is to answer if, given an a priori map made with a lidar sensor, can a vehicle equipped with a radar sensor find its position on that map. The process of relating/aligning two images or other datasets is called (co–)registration, and it will have to play a role in the localization of the vehicle. Question 1.a was generated out of an assumption that if datasets from two completely different sensors are to be aligned, it might be necessary to make said sets "look more alike", for example by making them the same dimensionality, and by taking into account what certain sensors might see that the other sensor is blind to, and vice versa. Discovering ways in which to accomplish this is the purpose of question 1.a. Answering question 1.b could be helpful as well, in making the two datasets look more alike. After all, the lidar maps available are all in 3D, and creating a radar map that is also 3 dimensional, might make relating the two datasets more robust for the purposes of localization.
Question 1.c lays the groundwork for a finalized system, that does not rely on premade maps that might have been made so long ago that they no longer properly describe the environment. New buildings get built, plants grow and can be cut down: The environment is in flux. To solve for this, the proposed solution is the creation of a continuously updated map. Though the system might start out with an a priori map, reliance on this should be reduced over time, until it is at some point no longer necessary at all. Whether this is possible is the purpose of question 1.c. If it can be proven that all of these things can be implemented using a cost effective 1D array, that one might find in cutting edge automobiles, then this system could, hypothetically, be implemented in real life in future cars, with the radar systems already installed within them. Therefore question 2 was answered, to guide this project in a direction in which the created system could hypothetically and practically be used in the future.
Finally, question 3 seeks to address to what degree a system with no a priori maps is able to localize itself while simultaneously mapping its environment, and outputting a 3 dimensional pointcloud of its environment. Because research in height detection for car radar is an ongoing field of study, this question was created, to see if besides localization with the radar, the system could aid with these height estimation techniques, these methods for making the radar map 3 dimensional. The reason this is being researched is because it can be desirable for the car radar to be able to tell the difference between a speed-bump/bridge, and a wall, so as to not engage the automatic braking system when it is not desirable.[16]

2.2. Report Structure

The structure of this report is as follows: First the basics of automotive SLAM are discussed, showing some of the techniques that can be used to reach the desired goals. Next, design choices are discussed, showing the reasoning behind the choices made so far. Then some simulation results are shown to provide proof of feasibility, at least in simulation. After that a real world experimental application of the current system is shown, where a preliminary 3D radar map of a street is created. Then we describe how to craft a lidar map that can be used as a prior data to further improve the performance of the system. Here it can be seen that lidar data can indeed be used to create a map that a radar system can use to localize itself, leading to novel methods of vehicle localization. And finally future work and the conclusion are discussed.

3

Fundamentals of radar SLAM

In this section the previous advancements in the fields of automotive radar, SLAM, OGM (Occupancy Grid Mapping), and various concepts that could be useful are reviewed. The purpose of this thesis project is ultimately to aid in the development of a radar based navigation system that is able to operate in environments where other sensors and GPS fail. To accomplish this multiple approaches have been carried out, such as: creating a map using the radar data, relating it to a previously created lidar point cloud, determining certain object properties with the radar such as elevation, using a SLAM algorithm to combine mapbuilding with localization, etc. This chapter, which is to serve as a literature review, is therefore divided into these general subjects, that could be used to accomplish the thesis goals. These subjects form the sections.

3.1. Frequency Modulated Continuous Wave (FMCW) radar

First the basics of FMCW radar are briefly discussed in this section. FMCW radar works by sending out a linearly modulated signal (a chirp), and then mixing the signal being sent out with the signal being received back. [20].
Then, one puts this signal through a low pass filter to leave only the beat signal, samples this signal, and applies spectral analysis (for example FFT) to find the distances to various reflectors. This is how a typical heterogeneous radar system works. The velocity of the reflectors can also be calculated by looking at the phase difference between consecutive FMCW chirps, and the angle of arrival can be estimated by looking at the phase difference between received signals at multiple receive antenna elements. This type of radar is often used in automotive radar because it has a relatively low complexity, is therefore relatively cheap to produce, and it has a high spectral efficiency.
Usually the aforementioned chirp signal, as illustrated in figure 3.1 is of the form:

$$s_t(t) = \exp(j2\pi(f_c t + \frac{1}{2}\alpha t^2), \tag{3.1}$$

where f_c is the carrier frequency and α is called the chirp rate,

$$\alpha = \frac{B}{T}. \tag{3.2}$$

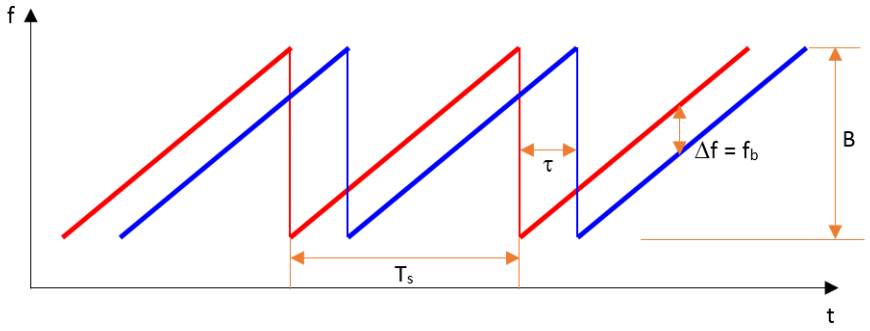


Figure 3.1: Visualization of the FMCW chirp signal, with the transmitted frequency in red, and the received frequency in blue. [34]

In this expression B is the bandwidth of the signal, and T is the chirp length in seconds. The signal that is received after reflection is a delayed version of the signal being transmitted,

$$s_r(t) = \exp(j2\pi(f_c(t-\tau+\frac{1}{2}\alpha(t-\tau)^2). \tag{3.3}$$

Here τ is the delay in seconds, equal to twice the distance to the reflector, r, divided by the speed of light

$$\tau = \frac{2r}{c}. \tag{3.4}$$

This received signal is then mixed with the signal being transmitted (equation (3.1)) to generate the following beat signal:

$$s_b(t) = \exp(j2\pi(f_c\tau+\alpha t\tau-\frac{1}{2}\alpha\tau^2). \tag{3.5}$$

This signal is sampled in what is called "fast time". Multiple chirps per antenna are saved in what will be referred to as a "radar data cube" (as seen in figure 3.2), along the axis called slow time. The signals from different antennas are placed along the "spatial sampling" axis.

The beat frequency of the beat signal is equal to the difference between the transmitted frequency and the received frequency, and can be extracted by using spectral estimation techniques along fast time,

$$f_b = \alpha\tau. \tag{3.6}$$

Finally, the distance to the reflector can then found as,

$$r = \frac{c\tau}{2} = \frac{cf_b}{2\alpha} \tag{3.7}$$

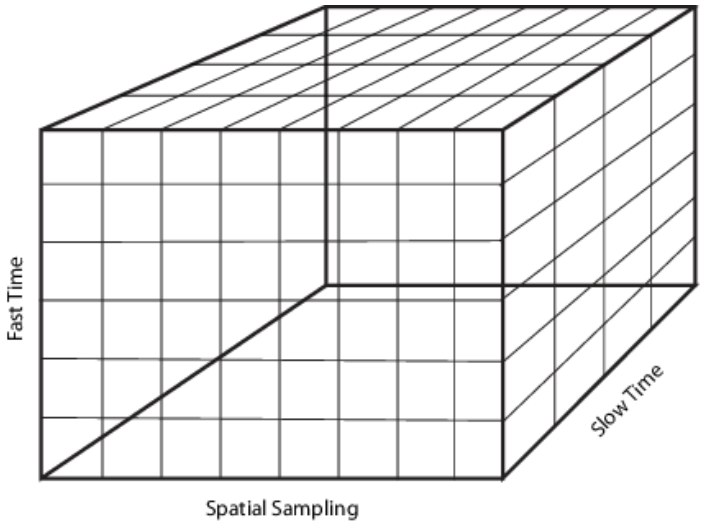


Figure 3.2: The radar data cube [33]

The velocity of the perceived target towards the radar can also be estimated, by applying spectral estimation along the slow time axis, to find the phase differences between consecutive chirps. This analysis yields the "Doppler frequency" f_d, which can then, given the carrier frequency and speed of light, be translated into a relative velocity,

$$v = \frac{c f_d}{2 f_0}. \tag{3.8}$$

Finally, from the data radar cube the angle of arrival of the returning signal can be extracted, by applying spectral estimation along the spatial sampling axis, to estimate the phase difference of the received signal between the different antenna elements in the system, which can be used to recover the angle of arrival. How these data are estimated can vary based on the radar's geometry, on the type of radar used. In this report a specific type of radar used in the automotive industry was examined, as the next section will describe.

3.2. MIMO (Minimum Input Minimum Output)

MIMO radar is a type of radar that synthesizes a virtual uniform linear array (ULA), using fewer antennas than would have been necessary in a standard ULA with the same aperture. As it needs fewer antennas, they are often cheaper to produce, and are currently being created for automotive applications. This type of 1D array was chosen for this thesis project.
To simulate a MIMO array, the following needs to be taken into account, beginning with the signal received by a hypothetical non virtual 1D linear array: If the signal is to be expressed which a hypothetical ordinary N element linear array would receive, as shown in figure 3.3, this signal can calculated as such: first, the distance of each element to the target (in the far-field) is

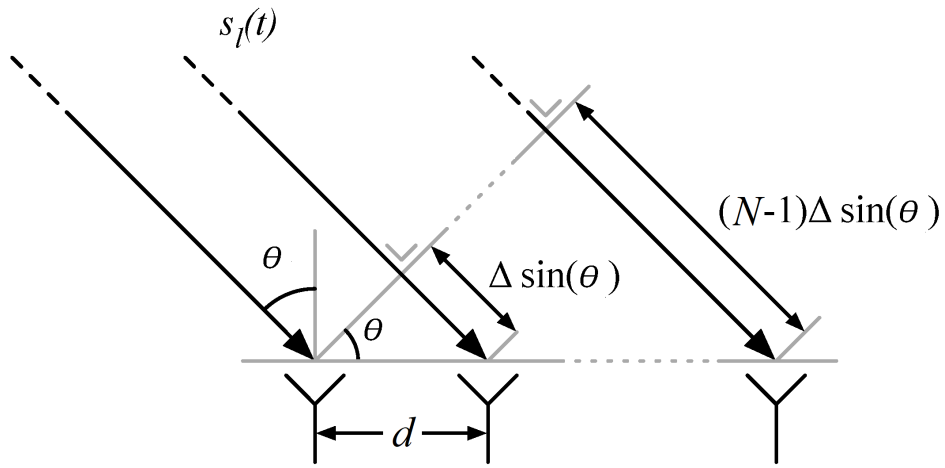


Figure 3.3: Array with angle of incidence denoted

$$r_M = r_0 + (M-1) * d \sin\theta, \tag{3.9}$$

Where r_0 is the distance from the transmit antenna to the target, θ is the angle of incidence (see figure 3.3), d is the inter element spacing, and r_M is the distance of the target to the M-th antenna. After defining antenna 1 to be the transmit antenna, the delay to each antenna would be expressed as such:

$$\tau_M = \frac{r_0}{c} + \frac{r_M}{c} \tag{3.10}$$

The left fraction indicating the delay from TX antenna to target, and the right fraction indicating the delay from the target to return antenna M, making the total, τ_M, the round trip delay. The signal received at each antenna after mixing would be, using equation (3.5):

$$s_M(t) = \exp(j2\pi(f_c\tau_M + \alpha t\tau_M - 0.5\alpha(\tau_M)^2)) \tag{3.11}$$

Expression (3.11) would hold true for an N element uniform linear array, but to simulate a N element MIMO array, with 3 TX antennas, and 4 RX antennas for example, some things need to be kept in mind. This is because 3 TX chirps must be sent, instead of just the one. In this case the amount of virtual array elements N would become $N_t N_r = N = 12$.
MIMO array signals are usually expressed with a steering vector that is the Kronecker product of the transmit and receive steering vectors. In the case that there are N_t transmit antennas, N_r receive antennas, the distance between the transmit antennas is $\lambda/2$, and the distance between the receive antennas is $N_t\lambda/2$, the steering vectors can be expressed as follows:

$$\mathbf{a}_{tx} = [1, e^{1i\pi\sin(\theta)}, ..., e^{N_t i\pi\sin(\theta)}], \tag{3.12}$$

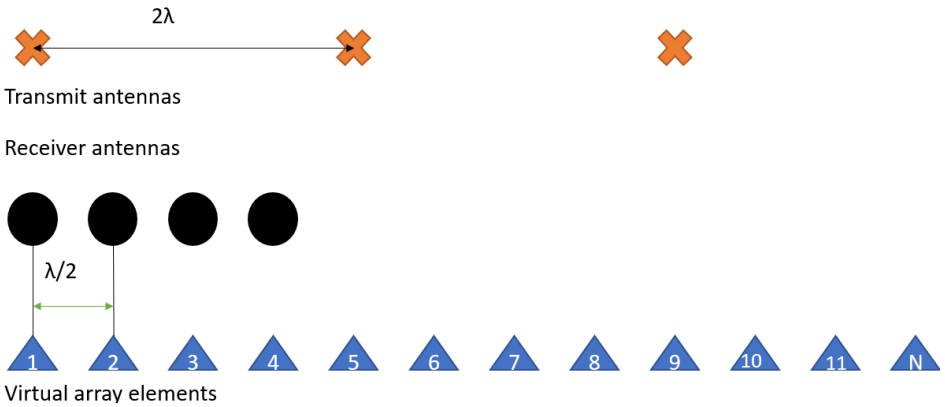


Figure 3.4: N virtual element MIMO array

$$\mathbf{a}_{rx} = [1, e^{1i\pi N_t \sin(\theta)}, ..., e^{N_r i\pi N_t \sin(\theta)}], \tag{3.13}$$

$$\mathbf{a} = \mathbf{a}_{tx} \otimes \mathbf{a}_{rx}, \tag{3.14}$$

where a_{tx} is the transmit steering vector, a_{rx} is the receive steering vector, and $\otimes$ is the Kronecker product.

$$s_M(t) = \mathbf{a} s_0(t) \tag{3.15}$$

The total set of collected signals $s_M(t)$ is the product of the total steering vector $\mathbf{a}$ and the signal received by the corner antenna $s_0(t)$. This means that if the target does not move (much) during the Coherent Processing Interval (CPI), when $D_0(t) \approx D_0(t + PRI) \approx D_0(t + 2PRI)$, one has effectively recreated a 12 element uniform linear array. If the target moves a lot between the chirps however, problems may arise.

3.3. Super-resolution spectral estimation

For the purposes of this project it was considered that super resolution techniques might be necessary, techniques that aim to provide resolution higher than an FFT would provide. This to increase the fidelity of the maps to be created and to increase the accuracy of the localization. The techniques that were considered are discussed here.

3.3.1. Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT)

ESPRIT is an approach for estimating certain signal parameters, that can, for example, be used to find the Direction of Arrival (DoA) of signals impinging on an array, or to create radar images.[22] [23] It is an example of a super resolution algorithm, that makes use of certain qualities of the signal subspace. In the case of ESPRIT, it takes advantage of the rotational invariance in signal subspaces caused by translationally invariant structure of the array. In other words: Because the radar array can be seen as a sum of 2 radar arrays, one of which is a translated (but not rotated)

version of the other, an underlying rotational invariance among the signal subspaces arises that can be exploited. ESPRIT is usually less computationally intensive than MUSIC, but more so than the FFT approach.

3.3.2. MUltiple SIgnal Classification (MUSIC)

MUSIC algorithm is another super resolution algorithm for estimating signal parameters. Like ESPRIT, it takes advantage of certain qualities of the signal subspaces. MUSIC makes use of the fact that the noise subspaces are assumed to be orthogonal. Though it is computationally expensive, there are methods to speed it up, such as Reduced-Dimension MUSIC [24]

3.3.3. Space-Time Adaptive Processing (STAP)

STAP is a technique that is adept at moving target detection, which is necessary for the application at hand. It's strengths lie in the rejection of interference, such as clutter or jamming.[25]
STAP involves a 2-dimensional filter technique using a phased-array radar with multiple spatial channels. The coupling of multiple spatial channels with pulse-Doppler waveforms is where the name "space-time" comes from. Applying the statistics of the interference environment, an adaptive STAP weight vector can be formed. This weight vector is then applied to the coherent samples received by the radar.

3.4. Simultaneous Localization And Mapping (SLAM)

SLAM, or Simultaneous Localization And Mapping, is the process of creating a map, and finding ones position on said map, at the same time. The mathematical basis that shows this problem is solvable is given in [6], which shows that a map created using SLAM can be made to converge to a correct map. SLAM often relies on acquiring "landmarks", certain stationary object on the terrain with very specific properties, such as shape, color and/or RCS (Radar Cross Section), or creating occupancy grid maps, or a point cloud based map. There are a lot of different SLAM algorithms, such as Incremental Smoothing And Mapping (ISAM) [7], FastSLAM [8], and Extended Kalman Filter (EKF) based SLAM [9]. They all generally rely on creating a map on the environment around the vehicle at a starting position, moving the vehicle, then mapping around the vehicle again and, through registration, finding the vehicles translation and rotation relative to the first measurement. We basically have to find how much the maps have to be translated and rotated for them to match up best to find the vehicle's movement, see figure 3.5. Then the two maps can be combined, and the process repeats, matching a third map with the previously created map, combining it, then matching a fourth, etc. This is the general idea, which can be improved by using one of the aforementioned specific SLAM algorithms, by fusing sensor data, such as from an Inertial Measuring Unit (IMU), and tracking the trajectory of the vehicle in question. [10]

3.4.1. Kalman filter

A popular solution to the SLAM problem is using a Kalman filter for the localization. This filter, works as follows: First, given the previous state of the vehicle (consisting of its velocity, its trajectory, its control-input (the gas pedal/brakes), and measurements from the IMU) an estimate is created of the current state of the system, the current pose/position/velocity of the car.

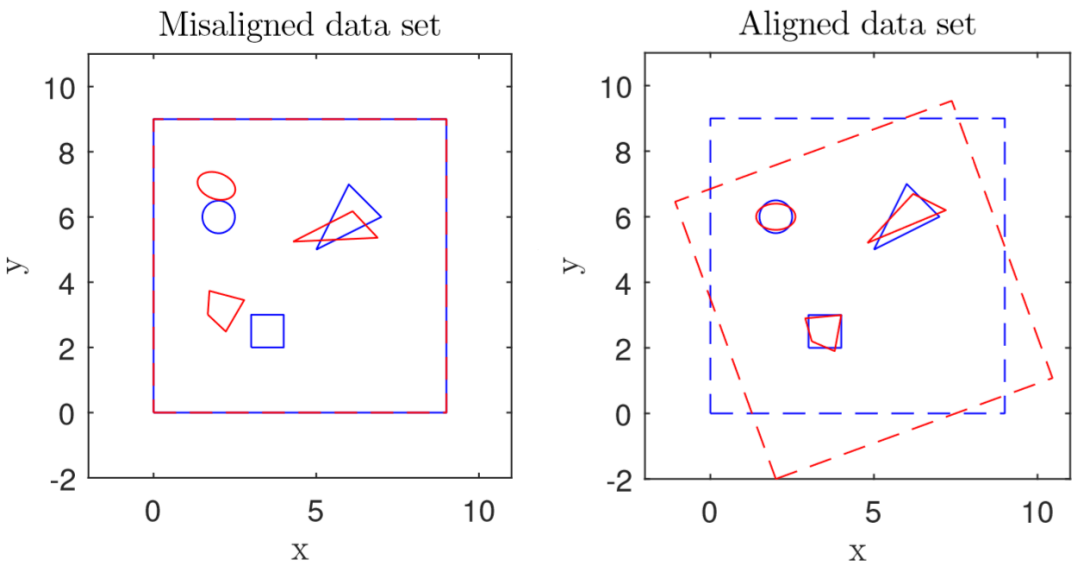


Figure 3.5: The process of registration, showing two example observations, one in red, and one in blue, that have to be aligned.

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{B}_k\boldsymbol{u}_k \tag{3.16}$$

Here $\hat{\boldsymbol{x}}_{k|k-1}$ is the predicted state at k, given the previous state at $k-1$, ($\hat{\boldsymbol{x}}_{k-1|k-1}$), the state transition model $\boldsymbol{F}$, and the control-input model $\boldsymbol{B}_k$ to which the control input vector, $\boldsymbol{u}_k$, is applied.

Then a measurement of the system is taken, the measurement being the cars current position based on the registration,

$$\hat{\boldsymbol{z}}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k, \tag{3.17}$$

here $\boldsymbol{H}_k$ is the observation model to map the true state space into the observed state space and $\boldsymbol{v}_k$ is the observation noise. Next a residual is calculated, based on the difference between the current observation and an estimated hypothetical noiseless observation, based on the predicted state in equation (3.16):

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}, \tag{3.18}$$

here $\tilde{\boldsymbol{y}}_k$ is the residual, the difference between the measurement and what we were expecting to measure. To some extent this provides a metric for how well the prediction step worked. If there was no noise and the current state were to be perfectly predicted in equation (3.16), then $\tilde{\boldsymbol{y}}_k$ would be zero.
Finally a prediction is made as to the actual state of the system: equation (3.19)

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \tilde{\boldsymbol{y}}_k \tag{3.19}$$

Where $\boldsymbol{K}_k$ is the Kalman gain, and $\hat{\boldsymbol{x}}_{k|k}$ is the estimated current state that is given by the Kalman filter. Optimal Kalman gain can be calculated as found in the literature [32], but that is outside of the scope of this project.

The problem with the SLAM application is that what is being solved is not necessarily a Linear problem, and a normal Kalman filter does not work for non-Linear systems. A solution to this is to use an EKF (extended Kalman Filter). In this filter the state transition/observation models do not need to be linear functions of the state, they can be non-linear functions, such as:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{w}_k \tag{3.20}$$

$$\boldsymbol{x}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{3.21}$$

Here the function f can be utilized to find the predicted state from the previous estimate, and h can be used to find the predicted measurement from the previous state.

3.4.2. Particle filter

The particle filter is another approach to tackling the SLAM problem[26]. Everything needed for solving the SLAM problem, in the most general terms, can be summarized as such[27]:

$$\rho(x_{1:t}, M | z_{1:t}, u_{1:t}) \tag{3.22}$$

Here $x_{1:t}$ represents the position across time, m is the map of the environment, $z_{1:t}$ represents the observations across time and $u_{1:t}$ is the control inputs.

The principle can be described as follows: The state of the system is modeled using particles, called samples. Every particle represents a hypothesis of the world at time "t" and is assigned a weight. The complete sum of all the samples is representative of the posterior distribution. The name of the technique that is used for the generation of samples for a given distribution is called the "importance sampling principle". In the case of the car each particle contains an estimate of the cars pose and landmarks around it.-
The idea can be simplified as: particles are used to model the state of the system. Each particle is a more or less likely hypothesis of the world. The algorithm recursively updates the particles with better samples as the car takes observations and travels over the roads." The sample space can be reduced by applying Rao-Blackwellization [28]:

$$p(x_{1:t}, \mathrm{M} | z_{1:t}, u_{1:t}, c_{1:t}) = p(x_{1:t} | z_{1:t}, u_{1:t}, c_{1:t}) \prod_{n=1}^{N} p(m_n | x_{1:t}, z_{1:t}, c_{1:t}) \tag{3.23}$$

With the vehicle path given, the landmarks (conditionally) independent, each of them is estimated by a short, constant size, extended Kalman filter(EKF). $c_{1:t}$ represents the correlation between the measurements and the "landmarks" on the map.

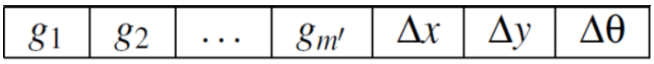


Figure 3.6: Chromosome that can be used in 2D SLAM [30]

3.4.3. Registration and the Genetic Algorithm

As mentioned before, to execute SLAM, (co-)registration is needed. This would be the art of estimating how much maps need to be moved for them to line up optimally. There are several ways to do this, Iterative Closest Point (ICP), genetic algorithm and Normal Distribution Transform (NDT), to name a few.

ICP Iterative closest point algorithm is a relatively computationally inexpensive registration algorithm. It tries to match points in the map to the closest points in current observation, and then, using a Least Squares approach, finds which rotations/translations minimize the mean square distance between all these points. [29]

NDT Normal Distribution transform relies on converting 1 of the 2 pointclouds to be registered into a continuous map/fuction, a sum of normal distributions, and then positioning the other map such that the points are as high up as possible on said normal distributions (in a 2D case). [19] The advantage of this technique is that it represents, and takes into account, the noisy nature of the maps.

Genetic Algorithm Genetic algorithms are a class of heuristic algorithms based on evolution, as found in nature[35][36]. It seeks to express the solution of a problem as a set of values, which together form an array called a 'chromosome', and to then have populations of these chromosomes 'breed', 'mutate', and 'die', and in the process iteratively moving towards an increasingly optimal solution. The most optimal solutions at every iteration will be allowed to breed, forming different combinations of properties that could be 'good', in the hope of getting closer to an optimal solution. And with set probability, as can be seen in nature as well, mutations can occur in the chromosomes, forming completely new properties, which might possibly be good.

Genetic algorithms in the case of SLAM make use of the fact that the movements of the car are not random. The car will never suddenly move upwards for example, or start driving sideways. Even the radius of the corners the car can take are limited. The way this is exploited is as follows: A population of chromosomes is created, each representative of a possible movement of the car between frames, and containing within it which radar targets between the 2 frames should be related, and which should be left out. A population of size n will provide n estimates, n guesses, of the translation/rotation between 2 frames. These guesses are tested with a fitness test, and the fittest guess is taken as the shift between 2 frames. Next, the genetic steps are executed. The population of chromosomes is made to breed, mutate and otherwise evolve. In a 2D example, the chromosomes could look like in figure 3.6:

In the chromosome shown in figure 3.6, g_m describes which targets between the 2 frames should be related, in essence it's a guess of which targets are the same target, but shifted because the car moved between the 2 frames. g_m says which target from the second frame, has to be related to the m-th target of the first frame. If, for example, g_3 equals 5, that means that the 3rd target in the

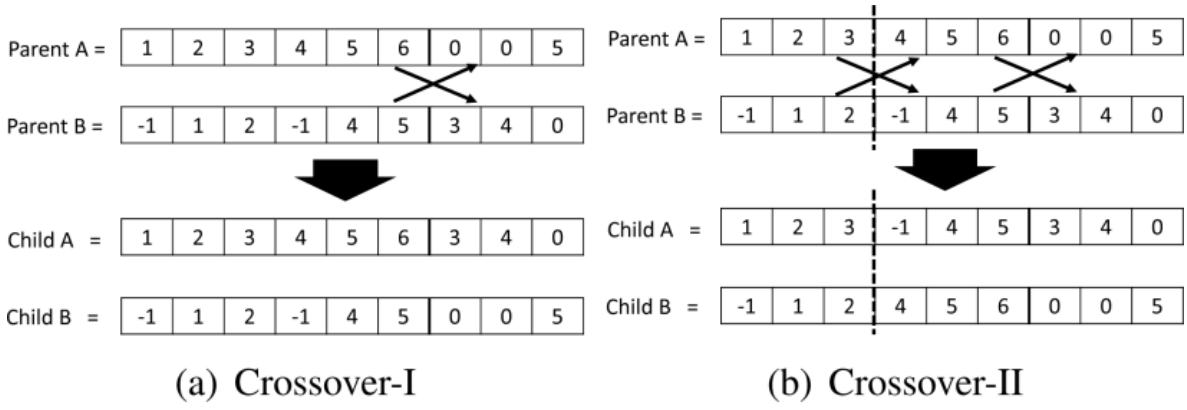


Figure 3.7: example of crossing over [30]

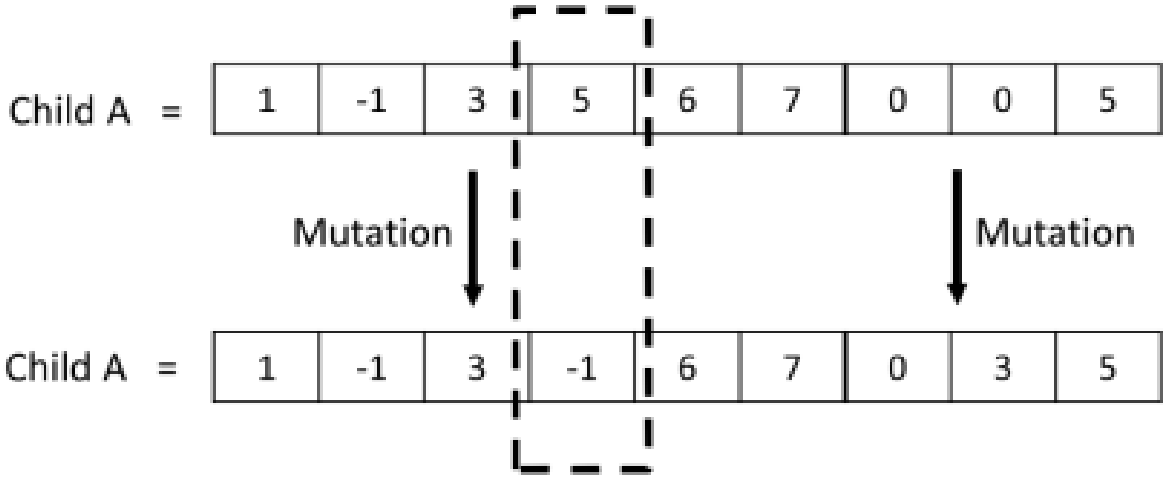


Figure 3.8: example of genetic mutation [30]

first frame must be related to the 5th target in the second frame. If the m-th target is to be left out and not related to anything at all, g_m can be given an illegal value, such as -1. ΔX and ΔY denote how much the 2 frames must be shifted in X and Y direction, and $\Delta\theta$ is the rotation between the 2 frames.

These chromosomes can start off quasi randomly, and as previously stated: between each location estimation genetic operations are executed. These fall into 3 categories: Mutation, Crossover/Reproduction and Death. In mutation, one of the genes, the elements of the chromosome, is changed/-mutated. In crossover/reproduction 2 chromosomes are copied, lined up, cut along a certain element, and then the cut off parts of the chromosomes are switched. Death needs to occur to keep the population size of the chromosomes in check, to keep it at a set size. To accomplish this, one simply kills off the least fit chromosomes to maintain a constant population size.

See figure 3.7 and equation (6.2). Note that the chromosome presented is but one possible option for the chromosomes, chromosomes are to represent possible solutions to a given problem, and those solutions can be formulated differently. We might, for example, use the angle of the turn, and the radius of the turn, to describe the cars' movement, instead of rotation and translation. So:

instead of the ΔX, ΔY and $\Delta\theta$, there would be $\Delta\theta$ and R elements, where R represents the radius of the turn, and $\Delta\theta$ represents the angle traveled allong the curve (the car moving in a straight line would be equatable to a very small angle, and a very large radius)

3.5. Height Estimation

One of the questions to be answered in this report is how to relate lidar and radar maps to each other. In this pursuit, height estimation might play a role, in that the lidar maps are 3D, and therefore making the radar maps 3D as well might make it easier to compare the two. This subsection describes the two main methods of estimating height with a 1D array. One of them relies most heavily on slow time data, the other on fast time data.

3.5.1. Vertical Doppler Beam Sharpening

Vertical Doppler Beam Sharpening is a technique for determining the height of a reflector from the ground, and it relies on the Doppler frequency shift induced by a targets velocity relative to the radar. [16] The idea is that the radar will have a certain velocity relative to stationary objects, which depends on the elevation (and azimuthal) angle to the object.

Take figure 3.9, for example. The measured velocity v_r, also known as the line of sight (LoS) velocity, is related to the velocity of the car as follows: equation (3.24)

$$v_r = v\cos\varphi \tag{3.24}$$

And so, given the measured velocity v_r, and the (assumed) known velocity of the car v, the angle φ is found to be: equation (3.25)

$$\varphi = \arccos\frac{v_r}{v} \tag{3.25}$$

The height of the target is then: equation (3.26)

$$h_t = h_s - R\sin\varphi = h_s - R\sin\arccos\frac{v_r}{v} \tag{3.26}$$

where R is the measured range to target. It has been shown that using this technique, the height of the target can be estimated with high accuracy [16].

In essence, this allows us to create a 3D map, using a 1D array. Not only does this help to discern objects suspended over the road (which the car can't observe anymore if it gets to close to them: the radar cant see traffic-lights when they're overhead), it also allows for better comparison between a lidar and radar dataset. There are numerous advantages to attempting this: Tunnels and hills can be identified, which can hinder localization as they are only visible from a certain distance/angle. Parked cars can be more easily identified. Fusion with 3D sensor data is aided, and the "richness" of the map increases. What that last point means is that in a 3D map more features of the environment can be identified, more separate objects can be observed, making the registration process more robust.

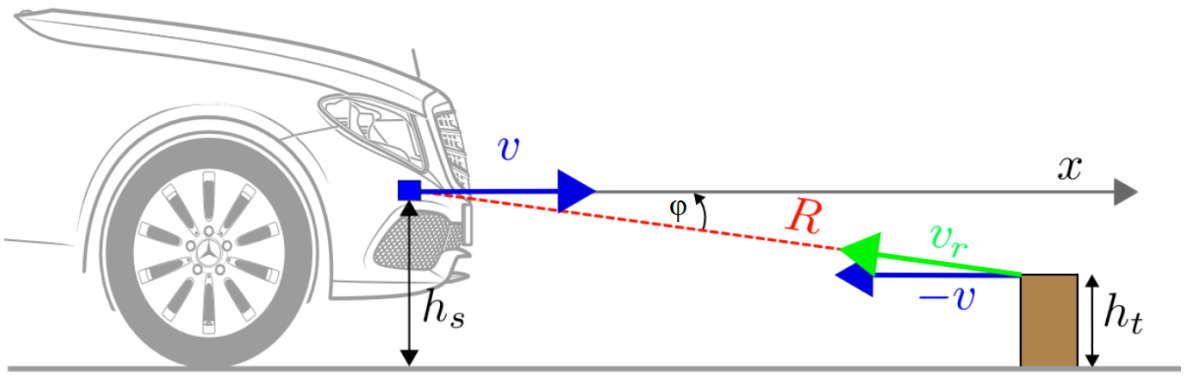


Figure 3.9: Blue vector: Velocity of the car, R: range to target, φ: angle between boresight and target, Green vector: target's relative velocity in the direction of the radar, h_t: height of reflector from the ground, h_s: height of radar from the ground [21]

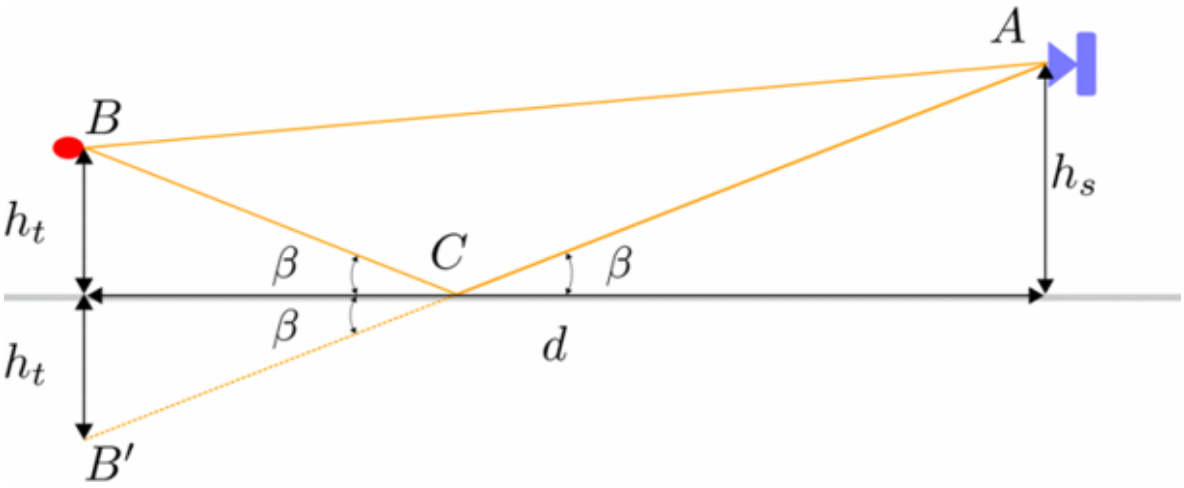


Figure 3.10: Diagram of multipath, A is the radar, B is the scatterer, and C is the point of impingement for the NLOS signal. [31]

3.5.2. Multipath Height Finding

Multipath Height Finding indicates that the height of a reflector is to be estimated by taking the the multipath properties of the signal into account [31]. It is based upon the assumption that the ground is flat, and the height of the radar from the ground is known. Figure 3.10 seeks to illustrate this.

A signal that hits B from the radar, can return and arrive along the line BA, and along the line BCA. The first is the Line Of Sight (LOS) component, the second the Non Line Of Sight (NLOS) component. In this figure, h_t indicates the height of the target from the ground, and h_s is the height of the radar from the ground. h_t can be determined as follows:

$$h_t = \frac{ACB^2 - AB^2}{4h_s}. \tag{3.27}$$

Assuming that the height of the radar from the ground h_s is known, and ACB and AB are found: h_t can be calculated. The exercise in multipath heightfinding is therefore finding out which received signals are LOS, and which are their NLOS counterparts. This is done by looking at the data along the range axis, in the fast time. One tries to find the highest peak along the range axis, assumes this to be the LOS component as it is assumed to be the most powerful, and then assumes that the very next peak following it, the next peak that is further away, is the NLOS signal belonging to it. In simple scenarios these assumptions will hold. Now that the perceived distance of the LOS component (AB), and the NLOS component are estimated, h_t can be estimated as well.
Issues this technique has is that it assumes there to be 2 perfect LOS and NLOS channels, that the ground is flat, and that no multipath effects more complicated than this take place. One has to establish, in noisy data, which peaks are the LOS components, and which peaks are the NLOS components that belong to them: Mistakes therein lead to errors.

4

Design choices for radar SLAM with 1D array

In this chapter some of the choices made throughout thesis are discussed. For many parts of the system, multiple choices, multiple options, are available, all with their own strengths and weaknesses. This section seeks to clear up why certain decisions have been made.

4.1. Registration

Registration can be seen as the art of aligning datasets/images. As discussed in chapter 3, we will focus on 3 main techniques: NDT, ICP and Genetic algorithm, and the choice between them. The genetic algorithm is different from the others, in that it technically is not just registering data, its actually a full fledged form of SLAM, in which registration cannot be seen separate from the rest of the system. Nonetheless, its registering ability will be discussed here. Starting off with the elimination of ICP. Iterative closest points' biggest advantage is its low computational complexity, it very quickly converges two maps together. Its main downside is that it doesn't deal with outliers or noisy data well. One way of dealing with this is attempting to remove outliers for example but for the time being, this aspect makes it a bad choice. Leaving the other two options. NDT and Genetic algorithm. Proposed is to not actually make a choice between the two, and implement both, as will be shown in the consequent sections.

4.2. Height extraction

To attain a form of 3-Dimensional awareness using a 1D array, a form of post processing is needed to estimate height/elevation-angle. The main methods to attain this are based on different qualities of the signal, due to scatterer height. The first technique is called DBS (Doppler Beam Sharpenning)[16], and it makes use of the fact that stationary scatterers at different heights cause different Doppler shifts. In effect, it makes use of a similar principle as front looking SAR (Synthetic Aperture Radar). The second method makes use of the multipath propagation [31]. Cars are often found on the road, and roads tend to be flat, meaning that the radar will usually receive a direct LOS(Line Of

Sight) return signal from a scatterer, and a NLOS(Non Line Of Sight) signal that bounced off the ground before returning to the radar. The difference in delay or phase between these signals can be used to estimate the vehicle height.
The pros and cons of these two approaches are as follows:

Doppler Beam Sharpening Method:

Pro:

- The Doppler approach does not require figuring out which received signals are LOS and which are NLOS, nor which belong together
- The height of the radar from the ground does not have to be known or even constant
- The ground does not have to be locally flat around the car
- It does not require a very high fast time resolution

Con:

- It requires the vehicle be moving
- It does need a very high Doppler resolution: Errors in Doppler estimation cause different errors in height estimation quickly

Multipath Method:

Pro:

- It can create a height estimate while the car is standing still.
- It doesn't require a high Doppler resolution

Con:

- It does require a very high range resolution: Errors in the Range (difference) estimate cause errors in the height estimation quickly
- LoS and NLoS peaks along the range axis have to be related by some algorithm, and in a situation with multiple multipath channels, and a lot of peaks, this can be hard and error prone
- The road has to be flat in the field of view around the car
- The height of the radar from the ground has to be continuously known

All in all, one of the most glaring differences is that one approach gathers a form of elevation angle estimation from the Doppler(DBS), and the other derives this elevation angle estimation from the Range (Multipath). This doesn't necessarily raise one method over another, but it is a good thing to keep in mind when designing the system and its trade-offs.

For this particular application the DBS method was chosen. The main reason is the requirement of relating the LOS peaks to the NLOS peaks, in an environment where many peaks may arise, because complex objects are being observed. For either approach a super resolution algorithms could be used, so that doesn't make one approach better than the other. And for this approach, we wish to test in an urban environment, that has many stationary scatterers in it, making it less

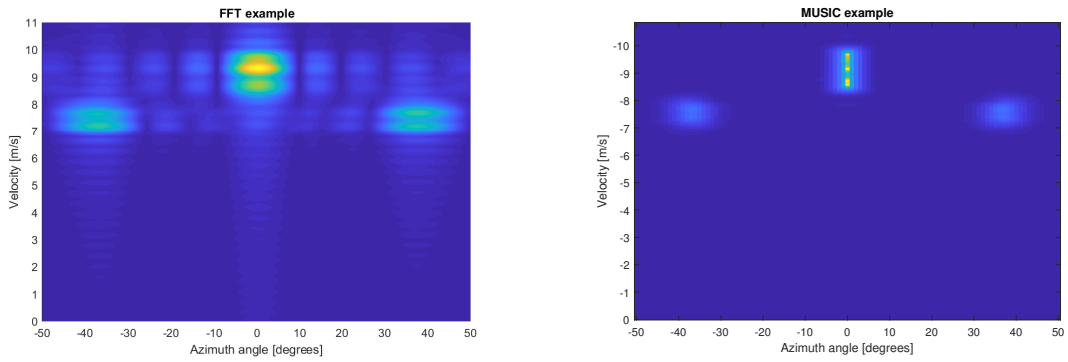
important to be able to take out moving targets.

4.3. Super resolution target acquisition

As was just described, for the concept to work, a relatively high resolution is required, either along the Doppler or the Range axis. To achieve this, a super resolution algorithm is required. The 3 methods that will be compared are MUSIC, ESPRIT, and STAP.

ESPRIT.
Of the 3 techniques this one is usually the least computationally expensive, but its also improves resolution least. It has been proven to be less accurate than MUSIC would be, in the same setup [37]. Another downside this technique has is that it only works for specific array configurations. Though we are planning to use a virtual Uniform Linear Array, with an even number of antennas (one of the array configurations for which ESPRIT works), different configurations might be used in the future. A sparse array for example, could be used to improve azimuthal angle resolution, but in a lot of cases, ESPRIT can not be used for this. For these reasons, we reject the use of ESPRIT in this particular application.

This leaves the most computationally expensive techniques. The problem is that their resolutions are hard to compare, as they are completely different imaging techniques. One is a so called Space-Time technique (STAP), the other is a subspace method (MUSIC), just like ESPRIT. To decide which to use, we looked at the average applications.
The main power of STAP is interference rejection, like clutter mitigation in MTI (Moving Target Indication). This could be useful in real world applications, to make sure that bad actors are unable to disable the system with a signal jammer for example. However, in the current stage of the research, we can safely assume that no one will be purposefully jamming our experiments.
MUSICs great strength lies in resolving targets close to each other, which is useful as it helps to gather more information about the structure of objects being observed.
Figure 4.1 shows images created using FFT, and using MUSIC, for comparison. As can be seen, the sidelobes are suppressed in MUSIC, and the 3 targets in the middle can now be separated.



(a) Image created using FFT

(b) Image created using MUSIC

Figure 4.1: Comparison of images created using FFT and MUSIC

4.4. Conclusion

Taking these things into account, we have currently decided to investigate MUSIC, to gather as much information about the shape of the world around the vehicle, though different approaches can be attempted in future research. One of the things to keep in mind while proceeding with the use of MUSIC, is the computational complexity, and how it might possibly be reduced[17] Furthermore, if the bandwidth, chirplength and pulse number are chosen correctly, the resolution might not have to be increased at all. For the height estimation the DBS approach was selected, and it is therefore necessary to have sufficient Doppler resolution going forward. Finally, for the SLAM itself, the genetic algorithm approach was chosen. Though more optimal solutions exist, finding an optimal SLAM solution is outside of the scope of this thesis, and the genetic algorithm will suffice.

5

Simulation Results

This section will briefly show some simulation results. For this simulation, 36 radar targets were simulated, around the cars trajectory, in the shape of a hexagonal prism. For these results, the genetic algorithm approach was used.
For the first simulation, the situation was as follows: The virtual car was 'driving' at 20 [m/s] in a circle with a 5 meter diameter, in this circle, a measurement frame is created every 1.8 degrees, making for 200 measurement frames for 1 entire circle. The car starts at X = 5 [m], Y = 0 [m], and completes 2 full circles during the entire simulation. The radar specifications can be seen in table 5.1, and a quick overview of the scheme of the simulation is shown in the flowchart in figure 5.1.

Carrier Frequency	78.5 GHz
Bandwidth	1 GHz
PRI	25.6 us
Number of Pulses	128
Beat Sample Frequency	40 MHz
Number of transmit antennas	3
Number of receive antennas	4
Max unambiguous range	76.8 m
Max unambiguous velocity	37.3 m/s

Table 5.1: Radar Specifications

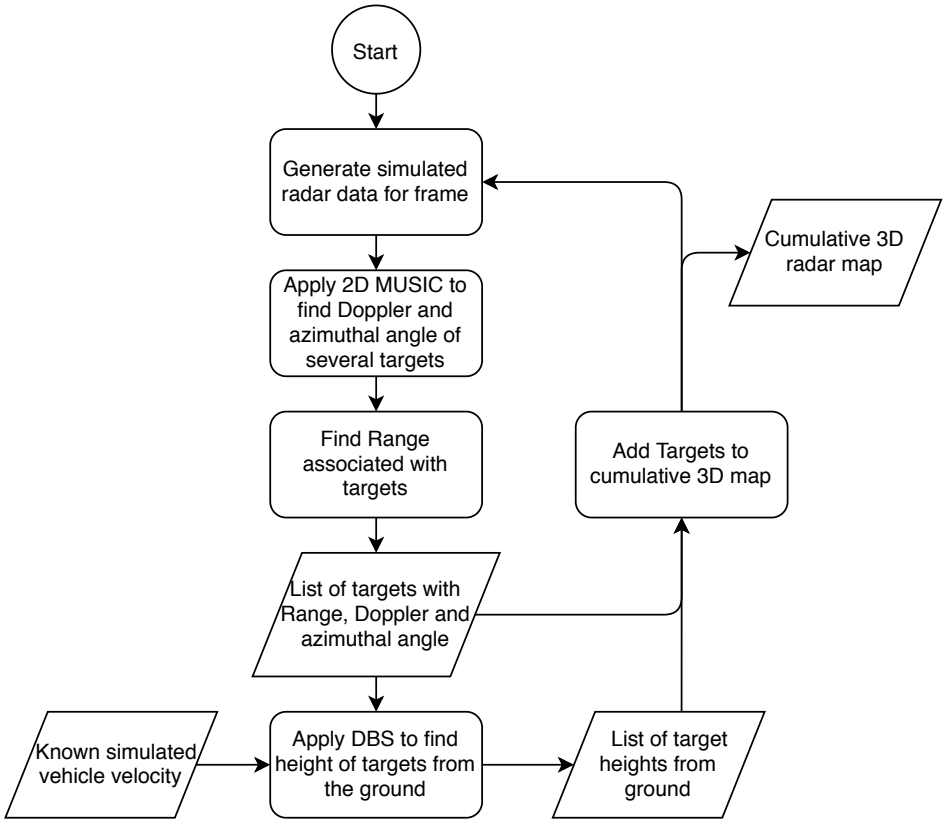


Figure 5.1: Flowchart illustrating the basic setup of the simulation

As can be seen in figures 5.2 and 5.3, the system manages to create a map that resembles the actual targets. There are several false points in the map however, this is mostly caused by the system wrongly estimating the range of a target. The trajectory is also estimated relatively well, and that is with a system for which much can still be improved in terms of performance, as will be discussed in section 5.1.

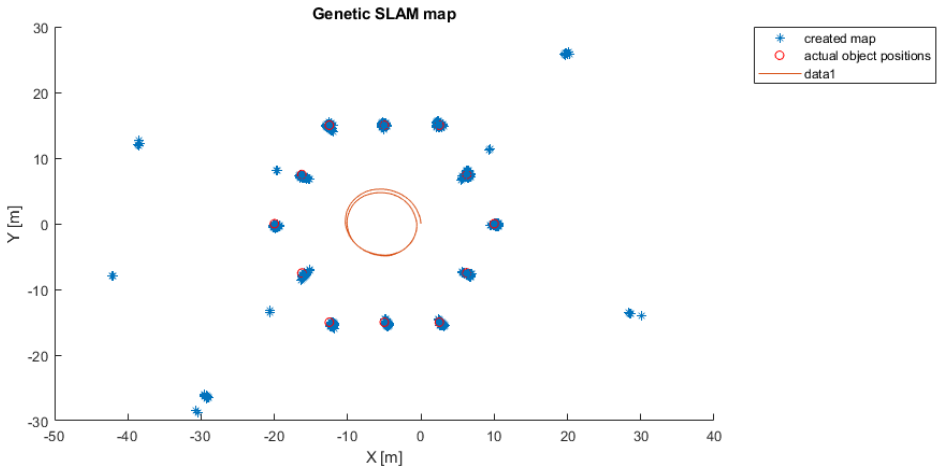


Figure 5.2: Top View of the map created, the red circle called "data1" is the estimated trajectory

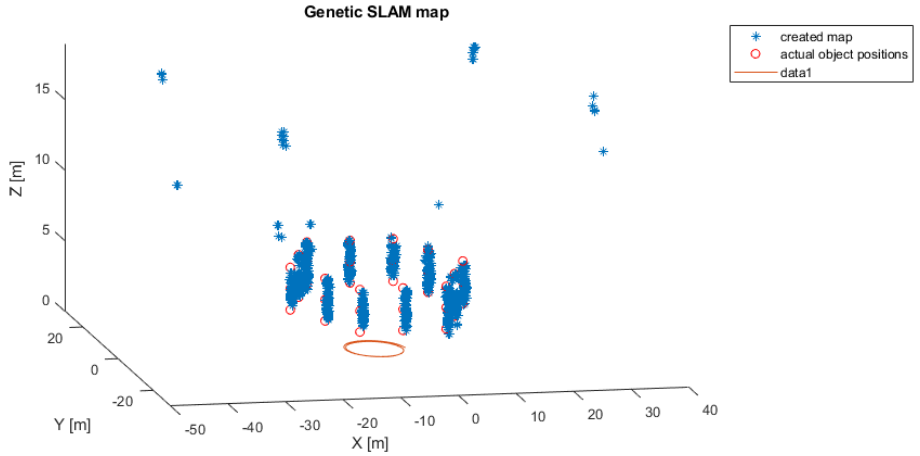


Figure 5.3: Isometric view of the radar map, the red circle called "data1" is the estimated trajectory

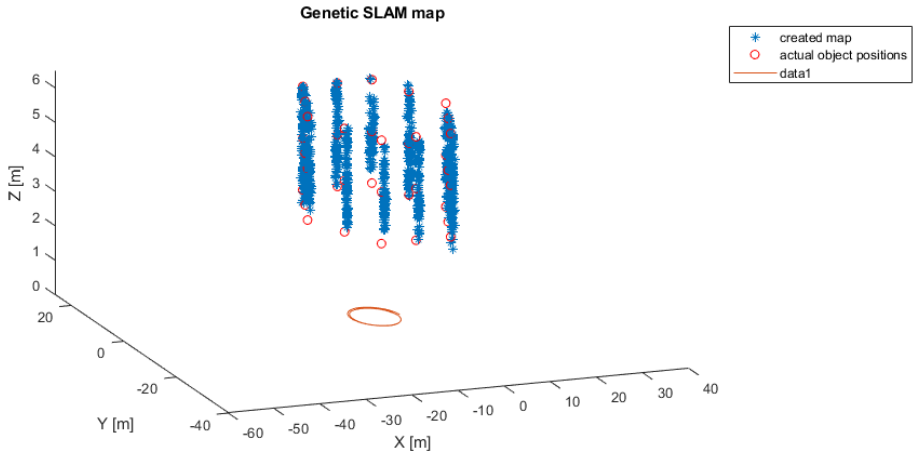


Figure 5.4: Isometric zoomed in view of created map, the trajectory is the red line called "data1"

It is worth observing the errors in localization during these simulations, as seen in figure 5.5. Here we can see that the errors stay within 1 meter. It is expected that there would be some steady state errors observed here, as the system can only localize itself on the map it is creating: it does not return an absolute position on earth. To get such a position, the SLAM map must ex post facto be aligned with actual coordinates on the earth, but as creating a system that returns latitude and longitude on the earth is outside the scope of this project, it is not executed in this thesis. Although the implementations of the systems created that use an a priori lidar map offer a solution to this alignment problem, in that the lidar map can be created such that it is already correctly aligned on the face of the earth.
Figure 5.4 shows a zoomed in version of this simulation, in which the wrong points can not be seen, to show the effect of changing 2 parameters, to make creating a map harder. This time, 2 circles are driven as well, except only 100 measurement frames are created per circle. And, to further pressure the system, the vehicle velocity has been lowered to 5 [m/s].

5.1. Conclusion

From these figures we can gather that this approach, with the genetic algorithm radar SLAM, can work, at least in simulation. We have achieved a system that can localize a vehicle equipped with an automotive MIMO radar, while mapping the environment at the same time. The genetic algorithm created, in concordance with the NDT, can be used to realize SLAM given a list of targets in every radar frame, and said target list could be generated by using MUSIC to hypothetically increase the precision of the system, if necessary.

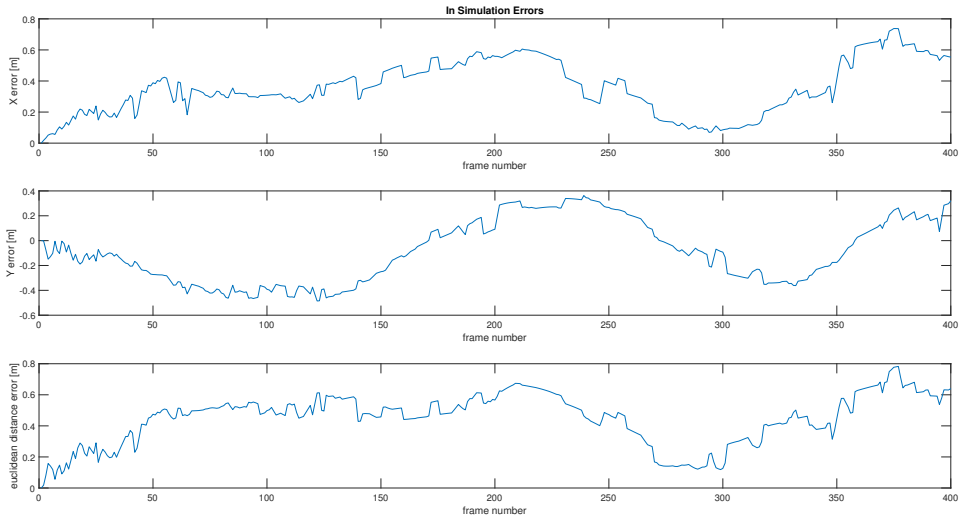


Figure 5.5: Errors in positioning along respectively X, Y, and total euclidean distance.

6

Experimental verification

A 3D SLAM system was created using the previous findings, and tested with real world data, to try and determine some of the possible future challenges, and to tailor the system to the given application of localizing a car with the data at hand. In this pursuit, we engineered several adaptations to the previously discussed techniques and algorithms.
First, the current system application for real world experiments will be discussed as it stands. After that, the output and results of the system will be shown and discussed, and then finally, some of the future challenges will be laid out.

6.1. Adaptations

This section describes how the current system works. The current system is, for the most part, a combination and implementation of modules that were previously discussed, made to work together.
The input for the current system is a list of targets (i) detected at every radar frame(j), $T_{i,j}$.

$$T_{i,j} = (R_i, \theta_i, v_i, M_i) \tag{6.1}$$

Where R_i is the range, θ_i, is the azimuthal angle, v_i is the perceived Doppler velocity and M_i is the RCS magnitude. Using these targets, a SLAM loop is run, with the previously discussed genetic algorithm. Some minor changes were made to this algorithm, to improve its stability, taking into account vehicle behavior of the car.

6.1.1. Adapted Genetic Algorithm

The genetic algorithm takes the same shape as described in chapter 3 and chapter 4, with 2 major changes: Chromosomal symmetry, and an updated breeding scheme. To start off with the chromosomal symmetry.
The chromosomes represent moves (rotation and translation) that the car can make in between radar frames, and, taking into account the symmetrical nature of the vehicle, this opens the genetic algorithm up to a constraint: For every high fitness move that is discovered its symmetrical

partner is saved as well. For example: if the genetic algorithm comes to the conclusion that the highest fitness move is 1 meter forward, 10 cm to the left, and rotated 10 degrees to the left, then the mirrored version of this move is saved as well, which would be 1 meter forward, 10 cm to the right, and 10 degrees to the right, see figure 6.1.

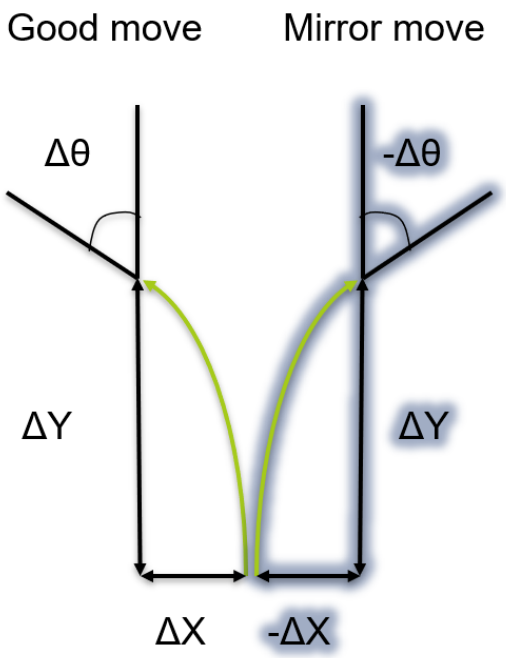


Figure 6.1: Figure depicting the mirror move principle now included in the genetic algorithm

This is done to increase the rate of convergence, and to account for the fact that often, after a car has (over)-corrected in one direction, it is likely to soon after steer in the other direction.
The second thing thing that was changed is the way chromosomes are bred, specifically: the way chromosomes are mutated. Previously, if a mutation occurs during the breeding stage, this entails that a random gene in the chromosomes is "reset" to a random value. The adaptation that was made is that now, instead of changing said gene to a random value, it is changed to a slight and random variation of the old gene value.

$$G_{new} = G_{old} + \Phi \tag{6.2}$$

Where G_{new} and G_{old} are the new and old values of the chosen gene in the chromosome respectively. Φ is a random value, with probability density function:

$$f(x) = \begin{cases} \frac{1}{2a} & \text{for } x \in [-a, a] \\ 0 & \text{otherwise.} \end{cases} \tag{6.3}$$

This continuous uniform distribution can be can be seen in figure 6.2. This way, instead of choosing mutations at random to try and find new good moves, this version of the algorithm tries to iterate on the moves that have already been proven to be good. In equation (6.3) the value of a can

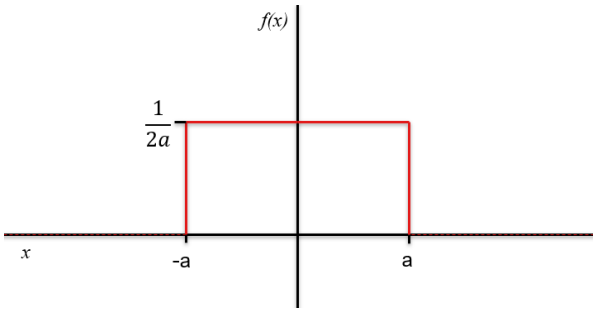


Figure 6.2: Figure depicting the probability density function of Φ

be tuned to increase or decrease the variance of Φ, with the aim of increasing the rate of convergence of the genetic algorithm.

6.1.2. Ego Motion

One of the most important factors to establish for Doppler Beam Sharpening [16], is the ego motion: The current velocity of the car. This is because in DBS the velocities of all detected targets relative to the car need to be compared to the car's velocity on the road. To accomplish this, there are at least two techniques/factors that can be used.

SLAM based ego motion estimation The first is the output of the SLAM algorithm. That is to say: the distance moved between frames, the trajectory, can be used as an estimate for the velocity. If a 2D SLAM algorithm is used for example, one in which DBS is not necessary, as the map being created is only 2D, the velocity estimate gathered from the estimated trajectory can be used as a velocity estimate for DBS, to create a second map that is 3D.

Radar target information based ego motion estimation The second technique relies on creating an ego motion estimate from the detected radar targets. First, all targets are assumed to be stationary targets, none of the detections are assumed to be false, and all targets are assumed to have no elevation angle relative to the radar, only an azimuthal angle. Then every target can be used to generate an estimate of the velocity of the car as follows:

$$\Omega_i = \frac{v_i}{\cos\theta_i} \tag{6.4}$$

In this list of velocity estimates, outliers have to be removed. These outliers can represent non-stationary targets, or fully false detections. After these estimates have been removed, an estimate of the velocity of the car can be created. This is done by analysing the distribution of the velocity estimates. This was done by first creating a histogram distribution for all the velocity estimates in a single frame. As can be seen in figure 6.4, there are several outliers in the set of velocity estimates $\mathbf{\Omega}$. For example, in the figure, it can be observed on the right side that there are targets that are perceived to have a positive velocity relative to the radar. This would indicate that the targets are moving away from the car, even thought the car is driving forward. Though this is not impossible, it is also a possibility that there were targets moving towards the car, such that the relative velocity

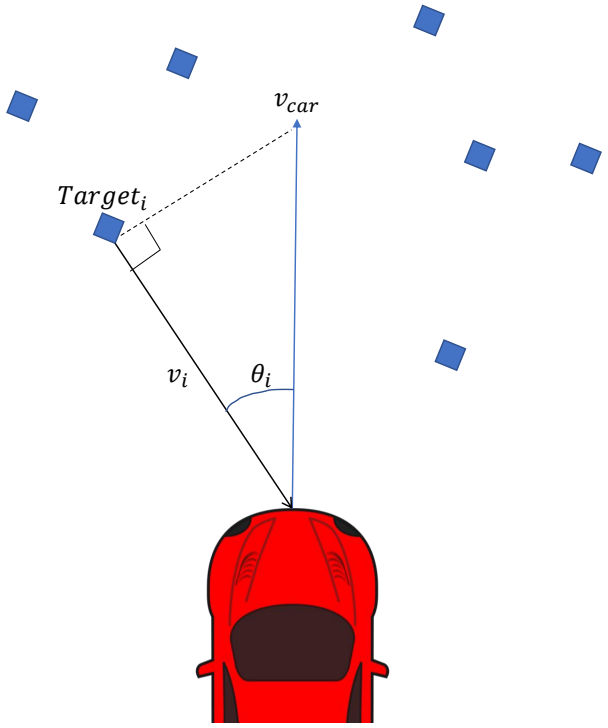


Figure 6.3: Diagram depicting a set of targets in front of the car, with measured velocities v_i, and azimuthal angles θ_i

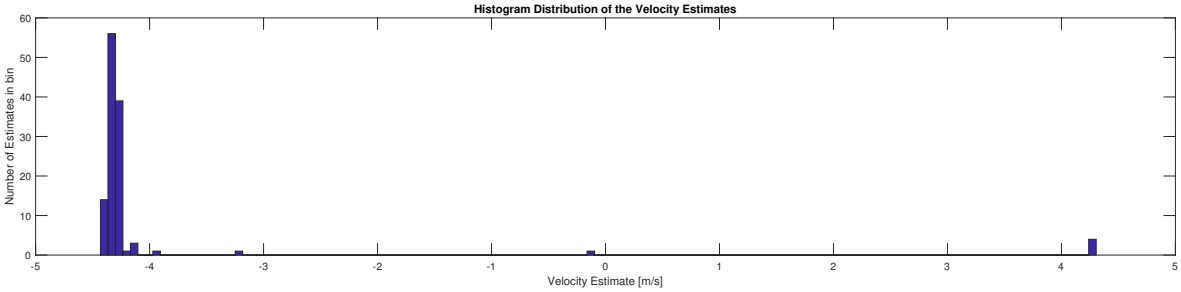


Figure 6.4: Example histogram of the Velocity estimates. Notice the outliers, even with a positive velocity on the right, indicating the target would be moving away from the car in theory.

between the car and the target is above the maximum unambiguous Doppler velocity, and because of aliasing the target wrapped around and shows an incorrect Doppler velocity.
In the histogram, the center of the highest bin is taken as the final velocity estimate, until it is further refined. There are several factors that can still be tuned that affect the outcome. The amount of bins in the histogram for example. By increasing the amount of bins, the amount of possible final velocity estimates increases, the resolution along the X axis of figure 6.4 increases as it were. This comes, on average, at the expense of resolution along the Y axis. If the amount of bins is increased, then the height of the highest bin is likely to decrease. This makes it more likely that there are multiple highest bins, of the same size, in which case the average of the centers of these bins

can be taken as the velocity estimate. Another factor that has to be taken into account when increasing the amount of bins is computation time: If the amount of bins is taken high enough, the system might fail as it is no longer able to calculate things within a reasonable time-frame.
After this process, a velocity estimate has been created for every frame in time, which can then further be refined by means of fitting. As can be observed in figure 6.5, in this estimation pro-

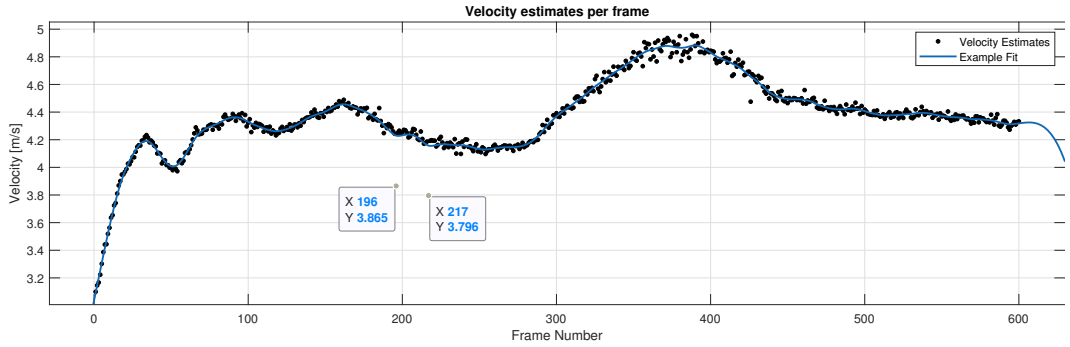


Figure 6.5: Figure depicting the output of the histogram based velocity estimator, with an example fit given in blue. The two estimates that are highlighted are assumed to be outliers.

cess, there are still outliers, such as the estimates for frames 196 and 217. Both these outliers and some of the noise/jitter can be removed by means of fitting. The example fit given is a smoothing spline fit $s(n)$, which is also used in the final results. This fit is generated by trying to minimize equation (6.5),

$$p\sum_{n}(v_e n - s(n))^2 + (1-p)\int(\frac{d^2 s}{dn^2})^2 dn, \tag{6.5}$$

where n is the frame number, $v_e n$ is the histogram velocity estimate for frame n, and p is the smoothing parameter. The smoothing process is executed manually, tuning the smoothing parameter p until a result such as observed in figure 6.5 is reached. In this process points that are suspected of being false outliers can also be removed manually. This method of fitting is called a cubic smoothing spline interpolation. A general description of the method is as follows: given a set of points (x, y), equation (6.6) is to be minimized[38].

$$\underbrace{p\sum_{j=1}^{n} w_j |y_j - f(x_j)|^2}_{\text{error measure}} + (1-p)\underbrace{\int \lambda(t)|D^2 f(t)|^2 dt}_{\text{roughness measure}}. \tag{6.6}$$

In equation (6.6), n denotes the amount of entries of x and y, and the integral in the roughness measure portion of the equation is to be executed over the smallest interval that still contains all entries of x and y, where $D^2 f$ is the second derivative of function f. Furthermore, the weights of the entries can be set by setting w_j, though all entries of w_j are set to 1 by default. In a similar manner, the importance of the roughness over the interval of the fit can be changed by adapting $\lambda(t)$, such that the roughness of the fit in certain locations can be made more or less important in the outcome of the entire integral, but $\lambda(t)$ is set to be a uniform 1 over all t by default. By increasing p, the error measure part of the equation becomes relatively more important in the outcome of the equation, and the fit will on average have a lower mean square error, although it will also be less smooth. Conversely, by lowering p, the roughness measure portion becomes more important relative to the error measure portion, and as such, the "roughness" of the final fit will

decrease, although the fit will have a higher mean squared error. As such, by tuning the value of p, one can make a weighted decision between smoothness and mean square error.

This estimate is limited in accuracy by the Doppler resolution, and therefore a system that has a high Doppler-Velocity resolution should be able to able to estimate the ego motion with a high accuracy as well. To separate between an elevation angle of 0 degrees, and φ degrees, right in front of the car (boresight), while driving g [m/s], a difference in velocity (Δv) must be measured that can be described as such:

$$g - gcos(\varphi°) = \Delta v. \tag{6.7}$$

Another factor that must be kept in mind is that this estimation carries within it a slight steady state error. This is caused by the fact that the elevation angles of the targets are not 0. This means that this velocity estimator is biased. This bias should be estimated and minimized in future work.

6.1.3. Proposed mapping method

Using SLAM and DBS, a 3D map of the environment can be generated. One problem that must be overcome in this process is the elevation angle ambiguity. From the Doppler information we can gather that a certain target is some distance Z above the horizontal plane of the radar, or below it. To resolve whether the target is above or below the plane of the radar, one could try to use a simplified version of Multipath Height Finding [31], but this assumes an ideal situation, and tends to fail at larger distances in an "object rich" environment.
Proposed is the following method to overcome part of the ambiguity. We assume that detected targets do not exist underground, and therefore, for example, if a target is found to either be 5 meters above the horizontal plane of the radar, or below it, we assume the target to be above the plane, as it is unreasonable to assume that a target was detected underground. This resolves a part of the ambiguous targets. See figure 6.6.

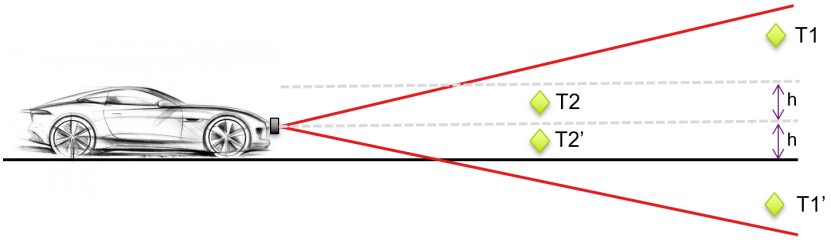
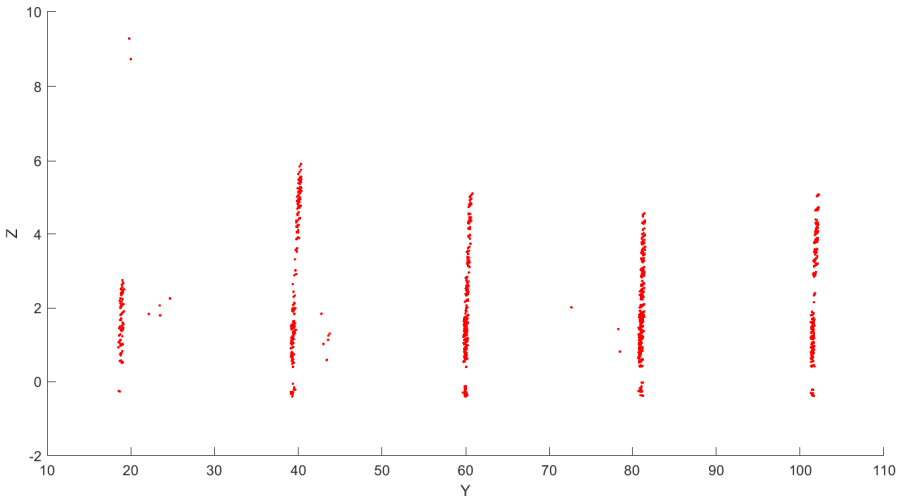


Figure 6.6: Figure depicting the problem with ambiguous target heights, the red lines indicate the beamwidth in elevation, h is the height of the radar from the road, and T1/T2 are the detected targets.

As can be seen in figure 6.6, targets like T1/T1' can instantly be resolved, as we can assume targets not to be underground, so T1 is correct, and T1' can be discarded. For T2/T2' however, this can not simply be done, as both possible target locations are feasible. To resolve this, we propose the following: Always disregard the higher target. The reasoning for this is as follows: We assume the area between the ground and the horizontal plane of the radar to be more densely populated by objects, like potholes, speed-bumps, manhole covers and curbs, than the area between the 2

segmented gray lines in figure 6.6. This means that in mapping, the layer between said aforementioned gray lines is always kept empty, see figure 6.7. Future mapping implementations might want to work around this, but for the purposes of mapping and localization, this does not harm the performance in a noticeable way.



(a) Figure depicting the hole created by the current mapping method, in 5 lantern poles

(b) Image of one of the poles

Figure 6.7: 3D mapping of 5 lantern poles with an image of one of the lantern poles

6.1.4. Current System

The combination of all these modules yields system shown in figure 6.8. In the first frame, when the radar starts working, the currently observed targets are used to create the cumulative map, to which the system will add data points from the next frame on. This is the initialization frame. Next, another radar measurement is taken, and a list of targets, with their respective Azimuthal angles, ranges, RCS's and Doppler velocities is created. Using the Range and Azimuthal angle information, a 2D map of the current targets is created. It is important to note, that in this implementation, the SLAM still takes place in 2D. It makes 2D maps of the current targets, compares these to the 2D cumulative map, and then adds to them. This can be changed, which will be discussed in the chapter on future work.
As stated: a 2D map of current targets is created, and fed into the genetic algorithm. From the currently observed target list, an ego-motion estimate is also created, as described in the Ego-Motion subsection, which is also fed to the genetic algorithm. The genetic algorithm then determines heuristically to what degree the car has moved between frames, the 2D current target map is rotated and shifted accordingly, and added to the cumulative map. Finally, using both the ego motion and estimated trajectory from the SLAM, another map is created, a 3D map, using DBS.

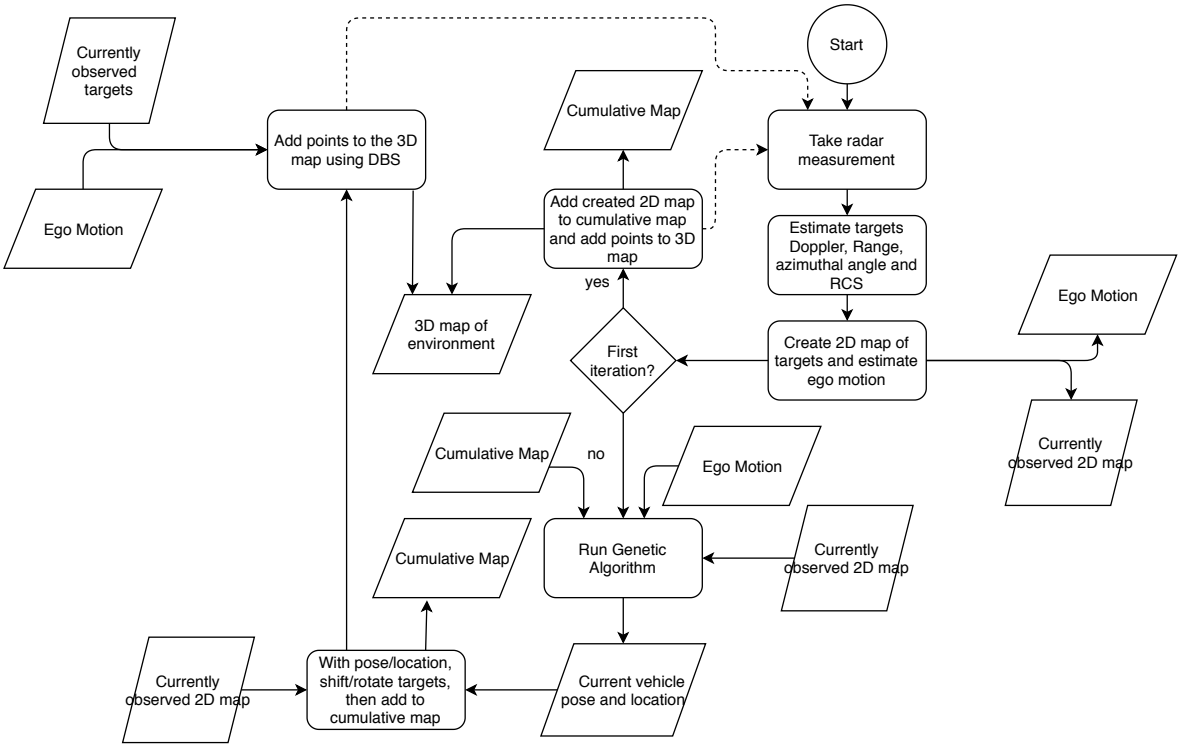


Figure 6.8: Flowchart depicting one of the current system architectures

6.2. Results

This section will show some of the results of the current 3D mapping system, in an attempt to show its feasibility. First, a 3D map of the whole environment will be shown.

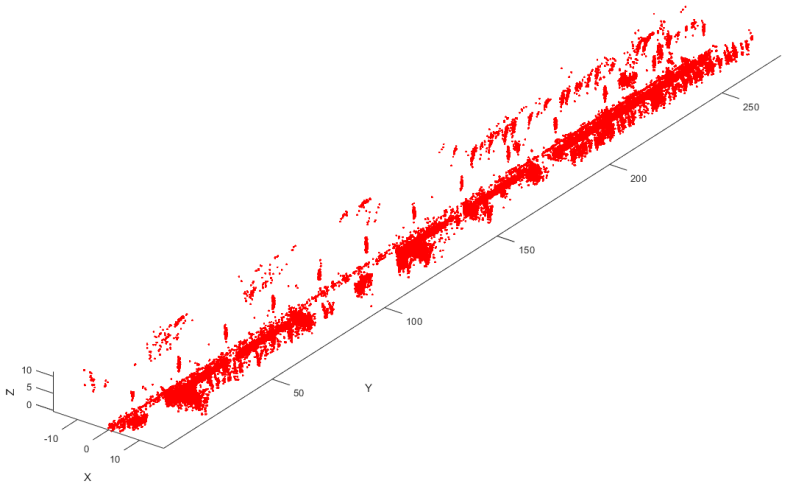


Figure 6.9: 3D image of entire environment

As can be seen from figure 6.9, a 3D image of the street the car drove through begins to appear, though there are still several things that can be improved. It is observed in the results that errors in the order of 0.01 m/s in the ego motion estimation, can cause aberrant errors in the height estimation. This means that the image can be improved by improving the methods for ego motion estimation, which are aided by the SLAM system. Furthermore, for these results, no MUSIC algorithm is applied to attempt to increase the Doppler resolution (as the implemented MUSIC algorithm did not sharpen the image), errors in which can cause the same magnitude of errors in the height estimation, as errors in the ego motion estimation can cause.
The next result example seems to show the system working correctly, though not as precise as is desired of the final product.

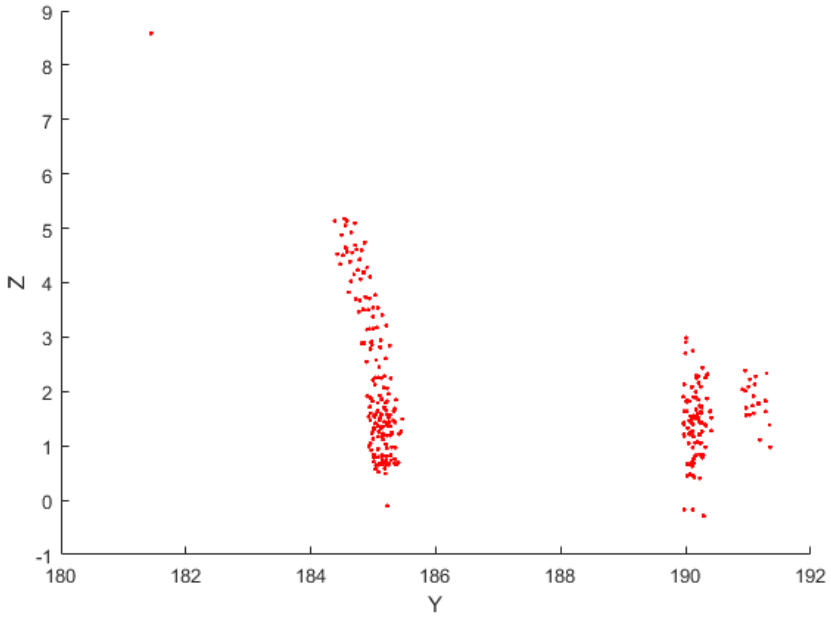


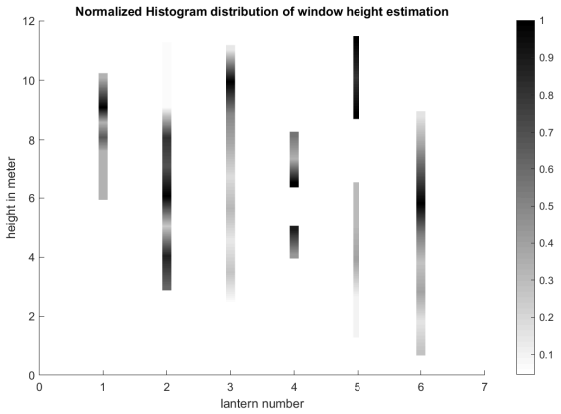
Figure 6.10: Side view of a lantern (on the left), and a group of people having a conversation (on the right)

(a) Image from the camera footage of the experimental run with the car, showing both the lantern and the group of men having a conversation.

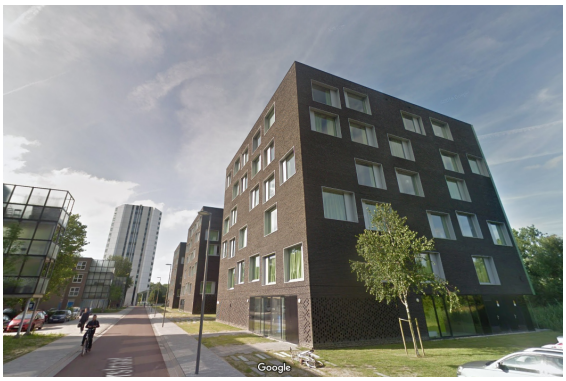
(b) Image displaying a different angle of the group of talking men, showing a 4th person standing about a meter further down the road, explaining the rightmost set of points in figure 6.10.

Figure 6.11: Comparison of images created using FFT and MUSIC

As is shown in figure 6.10, the current system seems to be able to distinguish between the height of a lantern, and the height of a group of people. Figure 6.11 shows what the scene looked like when the car drove past it. The lanterns are slightly under 6 meters tall in reality, when measured with a laser mater. The group of people is estimated to have a max height of 3 meters by the algorithm, which is incorrect, but can be the result of errors in one of 3 factors: The Doppler estimation of the radar target, the ego-motion estimate of the vehicle, and the azimuthal angle estimate of the target. Further improvement of the qualities of these 3 factors is necessary if more precise height estimation is desired.
Another part of the 3D radar map that indicates that the system is working can be found in the height data of several windows on buildings along the road. This is because the windowframes are quite deep and made of metal, making them corner reflectors with a high RCS, relative to the rest of the building. Therefore all radar targets detected along the side of these buildings come from these scatterers. First, the detected targets from the area of the map where the windows are located can be extracted from the 3D map, as seen in figure 6.12d and figure 6.12b, and the separate windows isolated. Then a histogram distribution of the height values of these windows can be generated, as seen in figure 6.12a. In this height estimation histogram set, we can notice first of all, that the leftmost histogram is located high up, and has no "lower" part, which would indeed be correct as it is the only window on the 3rd floor of the building that has no window below it. To further illustrate that the system has height estimating capabilities, is by overlaying the histograms on an image taken of the facade with a cellphone, as seen in figure 6.12c. Though the height estimations are going to be off, because of aforementioned error sources, such as errors in ego motion estimation and limited Doppler resolution, it can be observed that the histograms do align with the positioning of the windows on the facade. One last thing to note is that not all the windows are the same, some windows have a long white metal portion on the left side, some on the right side. This might explain why the top window might show up more brightly then the bottom, or vice versa, as the car was passing the facade from the left side to the right side of figure 6.12c. For example, the top window at "window number" 3 is more prevalently present than the bottom. Same thing goes for "window number" 5.

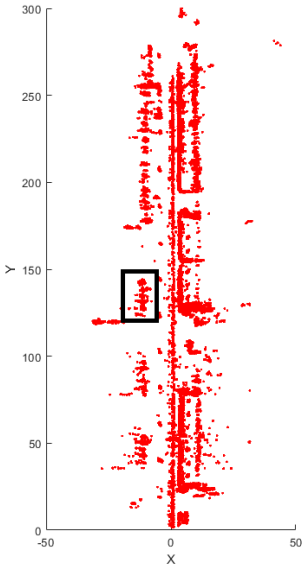


(a) Histogram distributions of the estimated heights of strong point targets possibly scattered from the frame of a window



(b) Image of the building being used to establish height estimation functionality, taken from google maps

(c) Height estimation of windows overlaid on the facade of the building itself. Note that the height axis does not align with the actual height of the building, it is merely included for illustration purposes



(d) Area of the radar map from which the windows of the building are extracted

Figure 6.12: Figures and images used to establish the working of the height estimation system

7

Incorporation of Lidar

This section will go over the the way in which lidar can, and was, added to the system, to aid in localization of the vehicle. 3D lidar pointclouds proved to be too large to be worked with simply, so a method of reducing the data was devised, and will be discussed here.

7.1. Lidar vs. Radar

In this thesis, all practical tests focused on one street on campus, on which the car can localize itself using only radar. But as discussed in the beginning of the thesis report, the eventual aim is to add in lidar data as well, as pre-made lidar maps are already available for all of the Netherlands (and most of the rest of the world), and using them as well could improve the fidelity of the system. Lidar data will effectively be used as a starting map, a map that is available before the first radar measurements are added to this map. As if the car had already gone past this street once, and build up a map in doing so with the radar. Except, this map is generated using a lidar sensor, which means there are multiple things to take into account.

The first and most important thing is that lidar and radar don't always see the same things. Radar might see through things that a lidar does not, and vice versa. Lidar might be able to see through a metal mesh, or though a window. Lidar detects objects, even when viewed at a very shallow angle, whereas the RCS of an object can be very dependent on the angle from which it is perceived. The lidar sees the road surface itself consistently, in most cases the radar does not. On the other hand, radar can see through rain better, can see through fog better, and depending on the frequency band being used, might even see through walls. These are all merely examples of situations in which the two sensors might perceive something different from one another.

Another thing to take into account is moving targets, or rather, the methods of removing them from the map. Where a radar has the Doppler information which can be exploited to take out moving targets, the lidar is not in possession of such information, and therefore a different method of removing moving targets must be devised.

Thirdly, as previously stated, the lidar data is immensely dense. Where, as an example, on an average run of the street the cumulative radar map that is generated consists of around 3500 points, the entire lidar map consists of over 55 million points. Manipulating that amount of data is not

feasible live. Therefore, the amount of data must be reduced.

Lastly, the lidar data, the pointclouds, are 3D, whereas the cumulative map being created by the radar for the purposes of localization, is only 2D. The radar has a very narrow elevation beamwidth, of around 4 degrees, as is common for most automotive radars, the lidar has a much wider Field of View in elevation. Besides this the lidar was placed on top of the car, whereas the radar was placed on the bumper. These geometric differences can also be kept in mind, when comparing and combining lidar and radar data in the same application.

7.2. Proposed Method

Bearing all these things in mind, the following method for crafting an a priori lidar map was devised.

For the purposes of understanding the chosen solutions it is essential to separate key problems.

1. The lidar map is 3D, and must become 2D for the purposes of the current SLAM.
2. The lidar sees the road, radar does not, therefore the road must somehow be removed.
3. The lidar pointcloud is too large, and must be reduced in size before it can be worked with.
4. The lidar data contains moving objects, such as cyclists, which need to be removed.

Starting off with problem 2, the chosen solution is: take only the points from the pointcloud in a certain height range. For example: take only the points from j cm below the lidar up, effectively extracting a half-space from the total 3D pointcloud, and choose j such that the road is selected out, which can depend on the height of the radar from the ground, for example. This takes out the road, and any aberrant objects that are high up from the ground, that the radar cannot see in most circumstances. This solution also affects problem 3, in that it reduces the amount of points already. And then,to solve problem 1, remove the height from the pointcloud, flatten the 3D pointcloud to 2D by doing so. This 2D pointcloud still has as many points in it as before this step.

The next step in the creation of a lidar map that can be used as an initial map in the SLAM system, a map that must have a manageable amount of points, is 2D, and has no moving objects in it, is a single solution. That is to say: there is a single chosen solution that simultaneously solves problems 3 and 4.
The solution is based on the statistical analysis, in the form of the creation of a histogram, being used to find the places in the pointcloud that have a high density of points. The flattened map is input into a 2D histogram function, with set bin coordinates. Effectively, a grid is placed over the 2-dimensional map, a grid consisting of 1/3 by 1/3 meter squares. Then, the amount of points within every square is counted. Finally, a threshold is set: Find the squares/resolution bins with at least n amount of points in them. The value of n in was manually tuned to gain a map in which the lanterns, sides of buildings, and tree trunks were all clearly visible and not removed by this part of the lidar map creation. An example of such a density/histogram plot can be seen in figure 7.1. As can be observed in said figure, things like walls, lantern-poles and and trees show up brightest. Note that in this plot the road has not been removed from the data, but as the lidar points on the road don't stack up in the Z direction, the road still appears less bright than any vertical object. The center coordinates of these bins, the "square feet" on the map that contain at least n amount of points, are saved. These coordinates form the new map. There are several factors that can be

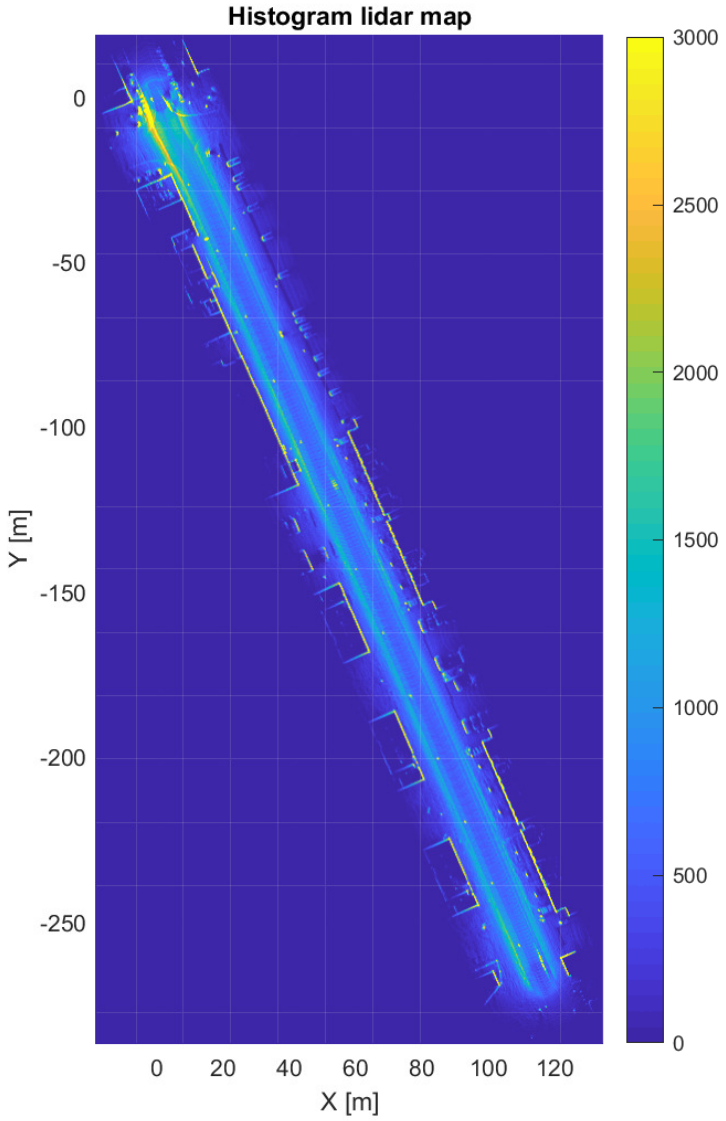


Figure 7.1: Histogram map of the lidar data, the color indicating the amount of lidar points per bin.

tuned, from the threshold n to the size of the tiles of the grid, but these are the values used as of now. This method inadvertently solves both problems 3 and 4. Through this approach the amount of points is reduced drastically, starting off with 52 million points, in this set-up we end up with a map of the street consisting of around 1500 points. This method of creating the map also favours stationary objects, as the lidar points will add up consistently in the same place for stationary objects. A moving object will show up on the map as smeared out over its trajectory while it was being observed, with their points spread out across the squares in the trajectory, instead of all ending up in a smaller amount of squares. Moving things show up in the original lidar map with a lower density per area, than stationary object. As such, choosing the correct tuning values ends up removing

all moving targets from the map.
To most easily allow the localization algorithm to lock onto the lidar map, it was necessary to first rotate it such that the street was oriented vertically Starting off with the map shown in figure 7.2, the map shown in figure 7.3 is created, for use in the SLAM algorithm.

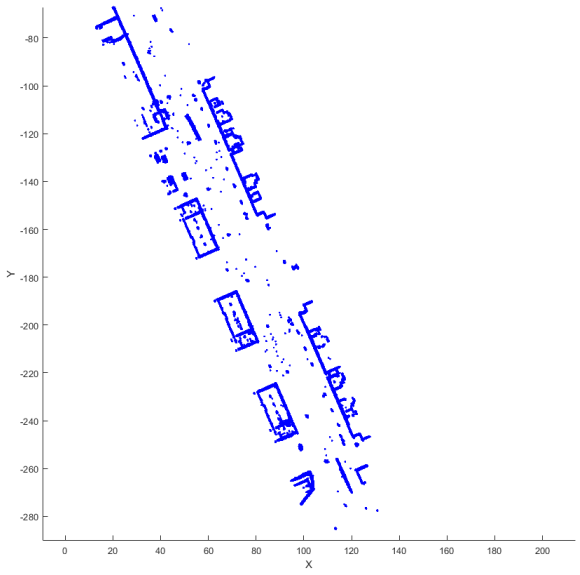


Figure 7.2: Overhead view of the lidar map of the street.

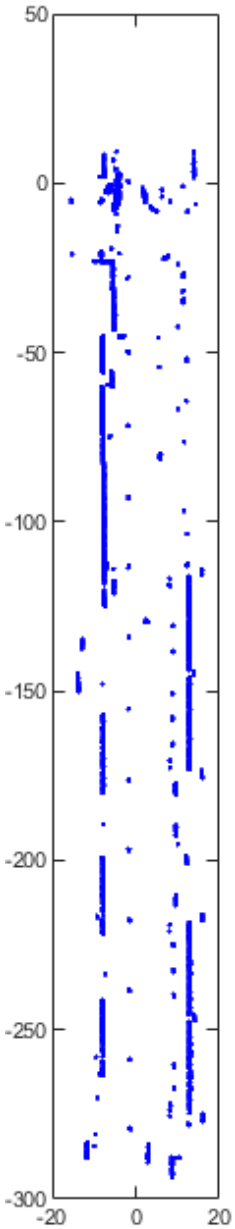


Figure 7.3: Overhead view of the a priori compressed lidar map of the street, to be used in SLAM

7.3. Implementation

This subsection describes how the now created a priori lidar map can be used in coordination with the already existing system. The final product should rely heavily on the lidar map at first, as the

lidar map has a higher fidelity than the radar map, given that the lidar itself had a higher precision, and a cutting edge GPS system was used to determine the trajectory. But, as the system adds more to the cumulative map, the system should start to rely more and more on the radar data, given that this map will better represent what a radar would perceive. This is accomplished by running the Fitness function, used in the genetic algorithm, twice. Once comparing the currently perceived radar targets to the a priori lidar map, generating the fitness value F_{R2L}, and once more comparing the currently perceived radar targets to the cumulative radar map, generating the fitness value F_{R2R}, and then combining the two values, with weights, into a new fitness value F_{new}, as shown in equation (7.1).

$$F_{new} = Q * F_{R2L} + (1 - Q) * F_{R2R} \tag{7.1}$$

As the amount of points in the cumulative map grows, so will the average value of F_{R2R}, and therefore, as time goes on, the value of F_{new} will become more dependent on F_{R2R} over F_{R2L}. The currently chosen value for Q, the weighting factor, is 0.75, though this can be tuned further in the future.

Using this new fitness function in the genetic algorithm, a new trajectory can be estimated. The Final result is shown in figure 7.4. As can be observed in the figure. the car is able to localize itself on the street, using both lidar and radar data.
This new system can then be represented by the flowchart observed in figure 7.5. In this new system, both the a prior lidar map, and the cumulative radar map are being related to the current radar data, to estimate the vehicles current location. As no mention of such a system can be found in the literature, it is indeed assumed to be a novel method of vehicle localization. However, another scheme has also been tested. In this scheme the radar data is not being saved in any way: the system only has access to the current radar observation. In any practical application this would mean that the system needs less memory, and will only ever have to download data from the cloud (the a priori lidar map, before driving off), and never upload it. Though it might come at the cost of precision, accuracy and stability, which at this moment in time is still under investigation, as more data and more research time than is currently available for this thesis would be required.
This streamlined version of the previous localization method can then be described as shown in figure 7.6. This method of vehicle localization has been shown to also be able to provide a location estimate, as shown in figure 7.7. To give an impression of the stability of both localization implementations that make use of lidar data figure 7.8 was created. It is interesting to note that both implementations show more likeness to each other than to the radar trajectory, which is known to have an inaccuracy in the orders of several meters.

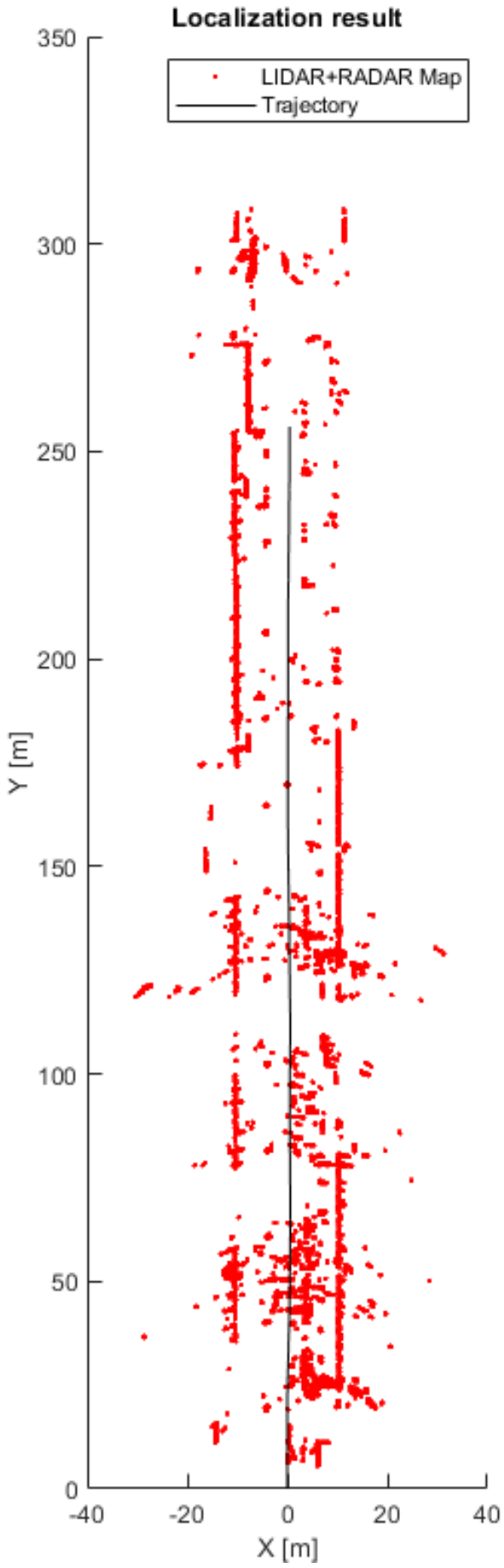


Figure 7.4: Overhead view of the final result.

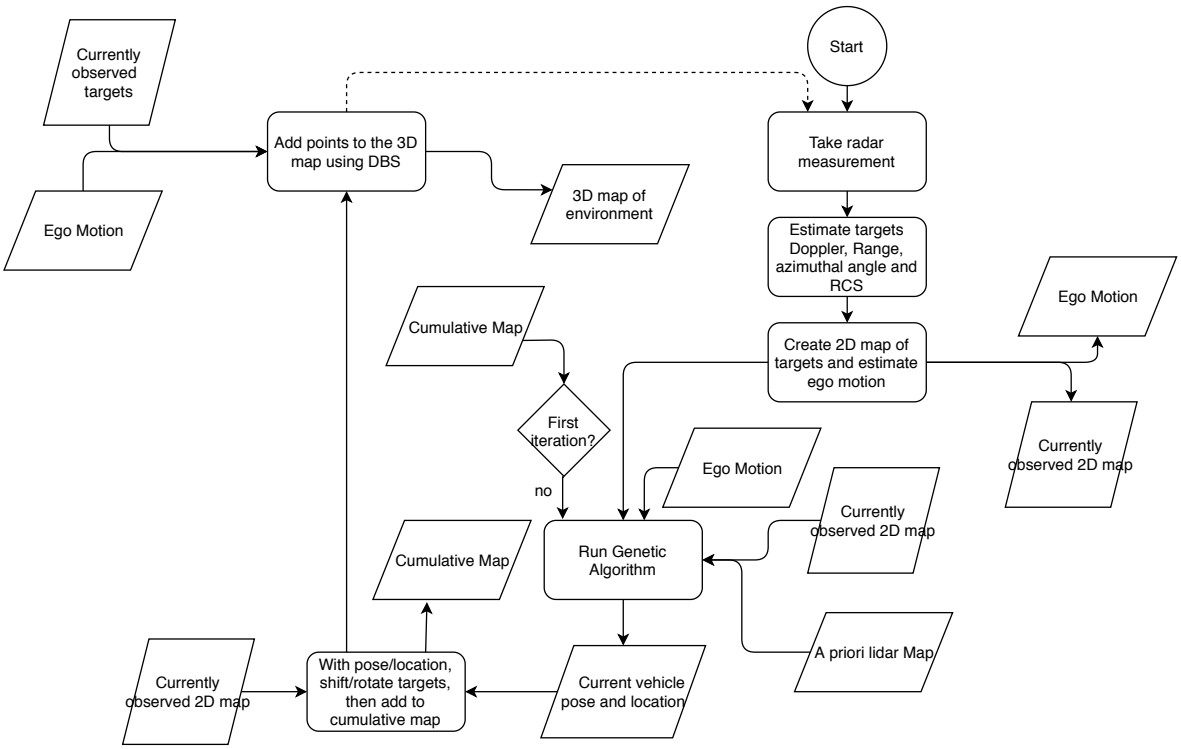


Figure 7.5: Flowchart depicting a method of localization of a vehicle with the radar, using an a priori lidar map

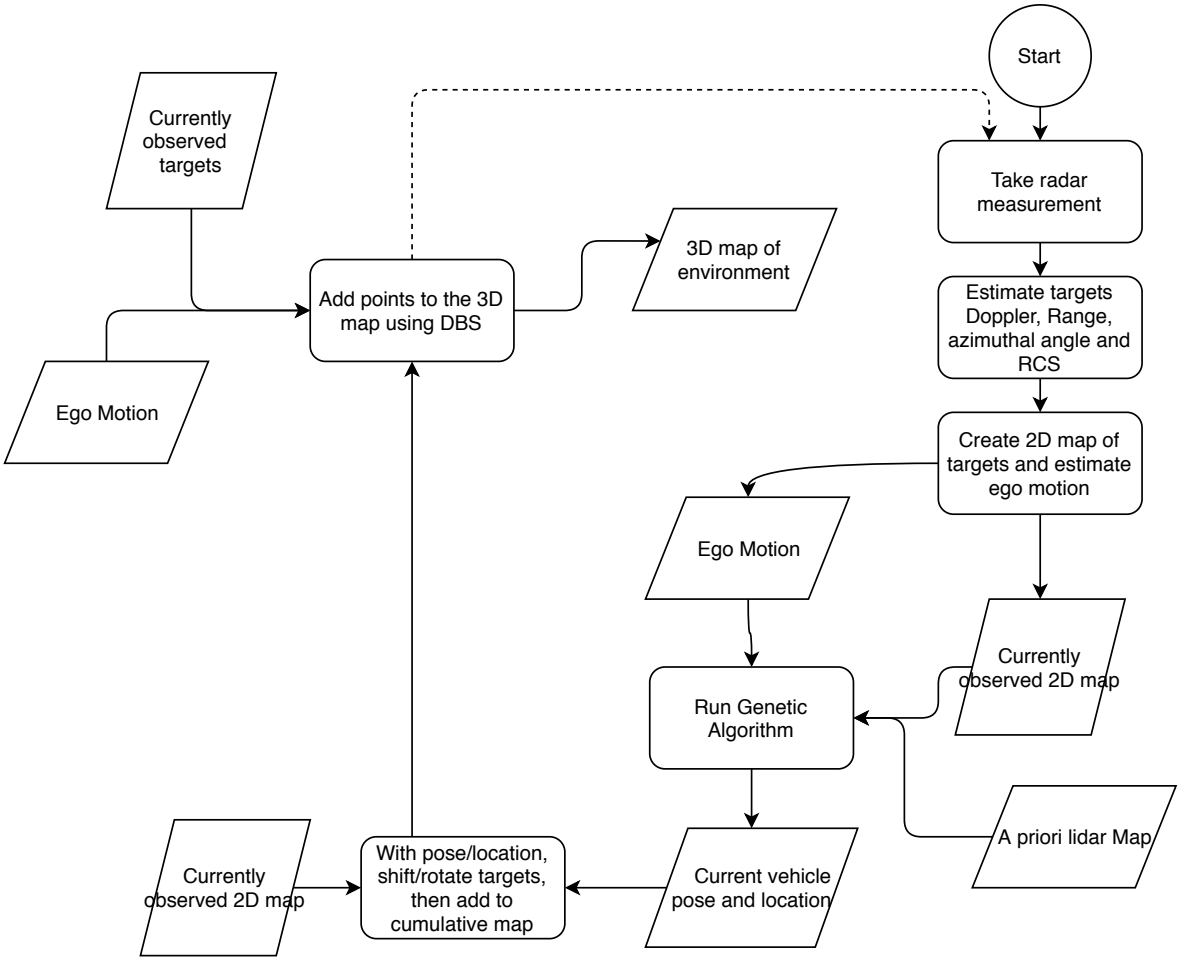


Figure 7.6: Flowchart depicting the 2nd method of localization of a vehicle with the radar, using an a priori lidar map

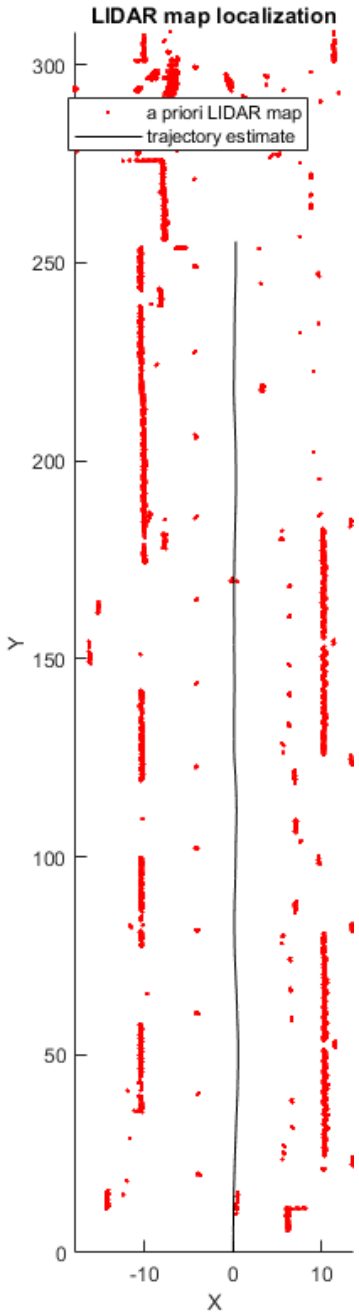


Figure 7.7: A priori lidar map with the estimated trajectory of the vehicle

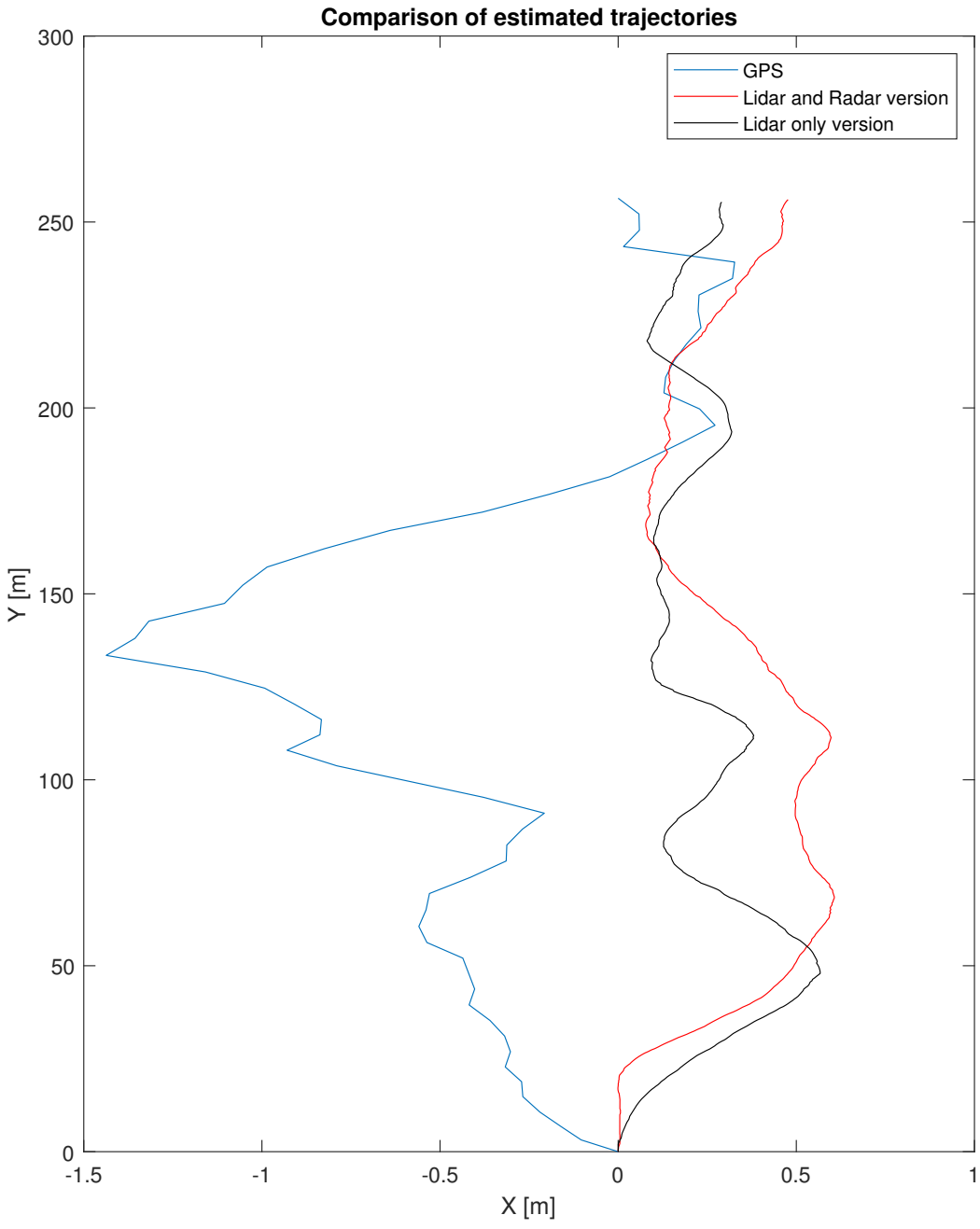


Figure 7.8: Figure comparing the estimated trajectories of the system that saves and uses both radar and lidar data in black, the system that saves no radar data in red, and the GPS trajectory in blue

8

Conclusion and Recommendations

In this final chapter I will explain the contributions I have made, and the systems I have developed. After that, I have several recommendations for future work that can be executed using the results and developed systems from this thesis.

8.1. Conclusions

In this thesis I have investigated and answered the following research questions:

1. Can localization be done by relating radar measurements to a lidar point cloud?
   (a) How does the radar point cloud relate to the lidar point cloud, and how can they be made more relatable?
   (b) Can a 3D radar point cloud be created using DBS(Doppler Beam Sharpening), and can it be used in localization?
   (c) Is it possible to use only the radar based point cloud in the long term for localization, if so: how?
2. Can this be implemented using a cost effective 1D antenna array?
3. Can pure-radar (without a priori maps) 3D SLAM(Simultaneous Localization And Mapping) be implemented using a cost effective 1D antenna array?

To this end, the following activities were performed:

- A review of the existing literature was executed, to establish different routes by which these questions may be answered, and to form the ground work from which novel applications could be developed in this thesis.
- Radar data from simulated targets was generated, to establish the functionality of the DBS algorithm, the SLAM algorithm, and target detection system.
- Target detection scripts were constructed, while also researching the possibility of including super resolution algorithms, to improve the precision of the system if it were found to not be precise enough.

- A DBS algorithm was implemented, and tested on the detected simulated targets, to prove the functionality of the DBS, before it was to be given real world data.
- A genetic SLAM algorithm was constructed, Which both allows the creation of maps from scratch, and can also be used to have the system lock onto existing maps provided to the system. This algorithm was then tested and proven to be functional using simulated data.
- The genetic SLAM algorithm was tested on real world data, creating the first 2D maps of the environment, and proving the systems ability to localize the vehicle on the road.
- An ego motion estimation algorithm was developed that allowed for ego motion estimation that is precise enough for the current applications, and was then used to improve the SLAM system.
- Using the ego motion estimates, the 2D maps were turned into 3D maps of the environment, using the DBS algorithm for the necessary height estimation, and the functionality of this height estimation was proven by isolating scatterers with known heights or configurations.
- Next, to establish that the system could lock onto an existing map created using a lidar sensor, a lidar map was created, and filtered to reduce the dimensionality from 3D to 2D, and to reduce the amount of points in it.
- The system was then adapted to accept a priori information about the environment, by providing it with said aforementioned a priori lidar map. It was proven to be able to lock onto it, and localize the vehicle on the road. Thus, another localization system was invented, one that makes use of both lidar and radar, unlike the previous SLAM system, that builds maps using only radar data.
- Finally, a last novel localization system was constructed that not only locks onto an existing lidar map, but also creates and maintains a cumulative map created using the radar data. This system can stop relying on the lidar data when a cumulative radar map has been created that is sufficiently populated after enough passes of the environment.

These points define an outline for the followed work flow during the thesis project. The results of these activities can be summarized as follows:

- A system to create 3D radar maps using an automotive radar with a 1D antenna array has been developed.
- A novel self localization algorithm for car localization, using both lidar and radar data was developed. It makes use of a previously created and processed lidar based map, on which it is able to localize the vehicle using the radar data available at any moment in time.
- A second novel localization algorithm and new approach to the localization problem was proposed and implemented, one that saves the radar data from every frame into a cumulative radar based map. This system can then localize itself by relating the current radar target observations to both the a priori lidar map and the radar map, both localizing the car, and adding to the cumulative map, as it goes along. If the radar map becomes sufficiently populated, the lidar map is no longer necessary, thereby avoiding this data becoming outdated, and creating a continuously self updating radar map.
- The algorithms/systems were all tested on real world data, and the fact that they work has been verified

Over the course of this project, there were various things that I found to be important that must be kept in mind. For one, accurate estimation of the ego motion is very important. Not only does knowledge of the ego motion of the vehicle help improve the performance of localization algorithms, it is also vital for the DBS based height estimation, with which the 3D maps are generated. The more accurate the ego motion estimate is, the more accurate the height estimations become. On a related note, it is also important to have accurate and precise Doppler measurements of the perceived radar targets. This aids both the ego motion estimation accuracy and precision, and the height estimation performance.
Another important finding is that it is in fact possible to relate radar data to a lidar map, but, to accomplish this, the lidar map has to be processed to make it have the same dimensionality as the radar map, and to reduce the amount of points in it to make the computation time between frames manageable. This was initially why I attempted to make the radar maps 3D, instead of making the lidar maps 2D. Though I found out that with the current setup, the latter approach is better for the built applications, though more about this will be discussed in section 8.2.
The height estimation is most accurate for targets that are not in right in front of the radar, but are instead above or below boresight. The further up a target is, the more accurate the height estimation becomes, until the targets are so high up that they are never in the main beam of the radar. To this end, using a radar with a narrow beamwidth in the elevation angle is a hindrance, but not one that can not be overcome.
Similarly, the faster the car drives, the greater the difference in received Doppler from a target right in front of the radar, and one high up, becomes. This causes the height estimation resolution to increase with the velocity of the vehicle. It also means that height estimation with the implemented methods is impossible if the car is standing still.

8.2. Recommendations

In this section I seek to consider what changes/adaptations/implementations might possibly be tried. While this system illustrated the feasibility of 3D SLAM with a 1D array, there are some things that can still be implemented. Such as: feeding the genetic algorithm 3D lidar and radar maps, and possibly allowing it to estimate the pitch and roll and height of the car, giving a more complete picture of the pose of the vehicle. This could also help in the SLAM, in that certain targets can appear and disappear depending on the vehicle pose, such as bridges and traffic signs that are not visible when the vehicle is underneath them, and knowing which points on the map are not visible to the car at its current pose and location can help you select which points on the map the currently observed radar targets should be related to, and which can be ignored.
Another idea might be to refine the Doppler/angular information using MUSIC algorithm or different target detection schemes, to improve the height estimation and ego motion estimation accuracy, or simply running an experiment with different radar settings, such as greater bandwidth or higher pulsenumber, thereby increasing the precision of the range and Doppler information gained from FFT based spectral analysis. These changes are to improve the fidelity of the map to be created, and to allow the system to estimate the pose of the car more accurately.

Future research questions could include:

1. Can RADAR measurements made with different cars or different radars be combined, does 3D mapping aid in this pursuit?

2. Can quasi stationary objects be taken into account, such as parked cars?

3. Can near stationary objects be taken into account, such as parked cars?
4. Can hills, tunnels, and other objects that can only be detected when close to them, or far away from them, be taken into account?
5. What is the best angle for the RADAR to be placed in the car?

In future research, the multipath approach might still be attempted, or a combination of both DBS and the multipath approach. The system should be tested in real time in a real life scenario, to establish if the system can run live. I also recommend testing various forms of SLAM, as I used the genetic algorithm approach merely for its broadness, and adaptability. With the Genetic Algorithm approach, almost no prior information about the problem being solved is necessary, such as a vehicle model for example. It was used to prove that hybrid lidar and radar localization, with only a radar attached to the vehicle, was possible. If the system is to be deployed in the real world, it would be beneficial to test which of the many existing SLAM algorithms provides the most optimal solution for these types of data.

Acknowledgements

For this thesis I received a lot of help from people from different departments, who I would like to thank for their contributions. I would like to thank the technicians and engineers who created and delivered to me the processed and unprocessed radar data, Nikita Petrov, Pascal Aubry, Peter Swart and Fred v.d. Zwan. I must also express my thanks to András Pálffy, who helped me get my hands on the lidar data for the environments I needed. Finally I wish to express my gratitude for the guidance and help of Alexander Yarovoy, the head of the MS3 department, and most of all my coordinator Faruk Uysal, who guided me throughout the entire process of finishing my Masters Thesis.

Bibliography

[1] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 2017, pp. 399-404. doi: 10.1109/ITSC.2017.8317943

[2] S. G. Yi, C. M. Kang, S. H. Lee and C. C. Chung, "Vehicle trajectory prediction for adaptive cruise control," 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, 2015, pp. 59-64. doi: 10.1109/IVS.2015.7225663

[3] V. T. Ferrão, C. D. N. Vinhal and G. da Cruz, "An Occupancy Grid Map Merging Algorithm Invariant to Scale, Rotation and Translation," 2017 Brazilian Conference on Intelligent Systems (BRACIS), Uberlandia, 2017, pp. 246-251. doi: 10.1109/BRACIS.2017.69

[4] G. Bresson, Z. Alsayed, L. Yu and S. Glaser, "Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving," in IEEE Transactions on Intelligent Vehicles, vol. 2, no. 3, pp. 194-220, Sept. 2017. doi: 10.1109/TIV.2017.2749181

[5] J. W. Marck, A. Mohamoud, E. vd Houwen and R. van Heijster, "Indoor radar SLAM A radar application for vision and GPS denied environments," 2013 European Microwave Conference, Nuremberg, 2013, pp. 1783-1786. doi: 10.23919/EuMC.2013.6687024

[6] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," in IEEE Transactions on Robotics and Automation, vol. 17, no. 3, pp. 229-241, Jun 2001. doi: 10.1109/70.938381

[7] P. Lv et al., "Autonomous navigation based on iSAM and GPS filter for AUV," 2017 IEEE Underwater Technology (UT), Busan, 2017, pp. 1-4. doi: 10.1109/UT.2017.7890279

[8] Hyun Chul Roh, Chang Hun Sung, Min Tae Kang and Myung Jin Chung, "Fast SLAM using polar scan matching and particle weight based occupancy grid map for mobile robot," 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Incheon, 2011, pp. 756-757. doi: 10.1109/URAI.2011.6146004

[9] X. Xie, Y. Yu, X. Lin and C. Sun, "An EKF SLAM algorithm for mobile robot with sensor bias estimation," 2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Hefei, 2017, pp. 281-285. doi: 10.1109/YAC.2017.7967420

[10] Markus Hammarsten, 3D Localization and Mapping using automotive radar, (2016).

[11] H. Yin and C. Berger, "When to use what data set for your self-driving car algorithm: An overview of publicly available driving datasets," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, 2017, pp. 1-8. doi: 10.1109/ITSC.2017.8317828

[12] J. Oh, H. Song and H. C. Shin, "Recognition of the upper structure using the RCS characteristic of the automotive radar," 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 2018, pp. 949-952. doi: 10.1109/ICOIN.2018.8343264

[13] T. Deißler, J. Thielecke, R. Salman, T. Schultze and I. Willms, "UWB radar object recognition for SLAM," 11-th INTERNATIONAL RADAR SYMPOSIUM, Vilnius, Lithuania, 2010, pp. 1-4.

[14] L. Sun, T. Vidal-Calleja and J. V. Miro, "Coupling conditionally independent submaps for large-scale 2.5D mapping with Gaussian Markov Random Fields," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 3131-3137. doi: 10.1109/ICRA.2017.7989358

[15] P. Feil, T. Kraus and W. Menzel, "Short Range mm-Wave SAR for Surveillance and Security Applications," 8th European Conference on Synthetic Aperture Radar, Aachen, Germany, 2010, pp. 1-4.

[16] A. Laribi, M. Hahn, J. Dickmann and C. Waldschmidt, "A new height-estimation method using FMCW radar Doppler beam sharpening," 2017 25th European Signal Processing Conference (EUSIPCO), Kos, 2017, pp. 1932-1396. doi: 10.23919/EUSIPCO.2017.8081546

[17] P. Gupta and S. P. Kar, "MUSIC and improved MUSIC algorithm to estimate direction of arrival," 2015 International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, 2015, pp. 0757-0761. doi: 10.1109/ICCSP.2015.7322593

[18] F. Peng et al., "Street view cross-sourced point cloud matching and registration," 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 2026-2030. doi: 10.1109/ICIP.2014.7025406

[19] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 2003, pp. 2743-2748 vol.3. doi: 10.1109/IROS.2003.1249285

[20] C. Ozdemir, Inverse synthetic aperture radar imaging with MATLAB algorithms. Hoboken (NJ): Wiley, 2012.

[21] A. Laribi, M. Hahn, J. Dickmann and C. Waldschmidt, "Vertical Doppler beam sharpening goes self parking," 2018 IEEE Radar Conference (RadarConf18), Oklahoma City, OK, 2018, pp. 0383-0388. doi: 10.1109/RADAR.2018.8378589

[22] A. Paulraj, R. Roy and T. Kailath, "Estimation Of Signal Parameters Via Rotational Invariance Techniques- Esprit," Nineteeth Asilomar Conference on Circuits, Systems and Computers, 1985., Pacific Grove, CA,USA, 1985, pp. 83-89. doi: 10.1109/ACSSC.1985.671426

[23] Y. Zhao, Y. Zhu, Y. Su and M. Yang, "A fast 2D-ESPRIT super-resolution imaging algorithm for linear array SAR," IET International Radar Conference 2015, Hangzhou, 2015, pp. 1-5. doi: 10.1049/cp.2015.1062

[24] H. Zhao, M. Cai and H. Liu, "Two-dimensional DOA estimation with reduced-dimension MUSIC algorithm," 2017 International Applied Computational Electromagnetics Society Symposium (ACES), Suzhou, 2017, pp. 1-2.

[25] Shengqi Zhu, G. Liao, Yi Qu and Zhengguang Zhou, "Space-time-range three dimensional adaptive processing," 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, 2009, pp. 2037-2040. doi: 10.1109/ICASSP.2009.4960014

[26] A. L. Ibáñez, R. Qiu and D. Li, "An implementation of SLAM using ROS and Arduino," 2017 IEEE International Conference on Manipulation, Manufacturing and Measurement on the Nanoscale (3M-NANO), Shanghai, 2017, pp. 1-6. doi: 10.1109/3M-NANO.2017.8286298

[27] H. Wang, H. Liu, H. Ju and X. Li, "Improvement for the Rao-Blackwellized Particle Filters SLAM with MCMC Resampling," 2009 International Conference on Computational Intelligence and Software Engineering, Wuhan, 2009, pp. 1-4. doi: 10.1109/CISE.2009.5366761

[28] Murphy, Kevin P. "Bayesian map learning in dynamic environments." Advances in Neural Information Processing Systems. 2000.

[29] C. A. Kapoutsis, C. P. Vavoulidis and I. Pitas, "Morphological iterative closest point algorithm," in IEEE Transactions on Image Processing, vol. 8, no. 11, pp. 1644-1646, Nov. 1999. doi: 10.1109/83.799892

[30] S. Feng, Y. Li and W. Zhao, "Study on the fast image registration based on genetic algorithm," 2011 International Conference on Uncertainty Reasoning and Knowledge Engineering, Bali, 2011, pp. 213-216. doi: 10.1109/URKE.2011.6007800

[31] A. Laribi, M. Hahn, J. Dickmann and C. Waldschmidt, "Vertical digital beamforming versus multipath height finding," 2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), Nagoya, 2017, pp. 99-102. doi: 10.1109/ICMIM.2017.7918866

[32] P. J. Hargrave, "A tutorial introduction to Kalman filtering," IEE Colloquium on Kalman Filters: Introduction, Applications and Future Developments, London, UK, 1989, pp. 1/1-1/6.

[33] "Radar Data Cube- MATLAB Simulink- MathWorks Benelux", Nl.mathworks.com, 2019. [Online]. Available:

[34] Emagtech.com, 2019. [Online]. Available:

[35] "IEE Colloquium on 'Applications of Genetic Algorithms' (Digest No.1994/067)," IEE Colloquium on Applications of Genetic Algorithms, London, UK, 1994, pp. 0-3.

[36] "IEE Colloquium on 'Genetic Algorithms for Control Systems Engineering' (Digest No. 1993/130)," IEE Colloquium on Genetic Algorithms for Control Systems Engineering, London, UK, 1993, pp. 0-1.

[37] O. A. Oumar, M. F. Siyau and T. P. Sattar, "Comparison between MUSIC and ESPRIT direction of arrival estimation algorithms for wireless communication systems," The First International Conference on Future Generation Communication Technologies, London, 2012, pp. 99-103.

[38] "Cubic smoothing spline - MATLAB csaps- MathWorks Benelux", Nl.mathworks.com, 2019. [Online]. Available: https://nl.mathworks.com/help/curvefit/csaps.html. [Accessed: 19- Dec- 2019].