Pascal Fischer

Automotive Self-Localization using Polarimetric Radar

Master Thesis
to achieve the university degree
Master of Science
Master's degree programme: Computer Science

submitted to

Fakultät für Informatik und Mathematik
Goethe Universität Frankfurt am Main

Supervisors

Prof. Dr. Visvanathan Ramesh
Prof. Dr. Detlef Kroemker
M.Sc. Fabio Weishaupt

AISEL - AI Systems Engineering Lab
Head: Prof. Dr. Visvanathan Ramesh

Frankfurt am Main, 21. November 2021

Affidavit

I hereby declare that I have written the present work independently and without the use of other than the indicated sources and resources. Likewise, I confirm that this work has not been used, even in part, for any other examination or study guidance.

21.11.2021	
Date	Pascal Fischer

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Ebenso bestätige ich, dass diese Arbeit nicht, auch nicht auszugsweise, für eine andere Prüfung oder Studienleistung verwendet wurde.

21.11.2021	
Ort, Datum:	Pascal Fischer

Abstract

The field of autonomous driving has made great progress in recent years, including the development of algorithms for self-localization. However, localization in areas without satellite reception is still an open research topic. With the help of conventional radar sensors and probabilistic model-based methods such as particle or Kalman filters, initial successes have already been achieved. This thesis deals with the development of a probabilistic model-based self-localization algorithm in the form of a particle filter. Here, in contrast to previous approaches, polarimetric radar sensors are used to investigate the advantages of polarimetry as an additional source of information. In the beginning, there is a general introduction to the operation of the used radar sensors and the necessary steps in signal processing. In the further course, the polarimetry and the developed methods for the detection of objects in the environment with the usage of polarimetry are explained in more detail. In addition, an algorithm for estimating the vehicles ego-motion is presented. The ego-motion estimation algorithm forms the first major component of the proposed approach and is used to estimate the vehicle's location based on its motion. To correct the error of the estimated proper motion, a second method is developed based on the detected objects by the polarimetric radars, which match the measured positions with objects in a self-generated map of landmarks. In the end, it is shown that polarimetry adds value and a comparison of the results with other methods shows that the developed approach provides a new state of the art.

Kurzfassung

Der Bereich des autonomen Fahrens hat in den letzten Jahren sehr große Fortschritte gemacht. Darunter auch die Entwicklung von Algorithmen zur selbst-Lokalisierung. Allerdings ist die Lokalisierung in Gebieten ohne Satellitenempfang immer noch ein offenes Forschungsthema. Mit Hilfe von herkömmliche Radarsensoren und probabilistischen modellbasierten Verfahren wie Partikel- bzw. Kalman Filter konnten bereits erste Erfolge verzeichnet werden. In dieser Arbeit geht es um die Entwicklung eines probabilistischen modellbasierten selbst-Lokalisierungsalgorithmus in Form eines Partikelfilters. Hierbei werden im Gegensatz zu vorherigen Ansätzen polarimetrische Radarsensoren verwendet, um den Mehrwert der Hinzunahme der Polarimetrie als zusätzliche Informationsquelle zu untersuchen. Am Anfang gibt es eine allgemeine Einführung in die Funktionsweise der verwendeten Radarsensoren und der nötigen Schritte im Bereich der Signalverarbeitung. Im weiteren Verlauf werden die Polarimetrie und die entwickelten Verfahren zur Erfassung von Objekten in der Umgebung unter Hinzunahme der Polarimetrie genauer erläutert. Zusätzlich wird ein Algorithmus zur Schätzung der Eigenbewegung des zu lokalisierenden Fahrzeuges vorgestellt, welcher die erste große Komponente des vorgeschlagenen Ansatzes bildet. Um den Fehler der geschätzten Eigenbewegung zu korrigieren, wird ein zweites Verfahren basierend auf den erfassten Objekten durch die polarimetrischen Radarsensoren entwickelt, welches die gemessenen Positionen mit Objekten in einer selbst erzeugten Karte von Landmarken in referenz setzt. Am Ende wird gezeigt, dass die Polarimetrie einen Mehrwert bringt und ein Vergleich der Ergebnisse mit anderen Verfahren zeigt, dass der entwickelte Ansatz einen neuen Stand der Technik bereitstellt.

Contents

1 Introduction **1**

2 Technical Background and Related Work **3**
2.1 Radar Sensors . . . 3
2.1.1 FMCW - Frequency modulated continuous wave radars 3
2.1.2 Range and Doppler . . . 5
2.1.3 Azimuth Angle . . . 8
2.1.4 Radar Cross-Section . . . 9
2.1.5 CFAR - Constant False Alarm Rate . . . 10
2.2 Radar Polarimetry . . . 12
2.2.1 Polarization States . . . 13
2.2.2 Polarization Ellipse . . . 14
2.2.3 Jones Vector . . . 15
2.2.4 Polarimetric Scattering Vector . . . 16
2.2.5 Polarimetric Covariance Matrix . . . 19
2.2.6 Polarimetric Coherency Matrix . . . 20
2.2.7 Eigenvalue decomposition . . . 21
2.3 Radar Gridmaps . . . 21
2.3.1 Amplitude Gridmaps . . . 22
2.3.2 Covariance Gridmaps . . . 22
2.3.3 PreCFAR Gridmaps . . . 24
2.4 Radar-based Automotive Localization using Landmarks in a Multimodal Sensor Graph-based Approach . . . 27
2.4.1 Graph-based SLAM . . . 27
2.4.2 Evaluation . . . 29

3 Measurement Setup **31**

4 Ego-Motion Estimation **33**
4.1 System Motion Equation . . . 34
4.2 Velocity Profile Analysis . . . 36
4.3 Handling Unambiguous Doppler Velocity . . . 37
4.4 Validation and Results . . . 39

5 Automotive Self-Localization **43**
5.1 Bayes Filter . . . 46
5.2 Particle Filter . . . 47

5.3 Motion Model 52
5.3.1 Kinematic Configuration 52
5.3.2 Probabilistic Kinematics 53
5.3.3 Velocity Model 54
5.4 Measurement Model 55
5.4.1 Sensor Model 57
5.4.2 Landmark Measurements 58
5.4.3 Sensor Model with Correspondences 60
5.4.4 Correspondence Determination 61
5.4.5 Straight line features 64
5.5 Monte Carlo Localization 67
5.6 Validation and Results 70
5.6.1 Limitations 77

6 Conclusion **80**

Bibliography **88**

1 Introduction

Autonomous driving has already made a lot of advances in the last few decades [1]. Already twenty years ago, prototypes could already demonstrate their computer-controlled motion in complex environments. In the meantime, with increasing sensor complexity, and computing power new approaches using *radio detection and ranging* (radar) *light detection and ranging* (lidar), and camera sensors are developed on prototypes and production vehicles. One important sub-task of automated driving is self-localization, which is the task of localizing the ego-vehicle with the exact position and -heading direction. Through different sensor fusion algorithms using Global Navigation Satellite Systems (GNSS), e.g. Differential GNSS (DGNSS) or sensor fusions of GNSS and an initial measurement unit (IMU), a precisely self-localization could be achieved. DGNSS is a method for increasing the accuracy of satellite-based measurement procedures, whereby correction data is determined via surveyed transmitters on the ground and passed on to the user via transmitting stations [2]. However, navigation in complex environments without GNSS reception is still an open research topic for solving the task of self-localization. This work is part of the research project Automated Cars and Intelligent Traffic in the City (@CITY) [3], which was instituted to analyze and develop ways to navigate in those environments. In order to solve this problem, there are different self-localization approaches e.g. particle filters or Kalman filters coming from probabilistic robotics [4]. This approach combines a motion model that predicts the vehicle's location only on the vehicle's current motion state and a measurement model that corrects the motion-based prediction error by comparing perceived objects in the surrounding with a map of landmarks. Many sensors e.g. lidar, radar, and camera sensors are able to perceive objects in the environment. In comparison to other sensors, radar sensors are operating well, even in harsh environments and are relatively cheap in mass production. As a replacement for expensive IMUs, radars are able to predict the vehicle's motion state on their measurements only. For a good self-localization, a map of only the static environment is needed. With radar sensors, it is also easy to exclude measurements from dynamic objects, and so-called gridmaps, which are a depiction of the environment from the bird's eye perspective, can be created. Therefore, they are chosen to be a good complement next to other sensors for autonomous driving in urban environments in this research group. In contrast to the advantages, there are also disadvantages to radar sensors;

for example, it is hard to extract semantic information for a single target[5]. To handle this, it is possible to accumulate several measurements into the recently mentioned radar gridmaps. Each pixel of such a gridmap is an incorporation of several independent radar measurements. This makes it possible to detect interesting and distinct areas in an environment to perform the self-localization[6].
In previous research[7], there are successfully developed self-localization algorithms with a probabilistic model-based approach using radar ego-motion estimation with an underlying motion model to predict the vehicles pose based on its motion. Further, radar gridmaps are used to detect areas of interest and compare them in a measurement model with a map of landmarks to correct the estimated pose given by the motion model.
In a classical approach, a radar simply transmits an electromagnetic wave with a single polarization and receives the response. But there is also an extension called polarimetric radar coming from polarimetric synthetic aperture radar (PolSAR) appications. The general principle is that the electromagnetic waves are transmitted and received in different polarization states. Therefore, not only the reflected energy of a target can be received, but also several energies from different polarizations. This makes it able to distinguish objects by different scattering behaviours [8].
In this work the advantages of using polarimetric radars in comparison to conventional radars in a probabilistic model-based vehicle self-localization algorithm are investigated. In the first step, a radar ego-motion estimation is developed to estimate the vehicle's motion state corresponding to previous approaches. Further, the concept of conventional radar gridmaps is extended by incorporating different scattering mechanisms using different polarization states of the received and transmitted electromagnetic waves. Finally, the extended gridmaps, called *covariance gridmaps*, are used to build a map of landmarks and to detect areas of interest in the surrounding during the localization process. To extend dynamic objects from the covariance gridmaps a new technique called *PreCFAR gridmaps* is used. Finally, a measurement model is developed, which corrects the estimated pose based on the estimated motion of the vehicle by comparing the measured interesting areas with the corresponding landmarks in a particle filter-based approach. In the last step, the advantages of using different polarization states in the localization algorithm are tested and a comparison with the state of the art of radar-based self-localization is made.

2 Technical Background and Related Work

2.1 Radar Sensors

Radar is the abbreviation of radio detection and ranging and describes an electromagnetic sensor for the detection and localization of reflective objects. The radar's underlying principle has not changed since Heinrich Hertz proved that electromagnetic waves are reflecting on metallic surfaces in 1886 [9]. In the monostatic case the radar sends out an electromagnetic wave by one or more transmitters. The wave will be partly reflected on hitted objects and scattered back to a receiver. If the receiver and transmitter are at the same position, it is determined as the *monostatic case* and otherwise as the *bistatic case*. After amplification of the signal, signal processing is used to decide whether it is a real target and to make estimations about the target with the measured time, frequency, phase, and amplitude of the backscattered wave.
Radar sensors are used in many applications, ranging from modern driver assistance systems such as *adaptive cruise control* (ACC) to pure target detection, aviation and defense technology. Common to all the technologies are the sensing capabilities of a radar sensor. It is possible to determine a targets distance R by measuring the time from sending to receiving of the electromagnetic wave. Further, the targets azimuth angle can be estimated, when multiple receivers are used to determine the incident angle θ of the received signal. Additionally, the targets relative radial velocity v can be acquired with the Doppler effect. Through the introduction of a local tangent plane, targets can be described on a Cartesian coordinate system relative to the car by transforming range and azimuth angle (R, θ) from polar coordinates into Cartesian coordinates [10].

2.1.1 FMCW - Frequency modulated continuous wave radars

Different radar principles exist that vary in the type of extracted target information from a received signal. The main difference between the radar principles is the way how the radar forms the transmitted wave and pro-

cesses the received wave, because the back scattered wave greatly depends on the transmitted signal. In this work, only frequency modulated continuous wave (FMCW) radars will be used and covered.

In comparison to classical pulse Doppler radar systems, FMCW radar systems transmit a continuous wave and the transmitted signal is modulated within the radars frequency band. The block diagram for such a radar is shown in Fig 2.1 First, a generator module produces a wave and gives it

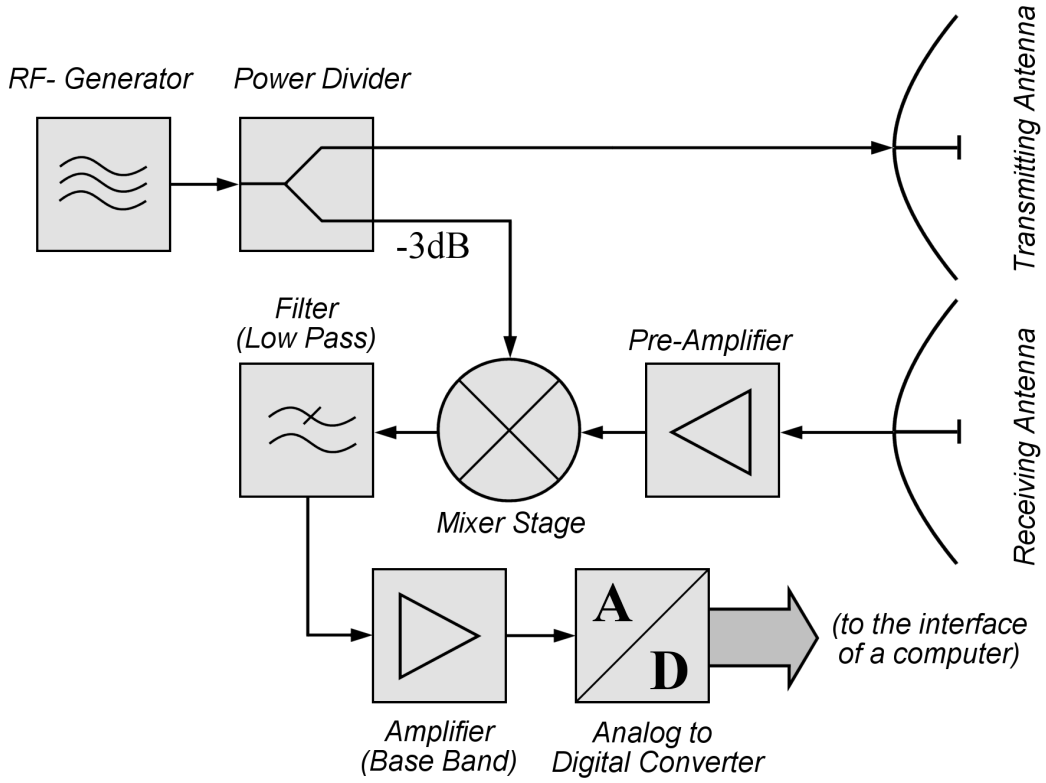

Figure 2.1: FMCW block diagram [11]

to the power divider, which splits the signal into two different outputs. In the first path, the signal is passed on to the transmission antenna, which sends out the wave. The receiving antenna receives the back scattered signal and gives it to the pre-amplifier to amplify it. The second signal path of the power divider and the received amplified signal are mixed in the mixer stage to generate a difference frequency between transmission and receiving path. The differential signal is then filtered, also amplified, and sampled in the analog to digital converter, so that it can be further processed.

2.1.2 Range and Doppler

Figure 2.2 shows an example of such a differential signal with the so-called sawtooth modulation. If the radial velocity between the radar sensor and the

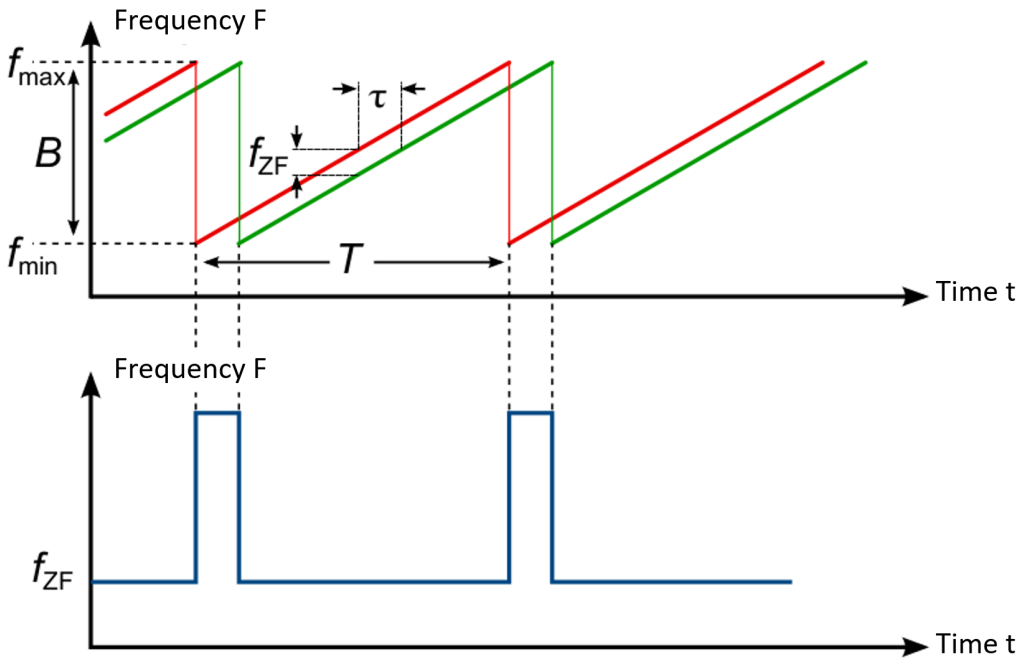

Figure 2.2: FMCW radar signal with sawtooth modulation figure 1 in [12]

reflecting object is zero, then the range R between them can be determined through the time difference by [13]

$$R = \frac{c_0 f_{ZF}}{2(\frac{df}{dt})} = \frac{c_0 \tau}{2}, \tag{2.1}$$

where

c_0 the speed of light = $3 \cdot 10^8 \frac{m}{s}$,
τ the delay time $[s]$
f_{ZF} the measured frequency difference $[HZ]$
df/dt frequency shift per unit of time (slope).

Suppose the reflecting object has a radial velocity with respect to the receiving antenna. In that case the echo signal gets shifted by the Doppler frequency and the radar measures an additional Doppler frequency f_D. To handle this, another modulation technique called *triangle modulation* is used, which is depicted in Fig. 2.3. With a triangle modulation the distance measuring can be performed on the combination of the so-called *up* and

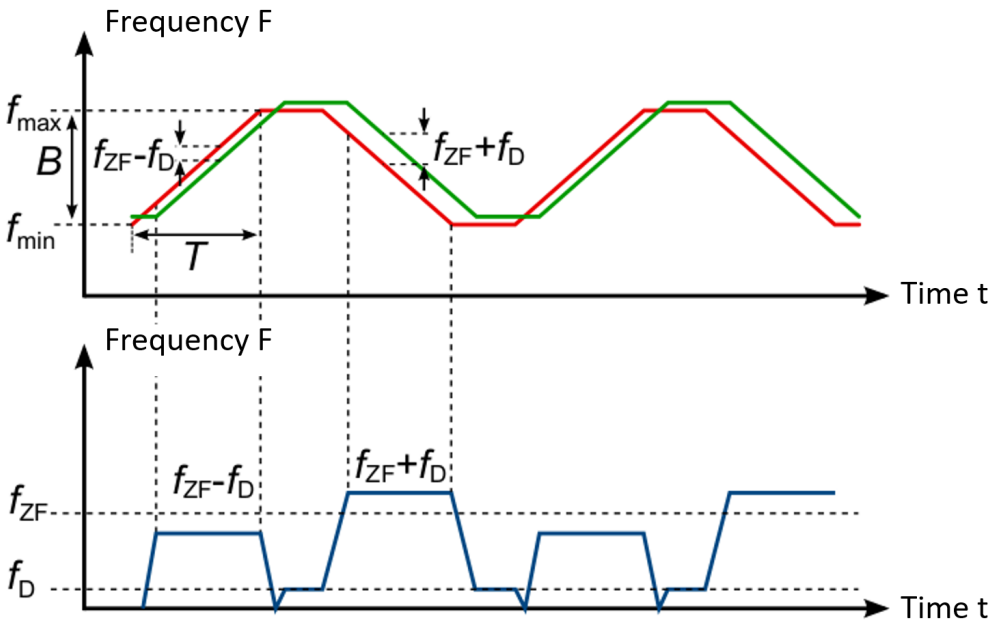

Figure 2.3: FMCW radar signal with triangle modulation. figure 2 in [12]

down chirps. Without a Doppler frequency, the amount of the frequency difference in the up chirp is equal to the measurement in the down chirp. A Doppler frequency shifts the echo signal in height. If the reflecting object is moving towards the radar, the signal will be moved up and if the reflecting object is moving away from the radar the signal will be moved down. This results in the difference of the frequency difference Δf and the Doppler frequency f_D in the up chirp, and the sum of these two frequencies at the down chirp. Using the information of both chirps, it is possible to determine the velocity and distance as the intersection of two linear equations. So the targets Doppler velocity v_D and distance R can be determined by [14]

$$f(R) = \frac{\Delta(f_1) + \Delta(f_2)}{2} \qquad (2.2) \qquad\qquad f(v_D) = \frac{|\Delta(f_1) - \Delta(f_2)|}{2}, \qquad (2.3)$$

where

$f(R)$ is the frequency as a measure of distance determination,
$f_{(v_D)}$ is the Doppler frequency as a measure of the speed determination,
$\Delta(f_1)$ is the frequency difference in the up chirp,
$\Delta(f_2)$ is the frequency difference in the down chirp.

To calculate the exact distance, the frequency $f(R)$ can be used in (2.1).

The FMCW radar that is used in this work applies a different modulation named *chirp sequence modulation* (CSM). The full process is depicted in Fig. 2.4 The CSM consists of a chirps sequence with equal linear chirps and

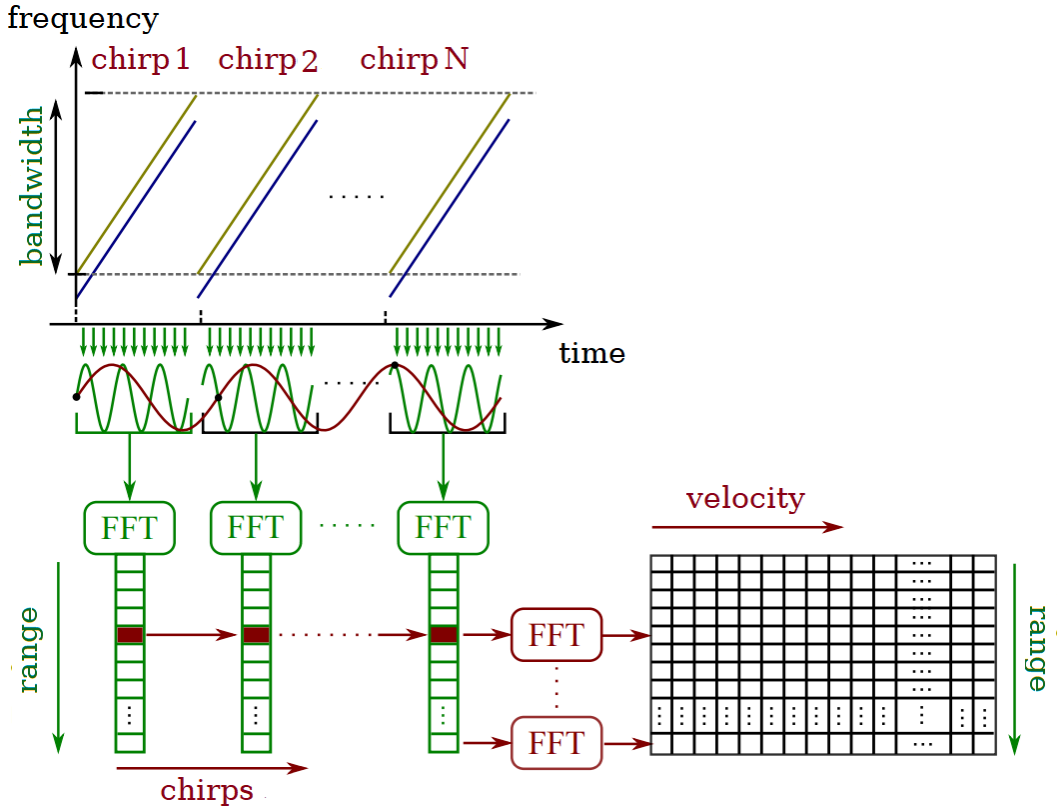

Figure 2.4: Determination of range (green) and Doppler velocity (red) from measured time signal. Fig 3.4 in [15]

the same time difference between all chirps. The duration of a chirp is so short, that the Doppler shift does not affect the received signal. Through a *fast Fourier transform* (FFT), the target's distance can be directly determined with a single chirp, and the Doppler velocity can be established with the use of a second FFT over all discrete values in every distance cell, over all chirps [16]. The result is a discrete 2D measuring space (r, v_D) for each antenna element with range r and Doppler velocity v_D. For each cell a complex amplitude value is given. The resolution in the different dimensions is important for the discretization of the measuring space and can be determined as follows, as given in [17], [18]. As the bandwidth increases, the resolving capability and accuracy of the distance measurement increase. The length of a distance cell ΔR is thereby inversely proportional to the bandwidth B:

$$\Delta R = \frac{c_0}{2B} \tag{2.4}$$

the maximum measurable target distance R_{max} is, besides the range limitation given by the radar equation, the intermediate frequency filter and is determined by the bandwidth of this filter $f_{ZF,max}$ as

$$R_{max} = \frac{f_{ZF,max} c_0 \Delta T}{2B}, \tag{2.5}$$

with

ΔT the chirp duration.

The maximum and minimum measurable velocities are described by

$$v^D_{max} = \frac{c_0}{4 f_0 \Delta T} \quad (2.6) \qquad\qquad v^D_{min} = -\frac{c_0}{4 f_0 \Delta T} \quad (2.7)$$

where

- f_0 is the carrier frequency.

The speed resolution Δv^D can be expressed as

$$\Delta v^D = \frac{c_0}{2 f_0 T_R N_R}, \tag{2.8}$$

with

- N_R is the total number of chirps.

So the speed resolution depends on the total number of chirps and the whole measuring range $[-v^D_{max}, v^D_{max}]$ will be divided by the total number into equidistant bins.

2.1.3 Azimuth Angle

In addition to the range and radial velocity, the azimuth angle can be determined. The common methods can be divided into three groups: *mechanical beamsteering* (MBS), *electronic beamforming* (EBF) and *digital beamforming* (DBF). In this work only DBF is used and the other two types are not discussed further. DBF systems consist of a spatial sequence of N_A transmitting and receiving antennas, whose signals are computationally combined to form desired radiation lobes. The targets azimuth angle will be estimated by *Direction of Arrival* (DOA) algorithms. For this, the phase difference from the impinging wave will be analyzed, depicted in Fig. 2.5. The maximal distance d_A between the antenna elements for the maximal detectable angle θ_{max} is determined from the Nyquist-Shannon sampling theorem by

$$d_A = \frac{\lambda}{2 \sin(\theta_{max})}, \qquad \text{with wavelength } \lambda. \tag{2.9}$$

With increasing positive or negative azimuth angle θ, the wave front hits the right or left antenna element first (figure 2.5 b)) and results in a sinusoidal curve of the phase over all antenna elements at one measuring point in time. (Fig. 2.5 c). The frequency of the course increases for larger target angles. The complex amplitude contains the information on which side the target is located. By means of fast fourier transform, the frequency and thus the target azimuth angle can be determined in this way. The related equations are summarized in [19].

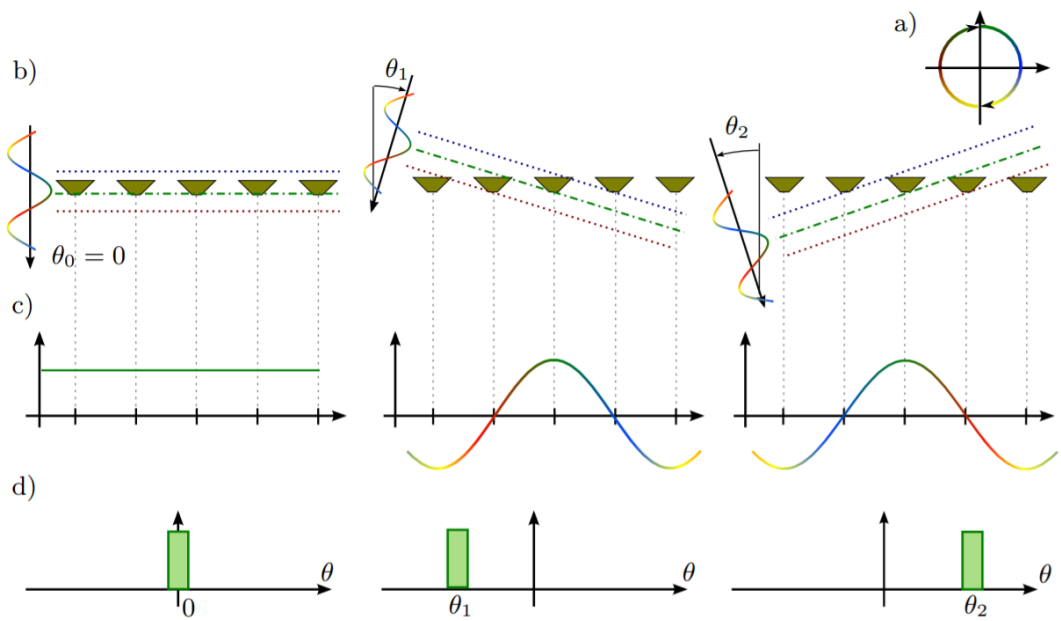

Figure 2.5: Example of an electromagnetic wave for three different target angles. Color-coded as complex phasor (a); wave representation in the plane with 5 sensor elements (b), measured amplitudes (c) and the result of FFT (d), Fig 3.2 in [15]

2.1.4 Radar Cross-Section

As a measure of the backscattered energy depending on the incident energy, it is possible to determine the radar cross-section (RCS) σ of a target, which is defined as in [13]

$$\sigma = 4\pi R^2 \frac{|P_S|}{|P_0|}, \tag{2.10}$$

where

R is the targets distance,
P_0 is the incident energy at the target,
P_S is the scattered energy at distance R, respectively.

More accurately, the RCS indicates the size of an isotropic reflecting surface that provides a radar echo equal to that of the object. The important value here is the on several factors depending backscattered power. On the one hand, the backscattered power P_S is partly defined by an object's contour, geometry and material. On the other hand, it is a function of wavelength λ, the polarization of the sent electro-magnetic wave and its incident angle. A few examples from the RCS of different geometric objects are presented in Table 2.1. It can be seen that there are objects, which are easier to detect

Geometry	**RCS**	**Symbols**
Sphere	πr^2	r : radius
Flat plane	$\frac{4\pi A^2}{\lambda^2}$	A: surface area
Dihedral	$\frac{8\pi a^2 b^2}{\lambda^2}$	a, b: side lengths
Trihedral	$\frac{4\pi a^4}{3\lambda^2}$	a: edge length
Cylinder/Dipole	$\frac{2\pi r}{\lambda}$	r: radius

Table 2.1: RCS for different objects [20]

than other objects because of their much higher RCS. Rearranging (2.10) for the scattered power leads to

$$|P_S| = \frac{|P_0| \cdot \sigma}{4\pi R^2}. \tag{2.11}$$

The formula shows that with a higher distance, the target's backscattered power decreases. This effect can be mitigated if either the target's RCS or the sent energy is higher. For target detection, it is easier to detect targets with a high RCS, but in defense applications, it is often necessary to lower a target's RCS to not be detected by radar systems.

2.1.5 CFAR - Constant False Alarm Rate

Not all measured reflections come from real targets but there is also noise. The *false alarm rate* (FAR) is the average number of 'false targets' that can be detected from the receiver in a given time. The main goal is to have a low as possible FAR while detecting nearly all real target's. So the targets need to be distinguished from the noise floor. Fig. 2.6 shows the usage of different thresholds to distinguish the targets from the noise floor. A higher threshold leads to a lower FAR but missing targets, and a lower threshold

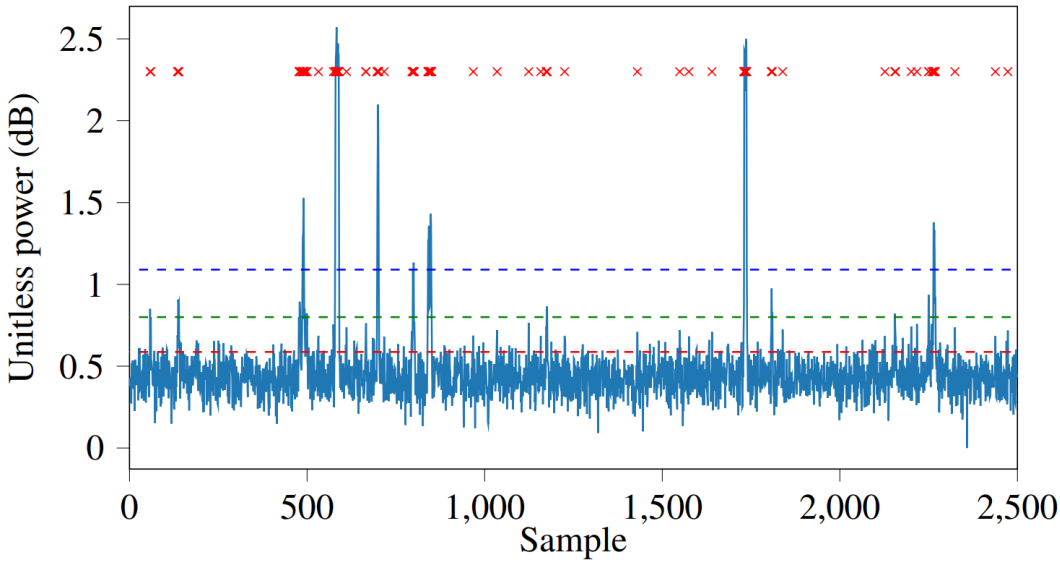

Figure 2.6: Real detections and false alarm are shown for different fixed thresholds. Higher threshold yields to a lower false alarm, while missing targets. Fig. 3.6 in [21]

leads to a higher FAR but detecting nearly all targets. Another technique is needed to get a better separation.
In comparison with the neighbourhood of a real target cell, the target cell should have a much higher amplitude. So instead of using only the single same threshold, it is useful to compare a target cell with its neighbourhood. The so-called CFAR detector [22] follows this approach. Figure 2.7 shows

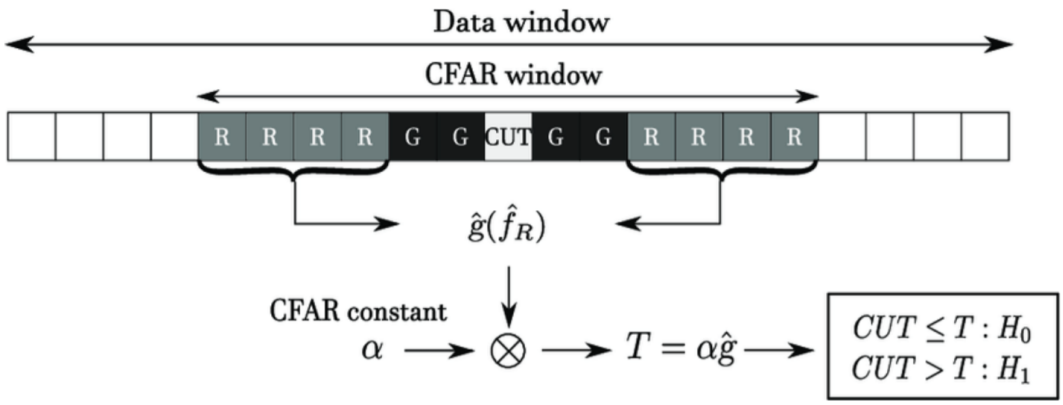

Figure 2.7: CFAR detection algorithm as a block diagram in the one-dimensional case. [23]

the block diagram of the CFAR detection algorithm for the one-dimensional case. For the application in this work, the CFAR detection algorithm is used in the two-dimensional case where the two dimensions are the horizontal x and y distance of a measured signal, but the one-dimensional case is easier to visualize. The algorithm considers for the *cell under test* (CUT) all neighbours inside a CFAR window. The cells directly next to the CUT are

determined as guard cells G and ignored, because a real target will usually appear in multiple cells. The cells around the guard cells are defined as reference cells R and will be used to estimate the noise floor $\hat{g}(\hat{f}_R)$. An additional CFAR constant α is multiplied with the noise floor to get the threshold T for the CUT. Under the H_0 hypothesis, the CUT is classified as part of the noise floor if the CUT is smaller than the noise floor estimate T. If it is higher the alternative hypothesis H_1 is true and the CUT is classified as a target. As the acceptable FAR is usually known, the threshold factor α can be calculated by [24]

$$\alpha = N({P_{FA}}^{-1/N} - 1), \tag{2.12}$$

where

N is the number of reference cells,
P_{FA} is the probability of a false alarm.

The remaining question is how the algorithm calculates the noise floor estimation $\hat{g}$. The answer is, there is more than one way to do this. The literature shows numerous ways to achieve a successful estimation for different applications. For this work only the cell averaging CFAR (CA-CFAR) is used. Using the CA-CFAR algorithm the noise floor $\hat{g}$, for the one-dimensional case is calculated using

$$\hat{g} = \frac{1}{N}\sum_{i=1}^{N} R_i \tag{2.13}$$

This noise estimation method is performant, but the CA-CFAR detector performs bad on clutter edges and targets which are close to each other.

2.2 Radar Polarimetry

The oscillation of transversal electromagnetic waves is determined to be perpendicular to the propagation direction of the wave, as depicted in Fig. 2.8 The displayed wave is called a linear polarized electromagnetic wave. In contrast to polarized waves, the electrical and magnetic waves of unpolarized electromagnetic waves do not stay in a certain direction. They oscillate in arbitrary directions perpendicular to the waves propagation direction. In Fig. 2.8 it can be seen that the electrical and magnetic waves are also perpendicular to each other. Polarized waves especially polarized light, are used in many applications e.g. sunglasses, which reduce the glare of bright surfaces or scientific usages in *polarimetric synthetic aperture radars* (PolSAR) applications, where they are used to determine different terrains by their specific polarization scattering. In the following a brief description

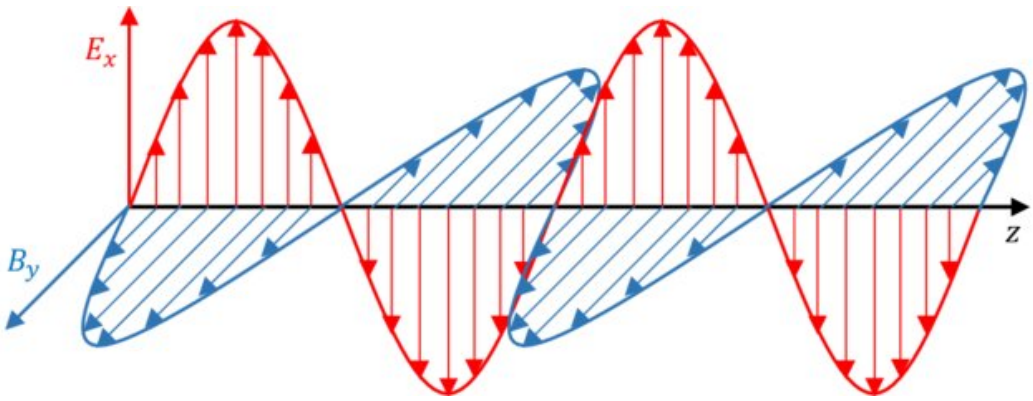

Figure 2.8: Electromagnetic wave with its electrical field E_x in blue, the magnetic field B_y in blue, and the direction of propagation in z. Fig. 1 in [25]

of different polarization states and their mathematical expressions will be discussed. That is followed by the so-called scattering vector and matrix which is used in the radar application of this work [26].

2.2.1 Polarization States

Assuming a lossless medium, electromagnetic sinusoidal waves, such as the one displayed in Fig. 2.8 can be described by [27]

$$\underline{\vec{E}}(z,t) = \begin{bmatrix} E_{0x}\cos\left(\omega t - kz + \theta_x\right) \\ E_{0y}\cos\left(\omega t - kz + \theta_y\right) \\ 0 \end{bmatrix}. \tag{2.14}$$

the electric field is composed of two orthogonal sinusoidal waves with, in general, different amplitudes and phases at the origin.There are different kinds of polarization states. Fig. shows a few examples. In general three types of polarization's can be specified. The first one is called linear polarization, where the phase difference of both components is deemed to be

$$\theta = \theta_x - \theta_y = 0, \tag{2.15}$$

meaning the two components are synchronized by phase. Fig. 2.9 shows a linear polarized vertical wave meaning the E_{0x} component of the wave is 0. In Fig. 2.9, a 45° tilted linear polarized wave is depicted. Here, both components of $\underline{\mathrm{E}}$ have an equal amplitude $E_{0x} = E_{0y}$ with a phase difference θ of 0 The second type is called circular polarization. In this case, the wave rotates circularly around the z axis as shown in Fig. 2.9. Both components also have an equal amplitude and the phase difference of both components is deemed to be

$$\theta = \theta_x - \theta_y = \frac{\pi}{2} + k\pi, \qquad \text{with } k \in \mathbb{N} \tag{2.16}$$

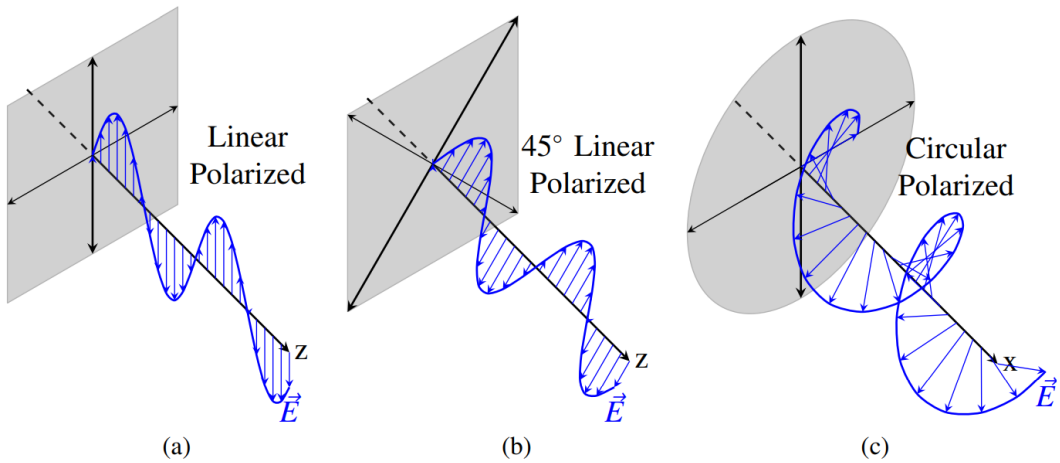

Figure 2.9: Polarized waves: (a) linear polarized wave (b) To (a) corresponding linear polarized wave rotated by 45° (c) right circular polarized wave. Fig. 3.9 in [21]

Taking the amplitudes of Fig. 2.9b and shifting them with $\theta = \pm\frac{\pi}{2}$ results in a circular polarized wave like the one in Fig. 2.9c . In all other cases, which don't follow the declarations of linear or circular polarization, it is the third case called elliptic polarization. In this case the wave describes a helical trajectory around the z axis[28].

2.2.2 Polarization Ellipse

As an attempt to describe the polarization, the so-called polarization ellipse was invented, which is shown in Fig. 2.10. It is a characterization of the wave

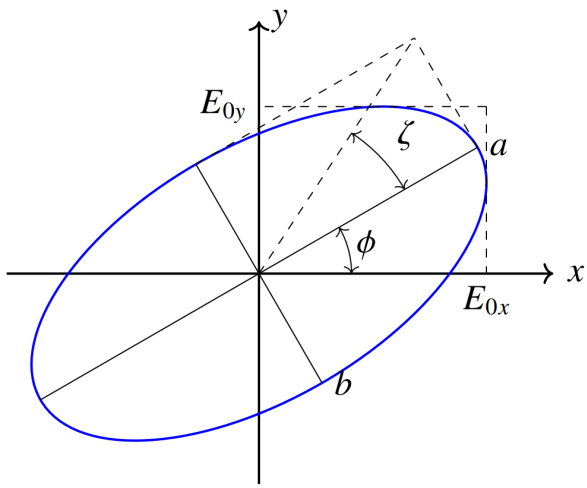

Figure 2.10: Polarization ellipse, with orientation angle ϕ and the aperture ζ. Fig. 3.10 in [21]

in the time domain, at a fixed position $z = z_0$. Because, from a practical

point of view, 3-D helical curves are difficult to represent and to analyze. According to the ellipse, a polarization state can be described by three different features [29]:

- The amplitude A of the wave, which can be calculated by

$$A = \sqrt{E_{0x}^2 + E_{0y}^2}, \tag{2.17}$$

- $\phi \in [-\frac{pi}{2}, \frac{pi}{2}]$ - the so-called ellipse orientation, which is derived using

$$\tan(2\phi) = 2\frac{E_{0x} + E_{0y}}{E_{0x}^2 - E_{0y}^2} \cos\left(\theta_x - \theta_y\right), \tag{2.18}$$

- and the ellipse aperture $\zeta \in [0, \frac{pi}{4}]$ described by

$$\sin(2\zeta) = 2\frac{E_{0x} + E_{0y}}{E_{0x}^2 + E_{0y}^2} \cos\left(\theta_x - \theta_y\right) \tag{2.19}$$

2.2.3 Jones Vector

The Jones vector is another representation of a plane monochromatic electric field. This form aims to describe the wave polarization using the minimum amount of information. The time–space vector, given in Equation (2.14) can be written as [30]

$$\begin{aligned} \underline{\vec{E}}(z,t) &= \begin{bmatrix} E_{0x}\cos\left(\omega t - kz + \theta_x\right) \\ E_{0y}\cos\left(\omega t - kz + \theta_y\right) \end{bmatrix} = \mathrm{Re}\left\{ \begin{bmatrix} E_{0x}e^{j\theta_x} \\ E_{0y}e^{j\theta_y} \end{bmatrix} \mathrm{e}^{-jkz}\mathrm{e}^{j\omega t} \right\} \\ &= \mathrm{Re}\left\{ \underline{E}(z)\mathrm{e}^{j\omega t} \right\} \end{aligned} \tag{2.20}$$

A Jones vector $\underline{\boldsymbol{E}}$ is then defined from the complex electric field vector $\underline{\vec{E}}(z)$ as

$$\underline{\boldsymbol{E}} = \underline{\vec{E}}(z)\Big|_{z=0} = \underline{\vec{E}}(0) = \begin{bmatrix} E_{0\mathrm{x}}e^{j\theta_x} \\ E_{0\mathrm{y}}e^{j\theta_y} \end{bmatrix}$$

Both definitions of a polarization state from the polarization ellipse as well as from a Jones vector are equivalent. As a result, the Jones vector can be formulated as a 2-D complex vector function of the polarization ellipse defined as

$$\underline{\boldsymbol{E}} = A\mathrm{e}^{+j\alpha} \begin{bmatrix} \cos\phi\cos\zeta - j\sin\phi\sin\zeta \\ \sin\phi\cos\zeta + j\cos\phi\sin\zeta \end{bmatrix}$$

where α is an absolute phase term. The Jones vector may be written in a more effective matrix form:

$$\underline{\boldsymbol{E}} = A\mathrm{e}^{+j\alpha} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\zeta \\ j\sin\zeta \end{bmatrix}$$

The Jones vectors and the associated polarization ellipse parameters for some canonical polarization states are presented in the following table:

Polarization State	**Unit Jones Vector $\hat{\mathbf{u}}_{(x,y)}$**	**Orientation Angle ϕ**	**Ellipticity Angle ζ**
Horizontal (H)	$\hat{\boldsymbol{u}}_{\mathrm{H}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$	0	0
Vertical (V)	$\hat{\boldsymbol{u}}_{\mathrm{V}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\frac{\pi}{2}$	0
Linear $+45°$	$\hat{\boldsymbol{u}}_{+45} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\frac{\pi}{4}$	0
Linear $-45°$	$\hat{\boldsymbol{u}}_{-45} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$-\frac{\pi}{4}$	0
Left circular	$\hat{\boldsymbol{u}}_{\mathrm{L}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ j \end{bmatrix}$	$\left[-\frac{\pi}{2} \cdots \frac{\pi}{2}\right]$	$\frac{\pi}{4}$
Right circular	$\hat{\boldsymbol{u}}_{\mathrm{R}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -j \end{bmatrix}$	$\left[-\frac{\pi}{2} \cdots \frac{\pi}{2}\right]$	$-\frac{\pi}{4}$

Table 2.2: Parameters for different polarization states. Page 38 in [30]

2.2.4 Polarimetric Scattering Vector

Understanding polarimetric waves, in general, is not enough. Similar to the RCS, the backscattering behaviour of such a wave by a target needs to be described. A polarimetric radar is able to capture the scattering behaviour of a target. In PolSAR applications the polarimetric scattering behaviour has been used for decades [31]. It is important to differentiate between the monostatic and the bistatic case. If the transmitter and receiver are located at the same point, it is called monostatic, and if the transmitter and receiver are placed at different locations, it is called bistatic. The impeding $\underline{\boldsymbol{E}}^I$ and scattered waves $\underline{\boldsymbol{E}}^S$ can be described as [32]

$$\underline{\boldsymbol{E}}^I\left(\hat{u}_\perp, \hat{u}_\parallel\right) = \begin{bmatrix} \boldsymbol{E}^I_\perp \\ \boldsymbol{E}^I_\parallel \end{bmatrix} \qquad (2.21) \qquad \underline{\boldsymbol{E}}^S\left(\hat{u}_\perp, \hat{u}_\parallel\right) = \begin{bmatrix} \boldsymbol{E}^S_\perp \\ \boldsymbol{E}^S_\parallel \end{bmatrix}. \qquad (2.22)$$

Here $\hat{u}_\perp$ and $\hat{u}_\parallel$are the unit vectors of an orthogonal basis, for a specific reference frame. The most common polarization bases are vertical and circular polarized waves, called E_{HV} and E_{LR} respectively. To enable the receiver to receive the complete power of the scattered signal, both orthogonal polarizations are transmitted alternately. For convenience, the two orthogonal

unit vectors will be called $\hat{x}$ and $\hat{y}$ in the following. The relation between the impeding wave $\underline{\boldsymbol{E}}^{I}_{(x,\hat{y})}$ and the received wave $\underline{\boldsymbol{E}}^{S}_{(\hat{x},\hat{y})}$ is described by

$$\underline{\boldsymbol{E}}^{S}_{(\hat{x},\hat{y})} = S_{(\hat{x},\hat{y})}\underline{\boldsymbol{E}}^{I}_{(\hat{x},\hat{y})} = \begin{bmatrix} S_{XX} & S_{XY} \\ S_{YX} & S_{YY} \end{bmatrix} \underline{\boldsymbol{E}}^{I}_{(\hat{x},\hat{y})}, \tag{2.23}$$

with

- $S_{(\hat{x},\hat{y})}$ is the 2 x 2 Sinclair scattering matrix [32].

In the bistatic case, the total scattered power (Span), for a scattering matrix S is then calculated as

$$\text{Span } = \text{Tr}\left(SS^{H}\right) = |S_{XX}|^2 + |S_{XY}|^2 + |S_{YX}|^2 + |S_{YY}|^2, \tag{2.24}$$

where

- $Tr(S)$ is the trace of matrix S,
- S^H is the hermitian operator, which represents the transpose and complex conjugate of S.

In the monostatic case it applies, that $S_{XY} = S_{YX}$. So it follows [33]

$$\text{Span } = \text{Tr}\left(SS^{H}\right) = |S_{XX}|^2 + 2\,|S_{XY}|^2 + |S_{YY}|^2, \tag{2.25}$$

Because of a more efficient calculation ability as in the matrix representation is needed, a vector form, called feature vector, is used for further calculations.

$$\underline{k} = \frac{1}{2}\,\text{Tr}(S\Psi), \tag{2.26}$$

where

- Ψ is a special basis set.

In literature, there are two existing special sets, the so-called Pauli and Lexicographic basis sets. The Pauli basis set is given by [34]

$$\{\Psi_{P,4}\} = \left\{\sqrt{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sqrt{2}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \sqrt{2}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \sqrt{2}\begin{bmatrix} 0 & -j \\ j & 0 \end{bmatrix}\right\}. \tag{2.27}$$

Combining (2.27) and (2.26) results in the so-called 4D-Pauli feature vector $\underline{k}$

$$\underline{k} = \frac{1}{\sqrt{2}}\left[\; S_{XX} + S_{YY} \quad S_{XX} - S_{YY} \quad S_{XY} + S_{YX} \quad j\,(S_{XY} - S_{YX}) \;\right]^{T}. \tag{2.28}$$

The second special basis set is the lexicographic basis set

$$\{\Psi_{L,4}\} = \left\{2\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} 2\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} 2\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} 2\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\right\}. \tag{2.29}$$

Combining (2.26) and (2.29) results in the so-called 4D-lexicographic-feature vector $\underline{\Omega}$

$$\underline{\Omega} = \mathrm{V}\left(S_{XY}\right) = \left[\begin{array}{cccc} S_{XX} & S_{XY} & S_{YX} & S_{YY} \end{array}\right]^T, \tag{2.30}$$

where

- V(S) vectorizes a matrix S to a column vector.

For the monostatic case, there is a different matrix set, with only three matrices. $\Psi_{P,3}$ in the monostatic case is given by

$$\{\Psi_{P,3}\} = \left\{\sqrt{2}\left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right] 2\left[\begin{array}{cc} 1 & 0 \\ 0 & -1 \end{array}\right] \sqrt{2}\left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array}\right]\right\}, \tag{2.31}$$

and for the Pauli feature vector follows

$$\underline{k} = \frac{1}{\sqrt{2}}\left[\begin{array}{ccc} S_{XX}+S_{YY} & S_{XX}-S_{YY} & 2S_{XY} \end{array}\right]^T. \tag{2.32}$$

The corresponding Lexicographic basis set also has only three matrices and is defined by

$$\{\Psi_{L,3}\} = \left\{2\left[\begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array}\right] 2\sqrt{2}\left[\begin{array}{cc} 0 & 1 \\ 0 & 0 \end{array}\right] 2\left[\begin{array}{cc} 0 & 0 \\ 0 & 1 \end{array}\right]\right\}. \tag{2.33}$$

Hence, the 3D lexicographic feature vector is described by

$$\underline{\Omega} = \left[\begin{array}{ccc} S_{XX} & \sqrt{2}S_{XY} & S_{YY} \end{array}\right]^T. \tag{2.34}$$

To preserve the received power, the cross-polarization element must be multiplied with $\sqrt{2}$, as (2.25) states [35].
The power of a backscatterer is calculated by the sum of its absolute squared elements, as shown in (2.24). The transformation with a basis set, may not alter the total power and hence
is true.

$$\mathrm{Span}(S) = \underline{k}^H \cdot \underline{k} = \underline{\Omega}^H \cdot \underline{\Omega} \tag{2.35}$$

Because both feature vectors Ψ_P and Ψ_L, have been derived from the same Sinclair scattering matrix, there must be a mathematical link between both representations. To not distort the total power, Considering (3.32), the transformation has to be done by a unitary transformation matrix U because the determinant of a unitary matrix has to be 1, to not distort the total power. In the bistatic case,

$$\underline{\Omega} = U_{4(P\to L)}\underline{k} \quad \text{with} \quad U_{4(P\to L)} = \frac{1}{\sqrt{2}}\left[\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -j \\ 0 & 0 & 1 & j \\ 1 & -1 & 0 & 0 \end{array}\right] \tag{2.36}$$

describes the transformation between the Pauli and lexicographic base, and the transformation from lexicographic to Pauli is described by [36]

$$\underline{k} = U_{4(L\to P)}\underline{\Omega} \quad \text{with} \quad U_{4(L\to P)} = U^{-1}_{4(P\to L)}. \tag{2.37}$$

For the monostatic case again, it applies

$$\underline{\Omega} = U_{3(P\to L)}\underline{k} \quad \text{with} \quad U_{3(P\to L)} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 1 & -1 & 0 \end{bmatrix}, \tag{2.38}$$

with another transformation matrix U_3. For the monostatic backward transformation from lexicographic to Pauli base follows then [37]

$$\underline{k} = U_{3(L\to P)}\underline{\Omega} \quad \text{with} \quad U_{3(L\to P)} = U^{-1}_{3(P\to L)}. \tag{2.39}$$

2.2.5 Polarimetric Covariance Matrix

Many objects have an anisotropic scattering behaviour, which means that different rotation and/or viewing angles lead to a different scattering vector Ω, but the scattering vector Ω is only useful to describe the scattering behaviour of a stationary target from a single point of view. Targets from the radar measurements of this thesis are collected by driving down a road; therefore, targets are necessarily seen from different angles. Using the second-order moments as a covariance matrix is a solution used in PolSAR applications, which allows to describing the scattering process in a stochastic model. This can be seen as an incorporation of several scattering mechanisms into a single mathematical expression. The covariance matrix is then described by

$$C = \frac{1}{N}\sum_{i=0}^{N} \underline{\Omega}_i \underline{\Omega}_i^H, \tag{2.40}$$

where

- N describes the total number of scattering vectors for a single object,
- H symbolizes the Hermitian transpose again.

The *probability density function* (PDF) of the scattering vector $\underline{\Omega}$ can be described through a multivariate Gaussian distribution with zero mean, assuming that a covariance is calculated by several random scatterers, as follow

$$p_{\underline{\Omega}}(\underline{\Omega}) = \frac{1}{\pi^q |\Sigma|} e^{-\underline{\Omega}^H \Sigma^{-1} \underline{\Omega}}, \tag{2.41}$$

where

- $|X|$ is the determinant for a matrix X,
- $\mathbb{E}\left(\underline{\Omega}\Omega^H\right)$ is the expected covariance matrix,
- $q = 4$ is the dimension of $\underline{\Omega}$.

As a consequence of the PDF of the scattering vector, the PDF of the covariance matrix can be described by the general complex Wishart distribution as

$$p_C(C) = \frac{n^{nq}|C|^{n-q}e^{-n\,\mathrm{Tr}\left(\Sigma^{-1}C\right)}}{\pi^{\frac{q(q-1)}{2}}|\Sigma|^n \prod_{i=1}^{q}\Gamma(n-i+1)}, \tag{2.42}$$

where

- n describes the degree of freedom,
- Γ represents the gamma function.

At this point, it should be mentioned that this distribution yields to the possibility of comparing two covariance matrices[38].

2.2.6 Polarimetric Coherency Matrix

Corresponding to the covariance matrix C, the so-called polarimetric coherency matrix T exists, which is derived from the complex Pauli scattering vector $\underline{k}$. Because of its derivation, the matrix is often also called the Pauli matrix. Both matrices describe the relationship between the different polarization channels. The calculation of the coherency matrix T is similar to the calculation of the covariance matrix in (2.40)

$$T = \frac{1}{N}\sum_{i=0}^{N}\underline{k}_i\underline{k}_i^H. \tag{2.43}$$

At this point, it is important to mention again, that the total power has to remain constant, so it applies

$$\mathrm{Tr}(C) = \mathrm{Tr}(T) = \mathrm{Span} \tag{2.44}$$

As mentioned, there exists a link between the feature vectors $\underline{\Omega}$ and $\underline{k}$, so there also exists a link between the covariance and the coherency matrix [39], which can be derived as

$$\begin{aligned} C &= \left\langle \underline{\Omega}\cdot\underline{\Omega}^H \right\rangle \\ &= \left\langle \left(U_{4(P\to L)}\underline{k}\right)\cdot\left(U_{4(P\to L)}\underline{k}\right)^H \right\rangle \\ &= U_{4(P\to L)}\left\langle \underline{k}\cdot\underline{k}^H \right\rangle U_{4(P\to L)}^{-1} \\ &= U_{4(P\to L)}TU_{4(P\to L)}^{-1} \end{aligned} \tag{2.45}$$

where

- $U_{4(P \to L)}$ is the special unitary matrix, given in (2.36),
- $\langle \cdot \rangle$ describes the ensemble average.

2.2.7 Eigenvalue decomposition

The covariance matrix C is positive semidefinite; this follows from the composition of the outer hermitian product of the same vector $\underline{\Omega}$. Hence, C can be decomposed to its eigenvalues and eigenvectors. The covariance matrix is also hermitian, which leads to the fact, that for every eigenvalue it applies $\lambda_i = \lambda_i^*$. Hence, the eigenvalues are real-valued [52]. Therefore, the following solution for the common eigenvalue problem[53] can be derived

$$\begin{aligned} C &= \begin{bmatrix} C_{11} & \cdots & C_{1q} \\ \vdots & \ddots & \vdots \\ C_{q1} & \cdots & C_{qq} \end{bmatrix} \\ &= U \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_q \end{bmatrix} U^H \\ &= \lambda_1 \mu_1 \mu_h^H + \cdots + \lambda_q \mu_q \mu_q^H \end{aligned}, \tag{2.46}$$

where

- λ_i, is the i-th eigenvalue,
- $\lambda_1 > \lambda_2 > \ldots > \lambda_q$
- μ_i, is the i-th eigenvector.

The important property of this decomposition is that the eigenvectors can be interpreted as a partial scattering mechanism of the object, while the eigenvalues are interpreted as the corresponding relevance of this scattering mechanism compared to the other eigenvalues.

2.3 Radar Gridmaps

A Radar Gridmap, in general, is a representation of spatially dependent information. They are useful in environments with no other absolute information, like GPS. For example they are widely used for robot localization in factories or warehouses. A single map is defined by the following three main parameters [21]

- The resolution, which determines the width and height of a single grid cell,
- The total size, which defines the number of grid cells in each direction,
- The layers, which hold the essential information, e.g. the RGB channels in a picture.

2.3.1 Amplitude Gridmaps

An approach introduced by Werber et al. [40], called *amplitude gridmaps* considers only the amplitude and doesn't include the polarimetric information. They represent the measured power in a single cell, like the example depicted in Fig. 2.11 . The measured uncertainties depend on the distance and angle; the single powers are not only averaged to calculate a cell value. The value for the cell on row u and column v is calculated by the inverse weight ranging,

$$G(u,v) = \frac{\sum_{k=0}^{i} \left(\frac{1}{r_{u,v}} \cdot A_{u,v}(k) \right)}{\sum_{k=0}^{i} \left(\frac{1}{r_{u,v}} \right)} \tag{2.47}$$

where

- $A(k)$ is the measured power of detection k,
- i is the number detections at $G_{u,v}$.

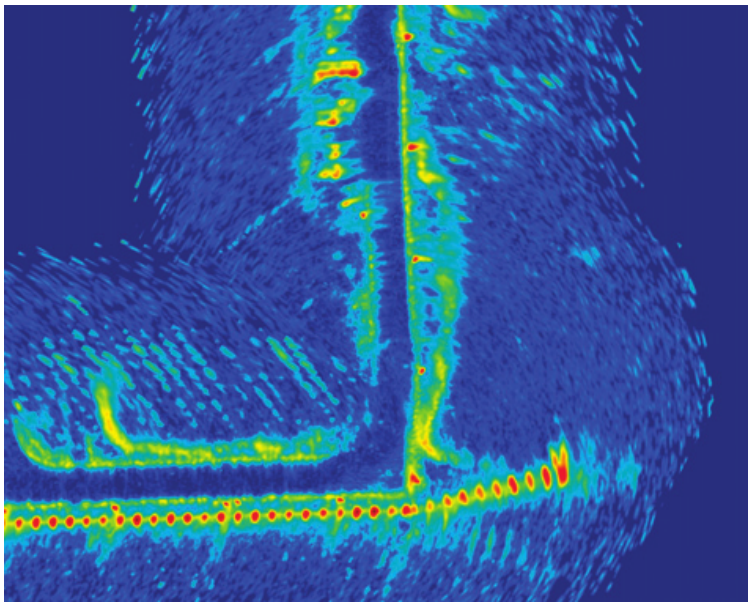

Figure 2.11: Example Amplitude gridmap with the lowest activations in blue and the highest activations in red [40]

2.3.2 Covariance Gridmaps

Another approach, first introduced by Weishaupt et al. called *polarimetric covariance gridmaps*[41] is more complex than the amplitude gridmaps, but it holds more information. It is a combination of polarimetric radar measurements and gridmaps to accumulate polarimetric information into urban

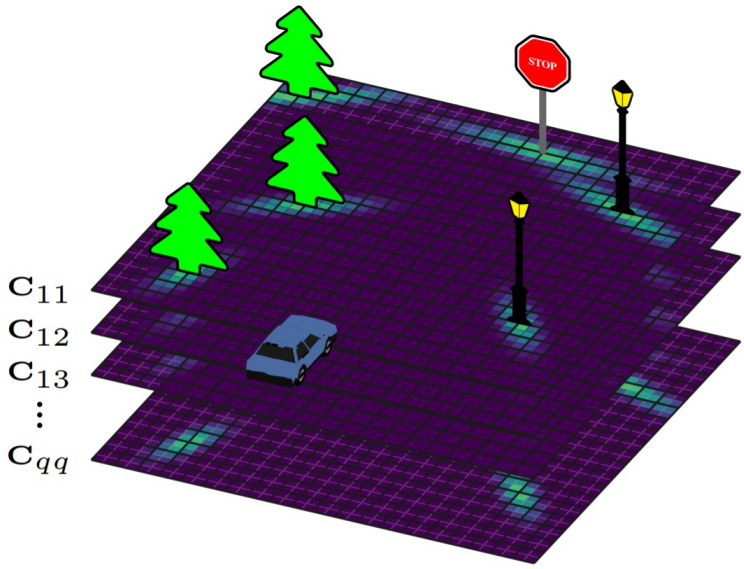

Figure 2.12: Conceptual overview of polarimetric covariance gridmapping. Each grid cell contains a complete covariance matrix constructed from the polarimetric feature vectors [41]

scenarios. A conceptual overview is depicted in Fig. 2.12. The creation of such covariance gridmaps is based on the covariance matrix C calculated by the retrieved scattering vector $\underline{\Omega}$. The accumulation is, therefore, similar to PolSAR applications. A single Grid Cell $\boldsymbol{G}(u,v)$ is then calculated by the average over all measured covariances at this cell. A simple average would not account for the target's distance, similar to the accumulation of amplitude gridmaps. So here again, an inverse range weighting is used. Therefore a single cell is described by [41]

$$\boldsymbol{G}(u,v) = \frac{\sum_{t=0}^{T} \frac{1}{d_t} \hat{\boldsymbol{G}}_t(u,v)}{\sum_{t=0}^{T} \frac{1}{d_t}}, \tag{2.48}$$

with

- T the last time a measurement for position (u,v) is available,
- $1/dt$ the corresponding distance between cell and sensor at time t.

At this point, all measurements are taken from different sensors in different coordinates relative to the world coordinates. So, all measurements have to be transformed first in the world coordinate system. Furthermore, the transformation of the measurement uncertainties is also important. Using the first order Taylor approximation, the uncertainty covariance matrix $Q_{\text{polar}} = \text{diag}\left(\sigma_r, \sigma_\theta\right)$, with the radial and, angular accuracy in polar coordinates σ_r, σ_θ, can be first transformed into Cartesian coordinates and then into

world coordinates by[41]

$$\boldsymbol{Q}_{\text{world,t}} = \boldsymbol{R}(\gamma)\boldsymbol{J}_t\boldsymbol{Q}_{\text{polar}}\,\boldsymbol{J}_t^T\boldsymbol{R}(\gamma)^T, \tag{2.49}$$

where,

- $R(\gamma)$ describes the rotation by angle γ,
- J describes the 2x2 Jacobian matrix derived from the first order Taylor approximation of the polar to Cartesian transformation.

The polarimetric covariance gridmap value at cell (u, v) at time t is therefore given by[41]

$$\hat{\boldsymbol{G}}_t(u, v) = \sum_{m=0}^{M} C_{m,t} e^{-0.5 \cdot d_{m,l}^T Q_{\text{world},t}^{-1} d_{m,l}}, \tag{2.50}$$

with

- $d_{m,t}$ is the distance vector between target m and the sensor at time t,
- The covariance matrix $C_{m,t}$, which is given by (2.40)

The generated gridmap for itself is not so easy to visualize. But using the eigenvalue decomposition of the covariance matrix it is possible to create so-called eigenmaps to examine different scattering mechanism in complex urban scenarios. The eigenvalue decomposition leads to $n \in \{1, \ldots, q\}$ different eigenmaps, where in each cell of such eigenmaps, the product of the corresponding eigenvector and eigenvalue of the initial covariance matrix from that cell represents the scattering mechanism. The result is an RGB image in which each polarization channel corresponds to a RGB color, with S_{LL} being red, S_{LR} green, and S_{RR} blue. The colors green, purple and white in such eigenmaps have a special role. Green indicates a strong odd bounce scattering, purple indicates even bounce scattering, and white signals a mixed scattering mechanism. Odd bounce scattering means the radar wave is reflected once or an odd number of times at an object and then reflected to the receiver and even bounce the same for an even number of times. It is mentioned that the colors described are only valid for the left and right-handed circular basis. For other polarimetric bases, the colors are different, depending on their definition. An example of such an eigenmap is shown in Fig. 2.13.

2.3.3 PreCFAR Gridmaps

To only get a gridmap of the static environment, it is important to exclude moving targets from the accumulation. The typical approaches using only detections from a sparse CFAR point cloud, result in suppression of low amplitude objects and partial masking of extended objects. Weishaupt et.

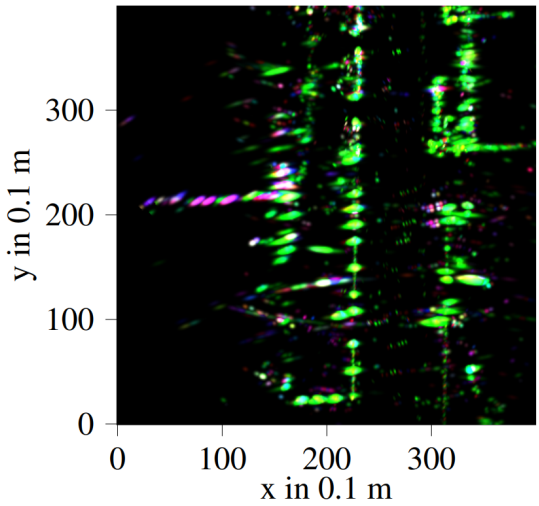

Figure 2.13: Example eigenmap λ_1 of a covariance matrix. Fig. 3.16 in [21]

al. introduced an approach called *PreCFAR gridmaps*[42] to represent the static environment as complete as possible. By including all measured backscattering power of static objects instead of using only detections from a CFAR point cloud. Assuming the data is given in a typical processed datacube s(r, ϕ, v_r) the static environment is located on a curved surface like depicted in Fig. 2.14. This is because the radar only measures the

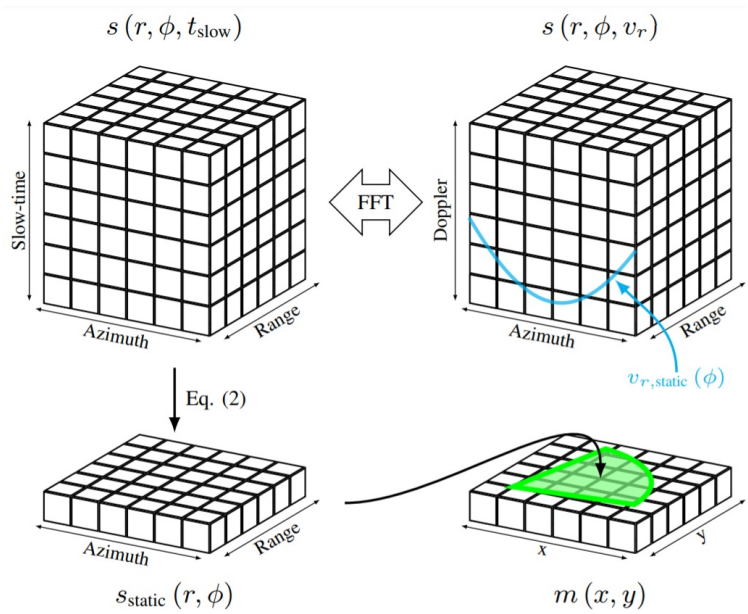

Figure 2.14: Conceptual visualization of the proposed signal processing approach for mapping the static environment as complete and dense as possible. Fig. 1 in [42]

radial projections of any velocity vector and therefore the expected Doppler velocity of a static object depends not only on the motion state of the object

but also on the relative azimuth angle under which it is observed. Provided that all quantities are measured in the car coordinate system, the expected radial velocity of a static object can be calculated by[42]

$$v_{r,static}(\phi) = -\mathbf{e}_r(\phi) \cdot (\mathbf{v} + \mathbf{\Theta} \times \mathbf{d}), \tag{2.51}$$

with

- the sensor's radial unit vector $\mathbf{e}_r$ pointing to the static objects,
- the ego-vehicle velocity vector $\mathbf{v}$,
- the angular rate vector $\mathbf{\Theta}$,
- the spatial displacement vector d.

Hence, to detect the static objects, one could select those Doppler bins of the fully processed radar data cube $s(r, \phi, v_r)$, which are closest to $v_{r,static}(\phi)$ This approach requires some interpolation for mitigation because the spectrum is not always aligned with frequency bins, and therefore amplitude discontinuities along the angular dimension would occur. Instead of the fully processed radar data cube, Weishaupt et al. start with an unprocessed Doppler dimension $s(r, \Theta, t_{slow})$ and use a technique based on the Fourier modulation theorem to avoid any additional interpolation and being still in slow-time domain. This allows modulating each azimuth slice by the expected Doppler frequency with an inverted sign, so that static objects would be located in the DC bin after a FFT in Doppler direction. The FFT, in this case, is then replaced by the following simple summation[42].

$$s_{static}(r, \phi) = \sum_{t_{slow}} s(r, \phi, t_{slow}) \exp\Big(-j2\pi \frac{v_{r,static}(\phi) t_{slow}}{2|v_r|_{max} T_{chirp}}\Big), \tag{2.52}$$

where

- $s_{static}(r, \phi)$ is an image-like representation of the static environment in polar coordinates,
- $|v_r|_{\max}$ is the maximum single-sided unambiguous radial velocity,
- t_{slow} is the sampling time of each chirp,
- T_{chirp} is the duration of each fast time cycle.

An illustration of this approach is shown in Fig. 2.14. At this point, it has to be mentioned that the resulting map is still in polar coordinates, and a transformation into Cartesian coordinates has to be done to allow an accumulation in the gridmap. In this work, a nearest-neighbor interpolation is used.

2.4 Radar-based Automotive Localization using Landmarks in a Multimodal Sensor Graph-based Approach

The approach of Jürgens et al.[7] provides the state-of-the-art for radar only based self-localization. In their paper *Radar-based Automotive Localization using Landmarks in a Multimodal Sensor Graph-based Approach*, they using a real-time graph-based SLAM approach to address the self-localization problem with automotive-grade radars. Besides the radar only approach, they also used a sensor fusion with contributions of lidar and camera sensors. To enable this, a single semantic landmark map is used and maintained for all sensors. Additonaly, to a radar only egomotion estimation they also testing an IMU to predict the vehicles motion state. Therefore, the contribution of this work is a graph-based SLAM system, that is able to perform on radar data alone as well as on a sensor fusion with a wide multimodal range of sensors. Fig. 2.15 shows an example of SLAM graph of Hamburg inner city Out of the untracked radar reflection points of a single

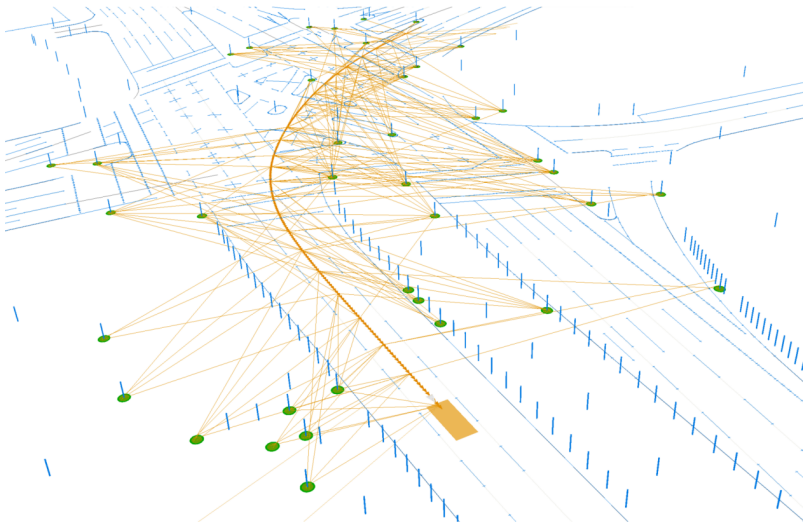

Figure 2.15: "SLAM graph of Hamburg inner city, run on radar data alone. Map in blue, landmark measurements in orange, map associations in green and localization pose in orange"[7], Fig. 1 in [7]

measurment cycle, the extraction of odometry information and landmarks is done in real-time.

2.4.1 Graph-based SLAM

The approach was previously used in [43] and [44] mainly on lidar, odometry and GNSS data. It is in general a graph-based SLAM framework using

landmark, odometry and global information with the following design objectives:

- "Fusion of information of different sensor modalities: radar, lidar, camera, odometry e.g. from inertial sensors and wheel encoders, and GNSS.
- Use of a single, semantic landmark map for all sensor modalities. It can be constructed by our system or pre-recorded by an external supplier.
- Modularity: Split of the system into the three parts: landmark extraction, odometry calculation and graph construction and minimization.
- Sensor redundancy: The system should perform stably with only a subset of possible sensor inputs, e.g. radar only or lidar and odometry information alone."[7]

The landmark-based SLAM aims to predict the poses of the vehicle and landmarks, that best fit to the past made sensor measurements as well as to the landmark map.There are three types of measurements: landmark position, odometry measurement or global pose measurement (e.g. GNSS). Written as a minimization problem, the state $\boldsymbol{x} = \left[\boldsymbol{x}^p \boldsymbol{x}^l\right]$ is defined with the vehicle poses $\boldsymbol{x}^p = \left[\boldsymbol{x}_1^p, \ldots, \boldsymbol{x}_P^p\right]$ and landmark positions $\boldsymbol{x}^l = \left[\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_L^l\right]$. The length of the sliding window is determined by the number of poses P and L is the number of landmark estimates. Therefore, the aim is solving

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \sum \boldsymbol{e}_i \left(\boldsymbol{x}, \boldsymbol{z}_i\right)^T \boldsymbol{\Omega}_i \boldsymbol{e}_i \left(\boldsymbol{x}, \boldsymbol{z}_i\right) + \mathcal{F}^{\text{map}} \left(\boldsymbol{x}^l\right),$$

with

- $\boldsymbol{e}_i$ is the i-th error function,
- $\boldsymbol{\Omega}_i$ is the i-th information matrix,
- z_i is the i-th measurement.

The information matrices $\boldsymbol{\Omega}_i$ as well as the error functions $\boldsymbol{e}_i$ differ for the different measurement types of odometry, global poses, landmark observations, and the landmark map associations they made by incorporating the measurement information z_i. Therefore, the following minimization problem can be specified

$$\mathcal{F}^{\text{map}} \left(\boldsymbol{x}^l\right) = \sum_i \boldsymbol{e}_i^{\text{map}} \left(\boldsymbol{x}^l\right)^T \Omega_i^{\text{map}} \boldsymbol{e}_i^{\text{map}} \left(\boldsymbol{x}^l\right). \tag{2.53}$$

Here, the error function is defined by

$$\boldsymbol{e}_i^{\text{map}} \left(\boldsymbol{x}^l\right) = \boldsymbol{x}_i^l - \boldsymbol{m}_i \tag{2.54}$$

, and m is the map with global positions of the landmarks. The approach performs three main steps. First, new measured objects are associated to

landmark states and added to the graph. Second, the landmark states are associated to the landmarks of the map. Last but not least, the problem is solved numerically.

2.4.2 Evaluation

The evaluation experiments differentiate between 3 types of the used approach based on the applied sensors. The first approach uses radar data only, that means radar for target detection during the localization process and from the map together with radar-based ego-motion estimation. The second approach uses a sensor fusion of radar and camera sensors for the target detections together with also radar-based egomotion estimation. The last approach uses a sensor fusion of radar, lidar and camera sensors for the detections and vehicle odometry instead of radar egomotion estimation. The experiments are performed on two different test vehicles a VW e-Golf 7

Figure 2.16: Vehicle sensor setup. The cars use the 6 radars on the corners and on the sides, the trucks 4, Fig. 2 in [7]

and a MAN semi-trailer truck, both depicted in Fig. 2.16. The setup for both cars is similar. For the vehicle odometry they use wheel encoders and an IMU. Also commonly, they have a close-to-series GNSS receiver, Velodyne lidars, a front camera for lane detection, and have a reference grade RTK system installed. An ADMA for the truck and for the car an Applanix. The used radars are for both 77GHz automative-grade radars. The cars have a a 360 ° coverage by 6 short range radars but the trucks only use 4 of them. An additional long range radar in the front is not used in the experiments. The positions of the radars are at a height of 40-70 cm. The data rate is 20 Hz, which also is used as the output frequency of the odometry.
The test scenarios are two different drives of Hamburg inner city and Hamburg terminal. The traffic was quite dense in the city and moderate at the terminal. Third-party, pre-recorded maps based on high dense lidar data are used. In the experiments, the driven paths for radar odometry and the

	68% city	95% city	68% termi.	95% termi.
radar-only lat.	0.25 m	1.07 m	0.75 m	> 2 m
radar-only abs.	0.32 m	1.41 m	1.16 m	> 2 m
radar & camera lat.	0.13 m	0.76 m	0.27 m	1.79 m
radar & camera abs.	0.20 m	> 2 m	0.98 m	> 2 m
full setup lat.	0.07 m	0.18 m	0.17 m	0.28 m
full setup abs.	0.15 m	0.28 m	0.24 m	0.45 m

Table 2.3: 68 and 95 percentile of the cumulative distribution function (CDF) of the absolute position errors for Hamburg inner city and terminal regarding to the different sensor fusions. Table 1 in [7]

localization system in the three cases radar-only, radar and camera localization, and the full sensor setup including the lidars and the vehicle odometry are compared. As ground truth the vehicles RTK reference systems are used. The driven paths are visualized in Fig. 2.17, where the absolute position error is indicated by the color coding and the blue paths come from radar odometry only. The final results are shown in Table 2.3.

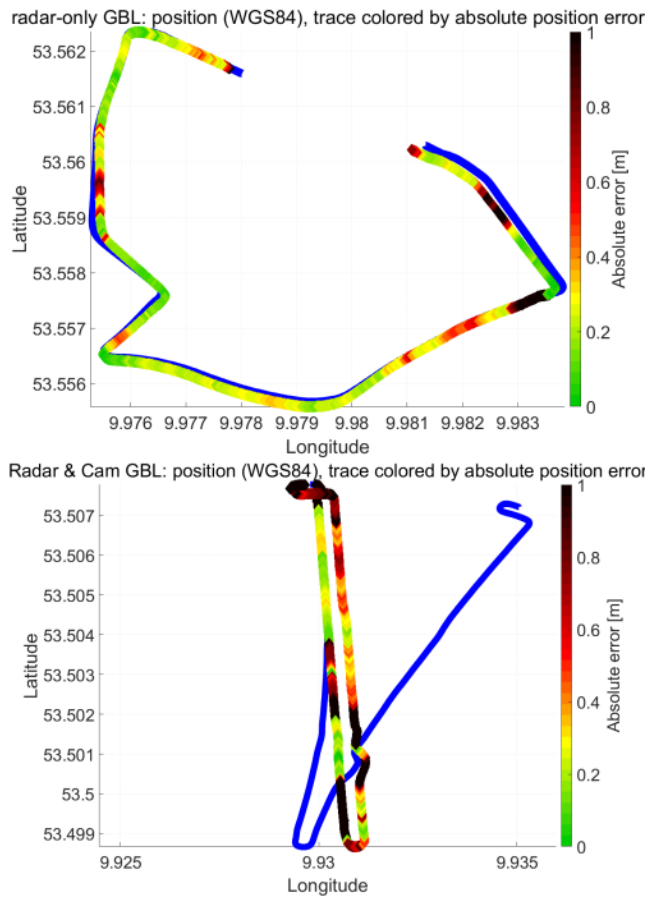

Figure 2.17: Driven paths for the city (top) and terminal (bottom) scenario. Fig. 4 in [7]

3 Measurement Setup

The measurement equipment for this work consists of three FMCW multiple-input multiple-output polarimetric radar sensors and a vehicle from Mercedes Benz. Figure 3.1 shows the Measurement vehicle from Mercedes Benz. The red arrows point to the radar sensors mounting positions. The sensors

Figure 3.1: Measurement vehicle, red arrows pointing to the polarimetric radar sensors.

include one sensor facing the vehicle's driving direction, while the left and right sensors face the side of the measurement vehicle at $\pm 60°$ respectively. The left and the front sensor are mounted upside down, while the right sensor is upright. The sensors are able to transmit and receive both polarizations in the $(\hat{l}, \hat{r})$ basis. The sensors are transmitting left-circular and right-circular polarized waves alternately but receiving both polarizations at the same time. Even though the sensors are quasi monostatic, the bistatic scattering matrix is measured. More details regarding the polarimetric radar sensors are shown in Table 3.1.

Using the measurement vehicle with the polarimetric radar sensors, three datasets are measured . The first one was recorded near the city Ulm and consists of four drives with alternating driving directions along a route of approximately 5 km, including driving direction changes. The vehicle's velocity was tried to stay between $\pm 5m/s$ in the unambiguous Doppler velocity range. Furthermore, a second dataset was collected near the city of Ingolstadt, Germany. Similar to the dataset recorded in Ulm, the route

Parameter	**Value**
Frequency band	77 - 79 GHz
Maximum Range	80 m
Range Resolution	7.5 cm
Field of View	$\pm 45°$
Azimuthal Resolution	$3°$
Unambiguous Radial Velocity	$\pm 5.15 \frac{m}{s}$
Radial Velocity Resolution	$0.15 \frac{m}{s}$
Update Rate	10 Hz

Table 3.1: Details of the used polarimetric radars

consists of approximately 3 km, again including changes of direction and a mixture between city and rural areas. For the ground truth velocities and locations, DGNSS and an inertial measurement unit (IMU) are used.

4 Ego-Motion Estimation

An accurate ego-motion estimation is not only necessary in driving dynamics control but also for precisely self-localization. A combination of inertial sensing (gyroscopes and accelerometers) and wheel-based odometry is the most common technique for vehicle odometry. The measured rotational movement and wheel-based odometry are enough in most cases but are insufficient if the vehicle drifts. This is compensated with periodic GPS absolute position updates. These systems provide an accurate motion estimation motion estimation on high-traction terrain and with good GPS reception, but in slippery terrain or during high-dynamic maneuvers wheel speed sensors contain non-systematic errors due to wheel slip and slide.
New possibilities for an instantaneous and accurate ego-motion estimation are coming from the growing use of Doppler radars and the continuously increasing measurement accuracy. This work follows the multiple-sensor ego-motion estimation approach of Kellner et al. [45], which is the extension of the single-sensor ego-motion estimation [46] and is based on the motion of the ego-vehicle relative to stationary radar targets. The approach is invariant against the vehicle's side-slide motions like drifts and is able to estimate the full 2D motion state, which includes the 3 degrees of freedom (DOF) *longitudinal*, *lateral velocity*, and *yaw rate*. The approach requires nothing more than the Doppler radars installation positions/orientations and could be applied with an arbitrary number of sensors.
The main idea is that all stationary targets have the negative 2D motion state to the vehicle and can be easily identified because they are probably the largest group, categorized by the movement. The algorithm consists of 3 parts and takes the mounting positions of each sensor and the measured target radial velocities and azimuth angles as input. In the first part, all sensor mounting positions, target information and the vehicle's motion state parameters will be rearranged in a simple linear equation (called the *ego-motion equation*) while assuming that the targets are stationary. Subsequently, stationary targets are identified by a statistical approach, performed by a *RANdom SAmple Consensus algorithm* [47] (RANSAC) on the ego-motion equation. In the last step, a least-square fit solves the ego-motion equation. For a 2 DOF estimation, at least two targets, and for a 3 DOF estimation, three targets, from at least two sensors have to be captured.

4.1 System Motion Equation

Compared to a moving radar sensor $j \in \{1, \ldots, M\}$, all stationary targets move in the opposite direction, like depicted in Fig. 4.1 The relative velocity

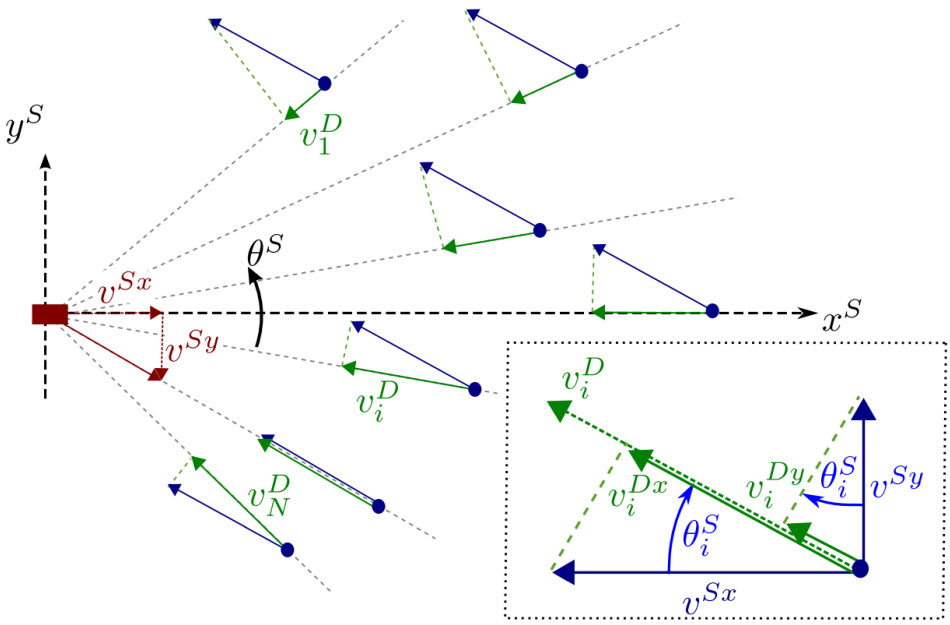

Figure 4.1: Sensor (red) and targets (blue) with longitudinal, lateral and radial(green) velocities in the sensor coordinate system, figure 2 in [45]

of a stationary target $i_j \in \{1, ..., N_j\}$ is equal to the inverse sensor velocity vector (v_j^{Sx}, v_j^{Sy}). The radar measures only the corresponding radial (Doppler-) velocities of this vector (v_{ji}^D), depending on the azimuth position $(\theta_{i,j}^S)$. The combination of all stationary targets from one sensor leads to [45]

$$\begin{bmatrix} -v_{j,1}^D \\ \vdots \\ -v_{j,N_j}^D \end{bmatrix} = \begin{bmatrix} \cos(\theta_{j,1}^S) & \sin(\theta_{j,1}^S) \\ \vdots & \vdots \\ \cos(\theta_{j,N_j}^S) & \sin(\theta_{j,N_j}^S) \end{bmatrix} \begin{bmatrix} v_j^{Sx} \\ v_j^{Sy} \end{bmatrix} \tag{4.1}$$

The resulting velocity vector (v_j^{Sx}, v_j^{Sy}) and the azimuth angles are in the local sensor coordinate system. The velocity information has to be transformed into a global coordinate system (v_j^x, v_j^y) to combine the velocity information of multiple sensors and to get an equation based on the velocity profile of the vehicle. Here the definition of the vehicle coordinate system from [48] [49] is used as a global coordinate system, like shown in Fig. 4.2. The origin is located at the center of the vehicle's rear axle. The x-axis is perpendicular to the rear axle and is aligned to the front. The y-axis is parallel to the rear

axle and points to the left in the direction of travel. The z-axis is aligned upwards and defines the direction of rotation in the right-handed coordinate system. So with the mounting orientation β_j follows [45]

$$\begin{bmatrix} v_j^{Sx} \\ v_j^{Sy} \end{bmatrix} = \begin{bmatrix} \cos(\beta_j) & \sin(\beta_j) \\ -\sin(\beta_j) & \cos(\beta_j) \end{bmatrix} \begin{bmatrix} v_j^x \\ v_j^y \end{bmatrix}. \tag{4.2}$$

The global measurement equation results from the combination of (4.1) and (4.2) with the global azimuth angle $\theta_{j,i} = \beta_j + \theta_{j,i}^S$ by

$$\begin{bmatrix} -v_{j,1}^D \\ \vdots \\ -v_{j,N_j}^D \end{bmatrix} = \begin{bmatrix} \cos(\theta_{j,1}) & \sin(\theta_{j,1}) \\ \vdots & \vdots \\ \cos(\theta_{j,N_j}) & \sin(\theta_{j,N_j}) \end{bmatrix} \begin{bmatrix} v_j^x \\ v_j^y \end{bmatrix}. \tag{4.3}$$

$$\mathbf{v_j^D} \quad = \quad \mathbf{M_j} \quad \mathbf{v_j}$$

The motion of an arbitrary rigid object is fully described by the reference point velocity (v_x, v_y) and yaw rate ω of the object. The sensor positions are

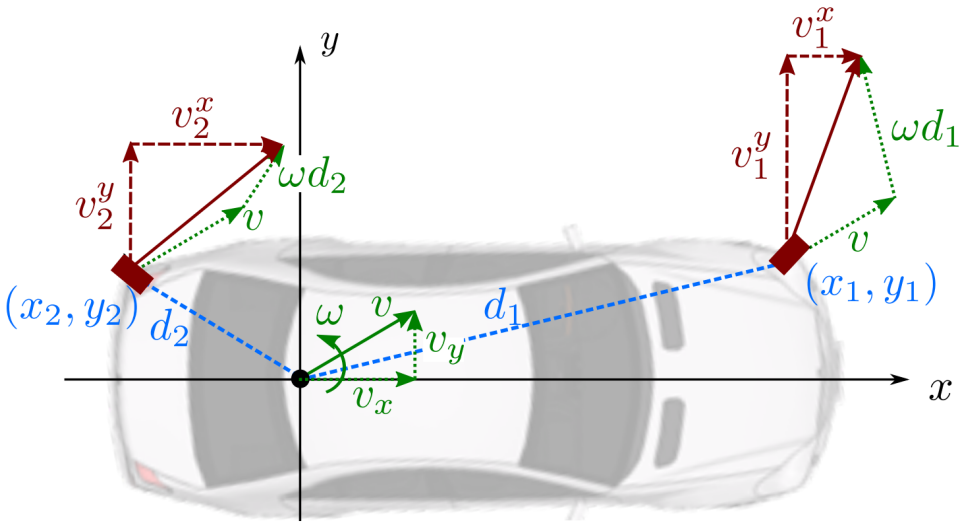

Figure 4.2: Vehicle motion specified at reference point (middle of rear axis) and the corresponding sensor motion (red - v_j^x, v_j^y) at position (blue - x_j, y_j), Fig 3 in [45]

described in vehicle coordinates as (x_j, y_j). The velocity vectors (v_j^x, v_j^y) of all sensors $(j = 1, \ldots, M)$ are given by [15]

$$\begin{bmatrix} v_1^x \\ v_1^y \\ \vdots \\ v_M^x \\ v_M^y \end{bmatrix} = \begin{bmatrix} -y_1 & 1 & 0 \\ x_1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ -x_M & 1 & 0 \\ y_M & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega \\ v_x \\ v_y \end{bmatrix} \tag{4.4}$$

$$[\mathbf{v}_1 \dots \mathbf{v_M}]^T \;=\; [\mathbf{S}_1 \dots \mathbf{S_M}]^{\mathbf{T}} \quad \mathbf{s}.$$

(4.3) can be rewritten with the chosen reference point for all sensors, if the Ackerman condition (no side-slip at the rear axis) is used, by deleting the last element of **s** and the last column of **S**, as [45]

$$\begin{bmatrix} \mathbf{v_1^D} \\ \mathbf{v_2^D} \\ \vdots \\ \mathbf{v}_\mathrm{M}^\mathrm{D} \end{bmatrix} = \begin{bmatrix} \mathbf{M_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{M_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{M}_\mathrm{M} \end{bmatrix} \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \\ \vdots \\ \mathbf{v}_\mathrm{M} \end{bmatrix}. \tag{4.5}$$

The complete system motion equation is a combination of (4.1) and (4.5):

$$\begin{bmatrix} \mathbf{v}_1^\mathrm{D} \\ \mathbf{v}_2^\mathrm{D} \\ \vdots \\ \mathbf{v_M^D} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1\mathbf{S}_1 \\ \mathbf{M}_2\mathbf{S}_2 \\ \vdots \\ \mathbf{M_j S_j} \end{bmatrix} \begin{bmatrix} \omega \\ v_x \\ v_y \end{bmatrix} \tag{4.6}$$
$$\mathbf{v^D} = \mathbf{R} \quad \mathbf{s}.$$

4.2 Velocity Profile Analysis

Assuming the major group of targets with the same velocity profile are stationary objects, the RANSAC algorithm is used to identify them. With the aim to estimate 3 DOF, three targets from at least two different sensors are randomly chosen in each iteration and the parameters of their velocity profile **s** are determined solving the system motion equation 4.6 for them. Estimating only 2 DOF (ω and v_x), two targets are randomly chosen, and the system motion equation will be solved without the last column of **R**. A threshold is used to identify a limited corridor around the fit of the current determined velocity profile. The error of the current fit is then calculated by the radial velocity errors of all targets inside the corridor according to the calculated error and/or number of inliers to limit the outliers influence on the error. After a certain number of iterations, the best fit, according to the calculated error, is chosen. Finally, all outliers are considered as either moving objects or clutter and excluded. Since the RANSAC's estimated velocity profile only depends on 2 (2DOF) or 3 (3DOF) single targets, a better result is expected using all inliers to estimate the velocity profile again. So finally, all inliers are used to calculate a final velocity profile with the system motion equation based on the method of least-squares (LSQ)[45].

4.3 Handling Unambiguous Doppler Velocity

For the used radars, the unambiguous determination of the Doppler velocity is only possible within the interval $[-5.15\frac{m}{s}, 5.15\frac{m}{s}]$, as explained in Section 3. Due to this, the Doppler velocity $v^D_{j,i}$ of a target $i \in \{1, \dots N_j\}$ of sensor $j \in \{1, \dots M\}$ outside the interval is measured as

$$v^D_{j,i} = \left((v^{D,R}_{j,i} + v^D_{unambiguous}) \mod 2v^D_{unambiguous} \right) - v^D_{unambiguous}, \tag{4.7}$$

with

- $v^{D,R}_{j,i}$ is the real Doppler velocity of target i of sensor j,
- $v^D_{unambiguous} = 5.15\frac{m}{s}$ is the measurable absolute unambiguous Doppler velocity.

Fig. 4.3 shows the unambiguous Doppler velocity folding. For the measured

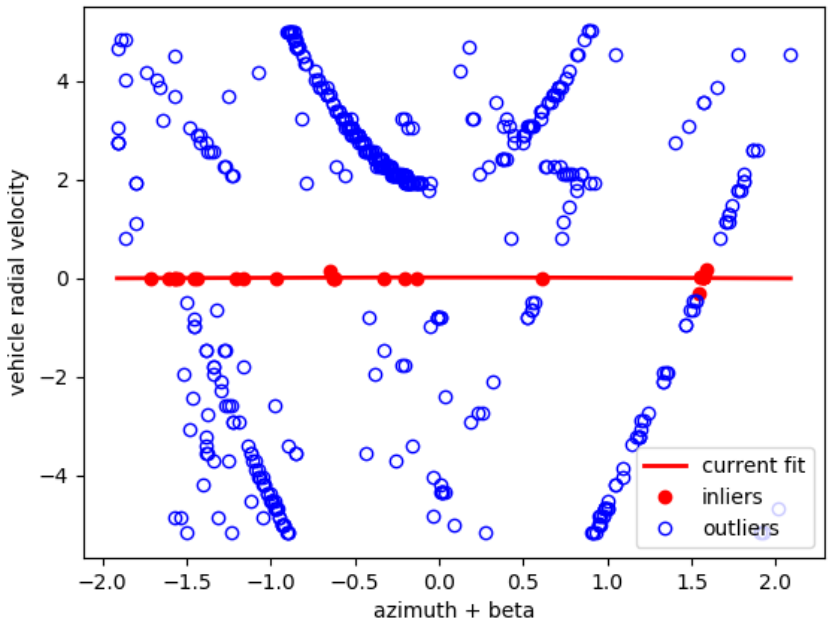

Figure 4.3: Failed ego-motion estimation example, with visible folding of the Doppler velocities due to the limited unambiguous range

targets from all sensors, the x-axis shows the azimuth angles, and the y-axis shows the Doppler velocities in the vehicle coordinate system. The blue points in the image correspond to the measured targets outside the corridor determined by the threshold of the RANSAC, and the red points correspond to the points inside. The red line corresponds to the estimated velocity

profile, which is in this case a fail. The U-shaped blue set of points in the area between approximately an azimuth angle between $[-0.8, 0.8]$ above the red line are shifted due to (4.7). Actually, they should be shifted downward with twice the unambiguous Doppler velocity. Fig. 4.4 shows the equivalent scene but with shifted radial velocities based on the estimated velocity profile. To

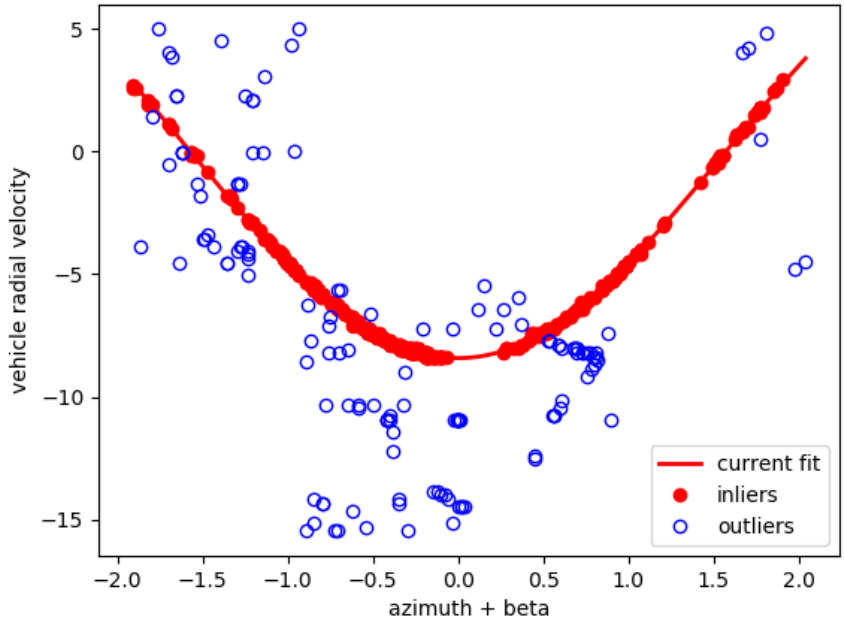

Figure 4.4: Successful ego-motion estimation example, with visible problem of shifted target Doppler velocities

avoid this problem, using (4.7) in RANSAC, the Doppler unfolding is also applied to the predicted Doppler velocities of the RANSAC and, before LSQ, all Doppler velocities of the targets are shifted downward by the associated shift of the RANSAC fit regarding the azimuth angle. Fig. 4.5 is another form of the plots used in Figures 4.3 and 4.4 with additional ground truth fit and by sensor separated targets, but it shows whats happening inside the RANSAC. There the Doppler unfolding is also applied to the fit of the RANSAC.

Due to the unambiguous Doppler unfolding and the proposed solution, another problem occurs. The proposed solution only works if the RANSAC choose targets, which are not affected by the problem of the unambiguous Doppler velocity. So if there are only a few targets with a radial velocity lower than the unambiguous Doppler velocity of $5.15\frac{m}{s}$, then the algorithm probably never chooses only the required targets with lower Doppler velocity. To avoid this, an additional part is added to the RANSAC, in which for each iteration and each chosen target all combinations of *wrapped-around* and *not*

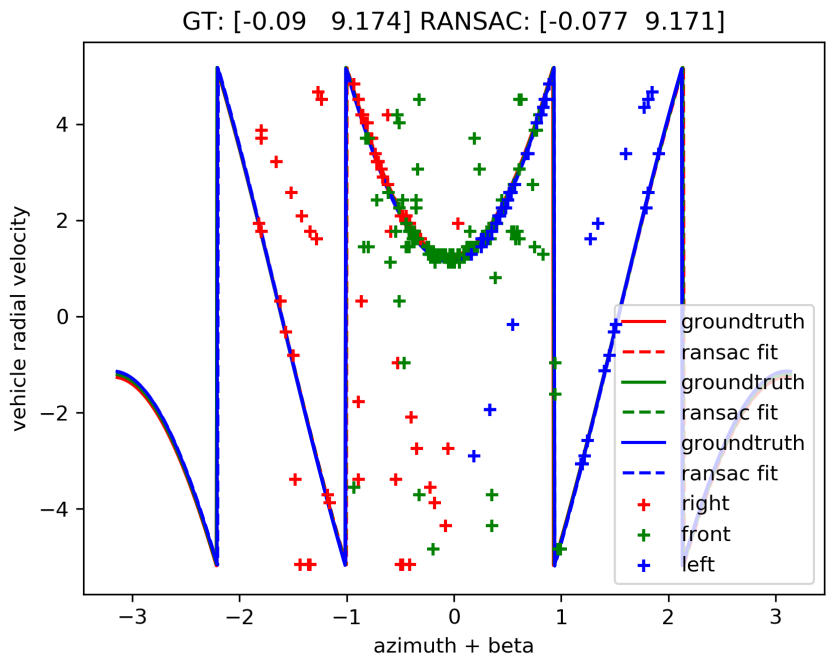

Figure 4.5: Successful ego-motion estimation example with Doppler shift from equation 4.7 on Doppler velocity estimations and ground truth

wrapped-around is applied and the minimum error of all combinations is selected. If a chosen target is assumed as wrapped-around then the Doppler velocity will be shifted down by the twice unambiguous Doppler velocity, and if not, then the target's Doppler velocity remains unchanged. There is a possibility that a one-time wrap-around is not enough, but in our case, it's fine because the car was not driving that fast than necessary. There is also the case, that the wraparound has to be done in the other direction, but our car was not driving so fast backwards. So the case of the simple wrap-around is sufficient.

4.4 Validation and Results

In the first step of the validation, the best threshold for the determination of inliers and the necessary number of samples used in the RANSAC has to be found. For this the 2DOF case is used and several thresholds between a minimum of $0.05\frac{m}{s}$ and a maximum of $1\frac{m}{s}$ are tested to find the right threshold. The number of samples, which defines the number of iterations of the RANSAC, is chosen as 200. The whole test results are depicted in table 4.1. The error of both the yaw-rate ω and x-velocity v_x decreases strictly to the minimum and increases then also strictly. The yaw-

rates error reaches its minimum at $0.01419\frac{m}{s}$ with a threshold of $0.3\frac{m}{s}$ and the x-velocity error at $0.01847\frac{m}{s}$ with a threshold of 0.2. the best overall threshold is at $0.2\frac{m}{s}$ and this will continue to be used in the following experiments. During the validation of the necessary number of minimum

	error	
threshold m/s	ω m/s	$\mathbf{v_x}$ m/s
0.05	0.01722	0.02731
0.06	0.01668	0.02505
0.07	0.01622	0.02337
0.08	0.01594	0.02216
0.09	0.01560	0.02108
0.1	0.01542	0.02038
0.2	0.01451	0.01847
0.3	0.01419	0.01931
0.4	0.01431	0.02087
0.5	0.01466	0.02323
0.6	0.01515	0.02588
0.7	0.01588	0.02873
0.8	0.01681	0.03223
0.9	0.01785	0.03565
1	0.01922	0.03994

Table 4.1: Ego-motion estimation threshold validation

samples within the RANSAC, the mean duration in seconds of an whole ego-motion estimation step is additionally measured for all numbers of tested samples. The minimum number of tested samples is 25 and the maximum is 1000. Doing this also, the 2DOF case is used. Table 4.2 shows the results of the validation. The result shows, that the number of samples does not affect the algorithms performance really much, because all errors with the usage of more than 100 samples are approximately the same. This shows that the necessary minimum number of samples is 200. The algorithm is also really fast and is even with 1000 samples faster than real-time by a fully vectorized implementation. In further experiments, the number of samples is set to 200.
Another important part is the choice of the best pair/trio of targets chosen in the RANSAC. Doing this, the mean L1 and mean L2 loss functions are tested. The whole error is then calculated by the loss functions error divided by the relative number of inliers regarding the current fit. Finally, the best fit with the minimum error under the top 10 fits regarding their relative number of inliers is chosen. During the error function validation, the ego-motion estimation is applied 10 times instead of 1 time on all frames. The results are depicted in Table 4.3 The result is clear; the L1 error function beats the

	error		
samples	ω m/s	$\mathbf{v_x}$ m/s	**duration** ms
25	0.01531	0.01888	1.10
50	0.01477	0.01859	1.77
100	0.01466	0.01853	3.15
200	0.01451	0.01847	5.69
300	0.01451	0.01848	8.89
400	0.01450	0.01851	11.58
500	0.01448	0.01849	14.07
750	0.01445	0.01850	21.30
1000	0.01444	0.01851	29.25

Table 4.2: Ego-motion estimation necessary minimum number of samples validation

error function	ω m/s	$\mathbf{v_x}$ m/s
L1	0.01521	0.01641
L2	0.01596	0.01753

Table 4.3: Ego-motion estimation error function validation

L2 error function in both x-velocity and yaw-rate.
The estimations of yaw-rate and x-velocity over all frames with the used L1 error function is depicted in figure 4.6 and 4.7. The predictions seem to be nearly perfect. There is no visible false estimation for the x-velocity but the estimations with an high absolute value of the yaw rate has a little bit lower estimated absolute value.
The result of the 3DOF case with 200 samples, a threshold of 0.2 and the L1 error function is shown in table 4.4.

DOF	ω m/s	$\mathbf{v_x}$ m/s	$\mathbf{v_y} m/s$
2	0.01521	0.01641	n.a.
3	0.1362	0.01675	0.5548

Table 4.4: Ego-motion estimation results

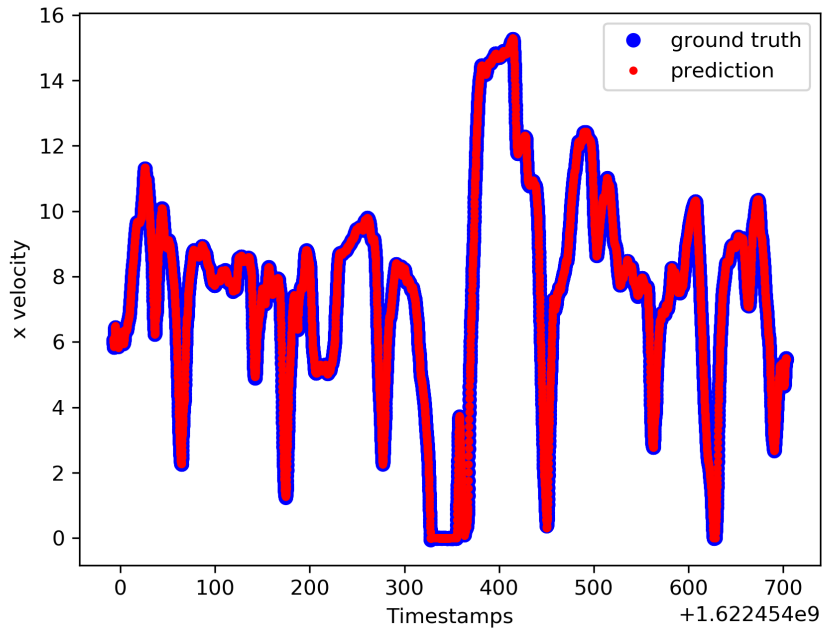

Figure 4.6: Ego-motion estimation depiction of x-velocity v_x over all frames for a whole drive

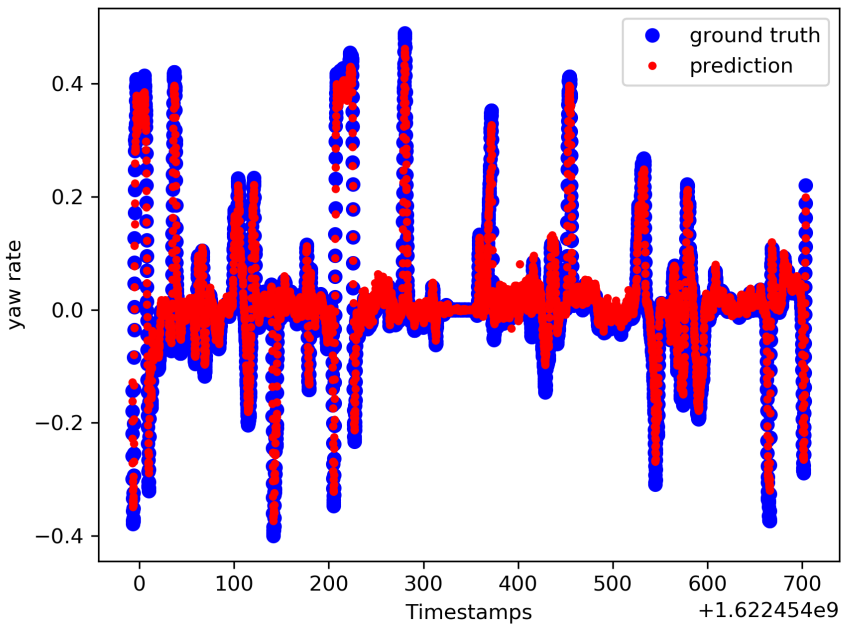

Figure 4.7: Ego-motion estimation depiction of yaw-rate ω over all frames for a whole drive

5 Automotive Self-Localization

Automotive self-localization in the case of this work is similar to a special case of mobile robot localization, which is the problem of determining a robot's pose relative to a given environment. The general localization problem is the most basic perceptual problem in robotics. Nearly all robotic tasks require the location of the known robot. The main goal of localization is to estimate the robots pose

$$x_t = (x \quad y \quad \theta)^T \tag{5.1}$$

at each time t in the global coordinate system of the map. Therefore it can be seen as a coordinate transformation problem, in which the localization describes the process of determining the correspondence between the map and the robot's local coordinate system. Knowing this transformation, the robot is able to express the other objects of the environment relative to himself in its own local coordinate system. Because of the uncertainties in sensors, the robots pose could not be sensed directly and must therefore be inferred from data. There are many localization types distinguished by the number of robots, a known initial pose or not, dynamic or static environments, the ability of a robot to be "kidnapped" and the way the robot move through the environment. In the case of this work the automotive self-localization problem is a form of so called *position tracking*, which means that the initial pose of the vehicle is known. The used PreCFAR covariance gridmaps leads to the fact that the environment is static and from the name, it applies that only a single vehicle pose has to be determined. The motion of the vehicle comes in general from its control and isn't affected by random moves or jumps in the environment, which makes the problem be an active localization problem instead of a passive localization problem[4].

Fig. 5.1 shows the interaction of a robot with its environment. The robots internal belief of the environment is depicted on the left. The internal belief is affected by the robots perception and its actions in the environment. In the case of this work, the map of the internal belief is a PreCFAR Covariance gridmap, and the introduced radars only do the perception. The only action of the vehicle, which changes the environment state is the vehicle's motion. In the following, the set of all environment measurements from timestamp 1 to $n \in \mathbb{N}$ is defined as

$$z_{t_1:t_n} = z_{t_1}, z_{t_2}, \ldots z_{t_n}, \tag{5.2}$$

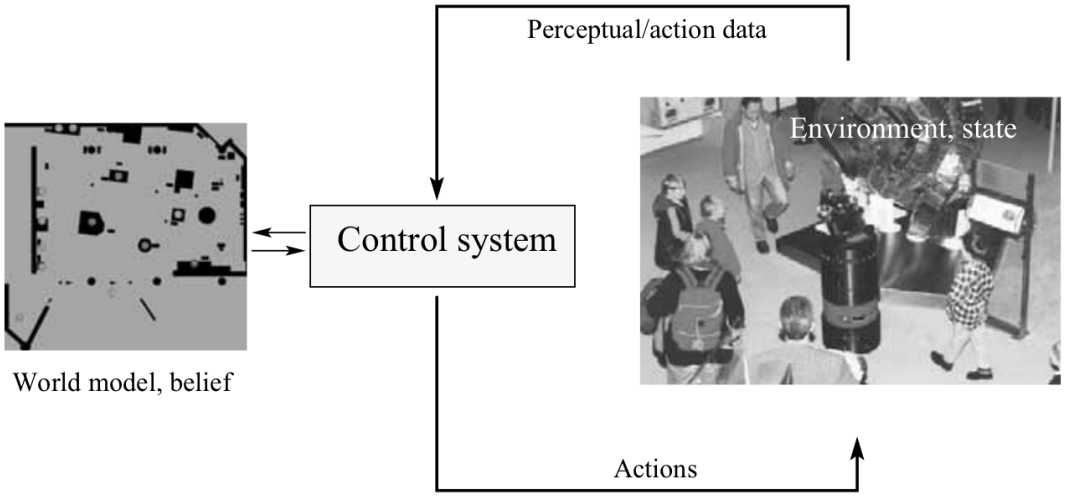

Figure 5.1: Robot environment interaction. Fig. 2.1 in [4]

and the set of all environment actions called controls from timestamp 1 to $n \in \mathbb{N}$ is defined as

$$u_{t_1:t_n} = u_{t_1}, u_{t_2}, \dots u_{t_n}, \tag{5.3}$$

The evolution of state and measurements are following probabilistic laws because of the motion and perception uncertainties. This leads to the fact that the state x_t has an underlying probability distribution. The state x_t could be conditioned on all past states, measurements, and controls. Hence, the underlying probability distribution might be of the following form

$$p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}), \tag{5.4}$$

assuming that the robot executes the control action before the measurement at a single timestamp t. An important fact is, if the state x is complete, then it is a summation of all states, measurements, and controls before, so that it suffices if the new state x_t is only derived by x_{t-1} and the control u_t. Expressed in probabilistic terms the following equality could be derived

$$p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t). \tag{5.5}$$

Corresponding to the probability distribution of the state one can derive the underlying probability distribution for a single measurement z_t, with again the following equality if the state x is complete

$$p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t | x_t). \tag{5.6}$$

In other words, the prediction of the measurement z_t only from the state x_t is sufficient if x is complete and no other past measurement, state or control is relevant. The probability $p(x_t | x_{t-1}, u_{t-1})$ is called *state transition*

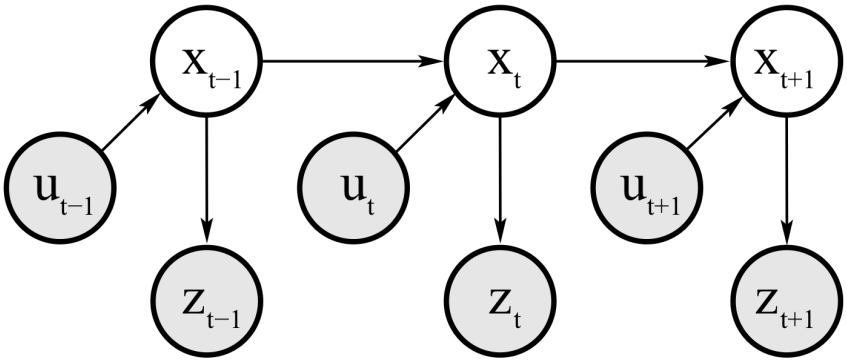

Figure 5.2: Characterization of the evolution of the states, measurements and controls in a Bayesian network. Fig.2.2 in [4]

probability and defines the stochastic evolution of the state environment depending on the controls u_t. In the case of this work, it can be seen as a noisy motion estimation model. The probability $p(z_t|x_t)$ is called the measurement probability and defines, which measurements are generated from the depending environment state x_t. It could be seen as a noisy projection from the state. Fig. 5.2 The stochastic dynamical system of the robot and the environment are described by the state transition probability and the measurement probability. Figure 5.2 illustrates the evolution of states and measurements, defined by those probabilities in a hidden Markov model also called dynamic Bayes network. The state at time t depends stochastically on the state at time $t-1$ and the control u t. The measurement z_t stochastically depends on the state at time t[4].

Another important concept is that of a *belief*, which is the internal knowledge from the robot of the environment state. The synonyms *state of knowledge* and *information state* are also used in the literature. The belief could also be defined as a conditional probability distribution. Belief distributions *bel* are posterior probabilities over state variables conditioned on the available data and defined as

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) \tag{5.7}$$

In comparison to the state transition probability the belief is taken after the measurement z_t and not before. It is useful to calculate another posterior before the measurement z_t is made but after the control u_t. The mentioned posterior is described by

$$\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t}). \tag{5.8}$$

The posterior $\overline{bel}$ is called *prediction* and the posterior $\overline{bel}$ is called *correction* or *measurement update*. This comes from the fact that $\overline{bel}$ predicts the state based on the previous state posterior, and *bel* corrects the state from $\overline{bel}$ by incorporating the measurement z_t[4].

5.1 Bayes Filter

The Bayes filter algorithm is the most general approach to calculate beliefs. The algorithm calculates the belief distribution *bel* from measurement z and control data u. The Bayes filter calculates the belief $bel(x_t)$ from the previous belief $bel(x_{t-1})$ and is, therefore, a recursive algorithm. In addition to the previous belief, the algorithm takes the most recent control u_t and the most recent measurement z_t as input. The output is the belief $bel(t)$. The so-called *update rule*, which is a single iteration of the Bayes filter algorithm, is depicted in Fig. The algorithm consists of two essential steps. The first one,

1: **Algorithm Bayes_filter(**$bel(x_{t-1}), u_t, z_t$**):**
2: *for all* x_t *do*
3: $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\ bel(x_{t-1})\ dx_{t-1}$
4: $bel(x_t) = \eta\ p(z_t \mid x_t)\ \overline{bel}(x_t)$
5: *endfor*
6: *return* $bel(x_t)$

Figure 5.3: Update rule of the Bayes filter algorithm. Table 2.1 in [4]

named the *control update* is processed in line 3. The belief $\overline{bel}$ is calculated over the state x_t based on the prior belief over state x_{t-1} and the control u_t. In general it is obtained by the integral (sum) of the product of two distributions. The prior assigned to x_{t-1} , and the probability that control u_t induces a transition from x_{t-1} to x_t . The second one, named the *measurement update*, incorporates the measurement z_t into the belief and is processed in line 4. For each hypothetical posterior state x_t, the belief $\overline{bel}(x_t)$ is multiplied with the probability that the measurement z_t has been observed. The result is normalized with the help of the constant η because the resulting product is not necessarily a probability and may not integrate to 1. The final belief $bel(x_t$ is then returned in line 6. Like any recursive algorithm, the algorithm requires an initial belief $bel(x_0)$ to compute the posterior belief. If the value of x_0 is known, a point mass distribution should be used, in which the correct value of x_0 is set to 1 and the rest is zero. If the value of x_0 is not known without any restriction, using a uniform distribution over the space of x_0 is the most common approach. For partial knowledge about the value of x_0, other distributions could be applied, but in general, the most common approaches are the ones described before. For most problems, the probabilities in lines 3 and 4 are not given. Hence, the basic Bayes filter algorithm can only be implemented for very simple estimation problems[4].

5.2 Particle Filter

An alternative version of the Bayes filter is the so-called particle filter. They approximate the posterior by a finite number of particles. The main concept of the particle filter is to use random state samples drawn from the posterior $bel(x_t)$ as an approximation of this posterior. An example for a Gaussian is illustrated in Fig. 5.4 A parametric form of representing the distribution is

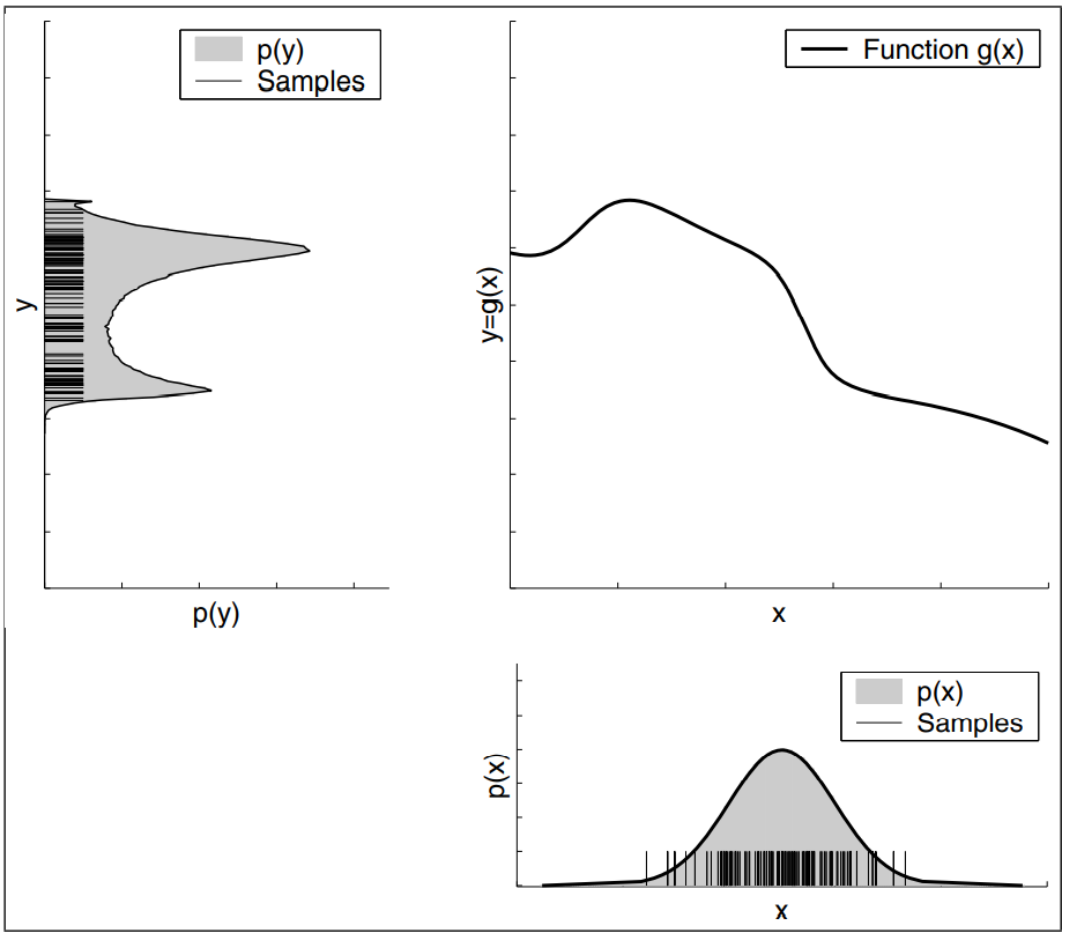

Figure 5.4: An example density function approximation used by particle filters. "The lower right graph shows samples drawn from a Gaussian random variable, X. These samples are passed through the nonlinear function shown in the upper right graph. The resulting samples are distributed according to the random variable Y"[4]. Fig. 4.3 in [4]

the exponential function that defines the density of a normal distribution. Instead of this, a particle filter represents a distribution by a set of samples drawn from this distribution, called particles. This leads to only an approximation of the distribution, but it can represent much more distributions than, e.g., Gaussians.The ability to model nonlinear transformations, as shown in Fig. 5.4 is also an additional advantage of the particle filter. An

additional advantage of the sample based representation is the ability to model nonlinear transformations of random variables[4].

The samples drawn from the posterior distribution in a particle filter are called *particles* and are denoted as

$$\mathcal{X}_t := x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]}, \tag{5.9}$$

with

- M is the number of particles in the particle set.

Each particle $x_t^{[m]}$, with $1 \leq m \leq M$ is an instance of the state at time t. So a single particle is a hypothesis of what the true world state may be at time t. The size of the particle set M is often large, for example $M = 1000$. The

```
Algorithm Particle_filter(X_{t-1}, u_t, z_t):
    X̄_t = X_t = ∅
    for m = 1 to M do
        sample x_t^[m] ~ p(x_t | u_t, x_{t-1}^[m])
        w_t^[m] = p(z_t | x_t^[m])
        X̄_t = X̄_t + ⟨x_t^[m], w_t^[m]⟩
    endfor
    for m = 1 to M do
        draw i with probability ∝ w_t^[i]
        add x_t^[i] to X_t
    endfor
    return X_t
```

Figure 5.5: The general particle filter algorithm, based on importance sampling. Table 4.3 in [4]

general approach behind particle filters is the approximation of the belief $bel(x_t)$ by the particle set X_t. So in an ideal case, the likelihood that a state hypothesis x_t is included in the particle set X_t should be equal to its Bayes filter posterior $bel(x_t)$.

$$x_t^{[m]} \sim p\left(x_t \mid z_{1:t}, u_{1:t}\right) \tag{5.10}$$

Corresponding to the Bayes filter algorithm, the particle filter is also a recursive algorithm. Because the particle set represents the belief, it follows that

the particle filter construct the particle set X_t recursively from the set X_{t-1}[4].

The simplest version of the particle filter algorithm is depicted in Fig. 5.5 The algorithm takes the previous particle set X_{t-1}, the most recent control u_t and the most recent measurement z_t as input. First, the algorithm processes each particle $x[m]_{t-1}$ to a new particle $x[m]_t$ of a temporary particle set $\overline{X_t}$ by incorporating the knowledge of the control u_t. The particle set $\overline{X_t}$ represents the belief $\overline{bel}(x_t)$. Second, it transforms the particle set $\overline{X_t}$ by the knowledge of the measurement z_t into the set X_t, which approximates the posterior distribution $bel(x_t)$. In more detail, the algorithm works as follows[4].

- In line 4 the temporary particle set is generated by successively sampling M hypothetical states $x[1]_t \dots x[M]_t$ for time t based on the previous particles $x[1]_{t-1} \dots x[M]_{t-1}$ and the control u_t. To perform this step, the ability to sample from the state transition distribution $p(x_t|u_t, x_{t-1})$ is necessary. After M iterations the result is the representation of the filter of $\overline{bel}(x_t$.
- In line 5 the corresponding importance factor $w[m]_t$ for each particle $x[m]_t$ is calculated. With the help of the importance factors the measurement z_t is incorporated into the particle set. The importance, as such, is the probability of obtaining the measurement z_t given the state of particle $x[m]_t$ defined by $w[m]_t = p(z_t|x[m]_t$. Assuming $w[m]_t$ as the weight of a particle, leads to an approximated representation of the Bayes filter posterior $bel(x_t)$ by the set of weighted particles.
- In lines, 8 through 11, the so-called *importance sampling* is implemented, which is the real "trick" of the particle filter algorithm. The algorithm changes the weights by frequencies by drawing with replacement M particles from the temporary set $\overline{(X_t)}$. Each particle will be drawn with the probability given by its importance weight. The resampling step transforms a particle set of M particles into another particle set of the same size, but the distribution of the particles change. The underlying distribution of the particles before the resampling step is the belief $\overline{bel}(xt)$, and after the resampling, the distribution is approximately the $bel(xt) = \eta p(z_t|x[m]_t)bel(x_t)$. This leads to a resulting sample set with many duplicates of particles $x[m]_t$ with high importance weight $w[m]_t$, but the particles with lower importance weights tend to vanish and are probably not contained in X_t.

To get a good understanding of the particle filter algorithm, it is necessary to take a further look at the resampling step. In general, one tried to solve the problem of computing an expectation over a probability density function f, but only samples generated from a different probability density function, g are given. For example, one might be interested in the expectation that $x \in A$. Assuming $f(x) > 0 \longrightarrow g(x) > 0$ it is possible to express this

probability as an expectation over g by

$$\begin{aligned} E_f[I(x \in A)] &= \int f(x) I(x \in A) dx \\ &= \int \underbrace{\frac{f(x)}{g(x)}}_{=:w(x)} g(x) I(x \in A) dx \\ &= E_g[w(x) I(x \in A)], \end{aligned} \tag{5.11}$$

with

- The indicator function

$$I(x) = \begin{cases} 1, & x \text{ is } \texttt{True} \\ 0, & x \text{ is } \texttt{False} \end{cases} \tag{5.12}$$

The importance sampling is able to make this transformation. Fig. 5.6 illustrates this process. The main goal is to obtain a sample from the distribution f. The problem is that sampling from f is impossible, but the ability to sample from another density g is available. The density g must be such that $f(x) > 0$ implies $g(x) > 0$, so that there is a non-zero probability to sample a particle from g for any state that can be sampled from f. Sampling from g (Fig. 5.6 b) leads to particles distributed by g not to f. Mathematically, for any interval $A \subseteq dom(X)$ the empirical count of particles that fall into A converges to the integral of g under A by

$$\frac{1}{M} \sum_{m=1}^{M} I\left(x^{[m]} \in A\right) \longrightarrow \int_A g(x) dx \tag{5.13}$$

The particles are weighted by the quotient

$$w^{[m]} = \frac{f\left(x^{[m]}\right)}{g\left(x^{[m]}\right)} \tag{5.14}$$

to offset the difference between f and g, which is shown in Fig. 5.6 (c). The vertical bars depict the magnitude of the importance weights.

They are the non-normalized probability mass of each particle. In particular, we have

$$\left[\sum_{m=1}^{M} w^{[m]}\right]^{-1} \sum_{m=1}^{M} I\left(x^{[m]} \in A\right) w^{[m]} \longrightarrow \int_A f(x) dx, \tag{5.15}$$

where the first term serves as the normalizer for all importance weights. This leads to the converging of the weighted particles to the density f, although they are generated from the density g[4].

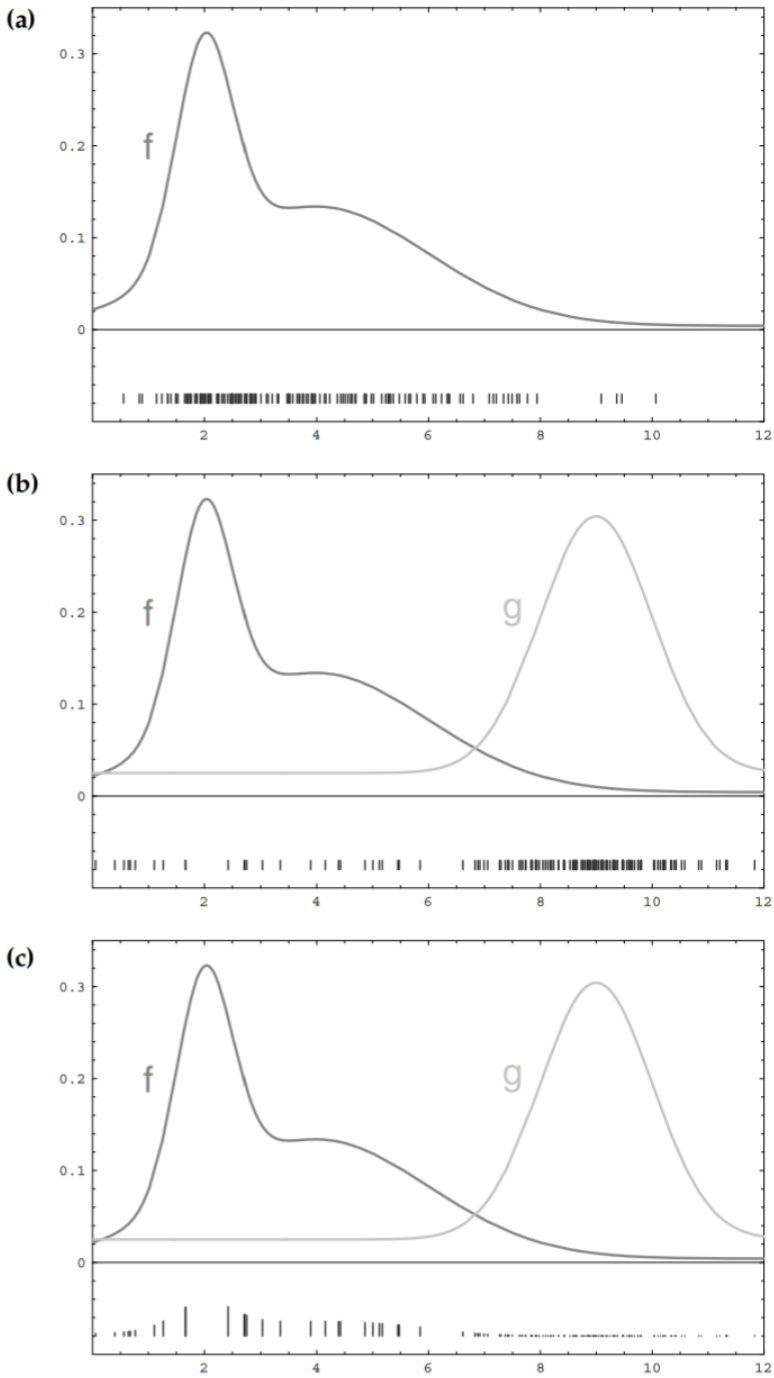

Figure 5.6: "Illustration of importance factors in particle filters: (a) Approximation of the target density f is sought. (b) Sampling from f directly is not possible, but instead of this samples are generated from a different density g. Drawn samples from g are shown at the bottom of this diagram. (c) A sample of f is obtained by attaching the weight $\frac{f(x)}{g(x)}$ to each sample x. In particle filters, f corresponds to the belief $bel(x_t)$ and g to the belief $bel(x_t)$" [4]. Figure 4.4 in [4]

5.3 Motion Model

The motion and the measurement models are the two remaining components for the implementation of the particle filter algorithm described thus far. The *motion model* comprises the state transition probability

$$p(x_t|u_t, x_{t-1}), \tag{5.16}$$

which is used in the particle filters prediction step. Corresponding to the motion model, the measurement model describe a probabilistic model of the sensor measurements

$$p(z_t|x_t), \tag{5.17}$$

which is essential for the particle filters measurement update. However, the motion model is the topic of the next chapter and will there be explained in detail[4].

5.3.1 Kinematic Configuration

The calculus, which describes the effect of control actions on the target objects configuration is called *Kinematics*. A *configuration* in the case of a rigid mobile object is described by six DOF. The three cartesian dimensions and the three Euler angles (roll, pitch, yaw) are relative to the global coordinate system. In this work the vehicles environment is considered as planar. Therefore, the kinematic state of the vehicle is described by only three variables, which defines the vehicles pose.
Figure 5.7 depicts a mobile robot operating in a planar environment. The

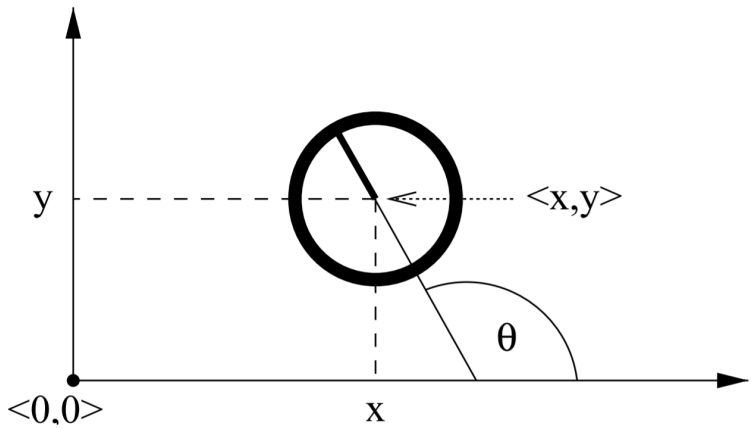

Figure 5.7: Robot pose relative to a global coordinate system. Figure 5.1 in [4]

pose consists of its two-dimensional planar coordinates (x, y) relative to the world coordinate frame, together with its angular orientation (θ). Therefore,

the pose is defined by

$$\begin{pmatrix} x \\ y \\ \theta \end{pmatrix}. \tag{5.18}$$

The orientation only is the so-called *bearing* or *heading direction*. In this definition, a vehicle with orientation $\theta = 0$ points into the direction of the x-axis and with orientation $\theta = \frac{1}{2}\pi$ points into the direction of the $y_a xis$, like shown in Fig. 5.7. Corresponding to the orientation only, a pose without orientation is called *location*. The fact that the vehicle's environment is considered as planar, leads to a description of the location by an only two-dimensional vector, which comprises the x and y coordinates of the vehicle[4]

$$\begin{pmatrix} x \\ y \end{pmatrix}. \tag{5.19}$$

5.3.2 Probabilistic Kinematics

The vehicle's motion is caused by uncertainties. Therefore the usage of an additional *probabilistic kinematics model* is necessary. In this case, the probabilistic kinematic model or motion model is equivalent to the motion model and plays the role of the state transition model. This model is the known conditional density

$$p(x_t|u_t, x_{t-1}) \tag{5.20}$$

To no coming into confusion, x_t and x_{t-1} are both robot poses and not just its x-coordinates. u_t is a motion command and is in this work the output of the ego-motion estimation from the previous chapter. This motion model describes the posterior distribution over kinematic states that a robot assumes when executing the motion command u_t at x_{t-1}.
An illustration of the kinematic model for a rigid mobile robot operating in a planar environment is, depicted in Fig 5.7 The robot's initial pose x_{t-1} is equal and the shaded area visualizes the conditional density $p(x_t|u_t, x_{t-1})$. In both cases, the darkness serves as a measure of the probability. The darker a pose, the more likely it is. The posterior pose probability is projected into x-y-space and the dimension of the robots orientation is missing in the depiction. In Fig. 5.7a, a robot moves only straight forward some distance and may collect translational and rotational errors as shown. Fig. 5.7b shows another example with a more complex motion command, which leads to a distribution with a larger spread of uncertainty[4].

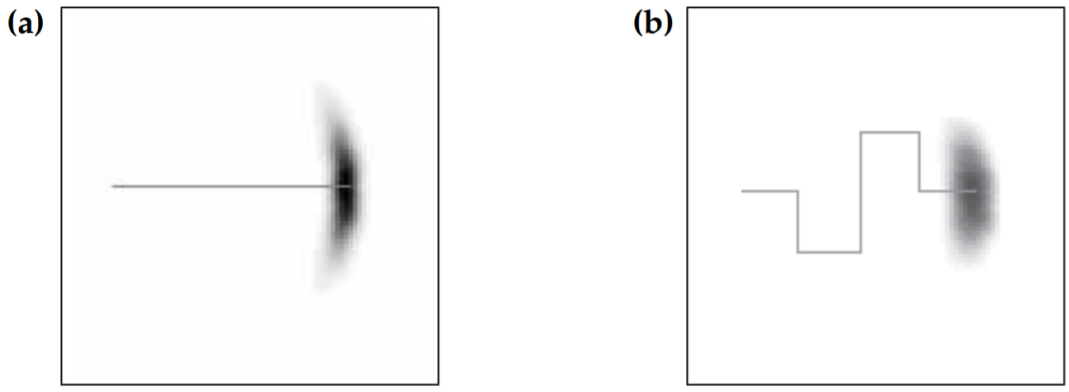

Figure 5.8: "The motion model: Robots trajectory illustrated by the solid line and posterior distributions of the robot's pose as shaded areas. The darker a location, the more likely it is. This depiction is only a projection in 2-D. The original density is three-dimensional, ad takes additional the robot's heading direction θ into account"[4]. Figure 5.2 in [4]

5.3.3 Velocity Model

The velocity motion model of this work assumes that the car can be controled through two velocities, a rotational and a translational velocity. The translational velocity at time t is denoted by v_t, and the rotational velocity by ω_t. This leads to

$$u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix} \tag{5.21}$$

It is also settled that positive rotational velocities ω_t correspond to a counter-clockwise rotation and positive translational velocities v_t to forward motion.

For particle filters, it suffices to sample only from the motion model $p(x_t|u_t, x_{t-1})$ instead of computing the posterior for x_t, u_t and x_{t-1}. Sampling from a distribution is different from calculating the distribution. For sampling, u_t and x_{t-1} are given, and their aim is to draw a random x_t from the motion model $p(x_t|u_t, x_{t-1})$. By calculating the density, x_t is also requested and has to be generated in a different way. Here the aim is to to compute the probability of x_t under $p(x_t|u_t, x_{t-1})$.
The sampling algorithm in Fig.5.9 is used to sample from the motion model. It generates random samples from the posterior distribution $p(x_t|u_t, x_{t-1})$ for a given pose x_{t-1} and control u_t. The algorithm requires x_{t-1} and u_t as input and outputs a random pose x_t drawn from $p(x_t|u_t, x_{t-1})$. More in detail, the algorithm adds noise to the control commands in lines 1-2, drawn from Gaussian normal distributions with zero mean and standard deviations of the corresponding error parameters of the kinematic motion model. The underlying ego-motion estimation of the motion model is only 2DOF, but it should be able to draw samples with translation in the lateral direction with-

Algorithm Sample_Motion_Model($\mathbf{u_t}$, $\mathbf{x_{t-1}}$)

1: $\hat{v} = v + \textbf{Sample_Random_Normal}(\sigma_v)$
2: $\hat{\omega} = \omega + \textbf{Sample_Random_Normal}(\sigma_\omega)$
3: $\hat{\gamma} = \textbf{Sample_Random_Normal}(\sigma_\gamma)$
4: $\theta' = \theta + \Delta t \hat{\omega}$
5: $x' = x + \Delta t \Big(\hat{v} \cos(\theta') - \hat{\gamma} \sin(\theta') \Big)$
6: $y' = y + \Delta t \Big(\hat{v} \sin(\theta') + \hat{\gamma} \cos(\theta') \Big)$
7: **return** $x_t = (x', y', \theta')^T$

Figure 5.9: Algorithm for sample poses $x_t = (x', y', \theta')^T$ from pose $x_{t-1} = (x', y', \theta')^T$ and control $u_t = (v, \omega)$ with additional motion noise from Gaussian distributions with zero mean and standard deviation of the corresponding noise parameter from $\sigma = (\sigma_v, \sigma_\omega, \sigma_\gamma)$

Algorithm Sample_Random_Normal(σ)

1: **return** $\frac{1}{2} \sum_{i=1}^{12} \textbf{Random_Int}(-\sigma, \sigma)$

Figure 5.10: Algorithm for sample numbers from an approximately normal distribution with zero mean and standard deviation σ [50]. **Rand_Int**(x,y) samples a number from a uniform distribution in $[x, y]$ like a pseudo random number generator, e.g. linear congruential generators [51].

out changing the yaw angle θ. That's the reason why in line 3 an additional noise value γ also sampled from a normal distribution with zero mean and an own hyperparameter for the standard deviation is created. In lines 4-6 the noisy command is then used to predict x_t from x_{t-1} and therefore creates a sample from $p(x_t | u_t, x_{t-1})$. Line 7 only returns the sampled pose x_t. Summarized, the sampling algorithm is an implementation of a simple physical robot motion model with integrated control noise in its prediction. The correctness of the motion model used from the sampling algorithm could be easily verified by studying a little bit about affine transformations and linear algebra [52][53].

5.4 Measurement Model

The last remaining component for the implementation of the particle filter is the *environment measurement model*. In the case of this work it describes the

process of formatting the measurements of the polarimetric radar sensors into the physical world by considering the noise in the sensor measurements. The model takes account of the uncertainties in the polarimetric radar sensors. The measurement model is formally defined by the conditional probability distribution

$$p(z_t|x_t, m), \tag{5.22}$$

where

- x_t is the robots pose at time t again,
- z_t is the measurement at time t,
- m is the map of the environment.

The polarimetric radar sensors measure, in general, more than one target. Therefore the number of measurement values that z_t contains will be denoted by K. Hence z_t can be written as

$$z_t = \{z_t^1, \ldots, z_t^K\} \tag{5.23}$$

z_t^K will be used to refer to an individual measured target (range, and azimuth angle). Through an independence assumption between the noise in each individual measurement, the product of the individual measurement likelihoods results in the probability $p(z_t|x_t, m)$ by

$$p\left(z_t \mid x_t, m\right) = \prod_{k=1}^{K} p\left(z_t^k \mid x_t, m\right). \tag{5.24}$$

This assumption is only true in the ideal case. There exist dependencies in general due to a a range of factors e.g. errors in the model m or approximations in the posterior. But for the sake of simplicity, violations of the independence assumption will not be considered yet.
For a good expression of the process of generating measurements, a specification of the environment in which a measurement is generated is needed. In the case of this work, a map of the environment is a list of objects in the environment with their features, the locations and covariance matrices. These objects are called *landmarks*, and their construction will be explained shortly. To easily handle the measurement model, a simple formalization of such a map m is helpful. So a map m is a list of objects in the environment together with their properties[4]

$$m = \{m_1, m_2, \ldots, m_N\}, \tag{5.25}$$

where

- N is the total number of objects in the environment,
- each m_n with $1 \leq n \leq N$ specifies the properties.

5.4.1 Sensor Model

In many robotics applications, features correspond to distinct objects in the physical world [4]. For example, in urban environments features may be traffic lights, street lamps, road signs, buildings, guard rails, curbstones, or bus and tram stops. In rural environments they may additional be tree trunks, guide posts, guard rails or bushes. Further such objects will be called *landmarks* to indicate that they are being used for the vehicle localization. The polarimetric radar sensors can measure the range and azimuth angle of the landmark relative to the vehicle's coordinate frame. With the usage of the introduced covariance gridmaps the polarimetric radars are also able to measure the polarimetric covariance matrix of a landmark. Denoting the range by r, the azimuth angle by θ, and the polarimetric covariance matrix by Σ, leads to the feature vector described by

$$f\left(z_t\right) = \left\{f_t^1, f_t^2, \ldots\right\} = \left\{\begin{pmatrix} r_t^1 \\ \theta_t^1 \\ \Sigma_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ \theta_t^2 \\ \Sigma_t^2 \end{pmatrix}, \ldots\right\} \tag{5.26}$$

The number of identified features at each time step is variable. However, like for the measurement z_t before it is assumed that there is conditional independence between the features

$$p\left(f\left(z_t\right) \mid x_t, m\right) = \prod_i p\left(r_t^i, \theta_t^i, \Sigma_t^i \mid x_t, m\right). \tag{5.27}$$

Conditional independence only applies if the noise in each individual measurement $\begin{pmatrix} r_t^i & \theta_t^i & \Sigma_t^i \end{pmatrix}^T$ is independent of the noise in other measurements $\begin{pmatrix} r_t^j & \theta_t^j & \Sigma_t^j \end{pmatrix}^T$ (for $i \neq j$). Under the conditional independence assumption it is much easier to develop an algorithm that implements a probabilistic measurement model. Now a sensor model for features can be devised. To recall, a map consists of a set of features, $m = \{m_1, m_2, ...\}$. Each feature comprises a location coordinate and a polarimetric covariance matrix. The location of a feature in the global coordinate frame is denoted by its coordinates $m_{i,x}$ and $m_{i,y}$. The measurement vector for a noise-free landmark sensor is in general defined by the standard geometric laws. The noise in the landmark perception will be modeled by zero-mean Gaussian noise on the range, azimuth angle, and the polarimetric covariance matrix. For the range and azimuth angle, the corresponding polarimetric radar sensors uncertainties are used as standard deviations, and for the covariance matrix a new hyperparameter is introduced. The resulting measurement model is defined for the case where the i-th feature at time t corresponds to the j-th landmark in the map. As a reminder, the vehicle's pose is given by

$x_t = \begin{pmatrix} x & y & \theta \end{pmatrix}^T$

$$\begin{pmatrix} r_t^i \\ \theta_t^i \\ \Sigma_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{\left(m_{j,x} - x\right)^2 + \left(m_{j,y} - y\right)^2} \\ \operatorname{atan2}\left(m_{j,y} - y, m_{j,x} - x\right) - \theta \\ \Sigma_j \end{pmatrix} + \begin{pmatrix} \varepsilon_{\sigma_r^2} \\ \varepsilon_{\sigma_\theta^2} \\ \varepsilon_{\sigma_\Sigma^2} \end{pmatrix} \tag{5.28}$$

Here $\varepsilon_{\sigma_r}, \varepsilon_{\sigma_\theta}$, and ε_{σ_s} are the said zero-mean Gaussian error variables with the corresponding standard deviations σ_r, σ_ϕ, and σ_s[4].

5.4.2 Landmark Measurements

The used landmarks in this work come from a previous work of Weishaupt et al. [54]. This approach, is to the best of the author's knowledge, the first approach that uses polarimetric information to extract landmarks. The advantage of using additional scattering information provided by polarimetric measurements, instead of only using amplitude or occurrence information like in [55] and [56], is shown in Fig 5.11. The resulting first Eigenmap of the polarimetric gridmap is depicted in the upper picture of Fig. 5.11 and an equivalent non-polarimetric representation in the middle of Fig. 5.11. Potential landmark candidates, extracted by the proposed algorithm, are encircled. Without polarimetry, there are only a few of the fence posts which are detected. In the first eigenmap of the polarimetric gridmap, one can see an additional even-bounce scattering at the post positions, which allows the algorithm to find all of them.
In general, the CFAR algorithm is used on PreCFAR covariance gridmaps with the Wishart distance as a distance measure to detect interesting areas. More precisely, the covariance matrix can be assumed to be complex Wishart distributed. As a derivation from a Bayes maximum likelihood classification, the used distance measure is described by

$$d_{\text{wishart}}\left(\mathbf{C}, \mathbf{\Sigma}\right) = \ln|\mathbf{\Sigma}| + \operatorname{tr}\left(\mathbf{\Sigma}^{-1}\mathbf{C}\right), \tag{5.29}$$

where

- $\boldsymbol{C}$ is the cell under test,
- $\mathbf{\Sigma}$ is the neighborhood covariance matrix,
- $|\boldsymbol{X}|$ is again the determinant of matrix $|\boldsymbol{X}|$,
- $\operatorname{tr}(\boldsymbol{X})$ is again, the trace of matrix $\boldsymbol{X}$.

Fig. 5.12 shows how the algorithm works. The cell under test $\boldsymbol{C}$ is compared to its neighborhood with the Wishart distance. The neighborhood covariance matrix $\mathbf{\Sigma}$ for a cell under test $\boldsymbol{C}$ is defined as a spatial averaging over the neighborhood's covariances. If the cell under test is of a larger amplitude or

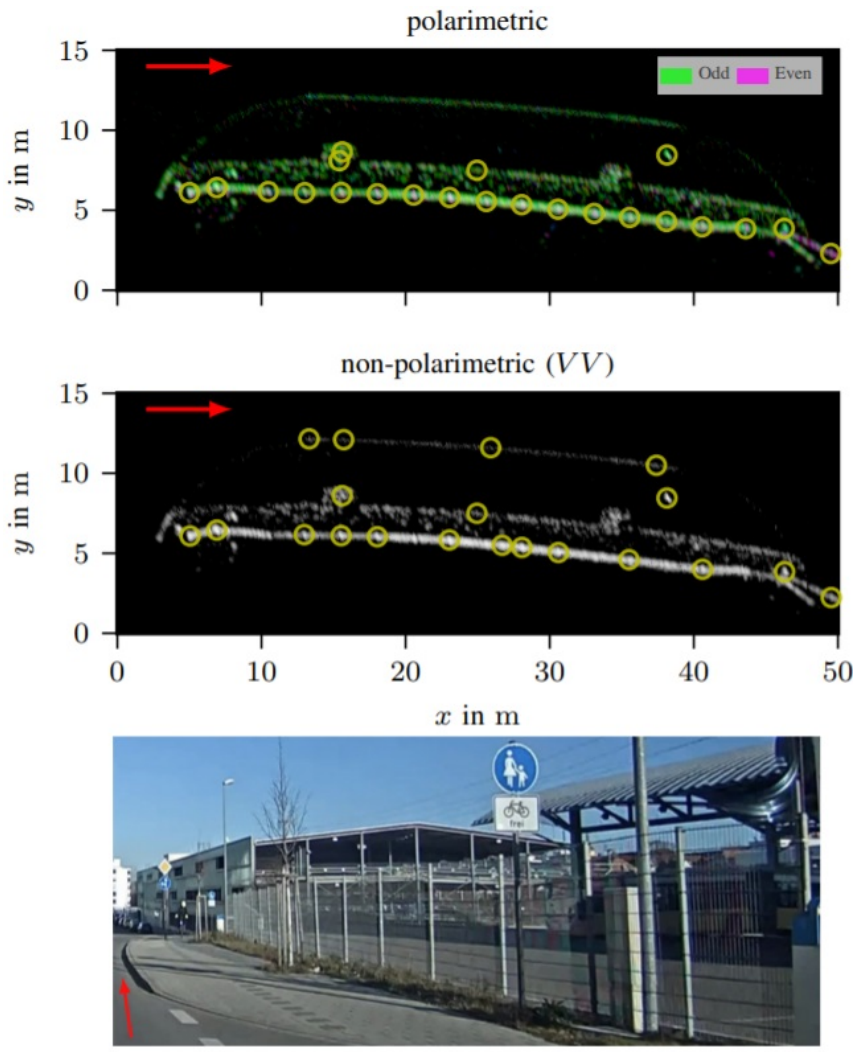

Figure 5.11: "Comparing the extractions of potential landmarks from full polarimetric (top) and non-polarimetric data (middle) shows the benefit of utilizing polarimetry. Landmark candidates extracted by the proposed algorithm are encircled in yellow. All fence posts (compare to photograph at the bottom) are only found with the help of polarimetric information. The red arrow indicates the driving vector."[54] Fig 1 in [54]

shows considerably different scattering behavior relative to the neighborhood, the distance indicates that by a large value, and the cell is therefore considered as an interesting area. The status from an interesting area to a landmark is raised, if the interesting area occurs in 3 of 4 different drives in the same environment[54].

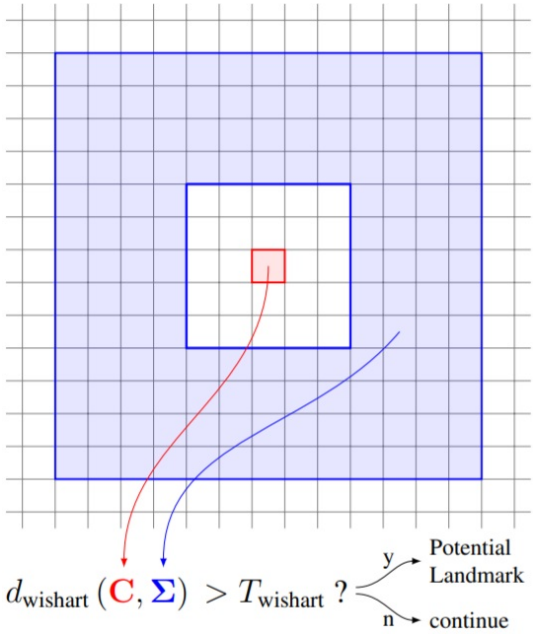

Figure 5.12: Algorithm for landmark detection, Fig. 2 in [54]

5.4.3 Sensor Model with Correspondences

As a result of the previously introduced approach, a map m of landmarks for the whole dataset is available, and with the same approach, the vehicle is able to produce interesting areas in every step of the localization process. But for using the produced interesting areas in the measurement model, the corresponding landmarks have to be identified. This leads to the so-called *data association problem*, which is a key problem of range/bearing sensors, like radar sensors. Due to some uncertainties with regards to the identity of a landmark, the problem occurs if landmarks cannot be uniquely identified. For the development of a sensor model, a correspondence variable between the measured interesting area, further only called feature f_t^i and the landmark variable m_j in the map. The variable is defined by c_i with $t \in \{1, \ldots, N+1\}$. N is the number of landmarks in the map m again. The observed feature f_t at time t corresponds to the $j-th$ landmark in the map if $c_t^i = j$ with $j \leq N$. An observed feature have no corresponding to a feature of the landmark if $c_t^t = N+1$. This case is really important for a good localization result because not every measured interesting area corresponds to a landmark of the map [4]. The algorithm for computing the conditional probability of a feature f_t^i with known correspondence $c_t^i \leq N$, given pose x_t and map m is shown in Fig. 5.13. In lines 3 and 4 the range and azimuth angle of the corresponding landmark relative to the vehicles pose x_t are calculated. Assuming independence in the noise, the probability of the measured ranges and bearing is then calculated in line 5.

Algorithm Landmark_Model_Correspondence(f_t^i, c_t^i, x_t, m)

1: $j = c_t^i$
2: $\hat{r} = \sqrt{\left(m_{j,x} - x\right)^2 + \left(m_{j,y} - y\right)^2}$
3: $\hat{\theta} = \operatorname{atan2}\left(m_{j,y} - y, m_{j,x} - x\right)$
4: $q =$ **Prob_Normal** $\left(r_t^i - \hat{r}, \sigma_r\right) \cdot$ **Prob_Normal** $\left(\theta_t^i - \hat{\theta}, \sigma_\theta\right)$
5: **return** q

Figure 5.13: Algorithm for calculating the likelihood of a landmark measurement. The algorithm requires an observed feature $f_t^i = \left(r_t^i \theta_t^i s_t^i\right)^T$, the correspondence variable c_t^i, the robot pose $x_{t-1} = (x \quad y \quad \theta)^T$ and the map m as input. Its output is the conditional probability $p(f_t^i | c_t^i, m, x_t)$. [4]

Algorithm Prob_Normal(a, b)

1: **return** $\frac{1}{\sqrt{2\pi b^2}} \exp\left\{-\frac{1}{2}\frac{a^2}{b^2}\right\}$

Figure 5.14: Algorithm for computing the probability from a of a zero-centered Gaussian distribution with standard deviation b

This algorithm is the implementation of Equation (5.28) without considering the polarimetric covariance matrix. This is because it only makes sense to use the polarimetric information in the correspondence determination in the actual implementation. The reason why is explained shortly.

5.4.4 Correspondence Determination

The determination of the corresponding features and landmarks is in this work done by the so-called *iterative closest point algorithm* (ICP) based on a non linear Least-squares fit[57]. Before the ICP could be applied, the features have to be transformed from polar to Cartesian coordinates by the algorithm in Fig.5.15 For time t pose x_{t-1} is used. Let $f_{xy}(z_t)$ be the relative to x_{t-1} transformed feature set in Cartesian coordinates in the global coordinate system.
Now there are two cartesian point sets the interesting areas $f_{xy}(z_t)$ and the landmarks m. The goal of the ICP is to find a transformation (rotation **R** and translation **t**) to apply to $f_{xy}(z_t)$ to match m as well as possible. In the first step of the ICP, a first version of the correspondences c_t has to be calculated. For this, there are two simple Algorithms available. The first

Algorithm Polar_To_Cartesian(f_t^i, x_t)

1: $x_f = x + r_t^i \cdot \cos(\theta_t^i + \theta)$
2: $y_f = y + r_t^i \cdot \sin(\theta_t^i + \theta)$
3: **return** $(x_f \quad y_f \quad \Sigma_f)^T$

Figure 5.15: Algorithm for transforming f_t^i relative to the pose x_t from polar to Cartesian coordinates in the global coordinate system.

one, depicted in Fig. 5.16 only uses the Euclidian distance to determine the correspondence. The second one, shown in Fig. 5.17, uses the product of the Euclidian distance and the Wishart distance to identify the correspondence.

Algorithm Correspondence_By_Distance($f_{xy}(z_t)_i$, m)

1: $(x_f \quad y_f \quad _f) = f_{xy}(z_t)_i$
2: $c_t^i = \underset{j}{\operatorname{argmin}} \sqrt{\left((x_f - m_{j,x})^2 + (y_f - m_{j,y})^2\right)}$
3: **return** c_t^i

Figure 5.16: Simple Algorithm for calculating c_t^i by using the minimum Euclidian distance

Algorithm Correspondence_By_Distance_Polarization($f_{xy}(z_t)_i$, m)

1: $(x_f \quad y_f \quad \Sigma_f) = f_{xy}(z_t)_i$
2: $c_t^i = \underset{j}{\operatorname{argmin}} \sqrt{\left((x_f - m_{j,x})^2 + (y_f - m_{j,y})^2\right)} \cdot d_{wishart}(\Sigma_f, m_{j,\Sigma})$
3: **return** c_t^i

Figure 5.17: Simple Algorithm for calculating c_t^i by using the minimum product of Euclidian distance and Wishart distance

Every iteration of the ICP will be treated as a least squares minimization problem. The function to minimize is the squared sum of distances between the interesting areas and the landmarks [57]

$$E = \sum_i \left[\mathbf{R} f_{xy}(z_t)_i + \mathbf{t} - m_{c_t^i}\right]^2 \rightarrow \min . \tag{5.30}$$

In each iteration, pose $\mathbf{R}, \mathbf{t}$ will be updated to how it is needed to transform $f_{xy}(z_t)$ to m. In the matching process, the ICP finds the pose $\mathbf{x}_{icp} = [x_{icp}, y_{icp}, \theta_{icp}]^T$ that minimizes the sum of squared lengths of the correspondences. The representation with a rotation matrix $\mathbf{R} = \begin{bmatrix} \cos\theta_{icp} & -\sin\theta_{icp} \\ \sin\theta_{icp} & \cos\theta_{icp} \end{bmatrix}$ and a translation vector $\mathbf{t} = [x_{icp}, y_{icp}]$. Further the notation

$$\mathbf{h}_i(\mathbf{x}_{icp}) = \mathbf{R}_\theta f_{xy}(z_t)_i + \mathbf{t} \tag{5.31}$$

is used and the error function is defined as

$$\begin{aligned} \mathbf{e}(\mathbf{x}_{icp}) &= \sum_{\{i,j\}\in C} \mathbf{e}_{i,j}(\mathbf{x}_{icp}) \\ \mathbf{e}_{i,j}(\mathbf{x}_{icp}) &= \mathbf{h}_i(\mathbf{x}_{icp}) - m_j = \mathbf{R}_\theta f_{xy}(z_t)_i + \mathbf{t} - m_{c_t^i}. \end{aligned} \tag{5.32}$$

Therefore, the minimization problem can be described by

$$\begin{aligned} \mathbf{x}_{icp}^{\text{query}} &= \arg\min_{\mathbf{x}_{icp}} \{\mathbf{E}(\mathbf{x}_{icp})\} \\ &= \arg\min_{\mathbf{x}_{icp}} \left\{ \sum_i \left\| \mathbf{e}_{i,c_t^i}(\mathbf{x}_{icp}) \right\|^2 \right\} \\ &= \arg\min_{\mathbf{x}_{icp}} \left\{ \sum_i \left\| \mathbf{h}_i(\mathbf{x}_{icp}) - m_{c_t^i} \right\|^2 \right\} \end{aligned} \tag{5.33}$$

For solving the non linear minimization problem the Gauss-Newton method is used. Therefore the function is linearized in the vicinity of $\mathbf{x}_{icp}$. Solving non-linear least squares is equivalent to solving[57]

$$\mathbf{H}\Delta\mathbf{x}_{icp} = -\mathbf{E}'(\mathbf{x}_{icp}), \tag{5.34}$$

where

- where $\Delta\mathbf{x_{icp}}$ is the increment of the argument $([\Delta x_{icp}, \Delta y_{icp}, \Delta\theta_{icp}]$ in this case),
- $\mathbf{H}$ is the Hessian of $\mathbf{E}$,
- $\mathbf{E}'(\mathbf{x}_{icp})$ is the derivative over the function that should be minimized.

The gradient is computed $\mathbf{E}'(\mathbf{x}_{icp})$ as follows:

$$\mathbf{E}'(\mathbf{x}_{icp}) = \mathbf{J}(\mathbf{x}_{icp})\mathbf{e}(\mathbf{x}_{icp}) \tag{5.35}$$

In Gauss-Newton method the function is linearized around the considered point, which allows computing the Hessian as

$$\mathbf{H}_{icp} = \mathbf{J}(\mathbf{x}_{icp})^T\mathbf{J}(\mathbf{x}_{icp}) \tag{5.36}$$

The Hessian and the gradient need the computation of a Jacobian. For computing the Jacobian, the derivative of the rotation matrix is required

$$\mathbf{R}'_{\theta_{icp}} = \frac{\partial}{\partial \theta_{icp}} \begin{bmatrix} \cos\theta_{icp} & -\sin\theta_{icp} \\ \sin\theta_{icp} & \cos\theta_{icp} \end{bmatrix} = \begin{bmatrix} -\sin\theta_{icp} & -\cos\theta_{icp} \\ \cos\theta_{icp} & -\sin\theta_{icp} \end{bmatrix}. \tag{5.37}$$

The Jacobian $\mathbf{J}$ can now be calculated by

$$\begin{aligned} \mathbf{J} = \frac{\partial \mathbf{e}_{i,j}(\mathbf{x}_{icp})}{\partial \mathbf{x}_{icp}} = \frac{\partial \mathbf{h}_i(\mathbf{x}_{icp})}{\partial \mathbf{x}_{icp}} &= \left(\frac{\partial \mathbf{h}_i(\mathbf{x}_{icp})}{\partial x_{icp}}, \frac{\partial \mathbf{h}_i(\mathbf{x}_{icp})}{\partial y_{icp}}, \frac{\partial \mathbf{h}_i(\mathbf{x}_{icp})}{\partial \theta_{icp}} \right) \\ &= \left(\mathbf{I}, \mathbf{R}'_\theta f_{xy}(z_t)_i \right) \\ &= \begin{bmatrix} 1 & 0 & -\sin\theta_{icp} f_{xy}(z_t)_{i,x} - \cos\theta_{icp} f_{xy}(z_t)_{i,y} \\ 0 & 1 & \cos\theta_{icp} f_{xy}(z_t)_{i,x} - \sin\theta_{icp} f_{xy}(z_t)_{i,y} \end{bmatrix} \end{aligned} \tag{5.38}$$

With the help of the Jacobian, the system of equations can be calculated and solved, which delivers the solution to the problem. First, the Hessian $\mathbf{H}$ and gradient $\mathbf{g}$ will be initialized by zeros

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.39}$$

For every corresponding pair of points do

$$\begin{aligned} \mathbf{H} &\rightarrow \mathbf{H} + \mathbf{J}^T \mathbf{J} \\ \mathbf{g} &\rightarrow \mathbf{g} + \mathbf{J}^T \mathbf{e} \end{aligned}, \tag{5.40}$$

and find the solution $-\Delta \mathbf{x}_{icp}$ to the least squares problem

$$\mathbf{H}\Delta\mathbf{x} = -\mathbf{g} \Longrightarrow \Delta\mathbf{x} = -\mathbf{H}^{-1}\mathbf{g} \tag{5.41}$$

Finally the correspondences c_t^i can be obtained by the iterative process which is provided by the algorithm in Fig[57]. 5.18.

5.4.5 Straight line features

The proposed measurement model is only able to handle point-shaped features like traffic signs, guide posts, or traffic signs. But there are also objects which are measured as straight lines, like buildings or curbstones. The extraction of *straight line features* (slf) is beyond the scope of this work and will not be further explained. But a map $ms \subset m$ of slfs is available, and they are also used in the localization process. So, for now, it is only to say that they can be extracted from a polarimetric covariance gridmap. Such slf fs could be mathematically described by 2 points $p_1 = (x_{p_1} \quad y_{p_1})$ and

Algorithm Correspondences_ICP($f_{xy}(z_t)$, m)

1: Find correspondences c_t^i by using algorithms in Fig. 5.16 or 5.17
2: Perform a single iteration by computing the least square fit for $f_{xy}(z_t)$ and m
3: Apply the found rotation to $f_{xy}(z_t)$
4: Repeat until correspondences don't change
5: **return** c_t^i

Figure 5.18: Algorithm for calculating c_t^i by using the ICP based on Least-squares.

$p_2 = (x_{p_2} \quad y_{p_2})$ in the Cartesian coordinate system. Therefore the feature vector for straight lines can be written as

$$fs\left(z_t\right) = \left\{fs_t^1, fs_t^2, \ldots\right\} = \left\{\begin{pmatrix} p_{1,t}^1 \\ p_{2,t}^1 \end{pmatrix}, \begin{pmatrix} p_{1,t}^2 \\ p_{2,t}^2 \end{pmatrix}, \ldots\right\}. \tag{5.42}$$

The number of identified slfs at each time step is also variable and here again, it is assumed that there is conditional independence between the slfs

$$p\left(fs\left(z_t\right) \mid x_t, ms\right) = \prod p\left(p_{1,t}^i, p_{2,t}^i \mid x_t, ms\right). \tag{5.43}$$

The association process of slfs is much more complicated than for point features. Hence, a proper straight distance metric is needed, which uses the position information for association. For this, the following requirements and the subsequent approach of [58] are used.
"For two input straight segments fs^i, fs^j it should be sensitive to both, differences in cross-distance (i. e. translations normal to the straight axes) and in orientation of the two slfs, ideally accounting for both of them linearly. Furthermore, it should not account for differences in length of the two slfs and not for translations along the slf axes directions as long as some overlap of the two slfs remains. These two requests for leniency regarding slf lengths and translations along the slf axes both stem from the application of associating a currently observed slf with a known slf ms_k. While the measured slf may only be a small currently observed segment, the slf from the map may be in fact a much larger structure
In order to fulfill these constraints, we propose a straight distance metric between the two straight input segments fs_t^i, fs_t^j defined by the four edge points p_1^i, p_2^i, p_1^j, and p_2^j consisting of two parts:

$$d_{\text{geom}} = d_{\text{convHull}} + d_{\text{proj}}, \tag{5.44}$$

First, the convex hull distance

$$d_{\text{convHull}} = \frac{A_{\text{convhull}}}{\text{mean}\left(\|fs^i\|, \|fs^j\|\right)} \tag{5.45}$$

as the fraction of the area of the convex hull of the four involved straight segment edge points $A_{convHull}$ divided by the mean length of the two input segments. And second, the projection distance d_{proj}.

$$\begin{aligned} d_{\text{proj}} = \min[&\text{proj}\left(p_1^i, fs^j\right), \text{proj}\left(p_2^i, fs^j\right) \\ &\text{proj}\left(p_1^j, fs^i\right), \text{proj}\left(p_2^j, fs^j\right)] \end{aligned} \tag{5.46}$$

with the function $\text{proj}(p_k, fs^i)$ returning the distance of the projection of p_k onto the underlying straight of fs^i to the closest point of fs^i."[58]
Not every measured slf corresponds to a slf landmark. For this, an algo-

Algorithm Correspondence_Straight_Lines(fs_t^i, ms)

1: $j = \arg\min_j \left\{ d_{\text{convHull}}\left(fs_t^i, ms_j\right) + d_{\text{proj}}(fs_t^i, ms_j) \right\}$
2: c_t^i= **if** ($d_{\text{convHull}} < T_{\text{convHull}}$ && $d_{\text{proj}} < T_{\text{proj}}$) **else** -1
3: **return** c_t^i

Figure 5.19: Algorithm for calculating the correspondence c_t^i of fs_t^i.

Algorithm Landmark_Model_Straight_Lines($fsra_t^i$, ms, x_t, c_t^i)

1: $\hat{fs}_t^i$ = **Polar_To_Cartesian**($fsra_t^i, x_t$)
2: $d_{\text{geom}} = d_{\text{convHull}}\left(fs_t^i, c_j^i\right) + d_{\text{proj}}(fs_t^i, ms_{c_j^i})$
3: q = **Prob_Normal**(d_{geom}, σ_s)
4: **return** q

Figure 5.20: Algorithm for calculating the likelihood of a slf landmark measurement. The algorithm requires an observed slf feature $fsra_t^i$ in polar coordinates, a pose $x_t = (x \quad y \quad \theta)^T$, the correspondence variable c_t^i and the map ms as input. Its output is the conditional probability $p(fsra_t^i | ms, x_t, c_t^i)$.

rithm is needed to filter slfs without correspondence and to find the correct matches. For slft this is much simpler than for the point features. The used

algorithm is depicted in Fig. 5.20. The algorithm chooses the slf landmark with the minimum sum of the convex hull distance and the projection distance to the slf fs_t^i as correspondence if both single distances are lower than the corresponding thresholds d_{convHull} and d_{proj}, else $c_t^i = -1$, which means there is no corresponding landmark for the slf.
In order to compute the likelihood of a slf landmark measurement, the algorithm in Fig. 5.19 is used. In general the measurements of the polarimetric radars are in polar coordinates and have to be transformed with a given pose x_t into cartesian coordinates to get a slf $\hat{fs}_i^i$. This is what is done in Line 1. After that, the distance d_{geom} is calculated in Line 2, which is then used to calculate the likelihood with the help of an also zero centered Gaussian distribution with a hyperparameter σ_s as standard deviation.

5.5 Monte Carlo Localization

1: **Algorithm MCL($\mathcal{X}_{t-1}, u_t, z_t, m$):**
2: $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
3: for $m = 1$ to M do
4: $x_t^{[m]} = \textbf{sample_motion_model}(u_t, x_{t-1}^{[m]})$
5: $w_t^{[m]} = \textbf{measurement_model}(z_t, x_t^{[m]}, m)$
6: $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
7: endfor
8: for $m = 1$ to M do
9: draw i with probability $\propto w_t^{[i]}$
10: add $x_t^{[i]}$ to $\mathcal{X}_t$
11: endfor
12: return $\mathcal{X}_t$

Figure 5.21: Monte Carlo Localization Algorithm based on particle filters. Table 8.2 in [4]

To finally perform the localization an algorithm called *Monte Carlo Localization* (MCL)[4], which represents the belief $bel(x_t$ by particles, is applied. Despite its youth, MCL is today one of the most popular localization algorithms in robotics. It is relatively easy to implement and works well for most localization problems. The big advantage of the MCL is that it is able to approximate almost any distribution of practical importance. It is not limited to a parametric subset of distributions, as like the also very popular localization approach of the extended Kalman filter[59]. However, Fig 5.21 shows the MCL algorithm, which is a modification of the particle filter

algorithm. Instead of sampling from $p(x_t|u_t, x_{t-1}^{[m]}$ and choosing $p(z_t|x_t^{[m]}$ as in the particle filter, the MCL algorithm sample from the probabilistic motion model and calculates the weights by using a combination of all parts of the measurement model. In general the MCL algorithm represents the belief $bel(xt)$ by a set of M particles

$$\mathcal{X}_t = \left\{ x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]} \right\}. \tag{5.47}$$

At the start point $t = 0$ all particles in $\mathcal{X}_0$ are equal to the initial vehicle pose $x_0 = \{x, y, \theta\}$. In each iteration the algorithm samples in line 4 from the motion model, using particles from the previous belief as starting points. As a short reminder the used control u_t in the motion model comes from the ego-motion estimation. After that the measurement model is then applied in line 5 to determine the importance weights. Here, first the weights for point features and slfs are calculated separately and are then multiplicated to get the final weights. The real magic of the MCL algorithm comes from Lines 8-10, where a new particle set is drawn from the sampled set from the motion model with the probability of the corresponding importance weights of each particle. The predicted position is defined at each step as the weighted average pose from all poses of the particle set. In Fig. 5.22, a

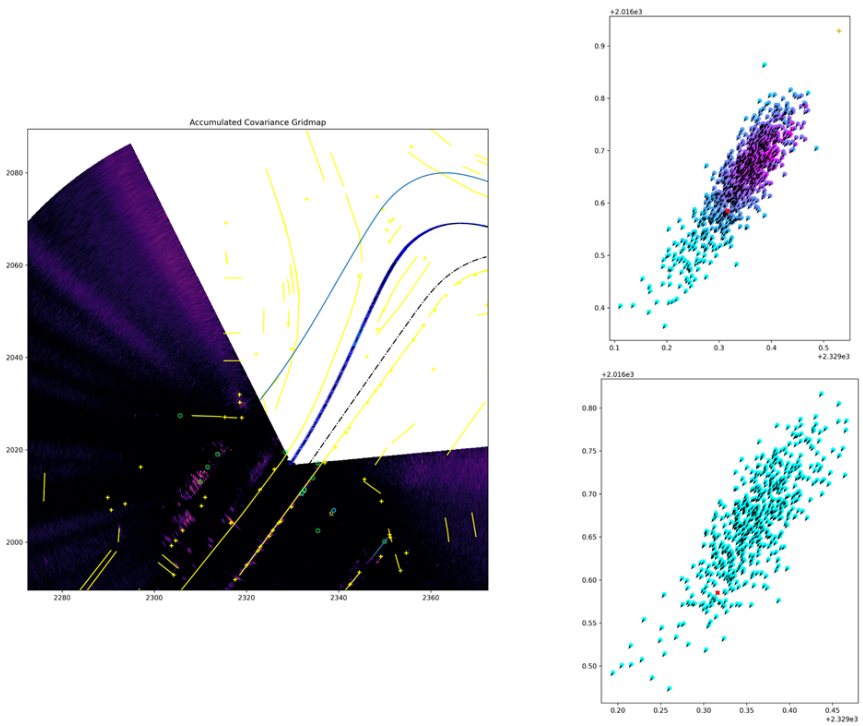

Figure 5.22: Localization example from the dataset of the city Ulm. With a visualization of the localization map on the left and the corresponding particle sets on the right with the colormap indicating the weights of the particles.

localization example from the dataset of Ulm is depicted. The left image shows an accumulated covariance gridmap for all three polarimetric radar sensors. The yellow lines and crosses are the corresponding landmarks of the map for the depicted extend. The green circles are the measured point shape features. The corresponding translated and rotated points of the ICP are the grey crosses. A green circle which is filled blue indicates that it is applied for the importance weights calculation, with the corresponding landmark, which is connected with a black line to the point feature. An only green circle depicts a measured point feature without correspondence. The small blue lines with depicted convex hull and projection distances indicate a slfs with corresponding slf landmarks. Here the green lines between the center of the slf landmark and the slf show the correspondence. An slf landmark without depicted convex hull distance and projection distance shows a slf without correspondence. The black dashed curved lines are the ground truth trajectory of the car, one for the drive in the front direction and the other for the back. The colorful curved line is the predicted trajectory of the car, with red indicating the biggest relative error and blue the smallest. The blue curved line is the trajectory of the ego-motion only. The error of the ego-motion here is not that big here because the picture is from about 30seconds after the start. The two images on the right of Fig. 5.22 show the corresponding particle sets before and after the resampling step. The color map indicates the importance weights of the particles. The red point in the pictures is the ground truth position and the yellow cross in the bottom image is the predicted position. It could be seen that the particles around the ground truth position have a higher importance weight in general. After the resampling step the range in the lateral and longitudinal direction of the particles shrinks, and the prediction is about 4cm next to the ground truth position.

Fig. 5.23 shows another example from the dataset of the city Ulm. In difference to the previous example, the accumulated covariance gridmap is separated from the rest of the environment. This is because Fig. 5.22 shows an older version of the localization algorithm, in which the covariance gridmap is accumulated over time by using the predicted pose of the vehicle. But while doing this, it is only possible to accumulate the covariance gridmap over a fixed time and then apply the localization only once for an accumulated covariance gridmap, discard it and then repeat. This is because while correcting the pose of the vehicle, the error in the gridmap is still remaining. In the final version in Fig. 5.23 the covariance gridmap is accumulated over the time relative to the pose of the ego-motion estimation. The extracted features will then be used relative to the predicted pose of the vehicle. For this, a sliding window approach for the accumulation is used so, that only the last measurements of a fixed size are accumulated in the covariance gridmap. Despite the completely wrong position of the

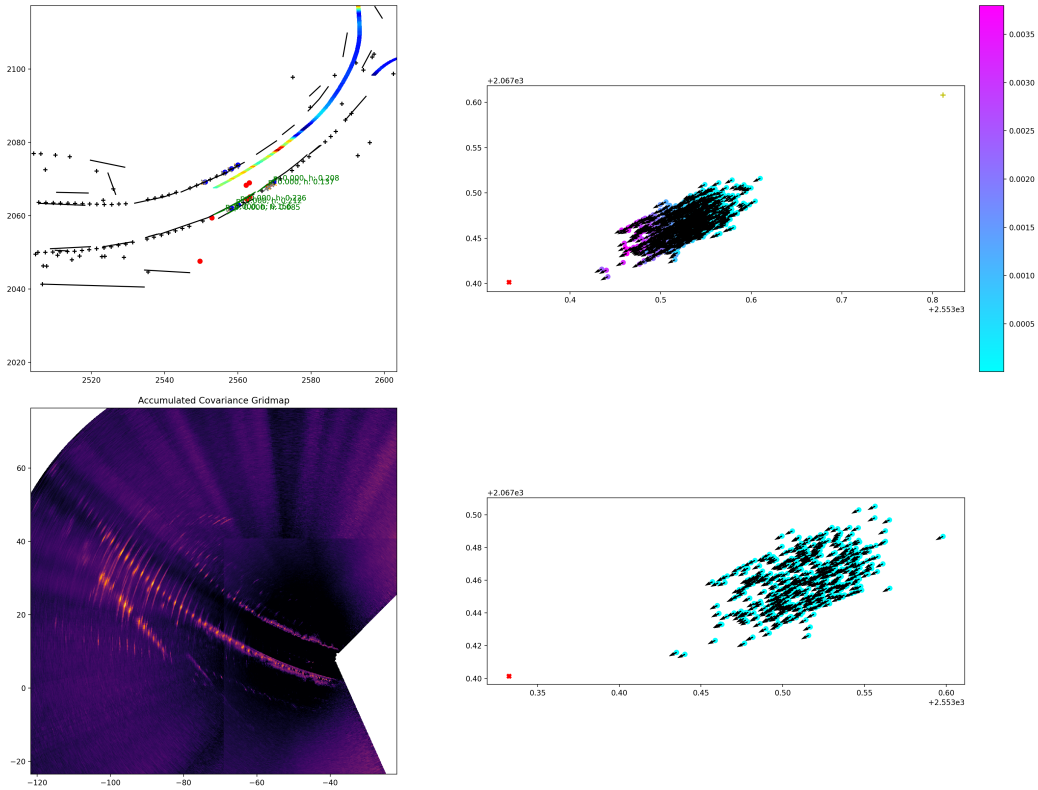

Figure 5.23: Localization example from the dataset of the city Ulm, with separated covariance gridmap on the bottom left image and the trajectories and features on the top. The corresponding particle sets are on the right with the colormap indicating the weights of the particles.

egomotion estimation, the sliding window approach leads to a covariance gridmap which is only affected by the egomotion estimations error from the steps inside the sliding window and is never affected by any update step of the localization.

5.6 Validation and Results

In the further validation, the number of particles is fix at 1000, because the computing time isn't that high, and the error seems not to decrease noticeable using a higher number. To determine the right accumulation time of the covariance gridmap for the localization. The error of the ego-motion estimation is first investigated. For this, the whole dataset of Ulm is used and the errors for different integration times between 1-5 seconds are chosen. In Fig.5.24 the result from this investigation is shown in the form of the complementary cumulative distributions (ccdfs) of the absolute position error for all different integration times. The result is obvious; a larger accu-

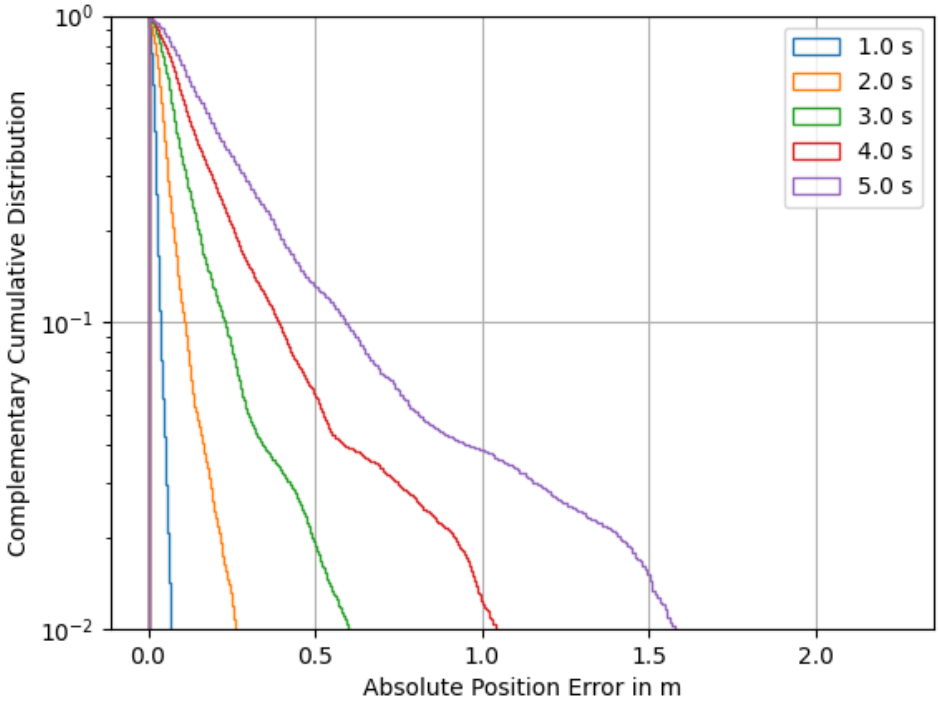

Figure 5.24: CCDFs of absolute position error of the ego-motion estimation on the whole dataset from Ulm for different integration times.

mulation time leads to a larger error and the with exponential growth. In 90% of the cases the error is for all values except 5s definitely under 50cm, and for 5s it is about 60cm. By integrating 1s, the error is in 99% of the cases under about 10cm for 1 second, 25cm for 2s, 60cm for 3s, over 1m for 4 and over 1.5m for 6s. To ensure a good localization, the error has to be as small as possible by accumulating as much as possible polarimetric radar measurements to provide better covariance gridmaps for the localization algorithm. The maximum error should definitely be under 50cm and it would be much better if it could be under 25cm. Therefore, further only the accumulation times under 3s are considered for the localization.

Fig. 5.25shows the CCDFs regarding the absolute position errors for different accumulation times for a small bag of the dataset from the city ULM. The full dataset has about 21.000 frames and the small about 10% of it. It could be seen that at the 50 percentile, they all not differ that much. But at the 95 percentile they part. The result is not that easy to interpret, but sorted in ascending order in 99% of the cases the absolute position error of 0.5s is under 20cm, for 2s under 25, for 1.5s under about 30cm, for 1s and 2.5s by about 30cm. In order to choose the right accumulation time the ccdf from the egomotion estimation as well as from the localization has to be

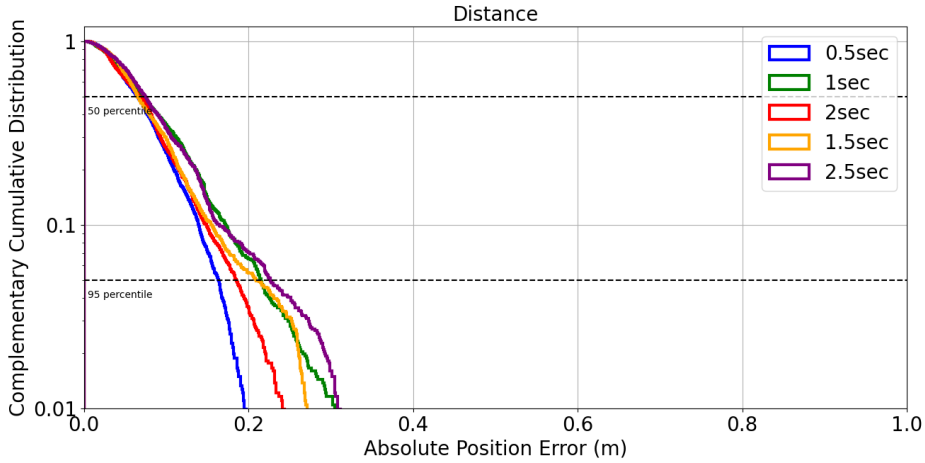

Figure 5.25: CCDFs of absolute position error of different accumulation times of the localization on the small bag of the city Ulm.

considered. From this point only 0.5s and 2s should be further considered. But including the fact, that with more accumulation time the quality of the covariance gridmaps increases, 2s is chosen in the following investigations. The main topic of this work is the investigation of the advantages of using

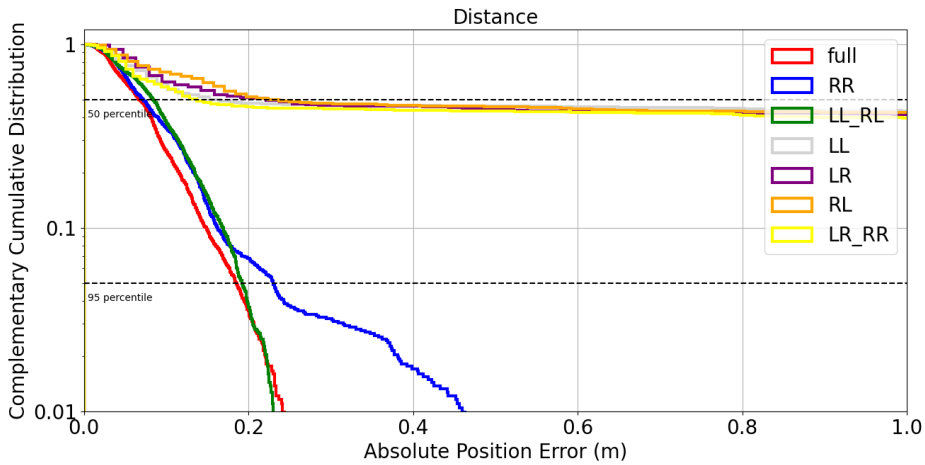

Figure 5.26: CCDFs of using different single, dual and full polarization channels of the localization on the small bag of the city Ulm.

additional polarimetric information in a radar-based localization. Hence, the usage of using polarization has to be validated. In cases before the full polarization is used, what means that all polarimetric radars send out and receive left circular and right circular polarized electromagnetic waves. Instead of using the full polarimetric scattering matrix, it seemed to be sin full to validate the 2 dual scattering behaviours, in which one sends out

left and right circular electromagnetic waves and receives only left circular electromagnetic waves (short: LL_RL). Corresponding to the first one, the second one also sends out left and right circular electromagnetic waves and receives only the right circular electromagnetic waves (short: LL_RL). This makes sense because an electromagnetic wave's polarization changes from left to right and from right to left by scattering from an object. Therefore, odd and even bounce scattering behaviour can be measured with the introduced dual scatterings behaviours, but they only receive left or right circular polarized electromagnetic waves. In addition to using the full polarimetric scattering matrix, or only the dual scattering it makes sense to validate single scattering behaviour. The odd bounce channels, which send out left and receive right and reverse (short LR and RL). In the left and right-handed circular bases, those channels are equal to conventional radars without polarization. Last but not least the even bounce channels, which send out the same polarization as they receive (LL and RR). Fig. 5.26 shows the ccdfs of the previously explained polarimetric scattering behaviours. All ccdfs, which go in the right direction instead of the bottom, lost the localization and at some point and never got it back. From this point it could be easily verified that all these polarimetric scattering behaviours failed. The only 3 remaining, are full, LL_RL, and RR. RR seems to work the worst and has a big outlier with about 50cm, while the others have a maximum absolute position error of about 25cm. The ccdf of the version using the full polarimetric scattering matrix is a litte bit steeper than LL_RL. Therefore, using the full polarization scattering matrix seems to be the best. In comparison with using the single polarizations LR or RL, which are equal to conventional radars, using the full polarization brings great added value.

Because of the possibility of errors in the given start position or errors during the localization caused by mistakes in the measurement or the ego-motion estimation, the approaches ability to convergence by a range of errors in the lateral and longitudinal direction is tested. To evaluate the convergence, 50 randomly chosen situations of the full dataset from the city Ulm are tested. Doing this the localization is applied with start positions additional errors of all combinations of lateral and longitudinal errors in the range of [-0.5m, 0.5m]. A test will be classified as *converged* if the error is under 20cm. If the error is higher than 2.5m, the test will be classified as *not converged*. Fig 5.27 shows the resulting convergence matrix of the localization. Here, the colormap indicates the mean time in seconds the localization needs to converge for all cases which are classified as converged and each combination of tested lateral and longitudinal errors in meter. The white number on each cell is the number of all tested situations, which converge under 20cm in percent. In general, it could be seen that with a higher absolute position error value, the convergence time increases, and the number of converged cases decreases. But a negative error value in the

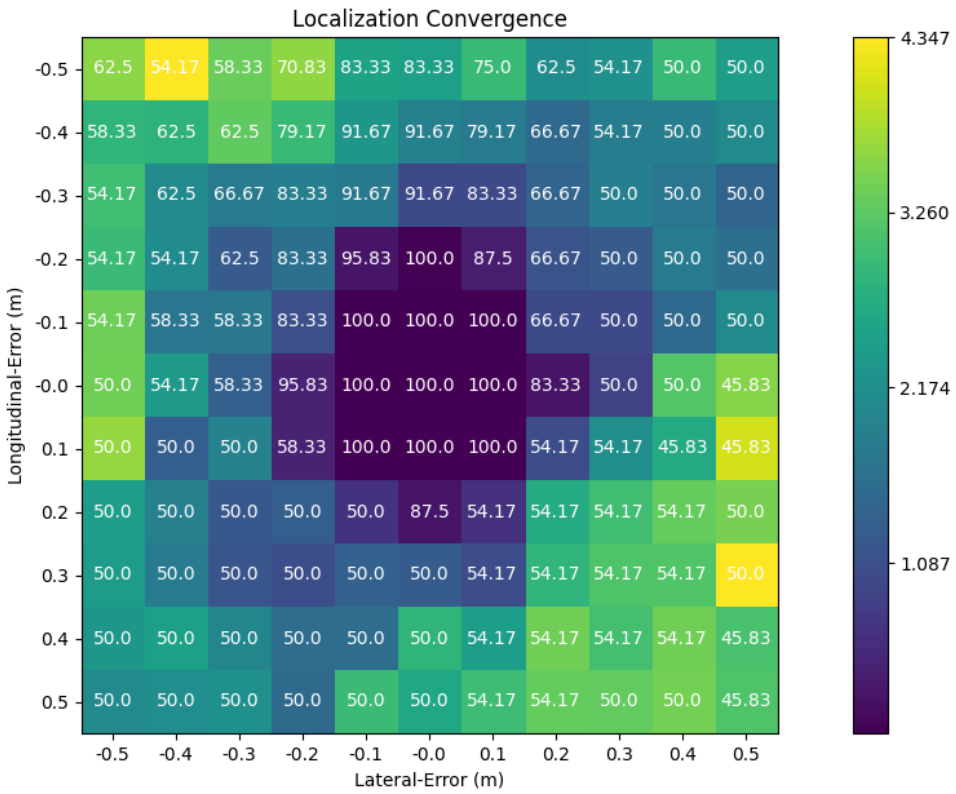

Figure 5.27: Convergence matrix of the localization on the full bag with lateral and longitudinal errors in the range of [-0.5m,0.5m]. The colormap indicates the mean time in seconds the localization needs to converge under 20cm for each combination of lateral and longitudinal error in meter. The white number on each cell is the number of all tested situations, which converge under 20cm in percent

longitudinal direction seems to affect the convergence not that much. In general the localization converges in a maximum duration of 4.347s and a minimum convergence rate of 45.83 %.
The final results of the full bag are depicted in table 5.1 and Fig. 5.28 The

Absolute Position-Error (m)				
mean	median	max	rmse	std
0.106	0.057	1.696	0.202	0.172

Table 5.1: Localization results of the whole bag from the city Ulm regarding to the absolute position error. 2s as accumulation time for the sliding window approach and the full polarimetric scattering matrix is used.

absolute position error is only 3 times over 50cm, with a mean of about 10cm and a median of 5cm. The maximum error is 1.75m, but this is only an outlier of a part from the bag where only fewer landmarks are available. The same applies to the other little outliers. In spite of this, the root-mean-square error is by only about 20cm and the standard deviation by 17cm. The full

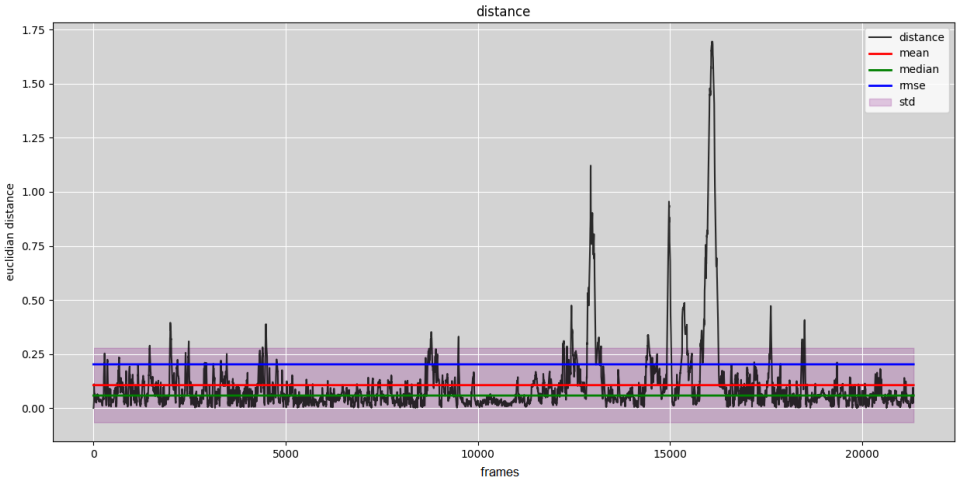

Figure 5.28: Localization results of the whole bag from the city Ulm regarding to the absolute position error over time. 2s as accumulation time for the sliding window approach and the full polarimetric scattering matrix are used.

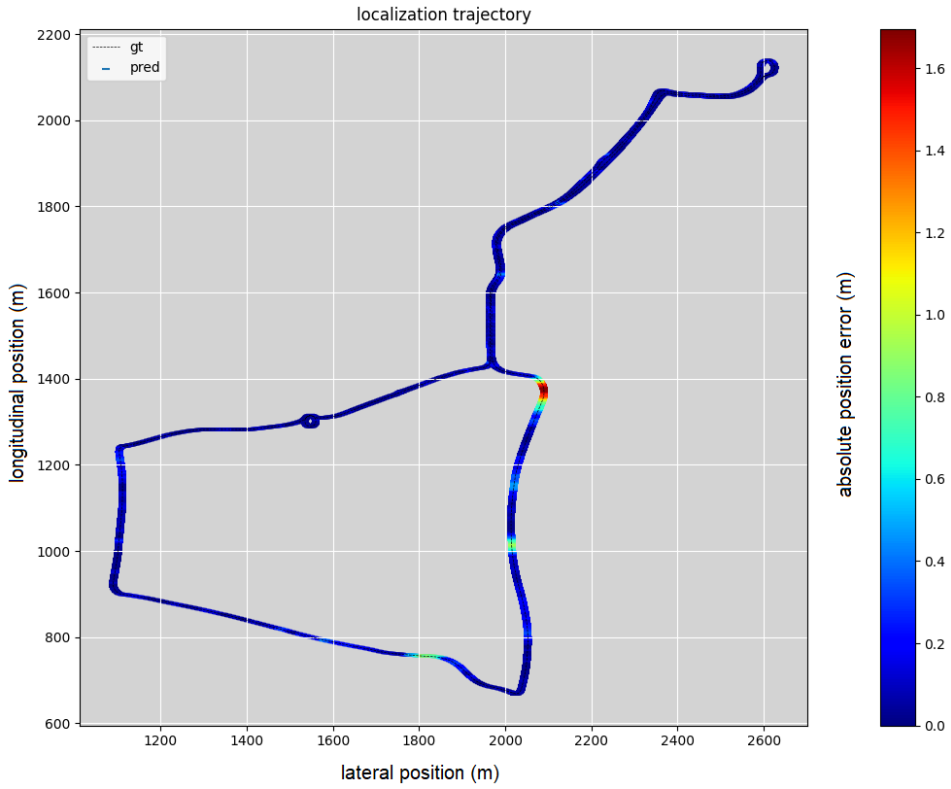

Figure 5.29: Localization trajectory of the whole bag from the city Ulm with the colormap indicating the absolute position error in meter for the specific position. 2s as accumulation time for the sliding window approach and the full polarimetric scattering matrix are used.

trajectory of the localization and the ground truth are shown in Fig. 5.29 from birds eye perspective.

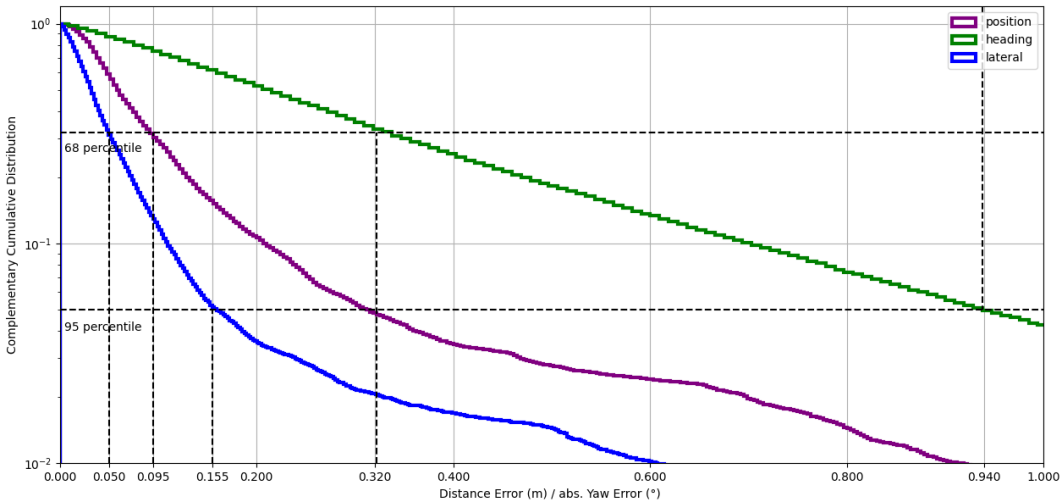

Figure 5.30: CCDF of the localization algorithm on the full bag of the city Ulm, using 2s accumulation time and full polarimetric scattering matrix. for absolute position error, lateral error in meter and heading error in degree

approach / error	68%	95%
radar-only lat.	0.25 m	1.07 m
radar-only abs.	0.32 m	1.41 m
radar & camera lat.	0.13 m	0.76 m
radar & camera abs.	0.20 m	> 2 m
full setup lat.	0.07 m	0.18 m
full setup abs.	0.15 m	0.28 m
this work lat.	0.05 m	0.15 m
this work abs.	0.09 m	0.32 m
this work yaw	0.32 °	0.94 °

Table 5.2: Comparison of the proposed approach and the previous state of the art approach [7] in the city environment with standard deviation of absolute position, lateral error in meter and additional yaw error in degree for 68 and 95 percentile.

Finally, the proposed algorithm has to be compared with the state-of-the-art approach from [7]. This approach uses a real-time graph-based SLAM approach to address the problem of localization with automotive-grade radars. The system also uses landmarks and radar-based odometry. Next to radar only, they also used a sensor fusion with radars + cameras and the full setup of radars + cameras + lidars + vehicle odometry instead of radar-based odometry. They tested their system in two scenarios: Hamburg

inner city and a container terminal in Hamburg harbor. To evaluate the system, they are using the absolute position and the lateral error in meter for the 68 and 95 percentile. In this case both parameters define the maximum error value of the CCDF for the specific parameter. For a fair comparison the proposed approach of this work is also validated on the same factors and is only compared to the results of [7] in the Hamburg inner city. The final CCDF for absolute position, lateral error in meter and yaw error in degree of the proposed localization algorithm using 2s accumulation time and the full polarimetric scattering matrix is shown in Fig. 5.30. Table 5.2 contains the from the CCDF in Fig. 5.30 following values for $\sigma 1$ and $\sigma 2$ together with the results of [7]. It could be easily verified, that the proposed approach beats the radar-only approach and radar + camera in absolute position and lateral error. The full setup with radar + camera + lidar + vehicle odometry also achieves worse results than the proposed approach of this work for the lateral error. For the absolute position error the proposed algorithm is about 2cm more accurate for the 68 percentile but about 4cm less accurate for the 95 percentile.

5.6.1 Limitations

Contrary to the very good results, the proposed approach is not perfect. Fig.

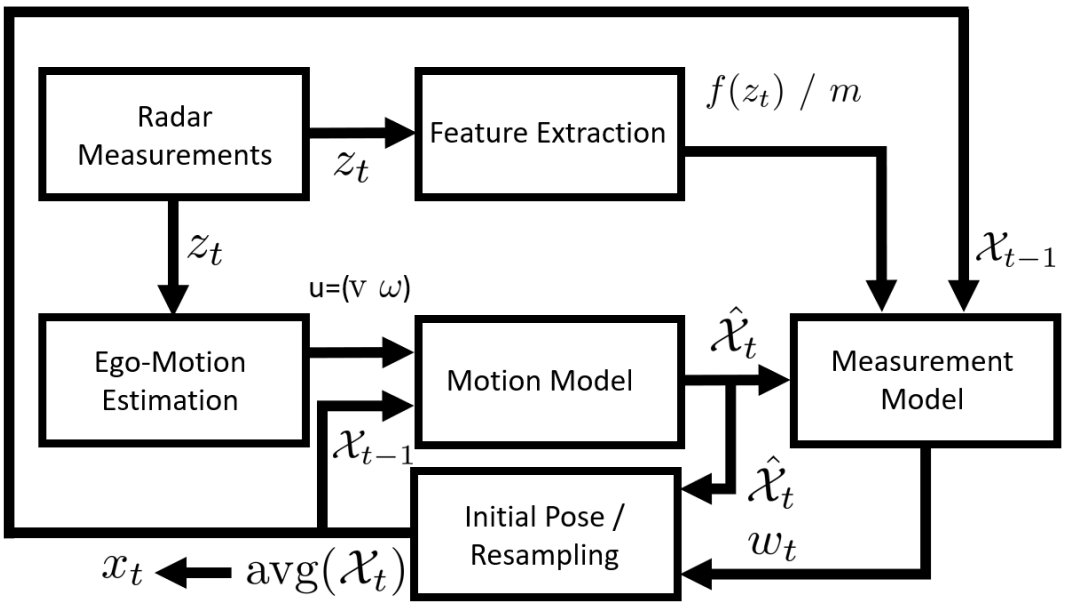

Figure 5.31: Block diagram of the proposed MCL algorithm.

5.31 shows a block diagram of the proposed MCL algorithm with the most important components. A single iteration starts with a previous pose $\mathcal{X}_t$ or the initial pose $\mathcal{X}_0$ at the start of the localization. Further, measurements z_t

are made from the radars. The measurements are then used to estimate the vehicle's motion state in case of the localization called by the ego-motion estimation. The estimated vehicle's motion state is then used as control in the motion model, which estimates the new poses of all particles assuming the motion of the vehicle is controlled by a translational v and a rotational velocity ω. In the meantime, features are extracted from the measurements in form of a PreCFAR covariance gridmap with the CFAR based approach for point-shaped objects and the Wishart distribution as measure. In addition to this, line shape objects are also detected from the covariance gridmap. After that, the detected interesting areas $f(z_t)$, the map of landmarks m and the pose estimates $\hat{\mathcal{X}}_t$ are given to the measurement model, which calculates the weights w_t for each particle given to it's pose, the measured interesting areas and the map of landmarks. Finally, the predicted poses $\mathcal{X}_t$ are calculated by the resampling step. Therefore, the vehicles pose is determined by the average pose of the predicted particle set $\mathcal{X}_t$.
Each component of the model has limitations that have to be mentioned. Some of them are capable of permanently disrupting the localization, which would have to be prevented in a real application. Starting by the initial pose x_0 or the predicted pose x_{t-1}, there could be errors in the x, y coordinates and the heading direction θ. That means the by start of an iteration assumed pose of the vehicle is wrong. The higher the error, the lower the ability of the localization to converge and the more time goes by to converge. Especially the initial pose x_0 has to determined without the proposed algorithm and will be probably come in a real application from a DGNSS algorithm. Therefore, without an accurate last determination of the vehicle's pose, the car could never start in a region without DGNSS reception. The next possible errors are coming from the polarimetric radars. There is uncertainty in all measurements. To avoid this the whole measurement model is probabilistic and incorporates the corresponding uncertainties for the measured ranges and azimuth angles. In addition to this, there comes another potential problem from the limitation of the perceptual ability of the polarimetric radars. The Doppler velocity is only the measured radial velocity. This leads to the problem that moving objects with a measured non radial velocity will be classifed as static and will therefore not be filtered out in the PreCFAR covariance gridmap. The Doppler velocity could also be a potential source of error for the ego-motion estimation. But not caused by a mistake in the measurement, but in the number of measured dynamic objects with the same radial velocity. Because the ego-motion estimation assumes that the biggest class of targets classified by the radial velocities are the static ones. In an environment of the vehicles that should ever be true, but potentially it could be false. The next limitation comes from the landmarks and measured interesting areas. The number of landmarks and extracted interesting areas as well as the threshold for the CFAR based algorithm is an important choice.

The interesting reader should take a look on the works of [54] and [21] in which this topic is discussed in detail. If there are more interesting areas than landmarks in the surrounding of the vehicle or a measured interesting area without a corresponding landmark the ICP is not working properly and the correspondence variables could not be determined correctly with a guarantee. To avoid this a filter is applied that exclude measured interesting areas in a single iteration, if they have a maximum distance of one meter to at least one of the landmarks. The ICP could also determine that more than one interesting area corresponds to a single landmark. This is definitely wrong and will causes false particle weights from the measurement model. All of these interesting areas which are determined as corresponding to the same landmark are not further considered in the iteration to prevent the false weighting. The last-mentioned restriction comes from the environment itself. If there are fewer or no objects in the environment, then there are fewer or no measured interesting areas and landmarks. In the case of no landmarks, the measurement model could not be applied and the localization algorithm works algorithm works then only by the motion model. This could not be long applied because the integrated error of the ego-motion estimation growth rapidly. With fewer landmarks a precise localization could also not be guaranteed because the more pairs of interesting areas and landmarks are determined, the more reliable the measurement model is.

6 Conclusion

With more and more advances in the field of autonomous driving, the demand for a vehicle self-localization algorithm without required GNSS reception increases. In probabilistic robotic approaches, e.g. particle filter or Kalman filter, a robot's motion estimate is fused with an observation-based approach using landmarks and gridmaps. There are sensors, e.g. ADMA sensors, which are able to estimate the vehicle's motion state very reliable. But they are really expensive and could, for this reason, not be considered for a serial production of an autonomous vehicle. Radars are also able to perform an ego-motion estimation. In addition to this, there are able to create gridmaps with excluded dynamic objects so that the gridmap is only of the static environment. Therefore, it is not surprising that radars are used in previous works to create a self-localization algorithm for vehicles. In PolSAR applications the polarization as additional information next to conventional radar sensing abilities, has already been used successfully. In this work a radar-based self-localization algorithm using additional polarimetric information was developed, which improves the performance of conventional radar-based localization algorithms.
In the first step, a radar-based ego-motion estimation was developed to estimate the robots motion state by 2DOF and 3DOF regarding to the underlying number of velocities of the estimated motion state. Additionally, a simple solution to handle the unambiguous Doppler velocity are developed. However, in the end all underlying parameters are evaluated and improved to make the best out of the radar based ego-motion estimation. As a result the 2DOF case seemed to be a more reliable prediction. Hence, a motion model was created, which assumes the vehicle is controlled by only two velocities (translational and rotational velocity), to estimate a pose from the control with additional noise regarding to the vehicles motion state. Corresponding to the amplitude gridmaps of conventional radars, polarimetric covariance matrices are used as the depiction of the environment. The used covariance gridmaps have in each gridcell a single covariance matrix. The matrices are built from the hermitian product of the polarimetric scattering vector as incorporation of several scattering mechanisms over time, to cover also the cases of anisotropic scatterers. On these gridmaps, a CFAR like approach is used to detect point shape objects during the localization and previously to create a map of landmarks. Another approach is used to detect also straight-line objects while localizing and also for a map of

landmarks. For both types of objects, a specific measurement model was created to calculate the probability of the polarimetric measurements to a given pose. The fusion of both models is the second big component of the localization, which is used to correct the prediction of the ego-vehicles motion only from the control given by the ego-motion estimation. Finally a Monte Carlo Localization algorithm is implemented which incorporates the motion and measurement model to perform the localization by a particle based approach. Finally the localizations parameters are evaluated. The average error of the ego-motion estimation for different periods of time and the error of the egomotion estimation for different accumulation times is validated with the aim of a less error as possible, by accumulating as much as possible to ensure good covariance matrizes. Following that, an investigation of the advantages of using full polarization against single or dual is done. It was shown that the full polarization works the best and using additional polarimetric information seems to improve the results of the localization. After that the localizations convergence is tested by using false start position poses. As a result the localization converges really fast, but with an increasing error the proportion of attempts that converges at all lowers. At the end the proposed localization algorithm is compared with the state of the art approach. The result shows, that our approach provides a much more precise localization then the previous state of the art approach for radar only based localization with radar ego-motion estimation and also another version with a fusion of radar and cameras. A third approach, which provides a fusion with radars, cameras, lidars and vehicle odometry reaches a little more precisely localization for one of four tested parameters of the comparision with our approach. Regarding to the other 3 parameters our localization works also better then the full sensor fusion of radar, lidar and camera with vehicle odometry.

List of Abbreviations

GNSS global navigation satellite system
ACC adaptive cruise control
radar radio detection and ranging
lidar light detection and ranging
MBS mechanical beamsterring
EBF electronic beamforming
DBF digital beamforming
DOA direction of arrival
FFT fast Fourier transformation
FMCW frequency modulated continuous wave
CSM chirp sequence modulation
FAR false alarm rate
CFAR constant false alarm rate
CUT cell under test
CA-CFAR cell averaging CFAR
IMU inertial measurement unit
GPS global positioning system
DOS degrees of freedom
RANSAC random sample consensus algorithm
LSQ least squares method
polSAR polarimetric synthetic aperture radar
PDF probability density function
@CITY Automated Cars and Intelligent Traffic in the City
ICP iterative closest point
slf straight line features
MCL Monte Carlo localization
CCDF Complementary cumulative distribution function

List of Figures

2.1 FMCW block diagram 4
2.2 FMCW radar signal with sawtooth modulation 5
2.3 FMCW radar signal with triangle modulation 6
2.4 chirp sequence modulation 7
2.5 Digital beamforming 9
2.6 CFAR - real detections and false alarms shown for different fixed thresholds 11
2.7 CFAR detection algorithm as block diagramm in the one dimensional case 11
2.8 Electromagnetic wave 13
2.9 Polarized waves 14
2.10 Polarization ellipse 14
2.11 Example Amplitude gridmap 22
2.12 Conceptual overview of polarimetric covariance gridmapping 23
2.13 Example of eigenmap λ_1 25
2.14 Conceptual visalization of PreCFAR Gridmap 25
2.15 Example SLAM graph of Hamburg inner city 27
2.16 Example 29
2.17 Driven paths for the city and terminal scenarios 30

3.1 Measurement vehicle 31

4.1 Sensor (red) and targets (blue) with longitudinal, lateral and radial(green) velocities in the sensor coordinate system 34
4.2 Vehicle motion specified at the reference point (middle of rear axis) and the corresponding sensor motion (red - v_j^x, v_j^y) at position (blue - x_j, y_j) 35
4.3 Problem of unambiguous Doppler velocity 37
4.4 Solved Problem of unambigoúous Doppler velocity 38
4.5 Ego-motion estimation example radial velocity over azimuth 39
4.6 Ego-motion estimation x-velocity over frames 42
4.7 Ego-motion estimation yaw-rate over frames 42

5.1 Robot environment interaction 44
5.2 Characterization of the evolution of the states, measurements and controls in a Bayesian network 45
5.3 Update rule of the Bayes filter algorithm 46

5.4 Example density function approximation used by particle filters 47
5.5 Particle filter algorithm 48
5.6 Illustration Importance sampling 51
5.7 Robot pose relative to a global coordinate system 52
5.8 The motion model: Robots trajectory illustrated by the solid line and posterior distributions of the robot's pose as shaded areas 54
5.9 Algorithm for sample poses $x_t = (x', y', \theta')^T$ from pose $x_{t-1} = (x', y', \theta')^T$ and control $u_t = (v, \omega)$ with additional motion noise from Gaussian distributions with zero mean and standard deviation of the corresponding noise parameter from $\sigma = (\sigma_v, \sigma_\omega, \sigma_\gamma)$ 55
5.10 Algorithm for sample numbers from an approximately normal distribution with zero mean and standard deviation σ [50]. **Rand_Int**(x,y) samples a number from a uniform distribution in $[x, y]$ like a pseudo random number generator, e.g. linear congruential generators [51]. 55
5.11 Landmark comparison polarimetric and non-polarimetric . . 59
5.12 Algorithm for landmark detection 60
5.13 Algorithm for calculating the likelihood of a landmark measurement. The algorithm requires an observed feature $f_t^i = \left(r_t^i \theta_t^i s_t^i\right)^T$, the correspondence variable c_t^i, the robot pose $x_{t-1} = (x \quad y \quad \theta)^T$ and the map m as input. Its output is the conditional probability $p(f_t^i | c_t^i, m, x_t)$. [4] 61
5.14 Algorithm for computing the probability from a of a zero-centered Gaussian distribution with standard deviation b . . 61
5.15 Algorithm for transforming f_t^i relative to the pose x_t from polar to Cartesian coordinates in the global coordinate system. 62
5.16 Simple Algorithm for calculating c_t^i by using the minimum Euclidian distance 62
5.17 Simple Algorithm for calculating c_t^i by using the minimum product of Euclidian distance and Wishart distance 62
5.18 Algorithm for calculating c_t^i by using the ICP based on Least-squares. 65
5.19 Algorithm for calculating the correspondence c_t^i of fs_t^i. 66
5.20 Algorithm for calculating the likelihood of a slf landmark measurement. The algorithm requires an observed slf feature $fsra_t^i$ in polar coordinates, a pose $\mathrm{x}_t = (x \quad y \quad \theta)^T$, the correspondence variable c_t^i and the map ms as input. Its output is the conditional probability $p(fsra_t^i | ms, x_t, c_t^i)$. 66
5.21 Monte Carlo Localization Algorithm based on particle filters 67
5.22 Localization example from the dataset of the city Ulm 68

5.23 Localization example from the dataset of the city Ulm, with separated covariance gridmap 70
5.24 CCDFs of absolute position error of the ego-motion estimation on the whole dataset from Ulm for different integration times. 71
5.25 CCDFs of absolute position error of different accumulation times of the localization on the small bag of the city Ulm. . . 72
5.26 CCDFs of using different single, dual and full polarization channels of the localization on the small bag of the city Ulm. 72
5.27 Convergence matrix of the localization 74
5.28 Final localization results 75
5.29 Localization trajectory of the whole bag from the city Ulm with the colormap indicating the absolute position error in meter for the specific position. 2s as accumulation time for the sliding window approach and the full polarimetric scattering matrix are used. 75
5.30 Final CCDF of the localization 76
5.31 Block diagram of the proposed MCL algorithm. 77

List of Tables

2.1 RCS for different object 10
2.2 Parameters for different polarization states 16
2.3 68 and 95 percentile of the cumulative distribution function (CDF) of the absolute position errors for Hamburg inner city and terminal regarding to the different sensor fusions. Table 1 in [7] 30

3.1 Details of the used polarimetric radars 32

4.1 Ego-motion estimation threshold validation 40
4.2 Ego-motion estimation necessary minimum number of samples validation 41
4.3 Ego-motion estimation error function validation 41
4.4 Ego-motion estimation results 41

5.1 Localization results of the whole bag from the city Ulm regarding to the absolute position error. 2s as accumulation time for the sliding window approach and the full polarimetric scattering matrix is used. 74
5.2 Comparison of the proposed approach and the previous state of the art approach [7] in the city environment with standard deviation of absolute position, lateral error in meter and additional yaw error in degree for 68 and 95 percentile. 76

Bibliography

[1] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: Approaches, lessons and challenges," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4649–4672, 2010 (cit. on p. 1).

[2] M. Wagner, *Lexikon Fachbegriffe Informationstechnik Datenverarbeitung Lexikon Fachbegriffe Informationstechnik Datenverarbeitung definitions of technical terms information technology*. LEHRMITTEL WAGNER/Autor (cit. on p. 1).

[3] D. Kreßel, *Automatisiertes fahren in der stadt*, last accessed 29.10.2021, 2017. [Online]. Available: `https://www.atcity-online.de/` (cit. on p. 1).

[4] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002 (cit. on pp. 1, 43–54, 56–58, 60, 61, 67).

[5] T. Zhou, M. Yang, K. Jiang, H. Wong, and D. Yang, "Mmw radar-based technologies in autonomous driving: A review," *Sensors*, vol. 20, no. 24, p. 7283, 2020 (cit. on p. 2).

[6] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, no. 3, p. 648, 2019 (cit. on p. 2).

[7] S. Jürgens, N. Koch, and M.-M. Meinecke, "Radar-based automotive localization using landmarks in a multimodal sensor graph-based approach," in *2020 21st International Radar Symposium (IRS)*, IEEE, 2020, pp. 373–378 (cit. on pp. 2, 27–30, 76, 77).

[8] H. Mott, *Remote sensing with polarimetric radar*. John Wiley & Sons, 2006 (cit. on p. 2).

[9] A. Ludloff, "Grundlagen und radargleichung," in *Praxiswissen Radar und Radarsignalverarbeitung*, Springer, 2002, pp. 11–54 (cit. on p. 3).

[10] R. Handbook, *Mi skolnik, ed*, 1990 (cit. on p. 3).

[11] *File:bsp2 cw-radar.png*. [Online]. Available: `https://commons.wikimedia.org/w/index.php?title=File%5C%3ABsp2_CW-Radar.png&oldid=446528599` (cit. on p. 4).

[12] D.-I. (C. Wolff, *Radar basics*. [Online]. Available: https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html (cit. on pp. 5, 6).

[13] I. D. Shirman, *Theoretische Grundlagen der Funkortung*. Militärverlag der DDR, 1977 (cit. on p. 5).

[14] N. C. Currie and C. E. Brown, *Principles and applications of millimeter-wave radar*. Artech House Norwood, MA, 1987 (cit. on p. 6).

[15] D. Kellner, "Verfahren zur bestimmung von objekt-und eigenbewegung auf basis der dopplerinformation hochauflösender radarsensoren," Doctoral dissertation, Universität Ulm, 2017 (cit. on pp. 7, 9, 35).

[16] M. Kronauge, C. Schroeder, and H. Rohling, "Radar target detection and doppler ambiguity resolution," in *11-th INTERNATIONAL RADAR SYMPOSIUM*, IEEE, 2010, pp. 1–4 (cit. on p. 7).

[17] H. Rohling and C. Moller, "Radar waveform for automotive radar systems and applications," in *2008 IEEE Radar Conference*, IEEE, 2008, pp. 1–4 (cit. on p. 7).

[18] W. Mayer, *Abbildender Radarsensor mit sendeseitig geschalteter Gruppenantenne*. Cuvillier Verlag, 2008 (cit. on p. 7).

[19] L. C. Godara, "Application of antenna arrays to mobile communications. ii. beam-forming and direction-of-arrival considerations," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1195–1245, 1997 (cit. on p. 9).

[20] H.-J. Li, Y.-W. Kiang, and W. Chen, "Radar and inverse scattering," in *The Electrical Engineering Handbook*, Academic Press, 2005, pp. 671–690 (cit. on p. 10).

[21] P. Will, *Investigation of polarimetric landmarks for self-localization in automated driving*, 2021 (cit. on pp. 11, 14, 21, 25, 79).

[22] H. Rohling, "Radar cfar thresholding in clutter and multiple target situations," *IEEE transactions on aerospace and electronic systems*, no. 4, pp. 608–621, 1983 (cit. on p. 11).

[23] A. Coluccia, G. Parisi, and A. Fascista, "Detection and classification of multirotor drones in radar sensor networks: A review," *Sensors*, vol. 20, no. 15, p. 4172, 2020 (cit. on p. 11).

[24] R. Srinivasan, *Importance sampling: Applications in communications and detection*. Springer Science & Business Media, 2002 (cit. on p. 12).

[25] F. Khoshnoud, M. B. Quadrelli, I. I. Esat, and D. Robinson, "Quantum cooperative robotics and autonomy," *arXiv preprint arXiv:2008.12230*, 2020 (cit. on p. 13).

[26] G. Waldman, *Introduction to light: The physics of light, vision, and color*. Courier Corporation, 2002 (cit. on p. 13).

[27] R. C. Jones, "A new calculus for the treatment of optical systemsi. description and discussion of the calculus," *Josa*, vol. 31, no. 7, pp. 488–493, 1941 (cit. on p. 13).

[28] A. Serbest and S. Cloude, "Radar polarimetry: A revision of basic concepts," *Direct and Inverse Electromagnetic Scattering*, pp. 257–275, 1996 (cit. on p. 14).

[29] W.-M. Boerner, W.-L. Yan, A.-Q. Xi, and Y. Yamaguchi, "On the basic principles of radar polarimetry: The target characteristic polarization state theory of kennaugh, huynen's polarization fork concept, and its extension to the partially polarized case," *Proceedings of the IEEE*, vol. 79, no. 10, pp. 1538–1550, 1991 (cit. on p. 15).

[30] J.-S. Lee and E. Pottier, *Polarimetric radar imaging: from basics to applications*. CRC press, 2017 (cit. on pp. 15, 16).

[31] W.-M. BOERNER, "Introduction to radar polarimetry–with assessments of the historical development and of the current state-of-the-art," *Electromagnetic Wave Interactions*, pp. 139–214, 1996 (cit. on p. 16).

[32] W.-M. Boerner, H. Mott, and E. Luneburg, "Polarimetry in remote sensing: Basic and applied concepts," in *IGARSS'97. 1997 IEEE International Geoscience and Remote Sensing Symposium Proceedings. Remote Sensing-A Scientific Vision for Sustainable Development*, IEEE, vol. 3, 1997, pp. 1401–1403 (cit. on pp. 16, 17).

[33] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 1, pp. 6–43, 2013 (cit. on p. 17).

[34] S. R. Cloude and K. P. Papathanassiou, "Polarimetric sar interferometry," *IEEE Transactions on geoscience and remote sensing*, vol. 36, no. 5, pp. 1551–1565, 1998 (cit. on p. 17).

[35] A.-Q. Xi and W.-M. Boerner, "Determination of the characteristic polarization states of the radar target scattering matrix [s (ab)] for the coherent monostatic and reciprocal propagation space by using the complex polarization ratio ρ transformation formulation," *JOSA A*, vol. 9, no. 3, pp. 437–455, 1992 (cit. on p. 18).

[36] S. R. Cloude and E. Pottier, "A review of target decomposition theorems in radar polarimetry," *IEEE transactions on geoscience and remote sensing*, vol. 34, no. 2, pp. 498–518, 1996 (cit. on p. 19).

[37] E. LÜNEBURG, "Principles of radar polarimetry," *IEICE Transactions on Electronics*, vol. 78, no. 10, pp. 1339–1345, 1995 (cit. on p. 19).

[38] J.-S. Lee and M. R. Grunes, "Classification of multi-look polarimetric sar data based on complex wishart distribution," in *[Proceedings] NTC-92: National Telesystems Conference*, IEEE, 1992, pp. 7–21 (cit. on p. 20).

[39] W.-M. Boerner, W.-L. Yan, A.-Q. Xi, and Y. Yamaguchi, "Basic concepts of radar polarimetry," in *Direct and Inverse Methods in Radar Polarimetry*, Springer, 1992, pp. 155–245 (cit. on p. 20).

[40] K. Werber, M. Rapp, J. Klappstein, M. Hahn, J. Dickmann, K. Dietmayer, and C. Waldschmidt, "Automotive radar gridmap representations," in *2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, IEEE, 2015, pp. 1–4 (cit. on p. 22).

[41] F. Weishaupt, J. F. Tilly, J. Dickmann, and D. Heberling, "Polarimetric covariance gridmaps for automotive self-localization," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, IEEE, 2020, pp. 1–8 (cit. on pp. 22–24).

[42] F. Weishaupt, N. Appenrodt, J. F. Tilly, J. Dickmann, and D. Heberling, "Precfar gridmaps for automotive radar," in *2021 18th European Radar Conference*, IEEE, 2021, pp. 1–8 (cit. on pp. 25, 26).

[43] D. Wilbers, L. Rumberg, and C. Stachniss, "Approximating marginalization with sparse global priors for sliding window slam-graphs," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2019, pp. 25–31 (cit. on p. 27).

[44] D. Wilbers, C. Merfels, and C. Stachniss, "Localization with sliding window factor graphs on third-party maps for automated driving," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 5951–5957 (cit. on p. 27).

[45] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using multiple doppler radars," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 1592–1597 (cit. on pp. 33–36).

[46] ——, "Instantaneous ego-motion estimation using doppler radar," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, IEEE, 2013, pp. 869–874 (cit. on p. 33).

[47] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981 (cit. on p. 33).

[48] M. Schorn, "Modelle zur beschreibung des fahrzeugverhaltens," in *Fahrdynamik-Regelung*, Springer, 2006, pp. 27–46 (cit. on p. 34).

[49] *Strassenfahrzeuge - fahrzeugdynamik und fahrverhalten*, Norm, Deutsches Institut für Normung e. V., 1994 (cit. on p. 34).

[50] G. Winkler, *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*. Springer Science & Business Media, 2012, vol. 27 (cit. on p. 55).

[51] J. E. Gentle, *Random number generation and Monte Carlo methods*. Springer, 2003, vol. 381 (cit. on p. 55).

[52] G. Fischer *et al.*, *Analytische Geometrie*. Springer, 1985, vol. 20017 (cit. on p. 55).

[53] M. Koecher, *Lineare algebra und analytische geometrie*. Springer-Verlag, 2013 (cit. on p. 55).

[54] F. Weishaupt, P. S. Will, N. Appenrodt, J. F. Tilly, J. Dickmann, and D. Heberling, "Robust point-shaped landmark detection using polarimetric radar," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2021, pp. 1–6 (cit. on pp. 58–60, 79).

[55] K. Werber, J. Klappstein, J. Dickmann, and C. Waldschmidt, "Interesting areas in radar gridmaps for vehicle self-localization," in *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, IEEE, 2016, pp. 1–4 (cit. on p. 58).

[56] M. Rapp, K. Dietmayer, M. Hahn, F. Schuster, J. Lombacher, and J. Dickmann, "Fscd and basd: Robust landmark detection and description on radar-based grids," in *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, IEEE, 2016, pp. 1–4 (cit. on p. 58).

[57] nbviewer, *Icp.ipynb*, last accessed 01.11.2021, 2021. [Online]. Available: `https://nbviewer.org/github/niosus/notebooks/blob/master/icp.ipynb` (cit. on pp. 61–64).

[58] K. Werber, J. Klappstein, J. Dickmann, and C. Waldschmidt, "Association of straight radar landmarks for vehicle self-localization," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 736–743 (cit. on pp. 65, 66).

[59] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960 (cit. on p. 67).