TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Computer Systems

IA70LT
Karl Laanemets 178191IASM

# INDOOR MAPPING USING AN AUTOMOTIVE MMWAVE RADAR

Master's Thesis

Supervisor: Mairo Leier
Researcher, PhD
Co-Supervisor : Priit Ruberg
Early Stage Researcher, PhD

Tallinn 2019

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

IA70LT
Karl Laanemets 178191IASM

# SISERUUMIDE KAARDISTAMINE MILLIMEETERLAINEALA RADARI ABIL

Magistritöö

Juhendaja: Mairo Leier
Teadur, PhD
Kaasjuhendaja : Priit Ruberg
Nooremteadur, PhD

Tallinn 2019

# Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Karl Laanemets

20.05.2019

# Abstract


**INDOOR MAPPING USING AN AUTOMOTIVE MMWAVE RADAR**

The goal of this thesis is to assess the suitability of an automotive radar for indoor mapping in conjunction with an inertial measurement unit. To accomplish this, a proof of concept based on a robotic solution is created. In conjunction, an introduction to the automotive radar technology and the concepts of probabilistic robotic mapping and perception is provided. This knowledge is used to solve the problem of simultaneous localization and mapping.

The thesis covers the issue of creating a map from incomplete knowledge of the surrounding environment, the state of the robot and imperfect sensors. Understanding the underlying uncertainty by representing it in a structured way is the main component when providing a solution to the simultaneous localization and mapping problem. The case of uncertainty that the sensor must operate in has to be considered when implementing a solution and therefore it requires the assessment of capabilities before use, even when used to augment other sensors.

The results of this work is a mobile platform using a low angular resolution radar and an inertial measurement unit that is used to create an occupancy grid map of a office environment. The solution succeeds in mapping parallel spaces in dominant directions of office hallways. The cases of circular movement and perpendicular surfaces however, require more effort, and this thesis provides recommendations and groundwork for further assessment and development.

The resulting solution could be further leveraged by modifying it for indoor mapping in case of difficult and harsh environmental conditions often faced by emergency workers, such as during building fires and after landslides, to assess the changes in building floor plans.

This thesis is written in English and is 72 pages long, including 3 chapters, 38 figures, and 0 tables.

# Annotatsioon


**SISERUUMIDE KAARDISTAMINE MILLIMEETERLAINEALA RADARI ABIL**

Käesoleva magistritöö hindab 77GHz sagedusriba madala nurgaresolutsiooniga radari sobivust siseruumide kaardistamiseks. Töö eesmärgiks on tutvustada tõenäosusteoorial põhinevat, kaardistamisrobotites kasutatavat metoodikat oma ümbruse ning oleku tuvastamiseks ja radaritehnoloogia tutvustust. Eelnimetatud põhimõtetele toetudes koostatakse kaardistamisrobot.

Eesmärgi saavutamiseks loodi liikuval alusel kaardistamisrobot, kasutadeks selleks madalaresolutsioonilist 77GHz sagedusribal töötavat radarit ning inertsiaalandurit. Töö tulemusena koostatud kaart näitab ruumi hõivatust, jagades kaardistatava ala diskreetseteks osadeks, mis luuakse võttes arvesse roboti asukohta ning radari mõõtetulemusi.

Välja töötatud süsteemi testiti sobivuse hindamiseks kontoriruumides, kasutades selle keskkonna enimlevinuid esemeid ning kõige tõenäoseimat läbimistrajektoori. Testi tulemused näitavad, et loodud lahendus on võimeline kaardistama seinu, liikudes nendega parallelselt, kuid kvaliteet langeb täisnurksete pindade ning ringtrajektooride puhul.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 72 leheküljel, 3 peatükki, 38 joonist, 0 tabelit.

# List of abbreviations and terms

| | |
|---|---|
| **3D** | Three Dimensions |
| **6DOF** | Six Degrees of Freedom |
| **AoA** | Angle of Arrival |
| **BPM** | Binary Phase Modulation |
| **CFAR** | Constant False Alarm Rate |
| **DOF** | Degrees of Freedom |
| **DSP** | Digital signal processor |
| **EKF** | Extended Kalman Filter |
| **FFT** | Fast Fourier Transform |
| **FMCW** | Frequency Modulated Continous Wave |
| **FOV** | Field of View |
| **IC** | Integrated Circuit |
| **IF** | Intermediate Frequency |
| **IMU** | Inertial Measurement Unit |
| **MCU** | Microcontroller Unit |
| **MEMS** | Micro-electro-mechanical systems |
| **MIMO** | Multiple Input Multiple Output |
| **OGM** | Occupancy Grid Map |
| **PCB** | Printed Circuit Board |
| **PDF** | Probability Density Function |

**PID** Proportional Integral Derivative

**PWM** Pulse Width Modulation

**RAE** Range-Azimuth-Elevation

**ROS** Robot Operating System

**RX** Receiver

**SLAM** Simultaneous Localization and Mapping

**SNR** Signal-to-Noise ratio

**TDM** Time Division Multiplexing

**TX** Transmitter

# Contents

# List of Figures

# 1 Introduction

The problem of Simultaneous Localization and Mapping (SLAM) has gained popularity in the field of research due to the rise of interest in self-driving vehicles, autonomous robots and drones. Solving the problem of mapping the surrounding environment requires sensing the environment in order to acquire information about it. The collection of information about the state of the world is in principle done using sensors. In the context of self-driving vehicles, autonomous robots and drones, the field of sensing has been dominated by lidar, high resolution radars, depth cameras in the form of stereo cameras. They are the mainly used as main sensing solutions for self-driving applications. Low resolution automotive radars have found use in being the good-to-have secondary sensing solution to complement accuracy of primary sensors. This limits the main use cases of radars in automotive sensing solutions to parking assistance, intelligent cruise control, collision warning and blind spot recognition systems.

Other use cases of radars cover the situations of difficult environmental conditions, such as rain and fog. In such difficult conditions, the effectiveness of the aforementioned primary sensors deteriorates. However, radar has proven to be excellent for these cases in harsh environmental conditions and has found its limited use in enhancing the driving experience. The augmental nature of radar has been the case for indoor localization and mapping and much research has not be done to use radar as a primary sensing solution for localization and mapping.

The thesis is divided into three chapters. The first chapter will give an overview of the starting conditions in the form of literature review, the task at hand and additional restrictions in the pursuit of the goal. The second chapter will introduce the theoretical principles required to understand the final solution of the defined task. The third chapter describes the implementation of a mapping robot based on the theoretical base laid in chapter two.

## 1.1 Related work

The main focus in this frequency band is on the automotive solutions that are often seen in vehicles for automotive safety systems. As an example, in [1] and [2], automotive radar systems are proposed as a augmental measure to other safety systems. However, some progress has been made in using those safety system radars for mapping parking spaces

as in [3] and under specific road conditions like described in [4] and [5].

This does not mean that providing indoor localization and mapping is not considered. Despite the interest, few solutions focus on probabilistic inference to create maps like in [6], which do not, again, handle the cases of simultaneous localization. The fully explored cases for mapping, include densely packed transmitter arrays for increasing the resolution of the 60GHz band as in [7] and [8]. They focus on the effects of the number of transmitter array components in building probabilistic indoor maps using frequency modulated continuous wave radars, while not considering localization. The solution in [9] focuses on high resolution radars to build probabilistic maps through walls and how they affect the quality of the map and propose a exploration scheme to best map perpendicular and parallel surfaces.

The main mapping and localization efforts have focused on creating digital signal processing proof of concept solutions for high resolution radars. These solutions focus on beacon based solutions, scattering them inside the environment. In [10] a millimeter wave simultaneous localization and mapping algorithm using a high bandwidth communication systems is used. It provides obstacle detection and dimensioning using virtual anchor nodes dispersed inside an environment. Similarly, in [11] and [12], a 5G communication system using millimeter wave is considered for determining the location and orientation of a radio terminal in an indoors environment. They propose a message passing estimator that provide localization capable of dealing with scatters and reflections inside the environment. For roboti mapping two radar processing methods are proposed in [13]: a lateration and an antenna gain method for localization. Using the two localization methods, the radar detections are used to create an occupancy grid map. It uses a 24 GHz frequency modulated continuous wave radar to provide localization and mapping in an indoor environment. Localization based on scattered radars is proposed in [14] by using triangluation, similarly to [11] and [13]. [15] focuses on non-line-of-sight propagation of the millimeter waves to provide localization and mapping using specular reflections of electromagnetic waves.

## 1.2 Problem definition and goals

As mentioned in the previous section, many current solutions rely on highly accurate radars or the scattering of multiple radars into the environment acting as beacons. Other solutions use integrated radar systems by duplication of sensors to reach the same goal. This leads to the use of pre-built radar systems of cars for mapping and localization[3], [4], [4]. This careful placement of sensors is acceptable, if the structure of the environment is known. However, in the case of robotic mapping, when the surroundings are not known,

this methodology is not feasible. For smaller robots this redundancy of sensors is often detrimental to its operation and a less complex solution is required. Taking the current situation into account, the purpose of this thesis is therefore, to explore the possibility of creating a solution for indoor mapping. This is achieved using a single lower cost radar with low angular resolution. The aim is to reduce the cost and power requirements of high resolution radars and the number of sensors. The proposed system should be tested experimentally to assess the suitability for mapping purposes. This thesis can be summarized as the completion of the following goals:

- introduce the theoretical basics of robotic mapping;
- explore the problem of robotic mapping and its solutions;
- give a overview of automotive 77 GHz radar technology;
- provide a proof of concept solution by incorporating the capabilities of an low angular resolution automotive radar and an inertial measurement unit;
- assess the suitability of the automotive 77 GHz AWR1843 evaluation radar module for indoor mapping.

To achieve the goal of providing a solution for mapping an indoor environment, certain limitations are imposed:

- only two dimensional planar setting is considered.
- the robot is moving on a flat surface and is not subject to slippage, pitch or roll;
- environment remains static;
- determining the physical movement of the robot for mapping is done using only a radar and an Inertial Measurement Unit (IMU);
- automotive, low angular resolution radar used in conjunction with an IMU for sensing.

# 2 Theoretical basis

This section will give a overview of automotive radar technology and form basic intuition for creating a mapping robot. It first focuses on the way that the surrounding environment is defined in robotics by introducing a way to represent the location and orientation of a robot in a structured way. It also touches upon the way to model the locomotion of a mobile robot. Next the introduction of modeling the robots own perception of the surrounding environment is made in the form of sensor models. Thirdly, the problem of estimating the location of a mobile robot is proposed. This is done through the lens of probabilistic state estimation for inferring the robots location based on its perception of the environment.

## 2.1 Robot pose

This section will give an overview of the representation of robot pose in the terms of orientation and translation of reference frames. The aim is to introduce a structured way to lay down the problems of mapping in later chapters. It requires a structured definition of the pose and locomotion of a mobile robot, discussed in the following section.

### 2.1.1 Representation of pose

A body can move according to its Degrees of Freedom (DOF) dictated by the composition of the body. Degrees of freedom represent the number of possible translations and rotations for that body. Translations describe the change in position in terms of a certain reference frame, while rotations dictate the orientation (attitude) between the reference frame axis. For example, a simple rigid body moving on a planar surface has six DOF: translation and rotation along and around each axis in 3D. This concept of six degrees of freedom is referred in robotics to as the pose of the body. The following sections will describe representation of orientation and position [16].

When describing pose, a reference is required to quantify the orientation and position of a body.Poses are always defined in relation to a reference frame be it a landmark, a traffic sign or by transformation from a previous pose for a moving vehicle. Defining a reference frame can be simply created by specifying a static point in space and defining basis vectors for representing location and orientation relative to that point. Figure 1

depicts two reference frames: a static $F$ and a moving $F_v$ with velocity $\overrightarrow{v}$. The latter signifies a fixed point on a moving rigid body while the former a fixed point in space [16].

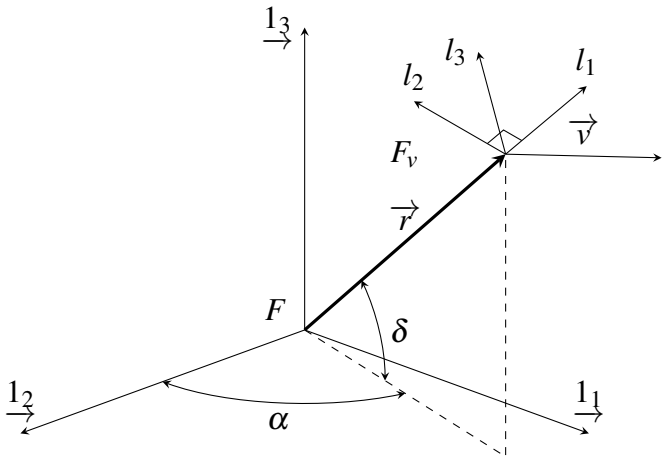



Figure 1. Basic configuration of reference frames with a single moving body.

For mapping, a reference frame is important when placing the measurements of a sensor when creating a map. This is because, to map the surrounding environment, the robot must be in motion. This proposes the need to know the change in orientation and location of the robot. As mentioned previously, reference frames provide a way to represent the state of a point in space in the form of translation and orientation. In mobile robotics a global reference frame for the purpose of simultaneous localization and mapping is usually defined. This global reference frame acts as a local landmark to keep track of the relative changes in orientation and location during locomotion. When solving the problem of simultaneous localization and mapping there is no map to define such a reference frame. Because of this the system initialization location chosen as the frame for localization.

In Figure 1 the vector $\overrightarrow{r}$ represents the translation of frame $F_v$ in relation to the fixed frame $F$. In a six degrees of freedom (DOF) system, the vector representation is provided in Equation (1). The row vector represents the components of the translation vector $\overrightarrow{r}$ and the column vector the basis vectors that define the static the reference frame $F$ [16].

$$\vec{r} = \begin{bmatrix} r_1 \; r_2 \; r_3 \end{bmatrix} \begin{bmatrix} \overrightarrow{1_1} \\ \overrightarrow{1_2} \\ \overrightarrow{1_3} \end{bmatrix} \tag{1}$$

Rotational motion of frame $F_v$ in relation to frame $F$ from Figure 1 can be described in three ways: rotational matrices, Euler angles or quaternions [16]. Rotation matrices

provide a verbose way to perform all needed transformations for determining the pose of a robot using linear transformations with linear maps. Euler angles on the other hand provide a way to express orientation in a more elegant way of sequenced rotations of reference frames. The downside of using the two solutions are the verbosity for the rotation matrices and the ambiguous way of expressing rotations for Euler angles, not ruling out the possibility of Gimbal lock. Quaternions provide the optimum for those two by expressing orientation in a unambiguous and compact way.

Quaternions are the generalization of complex numbers and can be represented as seen in Equation (2), where quaternion $\mathbf{q}$ is the sum of the product of four scalars $s$, $x$, ,$y$, $z$. The quaternion units $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ define the basis vectors, using which vectors can be constructed. The composing of vectors can be done by scaling the basis vectors using scalars.

$$\mathbf{q} = s + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \tag{2}$$

For example, representing a Eucleidian vector like $\overrightarrow{a} = (3,4,5)$ in Cartesian space as a quaternion can be written in the form of Equation (3).

$$\overrightarrow{a} = (3,4,5) = 3\mathbf{i} + 4\mathbf{j} + 5\mathbf{k} \tag{3}$$

Where $\mathbf{i},\mathbf{j},\mathbf{k}$ are the unit vectors representing the three Cartesian axes. Quaternions can be used to express rotations around vectors. The property of representing rotations around a vector in the form of quaternions can be leveraged to express rotations in Cartesian space. This is done by defining orthogonal unit vectors as the fundamental quaternion units when representing orientation and rotation of reference frames. The rotation of $\phi$ degrees around the unit vector $\overrightarrow{u} = u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}$ can be represented by a quaternion as seen in Equation (4).

$$\mathbf{q} = \cos\frac{\phi}{2} + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\phi}{2} \tag{4}$$

Equation (4) depicts a fixed rotation around a vector. This is not useful in a mapping solution as the orientation of the robot is in constant change. This raises the need to express the need for applying a series of rotations around a vector. Equation (5) shows a composition of rotations, where $q_1$ is the initial orientation of the vector and $q_2$ the applied rotation, both in quaternion form.

$$q = q_2 q_1 \tag{5}$$

Note that this is not commutative ($q_1 q_2 \neq q_2 q_1$), therefore rotations using quaternions cannot be applied in random order. However, the amount of rotations that can be applied in sequence is not limited. Therefore, representing a sequence of rotations can be done by applying quaternion rotations in sequence to a quaternion representing orientation. The resulting quaternion is the new orientation of the initial quaternion to which a new rotation can be applied. The noncommutative nature of quaternion rotations is similar to rotation matrices and Euler angles. The main argument for quaternions however, are that they are compact, easy to compose and avoid the problem of Gimbal lock.

The concepts of translation and orientation can be composed into a single model to describe the change of location and orientation from$(x,\ y, \theta)$ to $(x', y', \theta')$ called the odometry model. It composes of changes in translation and rotation expressed in Equation (6) [17].

$$\begin{aligned} \delta_{trans} &= \sqrt{(x'-x)^2 + (y'-y)^2}, \\ \delta_{rot1} &= \arctan(y'-y, x'-x) - \theta, \\ \delta_{rot2} &= \theta' - \theta - \delta_{rot1} \end{aligned} \tag{6}$$

## 2.2 Robot perception and sensor models

This chapter will give an overview of methodology to define how robots perceive the environment using different sensors as described in [16]. It will also give a theoretical overview of the sensors technology used in this thesis. First, the theoretical overview of the general sensor models is given. This is done in an abstract way and further details are provided in latter chapters in the terms of probabilistic models for mapping. After that, a more comprehensive introduction of radar technology is done to define the parameters of a radar and their implications on mapping.

### 2.2.1 Sensor models

Modeling is used to define the ways to perceive and represent the world. The purpose of sensor models is to map the sensor representation of the environment to a more usable and efficient way. Models provide a close enough of approximation for this goal to base decisions upon the results of measurement. Sensor models provide a way to represent the perception of the world in a quantifiable, general and compact way. These sensor models include perspective or stereo cameras, range-azimuth-elevation and inertial measurement units (IMU). The range-azimuth-elevation and IMU models are relevant to this thesis so they will be introduced in more detail.

First the representation of radar measurements are considered. In general, radar generates electromagnetic waves that cause reflection from objects in its area of propagation. These reflection points can be most accurately modeled using Range-Azimuth-Elevation (RAE) sensor model. This is caused by the working mechanism of the radar technology as it is capable of providing the range and Angle of Arrival (AoA) of a scanned object. A more detailed radar introduction can be found in section 2.3.

RAE observes a point in space using polar coordinates. This is very similar to the information of a typical radar reflection point, which is composed of the range of the target, its angle of arrival and the radial velocity of the target. Three main parameters of the RAE model are: range $r$, azimuth $\alpha$ and elevation $\varepsilon$ are shown in Figure 2 [16].

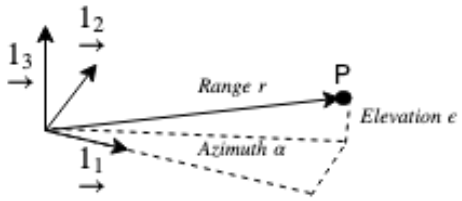



Figure 2. Range-azimuth-elevation visualization [16].

The model describes a three dimensional point in space fully in relation to a reference frame. In case of radars this reference frame is the radar and is called the sensor frame. The range vector composes of three components, one for each dimension of the Euclidean space shown in Equation (7).

$$r = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{7}$$

Since this thesis only covers the planar mapping problem, range $r$ and azimuth $a$ from Figure 8 are sufficient when presenting the measurements of the automotive radar. Therefore, to map sensor detection points from Caresian coordinates of point $P(x,y)$ in Figure 8 into polar coordinates $r$ and $\alpha$, the application of the Pythagorean theorem and simple geometry will result in the sensor model expressed in Equation (8) [16].

$$\begin{bmatrix} r \\ \alpha \end{bmatrix} = \begin{bmatrix} \sqrt{x^2+y^2} \\ \arctan(y/x) \end{bmatrix} \tag{8}$$

Next, the second sensor used in this thesis is described briefly, named Inertial measurement units IMU. They are Micro-electro-mechanical systems (MEMS) that provide the angular rate of change $\omega$ and acceleration $\alpha$ of three orthogonal frames by utilizing accelerometers and gyroscopes respectively. However, newer IMUs also provide orientational data by using magnetometers for measuring Earth's magnetic fields for achieving Six Degrees of Freedom (6DOF) [16], [18].

All the sensor measurements are done in the sensor frame, that is usually not overlaid with the vehicle frame. Thus a transformation from sensor frame to vehicle frame is required. In actuality, this transformation is usually just one of translation, since the measurement frame is fixed in relation to the vehicle frame, resulting in zero circular motion rate in relation to the vehicle. If the sensor frame offset from the vehicle frame is small enough, it can be ignored [16].

## 2.3 Radar

Radar (**RA**dio (Aim) **D**etecting **A**nd **R**anging), like other rangefinders: lidar, sonar and ultrasonic sensors, are a classical examples of sensors can be modeled using the RAE model. This chapter aims to give a detailed overview of the intrinsic parameters and technology of automotive radars. This is done to analyze and identify the relevant radar parameters for using it as a primary sensing solution. Radar uses electromagnetic waves to determine different parameters of an object by using the energy reflected back from it as seen in Figure 3. The returned signal is often called an echo in literature [19].

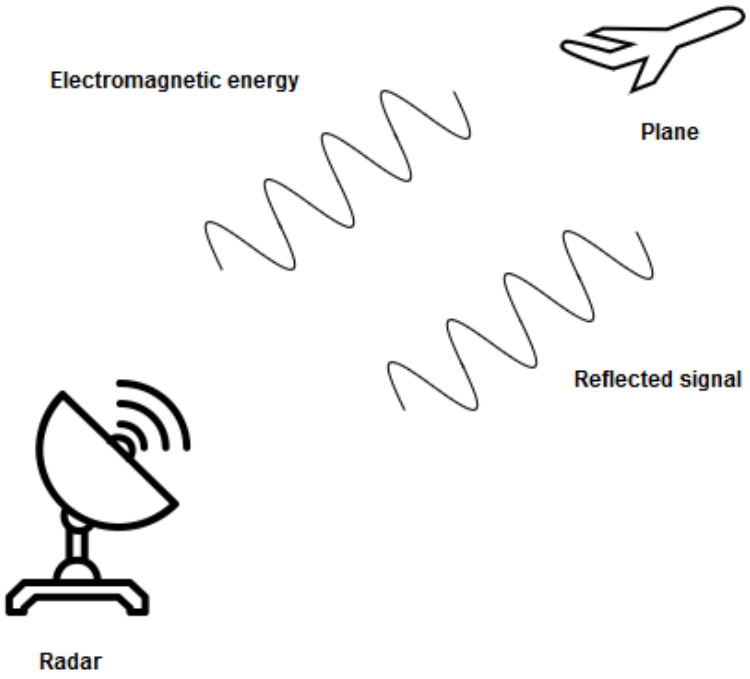



Figure 3. Simplified radar working principle.

Depending on the radar design, the waves transmitted can be directed into a desired direction, and thus the precise parameters of an object can be determined. These parameters include the relative distance of the object, its speed, dimensions, such as height or width, and the bearing of the object. Due to the constant nature of the speed of the electromagnetic wave, the distance of the object can be determined by measuring the time of travel of the wave. The time of travel increases linearly if the object is further away from the radar.

The main building blocks of a radar include a transmitter and a receiver. They are tasked with transmitting and receiving the electromagnetic waves used in determinign objects range from the radar. In addition to that, to create the waveform of the transmitted signal, a synthesizer is required. Synthesizers are electronic circuits that handle the creation of the transmitted signal. This is done by modifying the frequency of stable and accurate crystal oscillator. Radars have found use in different fields: military, where it was first used, geographical imaging, flight control and recently in automotive industry as a part of safety measure systems.

They provide various advantages compared to visual observation. Namely, the capability to operate in all weathers, be it fog or rain. Other significant upside of radar is that it does not require visible light to operate, making it versatile in its usage compared to video cameras used in autonomous vehicles.

Radars offer a more foolproof and often a cheaper solution, compared to ultrasonic or lidar systems, to implement safety systems in automotive applications:

- Collision warning
- Intelligent cruise control
- Parking-assist systems
- Blind spot recognition

Radars are classified based on the transmitted signal waveform and are mainly divided into four:

- Continuous Wave (CW) - the wave is continuously transmitted without modulating the waveform. Uses doppler effect as its crucial component [19].
- Pulsed - the waveform is transmitted without modulation after discrete time steps
- Frequency Key Shifted - the waveform is modulated in frequency by a modulating binary signal.

- FMCW - wave is transmitted continuously and modulated in frequency (phase)

The next chapters will introduce the working mechanisms of FMCW radars, which is the dominant class of radars used in automotive applications. The strength of radar include the high resolution provided in determining the distance and velocity of the tracked object. The next section will describe these parameters and strengths in more detail.

### 2.3.1 Detecting range of objects

This chapter will cover the basics of a single transmitter and single receiver frequency modulated continuous wave radar technology and how the different object defining parameters are resolved during signal processing. In this chapter, when using the word radar, single input single output FMCW radar is meant.

The radar processing chain comprises of a transmitter, receiver, mixer and a waveform synthesizer (Figure 4) [20], [21].

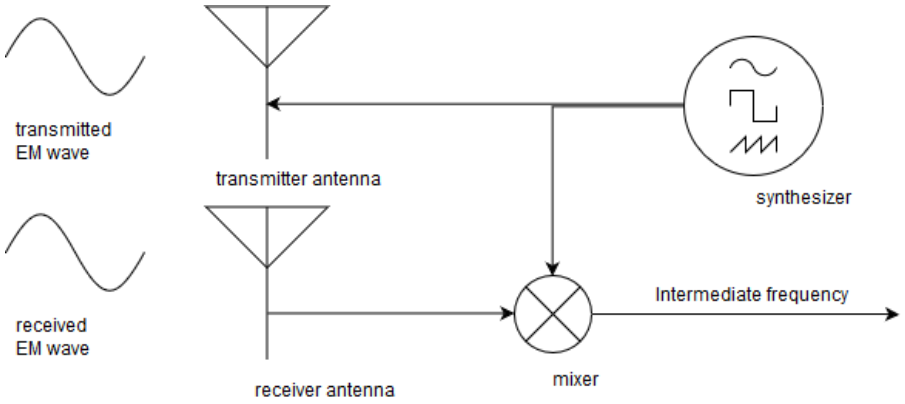



Figure 4. Basic FMCW radar processing chain.

The synthesizer handles the generation of the transmitted waveform, in the case of a FMCW waveform called a chirp. After the transmission and receiving process, the waveforms are processed by a mixer which produces a intermediate frequency Intermediate Frequency (IF) at a difference or beat frequency of the two waves [19] - [22]. Chirps can more easily explained by plotting frequency against time, seen in Figure 5 [20]–[22].

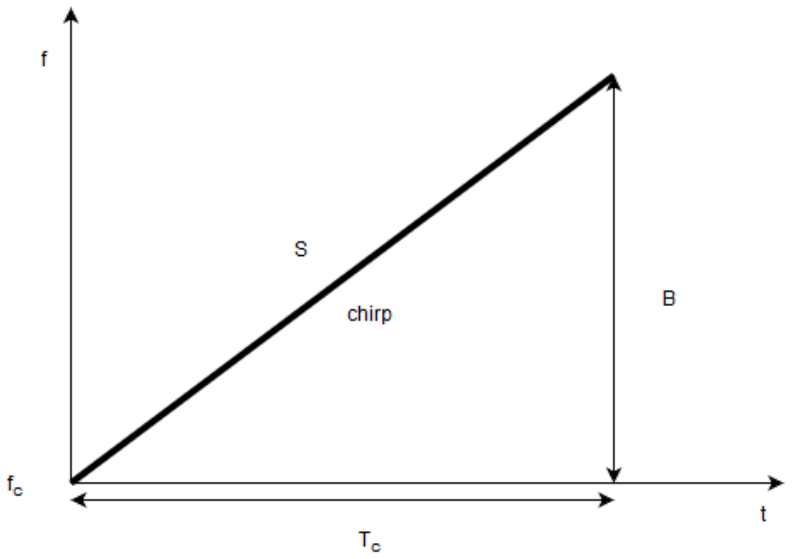



Figure 5. Chirp frequency plotted against time.

A linearly increasing waveform will start at a start frequency ($f_c$) and increases during its duration ($T_c$). This is also called ramping time. The resulting plot has two more defining parameters: bandwidth ($B$) and slope ($S$). The slope signifies the difference in the end frequency and the start frequency while the slope expresses the rate of change in frequency over time. The radar uses this waveform to determine the distance, velocity and angle of arrival by mixing the transmitted and received signals into a IF using a mixer [20] - [21].

The IF signal composed by the mixer is useful for the purpose of further signal processing. The nature of an IF signal is mainly multiplexing two signals that keeps the intrinsic properties and relations of the TX and RX chirps intact for amplification. It further eliminates the need for processing two signals simultaneously and thus simplifies the processing chain. The resulting IF is constructed by differentiating the instantaneous frequency and the phases of the transmitted and received signals [20], [22].

Two cases of range measurement will be considered: with one object for introduction and consideration for the case with multiple objects. For a single object, the resulting IF signal is a constant frequency tone, with frequency determined by the travel time $\tau_d$ times the slope of the transmitted signal $S$ as in Equation (9) [20] - [22].

$$f_{IF} = \tau_d \times S \tag{9}$$

The slope is a flat constant: it increases linearly between two frequencies and does not fluctuate (see Figure 5), therefore the tone frequency depends only on the time delay $\tau_d$

between the transmitted and received signals. Therefore, the distance is linearly proportional to the time delay between signals: the further the reflecting object is relative to the radar, the longer the delay and higher the frequency of the IF and vice versa. This provides a way to determine the distance of an object using a chirp [20],[22].

The time delay $\tau_d$ between the transmitted and received signals is two times the distance $d$ of the object over the speed of light $c$ as it travels to and from the object seen in Equation (10) [20], [22].

$$\tau_d = \frac{2 \times d}{c} \tag{10}$$

This chain of processing can be repeated with multiple objects, by calculating the IF of each receiving signal with a single TX signal. This provides a way of differentiating multiple objects in distance. However, the resulting IF tones overlap as they have the same validity period and thus provide a composite tone, consisting of multiple tones overlaid. This can be mitigated with frequency domain analysis using Fast Fourier Transform (FFT) as each distinct tone will produce a peak on separate frequencies, although two objects with the same range can produce a single peak. This can be mitigated by improving the range direction resolution of the radar or by using different dimensions for distinguishing between objects with same range, for example in velocity or AoA dimension.

First the case for improving the resolution of range measurement is considered. Range resolution depends on the frequency resolution of the FFT transform. When resolution is not enough, then the small difference in the two IF signal frequencies is lost, as they are bulked into one peak in the frequency domain.

This FFT frequency resolution $\Delta f$ is inversely proportional to the observation period $T$ of the signal Equation (11) [20].

$$\Delta f > \frac{1}{T} \tag{11}$$

For the IF signal this period of observation is the duration of the transmitted chirp $T_c$. This is because the travel delay time $\tau_d$ is negligible compared to the duration of the chirp signal transmission time and can be thus approximated infinitely small. Thus, two IF signals can be resolved if the IF signal duration is long enough. Increasing the period of the IF signal can be done by increasing the time that the transmitted and received chirps overlay. This can be done in two ways: by increasing the bandwidth of the chirps or increasing the chirp duration [20].

The second method, compared to improving the range resolution of the radar to distinguish two objects is done by differentiating them in the dimension of velocity. Measuring velocity of a single object can be achieved by transmitting two chirps with the interval of $T_c$. The resulting IF frequencies will have a peak in the same location, however the phase of the IF signal will vary. This phase variance is not unambiguous for determining the speed of the object, because the phase will overlap in pi radian when dealing with constant tone sinusoidal IF. Because of this using this method is only suitable if the absolute phase shift is less than pi radians. This sets the upper limit of the maximum velocity measured by that specific chirp pair for one object to pi radians to avoid ambiguities [20], [23].

For measuring the velocity of multiple objects, instead of transmitting two chirps in sequence, a chirp frame is used. This frame is composed of multiple chirps transmitted in sequence using Time Division Multiplexing (TDM) explained later. As the ranges of the objects are the same, the range-FFT will provide a single peak. However, for each chirp there will be a phase shift between two objects IF caused by objects motion. This can be inputted to a separate FFT from the range-FFT and be resolved as separate objects in the phase shift frequency. It is possible because there will be peaks for the most frequent phase shifts [20], [23]. These phase shifts are different if the objects have different velocities to each other. Therefore, to distinguish two objects using velocity, a sufficient velocity resolution is needed. Velocity resolution is inversely proportional to the duration of the transmitted chirp frame. The longer the frame duration, in other words the observation period of the signal, the better the resolution as was discussed with range resolution [20], [23].

### 2.3.2 Angle of Arrival Estimation

Now that the different objects with the same range have been resolved using velocity, they need to be placed somewhere. This is done with the AoA. It signifies the angle between the bearing of the radar and the object. It is resolved by using the phase shift of the reflected signals detected by multiple receivers. As mentioned before, change in distance of the object caused phase shifts in the peaks of the range or velocity FFT. This is used by placing the receivers at constant distance from each other and thus causing a constant phase shift between the received chirps with the distance l as seen in Figure 6. This controlled phase shift can be distinguished using FFT, from which the AoA can be estimated [20], [24].

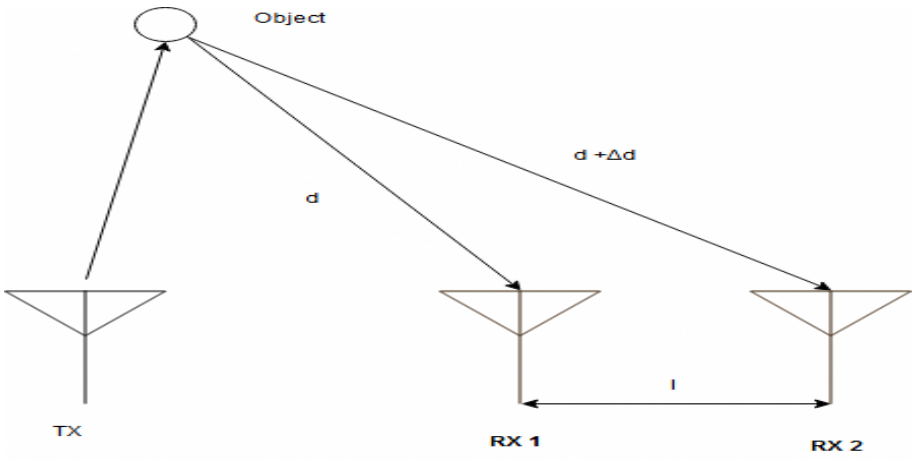



Figure 6. Geometrical representation of AoA estimation methodology.

This phase shift can be expressed as the change in distance caused by the spacing of the antennas divided by the wavelength as seen in Equation 12.[20], [24]

$$\Delta\phi = \frac{2\pi\Delta d}{\lambda} \tag{12}$$

Assuming planar geometry $\Delta d$ can be written from Figure 6 as a dependent of the angle of arrival $\theta$ Equation (13) [20], [24].

$$\Delta d = l\sin(\theta) \tag{13}$$

Using Equations 12 - 13, angle of arrival can be expressed as Equation 14 [20], [24].

$$\theta = \arcsin\frac{\lambda\Delta\phi}{2\pi l} \tag{14}$$

Since $\Delta\phi$ depends on the nonlinear sine, when $\phi$ has a small angle, the approximation of $\sin\phi \approx \phi$ can be done. [20], [24]

Because of this approximation, the angular estimation is more accurate, the nearer the object is to the bearing of the radar as seen in Figure 7 [20], [24].

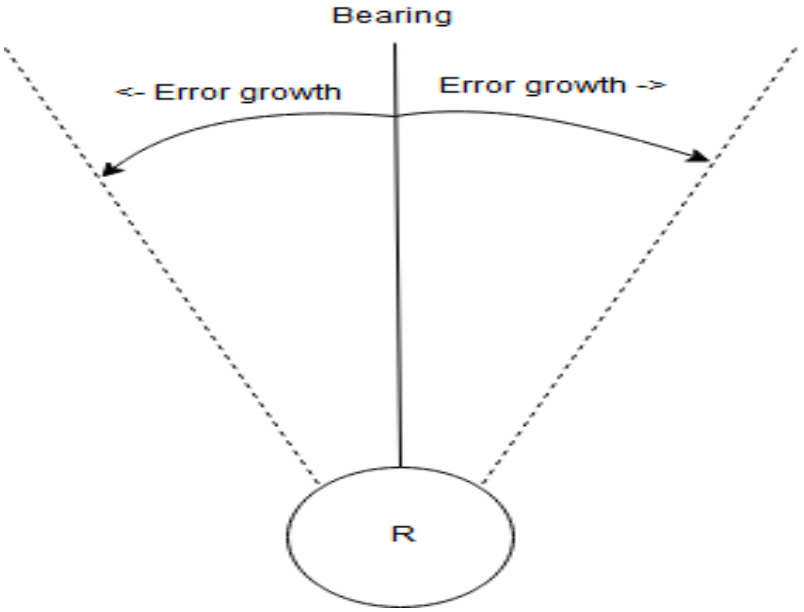



Figure 7. Geometrical representation of the change in angular error.

The maximum field of view for the radar is a crucial parameter to consider when using a radar for mapping applications. This is defined by the spacing of the antennas $l$ and the speed of light, as shown in Equation (15) [20], [24].

$$\theta_{max} = \sin(\frac{\lambda}{2l}) \tag{15}$$

Equation (16) represents the inversely proportional relationship between the angular resolution ($\theta_{RES}$) and the number of receiving antennas ($N$) used by the radar [20], [24].

$$\theta_{RES} = \frac{2}{N} \tag{16}$$

Thus the angular resolution can be improved by adding multiple receiver antennae to the transceiver of the radar. This is a very important parameter for a mapping application, as the accuracy of the dimensions of the map depend on the accuracy of the sensor used for that purpose. Angular resolution defines the horizontal resolution and thus accuracy of the map in the horizontal dimension relative to the radar [20], [24] In the following section, the methodology for resolving objects in a multiple transmitter and receiver configuration is considered to provide a better resolution.

### 2.3.3 Multiple Input Multiple Output Radar

The usefulness of placing multiple receiver antennae was covered in previous section, as it improves the angular resolution of the detection of objects. However the physical dimen-

sions of Integrated-Circuits (IC) are limited and therefore keeping the physical number of antennae minimal is necessary. This causes the need for a technique to create the same effect of adding antennae, but without doing it physically. A feasible way is to add virtual antennae to the processing chain by careful placement of the physical antennae and adding more transmitting antennae seen in Figure 8.

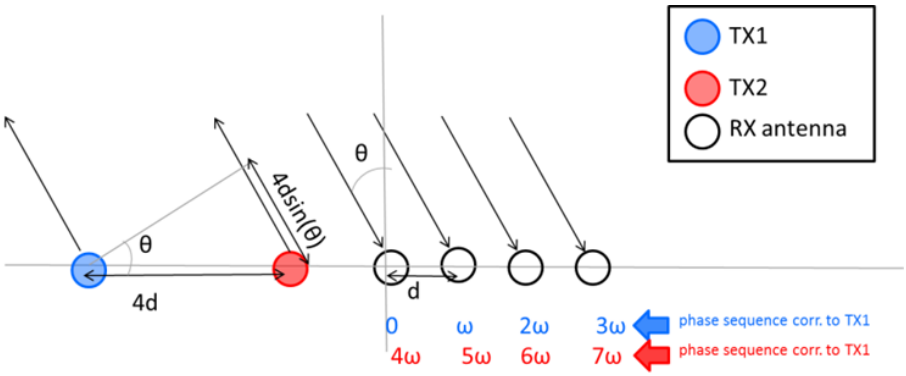



Figure 8. Basic MIMO configuration (2TX 4 RX) [25].

This method will be considered in this section compared to adding more physical receivers or cascading multiple transceivers to improve the resolution [25]. Adding one transmitting antenna to the previously mentioned 1 TX 4RX configuration has the same effect as composing a 1 TX 8 RX configuration. The computational complexity remains roughly the same, as with a single transmitter, because it uses the same phase shifts mentioned in the angle of arrival estimation chapter. By constructing a transceiver with a predetermined distance between the multiple TX creates a virtual RX array in conjunction with the physical array with which a angular resolution improvement is made. This method, however, requires the introduction of multiplexing the transmitter antennae transmission process to differentiate between the multiple transmitters. That is because when keeping the receiver array member number the same, there is no way for the processing chain to distinguish, by which transmitting array the reflected signal originated. The ability to distinguish between transmitters is however necessary to attribute the correct phase shifts and thus properly do the FFT [25]. The literature offers multiple waveform multiplexing solutions for this. There are mainly 2 methods used for multiplexing transmissions in FMCW radars: TDM and Binary Phase Modulation (BPM) [25].

For TDM the division is done in the time domain in blocks. A block consists of a sequence of chirps. Each chirp block is transmitted multiple times during a time window. The windows repeat during a chirp frame. Doing this offers separation of the chirps by using time which offers the benefit of using multiple transmitters to improve the resolution (Figure 9).

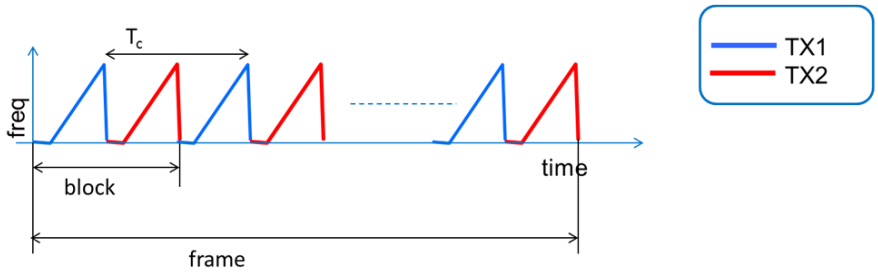



Figure 9. TDM-MIMO chirp scheme [25].

In addition, the scheme is simple to implement, but does not use the transmitter array efficiently. During transmission, only one transmitter antenna is active at a time. This efficiency loss can be mitigated using different modulation schemes such as BPM. This downside however, does not have a detrimental [25].

## 2.4 Pose estimation and mapping

Due to noise and uncertainty, perfect world modeling and capturing using sensors is not possible. There is always uncertainty and errors when dealing with actuation and measuring the results. Probabilistic representations give a good general method to incorporate as much information about the position and measurements taken by the robot, given the uncertainty of actuation and sensing. This can be done by using PDFs to incorporate a varying range of measurements into a distribution and thus express their occurrence in the internal world model of the robot with probable accuracy. PDFs incorporate uncertainty in the measurement by their nature, using variance and co-variance. Probabilistic models can thus be used to provide a better estimation compared to when noise and uncertainty is ignored.

### 2.4.1 Probabilities

In probabilistic robotics, sensor measurements, state of the robot and the environment, in the form of maps and process models, are described using probabilistic inference which in turn describes random processes using probabilistic laws. Those laws represent the dynamics between random variables and processes. In other words, random variables are a way to represent the state of random processes and their outcomes [26].

For example let us consider a random variable $X$ and its specific value that it takes as a result of a random process as $x$. For a coin flip the possible results are binary: head or tails. In this case the probability of either happening can be expressed as $p(X = tails) = p(X = heads) = 0.5$. This collection of possible outcomes are referred to as the probability distribution of the random variable as it fully defines all the possible

values that the random variable can take and each and every probability with which they occur in that distribution.

Probability distributions generally divide into two, based on the nature of the underlying processes: discrete and continuous. The case often seen in nature is a continuous probability distribution which is described by probability density functions as seen in Equation (17) [26], [27].

$$\int_x p(X = x)dx = 1 \tag{17}$$

For the set of random variable values to be a valid distribution it must add up to 1. This is verifiable intuitively as all the possible values of the random variable should define the whole probability range of the random variable noted by integrating over all of its possible values. The reason is that the distribution has to conform to the axiom of total probability and each probability has to be positive ($p(X = x) \geq 0$) [16]. When a probability of a value $x$ happening needs to be ascertained from a distribution, integration over a interval is required. The probability of variable landing inside the interval $[a,b]$ is a definite integral expressed in Equation (18) [16].

$$p(a \leq x \leq b) = \int_a^b p(x)dx \tag{18}$$

This integration can not be applied to the random variable itself by integrating in the interval $[x,x]$, because the integral would be zero. Integration over a larger interval is useful in robotics, since there is always an uncertainty associated with sensor measurements. In addition, sensors are not infinitely accurate and their precision is often given in discrete steps, providing a integration interval for Equation 18. One example of a probability density, and the one most relevant to this thesis is the normal density with the basic shape expressed parametrically in Equation (19) [27].

$$f(x) = e^{-x^2} \tag{19}$$

This shape is one of the most frequently occurring in nature and thus would be appropriate to express the processes and their uncertainty. Therefore, the term density is used over distribution as only the continuous case is considered, as opposed to the both cases covered by the term distribution [26]–[28].

To create the complete and more general normal density, normalization is required to ad-

here to the condition expressed in Equation (17), that the density integrates to one. This can be added by defining the curve using the mean $\mu$ and variance $\sigma^2$ of the random variable and then multiplying the curve with the normalization term expressed in Equation (20) [27], [28].

$$f(x,\mu,\sigma^2) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \tag{20}$$

The normalization term here is $\frac{1}{2\pi\sigma^2}$. The resulting standard curve with mean zero and variance of one is often referred to a Gaussian or the bell curve. In graphical form can be seen in Figure 10 [27].

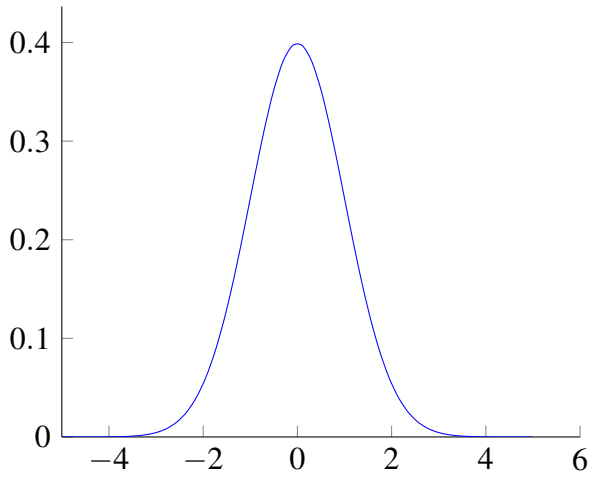



Figure 10. Standard normal (Gaussian) PDF ($\mu = 0$, $\sigma^2 = 1$).

The defining parameters of the Gaussian is thus the mean $\mu$ of the random variable distribution and its variance $\sigma^2$. The mean specifies the peak value of the distribution, in other words the most likely value of the random variable in the distribution seen in the first line of Equation (21) [27]. This most likely value has the highest probability of happening. The second line Equation (21) specifies the computation of variance, which describes the uncertainty, or the the dispersion of all the values of the random variable.

$$\begin{aligned} \mu &= \tfrac{1}{N}\textstyle\sum_{i=0}^{N-1} x_i \\ \sigma^2 &= \tfrac{1}{N}\textstyle\sum_{i=0}^{N-1}(x_i-\mu)^2 \end{aligned} \tag{21}$$

When modeling, multiple variables are needed to parametrizise the environment. The amount of variables can vary and grow exponentially quickly. This introduces the need to express Equation (20) in a multidimensional way. This is done using joint probabil-

ity. Joint probability is the multivariate version of the univariate probability distribution seen in Equation (18). For example, the case for two random variables, the probability probability that the $X$ takes the value $x$ and $Y$ the value $y$ simultaneously can be seen in Equation (22) [26].

$$p(x,y) = p(X = x \; and \; Y = y) \tag{22}$$

Another important definition, when talking about the probabilistic robotics is conditional probability expressed in Equation (23). It describes the chance of variable $X$ assuming the value $x$, given that $Y$ has the value $y$. Conditional probability helps in describing the probabilistic relationship of sensor measurements and their causes, for example an entity occupying an environment. The random variable $Y$ is called background information as it gives information about the cause of $X$.

$$p(x \mid y) = p(X = x \mid Y = y) \tag{23}$$

When $p(y) \geq 0$, then conditional probability can be expressed as in Equation (24) [26]. It is important when defining the Bayes rule in later sections for it defines the relationship between conditional and joint probabilities. The form in Equation (24) is useful as it helps to simplify the relations between sensor measurements for incremental updating of an acquired map, discussed in section 2.4.2.

$$p(x \mid y) = \frac{p(x,y)}{p(y)} \tag{24}$$

Given that the variables $x$ and $y$ are independent, then Equation (24) can be expressed as Equation (25) [26]. In other words, the independent variables do not give information about the occurrence of the other and simplifies the relations of measurements and the robot state described later in the thesis [26].

$$p(x \mid y) = \frac{p(x)p(y)}{p(y)} = p(x) \tag{25}$$

For expanding the single random variable $x$ in Equation (19), it is replaced by the vectors of random variables $\mathbf{x}$, mean $\mu$ and the matrix of co-variance $\sigma$ in Equation (26) [26].

$$f(\mathbf{x}, \mu, \Sigma) = \frac{1}{|2\pi\Sigma|} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{26}$$

The matrix $\Sigma$ in Equation (26) signifies the variances between the distributions of random variables expressed by the joint density function. In simpler terms it refers to the correlated variance of two variables [28]. For the multivariate case, the co-variance is expressed as a matrix of correlated variances seen in Equation (27) [28]. The rows and columns signify the deviation between all the random variables from $1...n$ and the diagonal the variance of the random variable itself. The variance is by definition the deviation of the random variable in its distribution.

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,n} \\ \sigma_{2,1} & \sigma_2^2 & \cdots & \sigma_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n,1} & \sigma_{n,1} & \cdots & \sigma_{m,n}^2 \end{bmatrix} \tag{27}$$

The probability of the same outcome given distinctly different causes is expressed in Equation (28) [26]. Theorem of total probability simplifies the calculation of a measurement occurrence probability caused by the occupancy of space and will be described in more detail in section 2.4.7. It is done by integrating multiple causes into one outcome using total probability Where $p(y)$ is the prior, $p(x)$ is the posterior and the conditional $p(x \mid y)$ is the likelihood of $x$ happening given $y$ has happened.

$$p(x) = \int p(x \mid y)p(y)dy \tag{28}$$

### 2.4.2 Probabilistic robotics

Conditional probability helps in defining the most important formula in the context of probabilistic robotics, named Bayes rule. This is done by applying commutativity ($p(x,y) = p(x \mid y)p(y) = p(y \mid x)p(x)$) to the conditional probability in the form of Equation (29) [17], [26], [29].

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} \tag{29}$$

In other words, the Bayes rule is a recursive way to incorporate new information into a distribution given the previous value and the occurrence of new information as it maps the conditional $p(x \mid y)$ to its "inverse" $p(y \mid x)$ as seen in Figure 11 [26]).

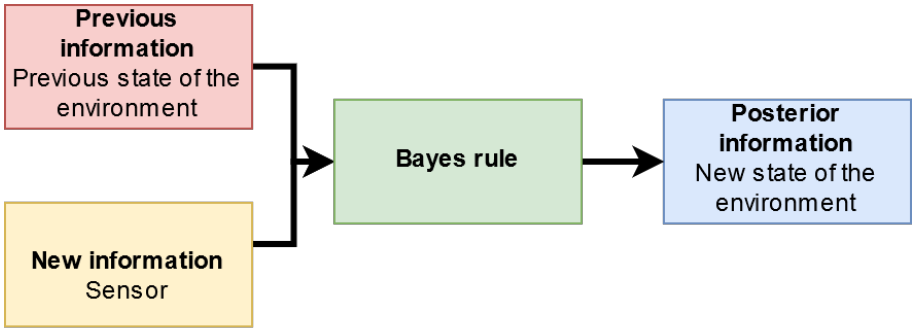



Figure 11. Bayes rule as the incorporation of new information based on the causal relationship between measurements and state.

This is useful since the likelihood of an event occurring is easier to predict than the the end result. In terms of robotics it helps to quantify a value $x$ from sensor data $y$. It holds great importance in occupancy grid mapping where constant updating of information is required in an iterative way [26], [29].

The Bayes rule can be considered a generative model [26] used to map the state into the measurement. It is often referred to as the model to map the state that causes the measurement $p(z,x)$ into the diagnostic probability $p(x,z)$. The diagnostic probability diagnoses the state $x$ based on the occurrence of the measurement $z$ in inverse[29]. This diagnostic property is what is meant when using the term mapping to the inverse in Equation (29) description. The Bayes rule can be extended to add an arbitrary number of conditions (background information) and the underlying formula and inference stays the same as in Equation (30) [29].

$$p(x \mid y,z) = \frac{p(y \mid x,z)p(x \mid z)}{p(y \mid z)} \tag{30}$$

The usefulness of this recursiveness, expansiveness and application will be discussed in the next chapter in the form of Extended Kalman Filter (EKF) state estimation using probabilities. Using this chapter as a basis, the main models used in estimating the robots state are: the state transition probability or the motion model: $p(x \mid z,u)$ based on measurements $z$ and controls $u$. The other model is the sensor model for mapping $p(z \mid x)$. The estimation of the motion will be covered in section 2.4.3 the probabilistic sensor model in section 2.4.7.

### 2.4.3 Probabilistic State Estimation

The previous chapters can be combined into a single concept of a robots state consisting of its pose from section 2.1.1 and sensor measurements described in section 2.2. This section

describes the basic probabilistic methods for estimating the robot's pose for solving the problem. Probabilistic state estimation aims to provide a more accurate estimation of a robots by fusing multiple sources of information using Bayes filters.

This fusion is done by predicting state measurements into the future based on process models that best describes the robot's movement, the corresponding uncertainty and noise. When a measurement of the robots state is done it is incorporated into the predicted measurement by a correction step. These two steps: predicting and correcting are what compose Bayesian filtering. As previously stated, in principle, the surrounding environment is Gaussian in nature. This Gaussian model can be estimated using a subset of Bayesian filter called the Kalman filters. Bayes filtering will be briefly described along the Kalman filter as the main way for estimating the location of a mobile robot.

Other Bayesian filters exist like Particle filters but they will not be considered because of their shortcomings. The downside of the particle filter is it does not provide a single location estimate, but a multimodal distribution, and makes the separate case of using occupancy grid mapping more resource heavy. This is because a map must be built for every possible location that the Particle filter outputs. Therefore, Particle filters are not considered due to their complexity of implementation and resource requirements.

When using Bayesian filtering to estimate the state of a mobile robot, the Markovian assumptions is made. Markov assumption, often called Markov property or complete state assumption, assumes that the currently measured state is independent of previous states. In other words it assumes that the currently observed state is sufficient to provide an accurate estimation of future states. This is the underlying assumption when using Bayesian inference for state estimation. Equation 31 expresses the Markov assumption made for a sensor model of the robot. In other words the measurements, given all past and current state $x_{0:t}$, past measurements $z_{1:t-1}$ and all control commands up to and including the current one $u_{1,t}$, the assumption is made that $z_t$ can be estimated unconditional of these past parameters except the current state, resulting in the right hand side of Equation (31) [26], [29].

$$p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1,t}) = p(z_t \mid x_t) \tag{31}$$

This is important when using occupancy grid mapping to build a map, as each cells occupancy probability is assumed independent by using the Markov assumption. This eases the computation of the measurement model The same assumption is made for the pose $x_t$ in Equation (32).

$$p(x_t \mid x_{1:t-1}, z_{1:t-1}, u_{1,t}) = p(x_t \mid x_{t-1}, u_t) \tag{32}$$

The Markov assumption can be satisfied by making few assumptions about the environment. Namely, the previous states can be discarded if the environment is assumed static, noise is independent and the model of the environment is perfect [29]. This assumption of completeness of information imposed by the Markov assumption is often violated in practice. It manifests when unmodeled dynamic variables or errors occur in the approximation of probabilistic models. However Bayesian filtering still remains robust and up to the task, despite these violations and thus is the most widely used method for robotic state estimation [26]. The Bayes rule can be used to express the the belief in the state of the system using conditional probabilities. Belief defines the most likely state $x_t$ of the robot given the sensor measurements $z_{1:t}$ and control inputs $u_{1:t}$ as seen in Equation (33) [26].

$$bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t}) \tag{33}$$

The belief, in contrast to a probabilistic distribution, does not sum up to one and therefore requires normalization. The Bayes rule can be applied to previous measurements to predict the next state. Thus the belief of state occurring in the future $x_t$ can be expressed as the conditional in Equation (34).

$$\overline{bel(x_t)} = p(x_t \mid z_{1:t-1}, u_{1:t}) \tag{34}$$

This is called the measurement measurement update or the correction, that predicts the state based on previous posterior [26]. Now, when applying Equation (33 to Equation (34) a Bayesian filter step called the correction or measurement update is made. Correction incorporates new measurements after completing the newest control input $u_t$.

This can be summarized in a single algorithm seen in Figure 12. It takes the previous belief in the state of the system $bel_(x_{t-1})$, latest measurement $z_t$ and control signals $u_t$ as an input. Then, for each variable of the state vector $x_t$(line 2 to 5) the prediction (line 3) and correction (line 4) steps are made. Line 3 showcases the prediction of the new state by the usage of the Theorem of Total probability, by predicting the future based on control inputs $u_t$ and previous states $x_{t-1}$.The probability $p(x_t|u_t, x_{t-1})$ is called the process model and will be discussed later. Line 4 shows the correction of the prediciton by applying the Bayes rule to incorporate measurement $z_t$. The term $\eta$ is the normalization term for the belief to be a distribution and add up to one [26].

This form of predict and update cycle from Figure 12 can be assigned to various dynamic

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

Figure 12. Bayesian filtering algorithm [26].

linear systems. The state transition probability $p(x_t \mid u_t, x_{t-1})$ in line 3 of Figure 12 requires a linear model to transition between the previous state $x_{t-1}$ and the current one $x_t$ given the control inputs $u_t$ as in Equation (35) [26], [28].

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \tag{35}$$

Where $A$ is called the state transition matrix. Matrix $B$ signfies the model that converts the input $u_t$ into an effect on the system. The term $\varepsilon$ describes the zero mean Gaussian uncertainty that comes with state transition that has the covariance matrix of $R_t$. These are again stochastic variables, in contrast to the state variables of other dynamic systems. Random variables have different rules for addition and multiplication compared to differential equations [26]. This transition matrix is the Newtonian kinematic equations in differential form for the odometry model of state estimation and the inverse sensor model for the mapping, both discussed later.

### 2.4.4 Kalman Filtering

Kalman filter is a subset of Bayesian filtering for continuous state spaces and is not applicable to discrete or hybrid states [26]. One thing missing for describing the full Kalman algorithm for a Gaussian, is the representation of the transition matrix to map the state $x_t$ to the measurement $z_t$, in other words the measurement update step of the Bayes filter. In Equation (36), $C_t$ is the measurement model and $\delta_t$ describes the measurement noise. The noise is with zero mean and covariance of $Q_t$.

$$z_t = C_t x_t + \delta_t \tag{36}$$

Equations (35-36) describe the predict and update steps of the Bayes filter and can be algorithmically expressed as in Gaussians for Kalman filter algorithm in Figure 13. In Figure 13 the full algorithm can be seen. Inputs defined are the previous mean vector of the Gaussian distribution, its previous covariance matrix $\sigma$ and the control inputs $u_t$ and current measurement $z_t$. Lines 3-4 describe the prediction of the new belief of a Gaussian

representation of the state of the system. Line 3 predicts the new mean using the state transition model from Equation (35). Line 4 predicts the new covariance matrix based on the state transiton model $A$ and its uncertainty $R_t$. Lines 6-8 depict the correction step of a Gaussian filter. Line 6 computes the Kalman Gain which is later used to scale the importance of the measurement versus the prediction. The Kalman gain is constructed using the measurement model, which incorporates all the uncertainty the measurement has. When the uncertainty is large the Kalman gain scales the final estimated mean and covariance more to the side of the prediction. In the case of large uncertainty of the state transition, it is scaled to trust the measurement more. Lines 7 and 8 update the mean and covariance based on the scaling provided by the Kalman gain and the newly acquired measurement $z_t$.

1 Algorithm Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
2 Prediction:
3 $\overline{\mu}_{t-1} = A_t \mu_{t-1} + B_t u_t$
4 $\overline{\Sigma_t} = A_t \Sigma_{t-1} A_t^T + R_t$
5 Correction:
6 $K_t = \overline{\Sigma_t} C_t^T (C_t \Sigma_t C_t^T + Q_t)^{-1}$
7 $\mu_t = \overline{\mu_t} + K_t (z_t - C_t \overline{\mu}_t)$
8 $\Sigma_t = (I - K_t C_t) \overline{\Sigma}_t$
9 return $\mu_t, \Sigma_t$

Figure 13. Parametrizised Kalman filtering algorithm [26].

Thus the Kalman filter assumes that all its components can be modeled by Gaussians which requires three assumptions [26]:

- State transition probability must be linear (Equation 35),
- Measurement probability must be linear (Equation 36),
- Initial belief of the state must be normally distributed (Gaussian).

If this is satisfied, then the resulting distribution after each iterative Kalman filter step is a Gaussian and Kalman filtering can be applied [26]. As the environment is not linear, the need to linearize the models around us is required to apply Kalman filtering. This is expressed in the form of Extended Kalman Filtering (EKF). Extended Kalman filters approximate the unlinear environment around a working point using linearization of differential equations. This is achieved by Taylor series expansion.

Using Extended Kalman filtering requires a state transition matrix and this is not always possible. This is due to the complexity of linearizing complex and highly nonlinear pro-

cess models. However, the scope of this thesis is movement on a two dimensional planar environment and this can be approximated using Taylor series expansion. This is done by approximating Newtonian equations of motion to compose a process model for the robot, leaving EKF as a valid choice [28].

Gaussian distribution based models, have certain limitations when using them to model the environment. In principle, world is not entirely full of normal distributions and this includes sensors and their noise. This can be a problem with small sample sizes, and is often the problem for Gaussian based inference when the distribution is not a Gaussian. However, using the central limit theorem it can be treated as such under the condition that a sufficient amount of samples is provided to accurately estimate the mean of the distribution. The other problem is that the non-linearity of the process in the world. Approximations are often required in the form of linearization around a working point when modeling a process for probabilistic inference. This leads to complex models that are computationally difficult to solve [28].

### 2.4.5 General Mapping Problem

Before describing how to build maps and what are the challenges in doing so, a representation of maps must be given to define the steps of composing it. Literature provides different ways to represent maps in autonomous robots. In practice, two most researched and used map representations are as follows [30]–[32]:

- Metric - different methods of map representation: feature based, volumetric maps, both are location-based.
- Topological - maps represented as topological graphs.Nodes in the graph correspond to places, situations or landmarks that are distinguishable. Direct-path arcs depict movement from location to location.

Feature based maps shape the environment at specific locations: places where there are objects. Geometric shapes and landmarks are often used to represent these features.The features must be unique in order to reduce uncertainty when later identifying them to provide localization. Identification is important to update the map at a later time step as many mapping algorithms depend on landmarks to localize itself. Volumetric maps describe every possible location in the enviroment. This is done by modeling the space in the environment as occupied or free. In other words,it specifies locations where there are entities and where there are none, covering all of the mapped area [26]. One classic example of volumetric maps are occupancy grid maps seen in Figure 14, which are location based. The world is discretizised into grids and their occupancy is determined

probabilistically. In contrast to the Kalman filter based solutions, Occupancy Grid Maps (OGM) are purely for building a discretizised representation of a environemnt and thus requires information of location from external sources [33], [34].

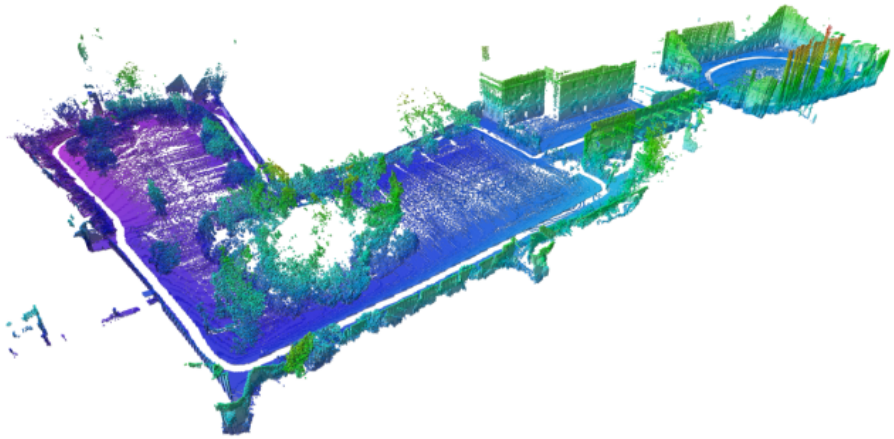


Figure 14. Example of a 3D volumetric grid map [35].

The main issues and problems faced when creating a map using autonomous robots as found in [34] and [26] and are described here briefly. When the robot traverses a path, it collects odometry data from various sensors to perceive the environment and reason about its location. The received data at its best is an approximation of the surrounding world and is littered with growing uncertainty and error as the time goes on. These errors accumulate because of previous error. Therefore, measurement noise is not independent of previous noise. This means that the dependency has to be taken into account when creating a map, otherwise it is considered trivial when noise and its dependence is ignored entirely. The dependence of noise is largely the main reason that mapping is still an active topic of research [36]. A problem with the dependence of states is the same for the locations of the world being mapped. One of the defining aspects of the surrounding environment is that it does not remain static. Both problems of dependent noise and states are mitigated by simplifying the environment to a static one and independent noise and this thesis makes the same simplifications.

The second difficulty is the size of the space where the hypothetical map can reside in. As an exact representation of the environment using computers is often not feasible nor possible, a discretized representation is used to approximate the environment. This is done in the terms of determining the occupancy of a grid cell in a approximate discrete grid representation of the location. However, this does not solve the problem as the number of variables is still large when the defined topological entities are detailed enough. This is often required when creating accurate maps, even on a planar surface.

The third problem is that mapping is a "chicken-or-egg" problem, which requires simul-

taneous problem solving to localize and map the environment, often referred to as simultaneous localization and mapping (SLAM). Mapping and localization can be divided into discrete and smaller problems on their own. Localization with reasonable accuracy can be achieved when an exact map of the environment exists. This is also true for mapping: an exact location is required to map an environment. In reality, however, having either present is unlikely when talking about autonomous robots and is therefore the final goal.

Fourthly, the correspondence of measurements - referred to as a data association problem. This can be described as comparing two measurements at different points in time and determining if they represent the same object or a topological entity. Data association is arguably the hardest problem to solve and no single representation is proposed by literature. The difficulty is largely caused because different sensors perceive the world differently and composing universal solution is highly difficult. The last case is robotic exploration. This includes finding the optimal trajectory to map a location. It is a problem because decisions have to be made based on incomplete information that causes the growth of uncertainty about the location of the robot.

### 2.4.6 Simultaneous localization and mapping

As mentioned previously mapping is a "chicken-or-egg" problem. It requires the simultaneous estimation of the position of the robot and the map of the environment. This dilemma is briefly called Simultaneous localization and mapping (SLAM), which is a problem that incorporates the mentioned problems into one set. It is best described by considering the case of an autonomous robot. SLAM problem arises when the robot has no knowledge of the complete map of its surroundings nor its location in it seen in Figure **??**. All the robot has is a set of measurements and control inputs [26]. In the following sections, different methodologies for overcoming the problems of solving the SLAM problem are introduced. It mainly focuses on the two most popular and well researched methods: Kalman filter based approaches and OGM using known poses [30], [32], [34].

Kalman filter approaches can be considered online SLAM algorithms as they estimate both the pose of the robot $s_t$ the map $m$ at a time $t$, in other words estimate the state vector seen in Equation (37):

$$x_t = (s_t, m)^T \tag{37}$$

The state estsimation is done by using Bayesian inference. Because of this, Kalman filters require three assumptions to hold for it to be effective in estimating the pose and map [34]:

- Process model must be linear with Gaussian noise - otherwise linearize models by Taylor series expansion resulting in a EKF.
- Perception model must be linear with Gaussian noise.
- Initial uncertainty must be Gaussian.

When these assumptions are asserted then, using conditional probabilities, the estimation of the posterior $p(s_t, m|z^t, u^t)$ is done. In other words: given a measurement $z^t$ and input $u_t$, what is the probability of observing the state vector $x_t$ composing of $s_t$ and $m$ [34], [26]. Depending on the context SLAM algorithms are divided into two. The first, called online SLAM, refers to incremental application of the Kalman update cycle that discards any previous measurement and control [26]. The other considers all previous measurements for every posterior and is called full SLAM. In addition to the state $x_t$ and map $m$, the correspondence of measurements to landmarks $c_t$ is done by modifying the posterior being estimated to: $p(x_t, m, c_t|z_{1:T}, u_{1:T})$. The posterior requires all measurements from time step 1 to $t$ and correspondences of landmarks $c$. Considering full SLAM is therefore computationally inefficient due to the sheer amount of parameters landmarks required to provide adequate localization. For example many modern state of the art EKF based SLAM algorithms require maps with tens of thousands of features, or more. The incremental nature of Kalman filter however, when discarding previous states, lends itself well for usage in online systems where constant estimation is required. The performance suffers from the scarcity of features, when not considering the computationally inefficient full SLAM [26]. Taking into account the online and full SLAM, there are two main distinctions between the assumptions made for those methods, based on correspondence:

- Mapping with known correspondence - each measurements correspondence to landmarks is known. This is the easier case and can be done in the context of radar by placing unique landmarks for localization.
- Mapping with unknown correspondence - requires optimization / correspondence estimation by using Expectation maximization (EM) algorithms to find a Maximum a posterior (MAP). Also possible to do online feature selection and map management in the form of provisional landmark list [26].

### 2.4.7 Occupancy grid mapping

This will provide an alternative to the EKF based SLAM problem solution by dividing the problem into two and then simultaneously solving them using a discretizised world model and a dead reckoning based solution to resolve localization. Occupancy grid mapping is a probabilistic method of building a map based on sensor measurement data. The method

was first introduced in the 1980s by Elfes [33]. The world is represented as grids with each grid having a probability of being occupied. Figure 15 depicts the original occupancy grid composed by Elfes for creating maps using sonar rangefinders. The crosses signify the estimated positions, in other words means of the pose of the mapping robot. The ellipses around the positions signifying the covariance of the pose distribution. The edges between the positions signify the sequence of movement, starting from the left. The map is represented in three colors: black, grey and white. The black curves signify the sonar measurements. The resulting measurements are curves because of the large uncertainty of the opening angle of sonars and other time of flight sensors. The white space signifies the free space and the grey unknown space. Usually, occupancy grid maps are two dimensional but the method is capable of mapping three dimensional space [34], [35].

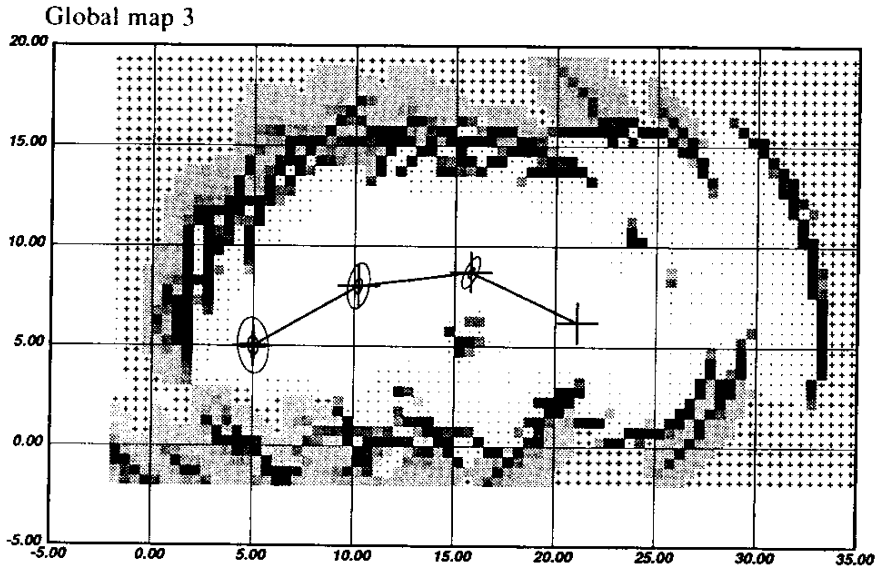



Figure 15. Occupancy grid map with relative robot positions [33].

The method lends itself well to range-azimuth-elevation type sensors by using a beam based models for determining the occupancy of each cell. This is suitable for solving the problem proposed in this thesis, as the radar can be approximated using beam based models. As occupancy grid mapping is probabilistic it can mitigate the problem of uncertainty in measurements. For example, when using sonar, a range is returned when a measurement is done inside the sonar cone. However, where the detection point resides inside that cone is not known. This uncertainty can be incorporated into the sensor model as a probability distribution. It handles the angular uncertainty by producing a possible distribution of detections inside that cone using Gaussians. Another uncertainty handled by the probabilistic approach is the cases of diffusal and specular reflections depending on the angle of the wave and material of the surface [34].

Like with the EKF based solutions, the method relies on Bayesian inference. Let a grid be described by its coordinates $(x,y)$ and the binary occupancy value $m_{x,y}$, having two possible values: occupied or free. The method requires the calculation of the Bayesian correction step over all the cells in the map, with each having a probability of being occupied: $p(m_{x,y}|x^t,z^t)$. Again this is a conditional probability, signifying the probability of cell with the coordinates $(x,y)$ being occupied [34], [26], [36].

For static maps and known poses the belief in the occupancy of a grid cell takes the form seen in Equation (38) by using Bayes rule and using the correction step of the Bayes filter. The term $s^t$ denotes the pose of the robot, $z_t$ the measurement and $m_{x,y}$ a grid cell of a map [34].

$$\frac{p(m_{x,y}|z^t,s^t)}{1-p(m_{x,y}|z^t,s^t)} = \frac{p(m_{x,y}|z_t,s_t)}{1-p(m_{x,y}|z_t,s_t)}\frac{1-p(m_{x,y})}{p(m_{m,x})}\frac{p(m_{x,y}|z^{t-1},s^{t-1})}{1-p(m_{x,y}|z^{t-1},s^{t-1})} \tag{38}$$

In Equation (38), the odds form: $\frac{p(x)}{1-p(x)}$ for the belief in the occupancy of the grid cell $m_{x,y}$ is used. The odds form depicts the odds of a random variable $x$ happening by dividing the probability of the event happening by its negate [26], [34], [36].

The second term on the right hand side in Equation (38) signifies the initial odds of the occupancy of each cell of the grid. When setting the occupancy of each cell 0.5 (unknown), this term is negated and disappears resulting in the multiplier 1 for the right hand side [34]. The third term on the right side is the *prior* probability from the previous step of the update. The resulting equation is thus Equation (39):

$$\frac{p(m_{x,y}|z^t,s^t)}{1-p(m_{x,y}|z^t,s^t)} = \frac{p(m_{x,y}|z_t,s_t)}{1-p(m_{x,y}|z^t,s^t)}\frac{p(m_{x,y}|z^t-1,s^{t-1})}{1-p(m_{x,y}|z^{t-1},s^{t-1})} \tag{39}$$

This form is the familiar Bayes filter correction step from Figure 12. The second term on the right hand side is the rightmost prior belief in the corrections step in Figure 12 line 4. This prediction is made for the occupancy of grid cell $m_{x,y}$. The first term is the measurement model depiced in Figure 12 line 4 as the measurement model $p(z_t|x_t)$.

In order to calculate the map posterior odds $\frac{p(m_{x,y}|z^t,s^t)}{1-p(m_{x,y}|z^t,s^t)}$,(line 4 in Figure 12) the only term that is needed to know to calculate the map posterior is the distribution $p(m_{x,y}|z_t,s_t)$ and is called *inverse sensor model*. The inverse model is the measurement model used by the sensor to map its detections to the map in Bayesian filters, by mapping the effect to the cause, while forward models, that signify the measurement model directly, map the cause (grid occupancy) to the effect (measurement) [34].

Figure 16 expresses the OGM algorithm for all cells expressed based on Equation (39). Forcomputational efficieny the implementations of occupancy grid mapping algoritms use log odds representation of the map, which is expressed as $l_{t,i}$ for each cell and $l_0$ expresses the initial distribution [34]. It takes the previous map belief $l_{t-1}$, state of the robot $x_t$ and the sensor measurement $z_t$ as its input. It then iterates over all the cells in the map and if the grid is in the field of $z_t$ it updates the occupancy probability as shown in line 4. As it is described in log odds form, multiplication of probabilities is substituted by multiplication and division by subtraction. If the cell is not in the perception field of the measurement, the previous occupancy probability is assigned as the new one. The inverse sensor model in the algorithm is a general function, however in reality it is specific to each sensor.

```
1 Occupancy grid mapping(l_{t-1,i}, x_t, z_t):
2 for all cells m_i do
3 |   if m_i is in the perception field of z_t then
4 |   |   l_{t,i} = l_{t-1,i} + inverse_sensor_model(m_i, x_t, z_t) - l_0
5 |   else
6 |   |   l_{t,i} = l_{t-1,i}
7 |   end
8 end
```

**Figure 16:** Occupancy grid mapping algorithm [26]

### 2.4.8 Probabilistic sensor models

Before discussing the inverse sensor model, the general representation of rangefinder sensor models for occupancy grid mapping is introduced. This model describes how measurements themselves are modeled using RAE. The inverse sensor model, in contrast, maps the measurements created by the general model inversely to a map. For rangefinders with low uncertainty the models used are called beam models. Beam models determine the effect part of the map building and can be considered sensor models, while the inverse model links the measurements to the map inversely. After introducing beam models, the inverse model is derived from it. This is relevant for radar as it can be considered a rangefinder, where the electromagnetic waves are approximated as beams. Beam-based inverse sensor models assume that measurement correspond to obstacle surfaces and the line of sight between sensor origin and endpoint does not contain any obstacles. Figure 17 depicts a single beam measurement using a RAE sensor with the measured range of $r$. The probability is depicted on the vertical axis, and the measured distance in the horizontal. The probability is one where the measurement mean is. The maximum probability spans over all the uncertainty in the range dimension, as it is not known where the measurement lies in that interval. The length of this span is defined by the resolution of measurement of the sensor and for radar it is the range resolution [33], [35].

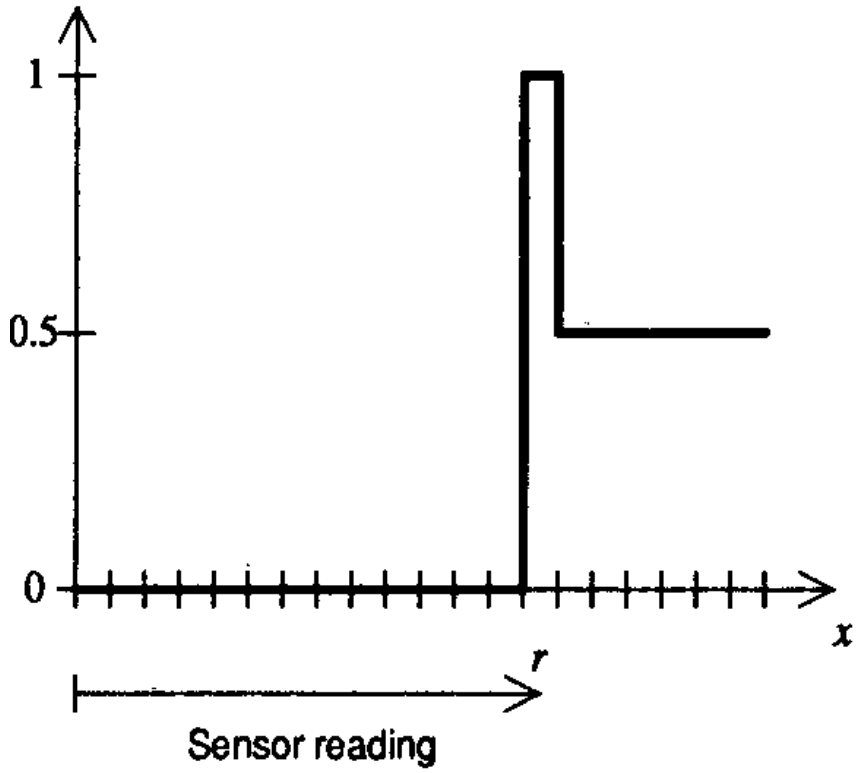



Figure 17. A single beam ideal sensor [33].

The beams are raycasted depending on the type of sensor. Ray casting simulates a beam until it hits an object, from which it reflects back to the sensor. According to literature, this is done only for the first reflection [34], [36]. Therefore, the model does not incorporate multiple reflections. Simulating multiple reflections would increase the computational load of the method and introduce dependency in measurements and their noise. This introduced dependency would cause the Markov assumption to be invalid and therefore causes the map creation to become impossible using occupancy grid mapping. The beam model using only one ray can be expanded to incorporate multiple beams in sequence, for better efficiency of the map creation process. For example, lidar has multiple beams, compared to ultrasonic rangefinders, so the beam model must incorporate multiple beams into single update cycle of the map. This is possible with Bayesian inference under the assumption that the Markov assumption between scans holds, meanin that each scan is independent and does not affect others. When taking into account the uncertainty, the beam model, when modeled using Gaussians can be seen in Figure 18. The Gaussian beam model takes into account the angular uncertainty in addition to the range uncertainty. The flat plane signifies the occupancy probability of zero. The higher the point on the plane is, the higher the probability of the measurement occurring. The angular uncertainty is expressed by the spanning of the distribution in the x dimension, while the range uncertainty is expressed in the direction parallel to the y dimension. The negative probability,

signified by the slopes in the plane, shows the growing probability of the location being occupied. The closer the cell is to the mean of the measurement the higher the probability of it being occupied.

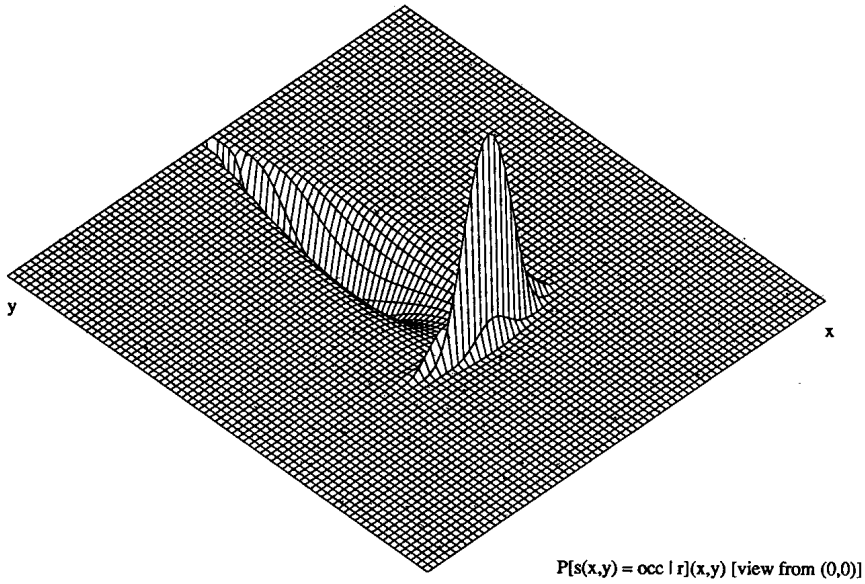



Figure 18. Geometric occupancy of a single beam Gaussian sensor [33].

As mentioned in section 2.2.1, sensor models are used to present sensors in a structured way. Occupancy grid mapping however requires differing models from direct sensor models, discussed previously, for sensor detections to build a map. This section builds upon previous knowledge of RAE model of sensors to accommodate it to a probabilistic discretizised representation of maps used in occupancy grid mapping. Still the previously proposed beam based RAE models are still crucial as the input to the inverse sensor model. Inverse sensor model maps the sensor measurements $z_t$ back to its cause grid occupancy $m_{x,y}$ as seen in Equation (40) [26], [34], [36], [37].

$$p(m_{x,y}|z_t, s_t) \tag{40}$$

This model of inference assumes that the world is static (Markov assumption is satisfied as change in the world creates dependency to the previous location), fitting the scope of this thesis. The other limitation is that this model assumes conditional independence of measurements in relation to grid cells. In other words, a single measurement represents only the occupancy of a single cell. When dealing with single beam models, this is true when noise is independent, but not with sweeping beams like sonar, lidar and radar. With lidar this is mitigated by defining the angular resolution and number of rays in the inverse model and ray casting multiple rays discussed previously. Each ray is assumed

independent, therefore satisfying the Markov assumption. Modern implementations have incorporated this into their framework to accommodate multiple beam rangefinders, like the Octomap framework [35].

This incorporation of multiple beams can be done the same way with radar model, the only difference being the reflective properties between the radar and lidar beams. Radar electromagnetic waves have the nature of penetrating wood, plastic and materials with low density, due to the wavelength that millimeter wave has. Also, there is high chance of secondary and non-line-of-sight reflections of the waves. The ray casting of the radar beams for secondary reflection detection will not be considered due to the dependency of noise and measurements it introduces. When using a multiple beam model for radar, the simplification of filtering secondary reflections can be made in place of modeling the reflections in the inverse model.

### 2.4.9 Comparison of mapping methods

When choosing a method to estimate the pose and creating a map, the two methods discussed previously: EKF based SLAM algorithms and OGM coupled with collecting odometry from different sources are considered. EKF methods require distinguishable landmarks that needs additional algorithmic processing to extract them from a radar point cloud. EKF based SLAM algorithms have a fundamental dilemma in this regard as described in [26]. When correspondence to landmarks are unknown, then maximum likelihood data association works well with dense maps that have millions of features to provide a location for the robot. However, EKF based solutions are weak in dealing with sparse maps, which is often required due to hardware constraints. In practice, they require sparse maps because the quadratic complexity of the update step. It sets unrealistic expectations for the computational power and memory requirements of a small mapping robot. Occupancy grid mapping still uses maximum likelihood data association to reason about the occupancy of every cell, but does not require uniqueness of each cell, the reason being that it does not provide localization. Because of this simplification of only providing a map, the uniqueness of grid cells is not required as the method does not need to distinguish between the first time it measures that cell and the following measurements. The resulting occupancy is independent between cells and their correlation to a previous measurement is not incorporated to the occupancy. The occupancy is only updated using Bayesian filtering. Therefore, the better method is the occupancy grid mapping method that does not require such requirements for density and correlation therefore working well for sparse measurement density. This requirement is achievable by a low angular resolution automotive radar. In that sense OGM scales better if the resolution of the map is not too high.

Other ways for creating a map using radar with EKF based algorithms using landmarks is to place highly distinguishable landmarks in the environment. For radar this can be achieved by using corner reflectors or other highly reflective surfaces. To guarantee uniqueness of landmarks, filtering based on the Signal-to-Noise ratio (SNR) of the received signal is done. This requires every corner reflector to have an unique SNR signature, which in practice is unfeasible to assume. It reduces the burden on data association side of the map construction. Providing unique landmarks does not ease the computational load of expectation maximization and the requirement for dense maps [4],[37].

Another algorithmic way of composing features is by forming shapes during mapping as presented in [1] and [3]. This solution does not provide relief for the density of the features, and even reduces the number even more, raising the uncertainty in the estimation of the map using expectation maximization. The raised uncertainty in turn causes the expectation maximization to converge slower.

On the other hand, occupancy grids do not require such restrictions. It does however, require a way of dealing with measurement error when building the map and collecting information about the relative poses of the robot. When using radars, this can be mitigated by choosing a suitable filtering configuration based on SNR and compensating for angular and range uncertainty in the inverse sensor model as discussed previously.

In general, occupancy grid mapping is unable to accommodate pose uncertainty, so an accurate positional information is required. This requires an outside source to reduce the uncertainty in placing the measurement. In addition, occupancy grid mapping assumes independent noise. It states that the noise does not influence the measurements in an accumulating way. In terms of the Bayes filter iteration, it means that noise can be applied discretely at every step according to a model and it does not have any residual effect to the next step. This requirement is due to the importance of Markovian assumption in Bayesian filtering. Despite these restrictions, occupancy grid mapping is robust and simple to implement and does not require elaborate set up of landmarks to function. Because of this, occupancy grid mapping using external robot pose data is the chosen method to generate maps in this thesis.

# 3 Implementation

This chapter will consider the hardware selection and implementation of the previously described restrictions and methods for creating an extensible mobile solution for mapping indoor environments.

When using a radar, linear movement is allowed due to the requirement of providing translation inside the environment using only radar. This is caused by the fact that radar provides only relative speed of the reflection points in relation to the radar, limiting number of possible test cases. In addition to that, the speed provided by the radar is a radial velocity as per RAE model. This imposes additional requirements to processing of the point cloud for linear velocity. One solution for acquiring robots linear velocity is to provide a projection of the measured velocities onto the bearing of the radar. This is not feasible as the AoA accuracy degrades as mentioned in section 2.3.2. This leaves a error bound in terms of degrees to filter out angles with too large deviation from the bearing to fight the rising approximation error caused by the increasing AoA.

Using point clouds as the source of linear speed introduces the need for measuring the speed of the environment with as high resolution as possible. This is defined by two parameters of the FMCW radar: the radial velocity resolution and the range resolution. The configuration of the radar will therefore focus on these two, and the generation of as much valid reflection points as possible to not influence nor degrade the map creation quality.

The chosen IMU supports the measurement of linear acceleration using an integrated three-axis accelerometer. Using this information can be considered to refine the estimation of the linear translation of the mapping robot. However, the double integration required to provide translational offset makes the accumulating error in measurements have a detrimental influence on the estimation quality. Integrating twice from acceleration makes the system unstable, which is not correctable by fusing linear speed measured by the radar using Kalman filtering.

## 3.1 System overview

This section covers the final system, taking into account the limitations and tools described in previous sections. Two different views are considered: the hardware and then the software architecture.

### 3.1.1 Hardware architecture and components

The block diagram defining the relationships between hardware can be seen in Figure 19. A Raspberry Pi 3 B+ is used as the central processing unit to run the solutions software stack, introduced in the next section. The mobile base is serviced by two Parallax continous rotation servos to provide translation for mapping. The servos offer up to 50 RPM for wheels, with linear response to Pulse Width Modulation (PWM) [38]. The resulting physical setup looks like seen in Appendix 6.

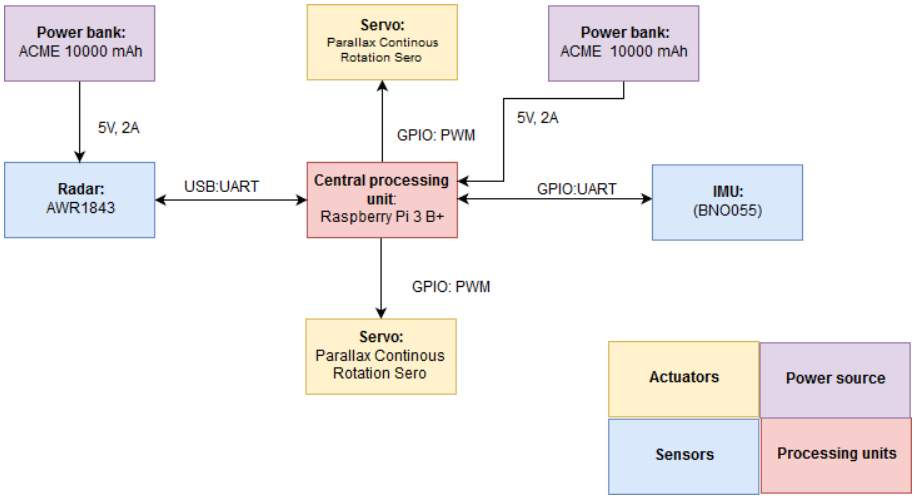



Figure 19. Hardware architecture overview.

The IMU provides 13 diferent operation modes, divided into two categories: non-fusion mode and fusion mode expressed in Appendix 8. The non-fusion modes offer raw data from the gyroscope, accelero- and magnetometers, which will be subject to further filtering to achieve relative or absolute orientation. The fusion modes filter the different sources into relative (in relation to previous orientation) and absolute orientation, depending on the configuration. These configuration can be observed in Table **??** on rows starting from IMU operating mode. The most versatile of these configurations, NDOF, provides absolute orientation by fusing all sources of information into 6DOF. Therefore, this operating mode will be used to provide orientational data. The difference between the two NDOF configurations is the power consumption and because of this the frequency at which the measurements are taken [18]. The fused sensor output can be formatted as Euler angles or quaternions. Because of their unambiguous nature compared to Euler angles, quaternions

will be used to represent orientation and rotation.

The AWR1843BOOST evaluation module provides a Printed Circuit Board (PCB) solution for the AWR1843 IC sensor module. It implements the FMCW radar technology described in section 2.3 using a PCB with a onboard-etched antennae (Figure 20).

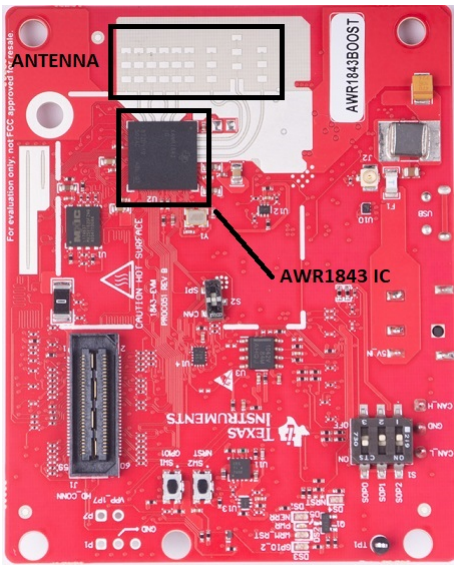



Figure 20. AWR1843 evaluation module [39].

The IC is composed of two main subsystems: the Digital signal processor (DSP) C674x and the user programmable Microcontroller Unit (MCU) Cortex R4F seen in Appendix 7. The MCU provides the interface for user programmable applications on the DSP. An example would be higher level clustering using Kalman filters for further processing of the DSP detections, which is provided by the manufacturer [39].

The processing chain of the DSP can be found in Figure 21. It provides the signal processing to acquire detections as per discussed by the FMCW radar theory provided earlier. In additon, a Constant false alarm rate (CFAR) peaking algorithm is imposed on the range FFT to eliminate false detections caused by noise, clutter and interference. The transmitted chirps are digitalized using an analog-digital-converter. The DSP front end will provide the intermediate frequencies of the recieved chirps as analog signals. The Range Proc and Doppler proc are the range and velocity resolution FFT mentioned in section **??**. The result is a two dimensional detection point, with an attached radial velocity. Neibhouring detections are peaked into a single detection based on SNR and CFAR. The peaked detection points are resolved in a third FFT determining their AoA. The final stage before moving the results into the shared memory between the front end of the ARM processor, is clustering, which is not utilized in this thesis.

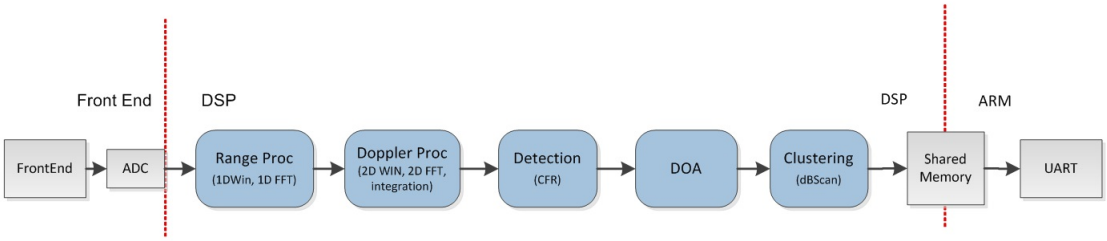



Figure 21. AWR1843 signal processign chain [40].

The evaluation module PCB has onboard-etched antennas: 3 transmitters and 4 receivers, for providing a 3D range and angle of arrival estimation (Figure 22). While the theoretical Field of View (FOV) of the processing chain is 180 degrees, the physical design of the antennae limits this to around 100. The elevation is approximately $\pm 15$ degrees from the normal angle of the antenna. The antenna are spaced with the distance of one or half wavelengths $\lambda$.

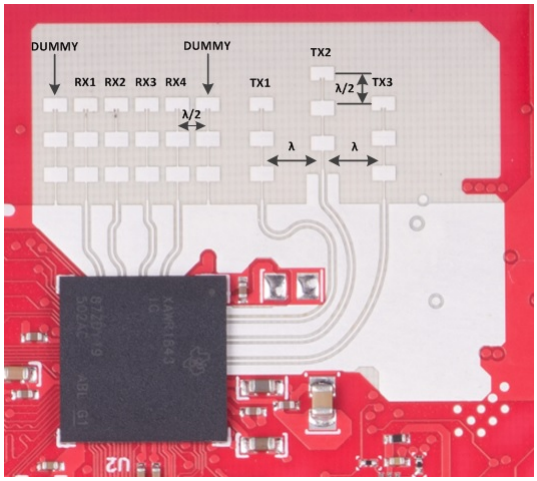



Figure 22. AWR1843 antenna configuration [39].

### 3.1.2 Software architecture and tool overview

For the map composing and robot state estimation the robot operating system (ROS) is used. ROS is a collection of software for creating robots. The software stack was first created to mitigate difficulties when developing robots and to allow fast prototyping. It provides software distributions similar to GNU/Linux distribution. These software bundles are composed of packages, that can be executed [41],[42].

ROS offers modularity in the form of nodes. Nodes are instances of ROS programs provided by different packages. Each node is either a producer or a consumer based on the context. The main units of communication and messaging are called topics. Topics offer

publication and subscription facilities for nodes. Basic information exchange scheme of the robot operating system can be seen in Figure 23.

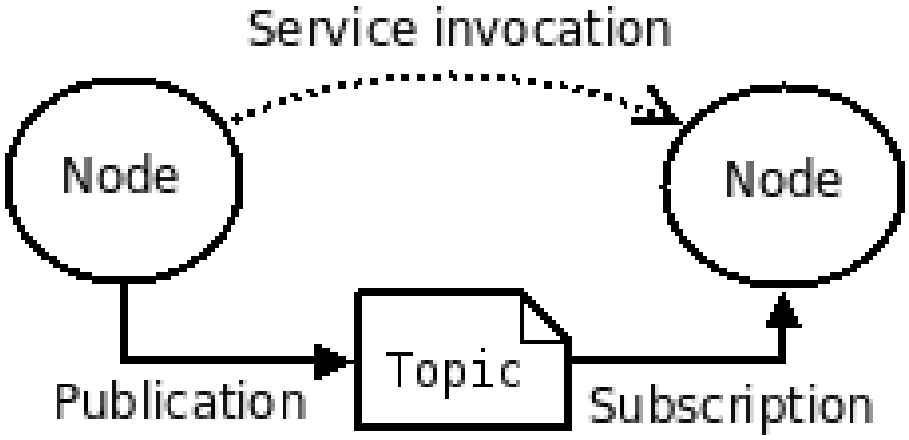



Figure 23. Basic computational graph level members in ROS [43].

ROS offers advanced modules to better facilitate more strict real-time timing restrictions in the form of services and nodelets, however they are not necessary to understand the basic concepts and capabilites of ROS used in this thesis. For estimating the linear change in the robots pose and the angular change measured by the IMU, the robot localization package is used. Robot localization is a ROS software package used to provide odometry for moving base robots by the utilization of Extended Kalman Filtering for fusing different sources of sensor information. The package utilizes a 12-dimensional state vector for 3D state estimation expressed in Equation (41) [44]. The vector provides estimation on all 6DOF by providing location data in three dimensions $X$ ,$Y$ ,$Z$, orientation of all three axis in the form of roll, pitch, yaw. It further estimates the rate of change of these translational and orientational parameters in the form of first and second order derivatives.

$$x = (X, Y, Z, roll, pitch, yaw, \dot{X}, \dot{Y}, \dot{Z}, \dot{roll}, \dot{pitch}, \dot{yaw}, \ddot{X}, \ddot{Y}, \ddot{Z}) \tag{41}$$

This provides a suitable interface for jointly estimating the pose of the robot using orientation from an IMU and linear velocity from the radar pointcloud. The package used for building an occupancy grid map is Octomap, which is a ROS compatible maximum-likelihood occupancy grid mapping framework. It uses octrees as a memory efficient way to create 2D and 3D maps alike. In addition, it finds the the most likely map considering the measurements by beam sensor model using maximum likelihood estimation. Figure 24 depicts a reconstruction of a picture using octrees [35].

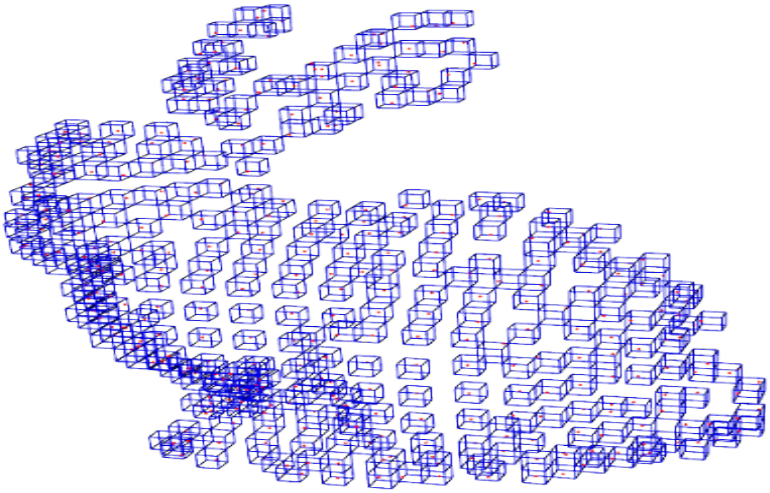


Figure 24. Octree representation of a bunny [45].

The sensor model used by Octomap is a inverse sensor model for a beam range finders. It is suitable for radars, if proper precautions are taken into consideration, as discussed previously. Three refrence frames are defined for the mapping robot: the map, odometry and the robot vehicle frame: *base_radar_link_1* in Figure 25. This is done because of the adherence requirements to REP103 [46] and REP105 [47], so that it is compatible with other software packages provided by ROS used in this thesis, namely the Octomap and robot localization packages. It is done to further facilitate reuse for future work (Figure 25).

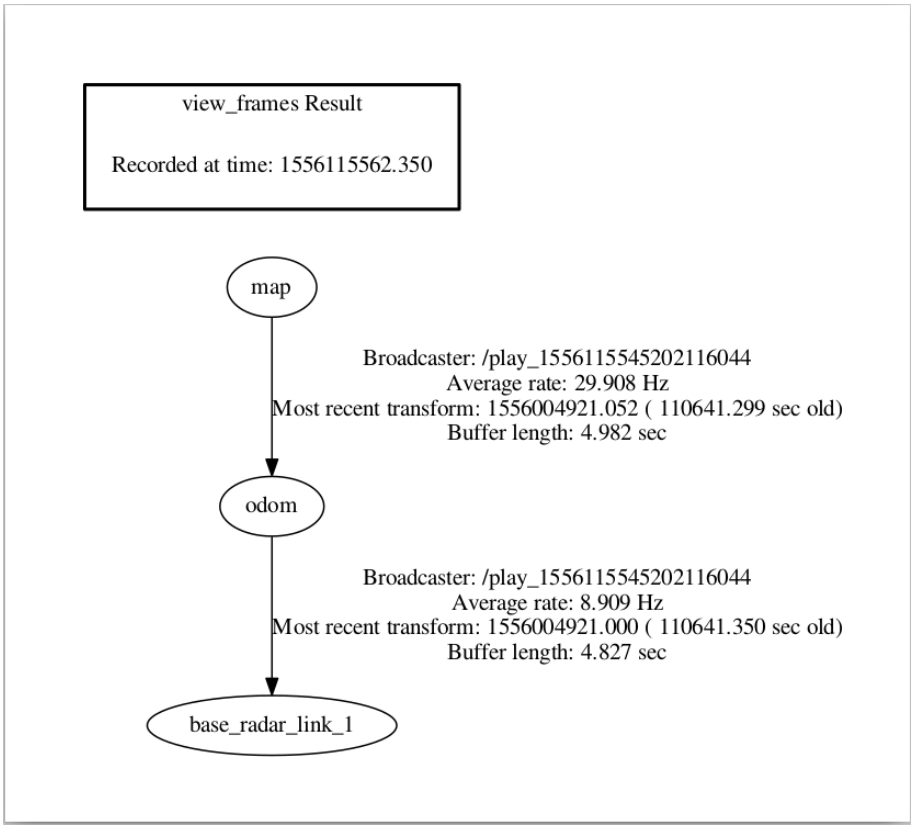



Figure 25. ROS reference frame hierarchy.

The global reference frame is *map*, which is acquired during system initialization and remains static for the whole operation period. The odometry frame *odom* is static to the *radar_base_link* frame and moves in the map frame. The dynamic reference frame is *base_radar_link*, which is the reference frame for the moving base, where all the sensors and actuators are attached. Each arrow represents a required transformation between the reference frame. The orientation transformations are published by the *transform_broadcaster* node to the */tf* topic, while the *ekf_se* node publishes the required base movement, based on the odometry messages provided by the *mmWave_octomap* node, which calculates the linear self-velocity of the base. The *base_movement* node moves the base along using PID control, where the initialization orientation is the set value. The final node is the *mmWave_Manager* nodelet provided by the radar manufacturer driver *ti_mmwave_rospkg*. [47], [48]

The full hierarchy of the system can be found in Figure 26. Each ellipse signifies a node and each square a ROS topic. Arrows where a node is the parent of a topic, signifies their publishing relationship, while the other way around signifies the subscribing nature. A more detailed figure, along with passed messages between nodes can be found in Appendix 4.

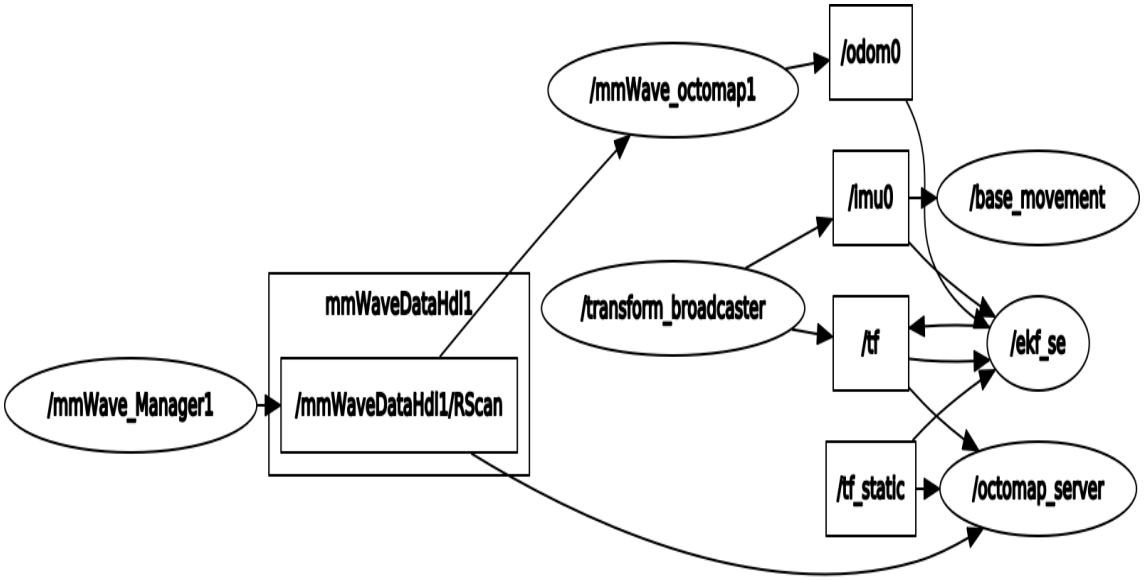



Figure 26. ROS node message passing hierarchy.

### 3.1.3 System configuration

Two sensors are used in the sensing of the state of the robot: the IMU and the AWR1843. For the IMU, the NDOF configuration mode as seen previosly in Appendix 8 is used in quaternion form as opposed to Euler angles, due to the disambiguous way quaternions represent rotations. The UART communication and data accumulation is a modified source code provided by the driver[49].

Configuring the radar requires consideration for two parameters: the determination of moving velocity of the robot and finding the suitable configuration for Octomaping package. Due to the requirement of using only radar for self-velocity of the robot and the usage of AWR1843, it further restricts the usage of reflection points for self-velocity. As the radar provides only radial velocity relative to the radar itself, there is the need to decompose it to it's components on the $x$ and $y$ axis. They are used to compose the self-velocity of the radar based on points not near the radar bearing. To compose the self-velocity, the x-axis component of the reflection points radial velocity can be used. This is done by reversing the sign of the component, attributing the velocity from the point to the self-velocity of the robot. This can be done on the assumption that the environment is static and that the radar does not move circularly. This assumption is valid under the conditions of Test case 2 in section 3.2 and thus this self-velocity is only used under that test case. The x-axis component is considered because as per REP 103, the forward facing axis that coincides with the bearing of the radar is the $x$ axis not the $y$ axis.

However, the projection onto the radar bearing of radial velocities is not feasible, due to

the low angular resolution of the radar and the increasing angle estimation error when the angle deviates from the bearing as discussed previously [50]. Therefore, only the bearing radial speed vectors are taken into consideration, where the error of angle estimation is the smallest and no projection of the vector is required.

For this certain error bounds are defined, where the angular deviation of the reflection points are inconsequential. To further improve the accuracy of the radar, the configuration will limit the y axis spread of the reflection by peaking the range detections into a single detection point using Constant False Alarm Rate (CFAR). This is achieved by enabling the Range peak Grouping and Doppler Peak Grouping in the configuratin as seen in Appendix 3. To generate as much of reflection points for estimating the self-velocity, the static clutter removal is turned off. This is done using the provided configuration interface for the AWR1843. The resulting restrictions can be seen in Figure 27.

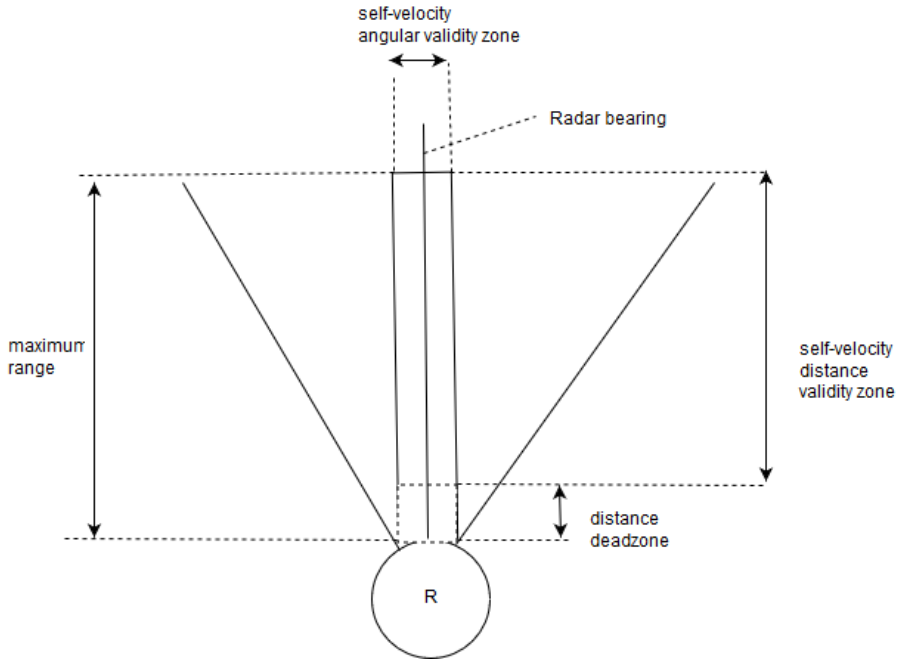



Figure 27. Self-velocity limitations.

The self-velocity angular validity zone is defined at 0,1 radians as a bound of deviation from the radar bearing. The distance validity zone is defined in meters resulting from setting the value 0,01 meters to the distance deadzone and subtracting it from the maximum unambiguous range of 16,16 meters (Appendix 3).

When using the range laser finder inverse model for approximating radar scans, proper precautions must be taken. That is because of specular reflections caused by the properties of the electormagnetic waves. Since laser range finders assume line of sight and that no specular reflections reach the sensor again, the radar's specular reflections of electromagnetic waves and non-line-of-sight propagation have to be taken into consideration.

The solution is to limit the range of the reflection points that are used in mapping, to guarantee that only the first reflections are used. Therefore, the range of the points that are taken into account are limited to the maximum length of 1 meter for both test cases (Appendix 1). For the base operation of the radar, the visualization demo binary from the software development kit version 3.1 was used [50]. To configure the radar, the mmWave demo visualizer graphical user interface was used (Figure 28) [51], [52] .

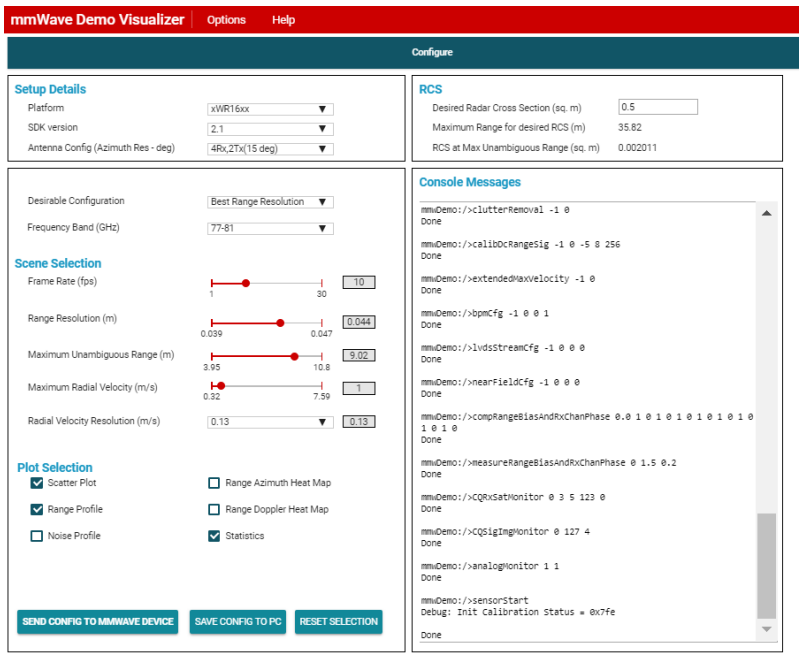



Figure 28. A screenshot of the configuration interface.

The best range resolution configuration is chosen for of the map. As seen in the radar configuration file (Appendix 3) the best possible resolution for the range dimension is achieved by the 4 GHz bandwidth option, as discussed in the FMCW radar theory. The resulting resolution is 3.9 centimeters, which is the best resolution provided by the 77-81 GHz frequency band. The resolution of the map depends on this variable and should be chosen as small as possible for most accurate mapping. For accurate self-velocity estimation, the velocity resolution is set as low as possible, resulting in the value of 0.15 meters per second. This parameter is limited by the sampling rate of the analog-digital converter. The chosen velocity resolution coincides with the moving speed of the wheels of the robot, given that the motors provide 50 rotations per minute and the circumference of the wheel is around 18,85 centimeters, resulting in the linear speed of: $\frac{18{,}85\ cm \times 50\ RPM}{60\ minutes} = 15.71\ cm\ per\ second$. This involves innacurate speed estimation during the acceleration of the robot. The influence will be minimized by accelerating to the top speed as fast as possible.

For the Octomap package configuration, each cell in the grid is a 5cm square that was

tested as the computationally smallest value that the Raspberry Pi could handle. The final configuration in Appendix 1. The chirp configuration is composed using TDM-Multiple Input Multiple Output (MIMO) chirp multipelxing scheme to provide reflection points (Figure 29)

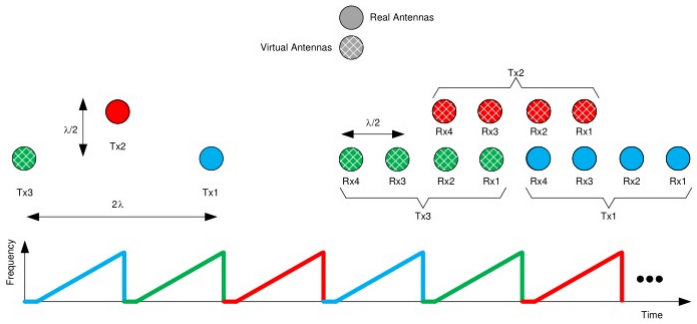



Figure 29. AWR1843 chirp multiplexing scheme and antenna configuration [40].

As presented in section 2.4.6, occupancy grid mapping requires the location of the robot as an input. In other words, a transformation between the odometry and radar frames is required because of the change of pose of the robot. In section 2.4.3, Kalman filtering can be used to fuse different sources of noisy measurements to provide a refined estimate of the robots pose. This is achieved by utilizing the ROS package *robot_localization*.

Considering the limitations presented in section 1.2, the *robot_localization* package configuration is used for the estimation of a two dimensional case, hence the two_d_mode is set to be active as seen in Appendix 2. In addition two odometry sources are provided by the usage of topics *odom0* and *imu0*. The IMU configuration requires the setting of few parameters. Since the IMU provides absolute orientation and not relative, the odometry soruce's relative parameter is set to false. In addition, as the IMU does not remove gravitational acceleration, it needs to be removed by setting the parameter *imu0_remove_gravitational_acceleration* to true. The parameters provided for the state vector for estimation are by the IMU are the yaw, pitch and roll and their rate of change in the form of first derivatives.

Since the radar provides only linear acceleration in terms of odometry, the state vector parameters are set only for the derivation of the x-component marked as ($\dot{X}$). The configuration requires the confirmation of the reference frame setup, which is discussed previously and and configured as described by in Appendix 2.

## 3.2 Experiments

This section will cover the methodology used for mapping indoor environments and experimental results. It covers the methodology used to map an office enviroment using a mobile base robot. The test scenarios are described in detail along with the analysis of the results. In addition, future improvements are discussed based on results.

For moving the mobile base, two continous rotation servo motors are used that are controlled by using PWM. For keeping the bearing straight, Proportional Integral Derivative (PID) control is used, where the system intialization orientation is chosen as the setpoint. The error for the bearing is represented as a unit quaternion Equation (42).

$$q_r = r_k\mathbf{k} + w = 0.05\mathbf{k} + 0.9987 \tag{42}$$

Equation (44 shows the experimental value of the quaternion chosen in conjunction with the PID variables, by visual tuning, where the PID control was successfully tuned to move straightwards and did not oscillate uncontrollably during the traversing of the linear path. This resulted in the PID components as seen in Equation (43).

$$\begin{aligned} K_p &= 10 \\ K_i &= 1.5 \\ K_d &= 2 \end{aligned} \tag{43}$$

Using Equation (4), the unit quaternion limitation $r_k$ can be translated into Euclidean angle, which represents the bounded error around the z-axis of the moving robot base *frame base_radar_link_1* as in Equation (44).

$$2\arccos(w) = 2\arcsin(r_k) = 5.74^\circ = \pm 2.87^\circ \tag{44}$$

### 3.2.1 Circular motion

The traversing of indoor environments compose of two separate actions: linear motion in the dominant direction of movement. Inn other words, along corridors, and circular motion to traverse between the dominant directions of movement. Based on this, two experiments will be defined for these occurrences separately. An occupancy grid map will be built. The results will be assessed visually. The first test case will consider an enclosed

rectangle, to evaluate the effect of circular motion of a stationary robot on a FMCW radar seen in Figure 30. The material of the walls are a highly reflective aluminium plates.

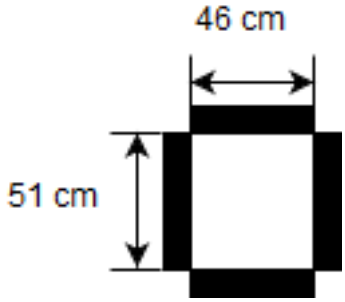



Figure 30. Volumetric floor plan for test case 1.

The motion plan for the circular movement test case can be seen in Figure 31. It considers a simple rotation, with an angular rate of change $\omega$. For this test case, the self-velocity model discussed previously is not considered nor used, due to the approximation error introduced by the increase of the angle of arrival of the reflection points.

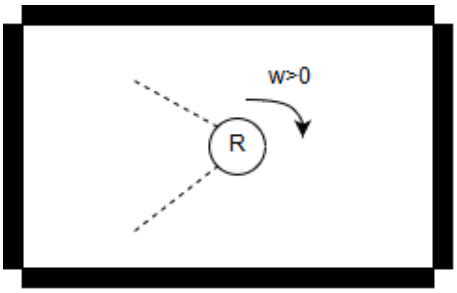



Figure 31. Motion plan for test case 1.

The the results of the circular movement case are depicted in Figure 32 as a projected occupancy grid map from octrees. The black cells signify occupied space, white unoccupied and grey unknown. The results show the specular reflection influence on mapping when there is highly reflective materials. The radar is placed inside the small area enclosed by them. This is showcased by the squares mapped behind the line of sight cells. It is not the case that the wavelength penetrates the aluminum sheets as shown in [53], but rather the result of specular reflections manifesting as false detections. This would not be mitigated by further reducing the range of valid reflection points for mapping. Reducing the range threshold even further would result in the mapping configuration becoming dependent on the physical dimensions of the mapable environment, which is not preferred.

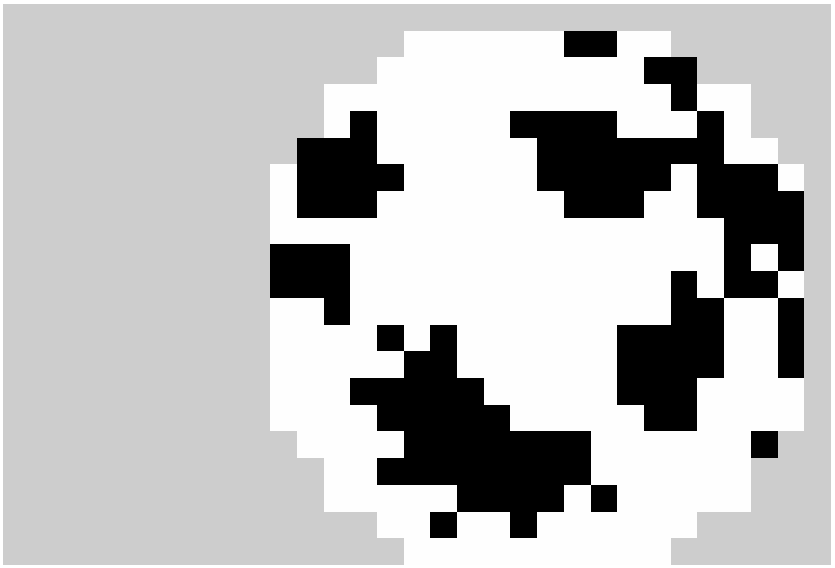


Figure 32. Resulting map from circular motion.

Another reason why the velocity model is not introduced is because error is introduced by the imperfect actuation of the wheels for rotational motion. This motion would introduce parasitic velocity, amplified by the angle of arrival estimation error, that would cause linear translation towards the wall as seen in Figure 33.

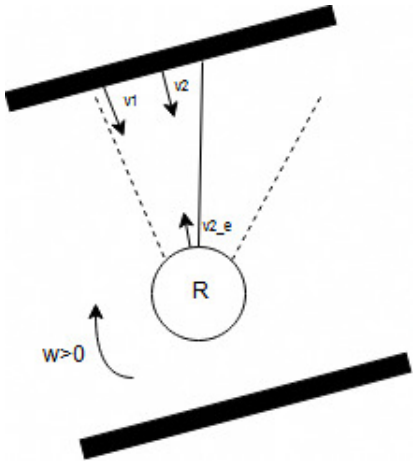



Figure 33. Parasitic velocity v2_e introduced by imperfect actuation.

Introducing the linear velocity for circular motion would need further filtering. This is to correct the curving and increased distance caused by parasitic velocity. The parasitic velocity test case setup can be seen in Figure 34.

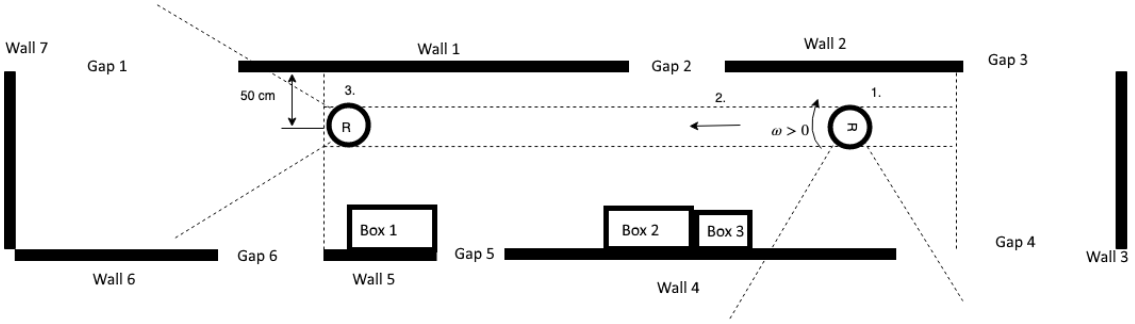



Figure 34. Parasitic velocity locomotion plan.

The resulting map looks like seen in Figure 35. The black cells show the occupied grids, while white free space and grey unknown. It is evident that parasitic velocity introduced by rotational motion has detrimental influence on the quality of the map. In the bottom right corner of the map, a curvature can be seen in Gap 4 mapping, caused by circular motion in conjuction with the radar velocity determination method. The curve is not aligned with the wall 4 and boxes 2 and 3, because of the parasitic velocity. Thus only the rotational information of an IMU will be considered.

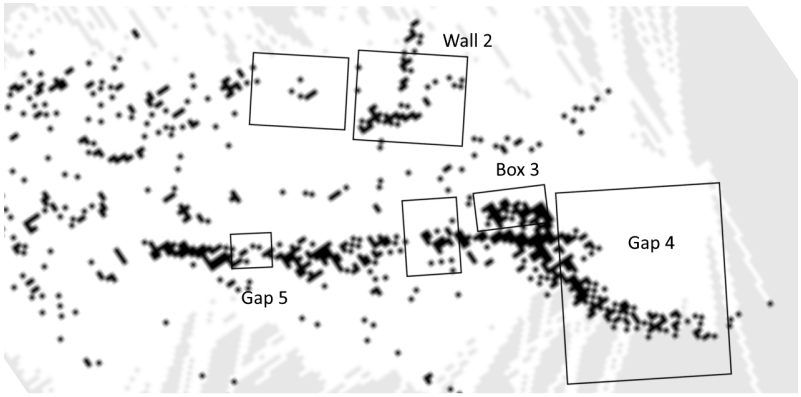



Figure 35. Parasitic velocity test case map.

### 3.2.2 Traversing dominant directions of hallways

The floor plan for testing the dominant direction of movement for indoor environments can be seen in Figure 36. The black bars signify the expected occupancy of the five centimeter grid cells. The environment composes of simple parallel walled corridor, consisting of plastic divisions, penetratable with the 77 GHz wavelength. The wall parallel to the moving robot are to asses the capability of mapping reflection points in the height dimension. This is done to detect the transition between different halls by detecting gaps in between parallel walls.

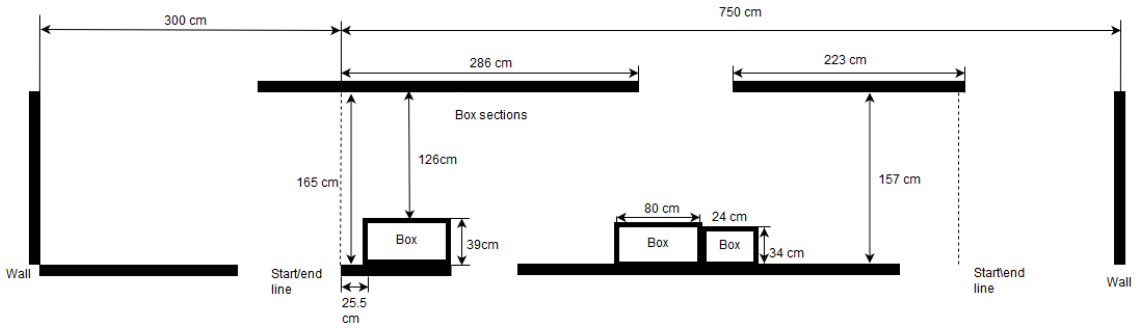



Figure 36. Volumetric floor plan for test case 2.

The perpendicular walls at the ends of the corridors, that are made of concrete, are useful for determining the self-velocity of the radar. The concrete walls are not penetrable by the wavelength and therefore are suitable to provide a reference for measuring linear speed. In addition, the perpendicular walls, in conjunction with the boxes, are used for assessing the capabilities of mapping of fully perpendicular surfaces in terms of width, when directly moving towards them. This is done in order to see the capability of determining gaps in terms of doors or changes in the width of the corridor. The introduction of boxes is made to ensue the capability of mapping perpendicular surfaces under differing angles of arrival in relation to the radar and assess the impact of the angular resolution of the radar in terms of results.

For the case of dominant directions of hallways, both assumptions for using the self-velocity motion model discussed previously, are guaranteed, so the velocity model is used. The path of the robot inside the structured environment can be seen in Figure 37.

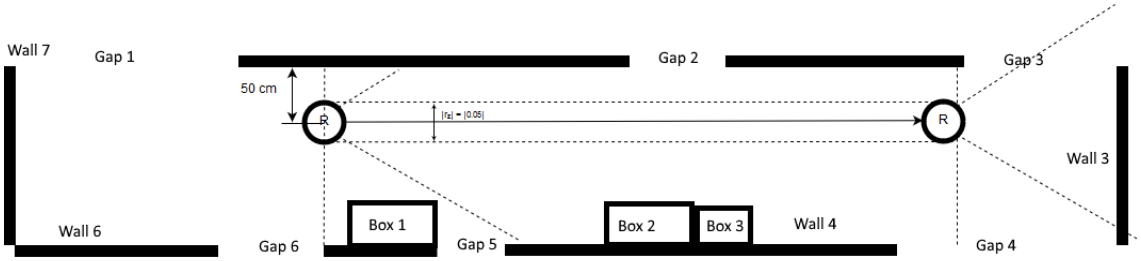



Figure 37. Locomotion plan for mapping dominant directions of hallways.

The trajectory is chosen based on the requirements described previously: to map differing and relevant use cases. The boxes are required to be in the field of view of the radar during the traversing. The result of the dominant direction case can be seen in Figure 38 as a projected occupancy grid map from octrees.

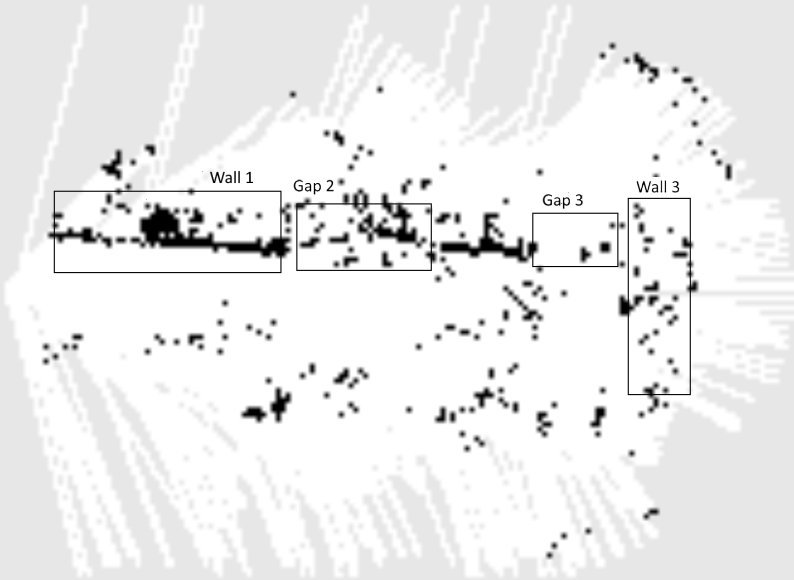



Figure 38. Resulting map from linear movement along a hallway.

Both results display the downsides of using the line of sight model of laser range finders as it assumes the shortest path back to the sensor. The results, however, show secondary reflections originating from specular reflection across multiple objects as seen in mapping Wall 1, that are still within the determined threshold of 1 meter for eliminating secondary reflections. The most evident result of secondary specular reflections from surfaces can be seen in the circular motion test results. The reflections bounce directly behind the radar and cause mapping in the length dimension, although the material is not penetrable for the wavelength. The hallway example offers the observation where the left hand side of the trajectory is mapped through plastic, namely Walls 1 and 2, which is consistent with the properties of the wavelength. Still, the sections acting as hallway walls are mapped as seen in mapping of the Walls 1 and 2 in Figure 38 [53], [54].

The gap determining assessment in perpendicular walls and parallel plastic divisions show differing results. The parallel case show some success in terms of mapping partially the two gaps in left hand side of the trajectory but is still overly noisy. Gap number 2 at the halfway point is overly noisy, while gap 3 at the end of the trajectory show better results. The dimensional mapping of the perpendicular wall at the end, namely Wall 3, of the trajectory proves too noisy for visual assessment. This is consistent with the low angular resolution of the AWR1843, providing only the field of view of around 100 degrees with steps of 14.5 degrees. The results show the success of mapping parallel plastic walls on the left hand side of the trajectory. However, the angular resolution did not provide the capability to map the right hand side, boxes 1 to 3 and gaps 4 to 6, where few detection points were made. This is caused by the differing angle of arrival for the detection of boxes. It is consistent with the left hand side mapping of Gap 2, as the walls are not aligned per-

fectly. As the robot passed Gap 2, the angle of arrival of detection points changed and the quality of the map deteriorated as seen in the quality of the mapping of Wall 2. For Wall 1, the angle of arrival of the parallel wall before Gap 2 stayed at a constant angle and is mapped adequately. The overall noisiness of the results provides motivation for further algorithmical solutions for filtering specular and secondary reflections.

### 3.2.3 Future work

In conclusion, these results provide visual success when mapping parallel surfaces. There is a chance for improvement when considering circular movement, given that the predictability of the reflection path can be better approximated. The improvent can be achieved by using alternative models.

To provide a more accurate and consistent map, the duplication of the radar sensors or the use of a different physical antenna configuration should be researched. The antenna used in this thesis does not provide the field of view nor the resolution to build highly accurate maps. To further improve the quality of a map, a filtering algorithm based on SNR can be used. Doing this would reduce the effect that the non-line-of-sight propagation and secondary reflection of electromagnetic waves have on the resulting map. The curvature caused by circular motion could be corrected by compensating for the parasitic velocity, or using different odometry sources for detecting translations of location. Alternative odometry solutions should be used to be more versatile in mapping the environment and not only use linear movement for mapping flat surfaces.

A single radar, with additional data association and filtering, is very suitable to be used as a filter of dynamic entities in the environment, like moving objects. This is because the FMCW radar is capable of measuring the relative speed of the entities. This can be realized using alternative sources of self-velocity to differentiate between dynamic and static objects in the environment. In addition, an alternative sensor models for the radar can be considered. This should model the FMCW radar type in terms of linear speed to map dynamic environments. Further testing could be conducted to improve the system to handle more harsh environmental conditions to leverage the versatility of the frequency band.

# 4 Conclusion

This thesis has proposed a proof of concept for robotic mapping solution using 77 GHz automotive radar and inertial measurement unit on a mobile base. The concepts of probabilistic robotic mapping and FMCW radar technology have been introduced to select the suitable methodology for creating a map. The robot composes a two dimensional online occupancy grid map as its representation of the environment. The map models the world as discrete cells that are either occupied, free or with unknown occupancy. The composed solution was assessed using two experimental test cases in an office environment.

The solution was composed by leveraging an existing software platform for building autonomous robots: the Robot Operating System. The solution utilizes continous rotation servos to move the robot. The sensing of the environment is done using a low angular resolution automotive radar and an inertial measurement unit. The sensors and actuators were interfaced using a general purpose personal computer, in order to asses the suitability for indoor mapping. An Extended Kalman filter based data fusion was used to provide localization for the mobile robot. The sources of sensing locomotion were limited to the usage of the radar to provide linear speed and the inertial measurement unit for rotation.

The resulting solution was tested in two structured indoor environments: while traversing dominant directions of hallways and in circular motion, the two most common directions in indoor locomotion. Based on the experimental results, the solution shows potential in mapping surfaces parallel to the moving direction of the robot. This is the case when the wall is at a constant angle in relation to the radar. When the angle of the wall changes, in example the motion is not perfectly linear, the low angular resolution of the radar causes the map quality to drop. In the case of rotational motion, secondary reflections in tight spaces cause false detections, making the resulting map highly noisy as it requires further signal processing.

# References


[1] L. Danielsson, "Tracking and radar sensor modelling for automotive safety systems", Göteborg, Sweden, 2010.

[2] M. Li, Z. Feng, M. Stolz, M. Kunert, R. Henze, and F. Küçükay, "High resolution radar-based occupancy grid mapping and free space detection", 2018-03, pp. 70–81. DOI: 10.5220/0006667300700081.

[3] K. Brötjefors and J. Gideflod, "Mapping of parking areas using radar sensors", Gothenburg, Sweden, 2018. [Online]. Available: http://publications.lib.chalmers.se/records/fulltext/255156/255156.pdf (visited on 2019-04-24).

[4] M. Hammarsten and V. Runemalm, "3d localization and mappingusing automotive radar", Gothenburg, Sweden, 2016. [Online]. Available: http://publications.lib.chalmers.se/records/fulltext/238490/238490.pdf (visited on 2019-04-24).

[5] K. Werber, M. Rapp, J. Klappstein, M. Hahn, J. Dickmann, K. Dietmayer, and C. Waldschmidt, "Automotive radar gridmap representations", 2015-06. DOI: 10.1109/ICMIM.2015.7117922.

[6] A. Guerra, F. Guidi, A. Clemente, R. D'Errico, L. Dussopt, and D. Dardari, "Application of transmitarray antennas for indoor mapping at millimeter-waves", Paris, France: IEEE, 2015-11. DOI: 10.1109/EuCNC.2015.7194044.

[7] F. Guidi, A. Guerra, and D. Dardari, "Millimeter-wave massive arrays for indoor slam", Sydney, NSW, Australia: IEEE, 2014-06. DOI: 10.1109/ICCW.2014.6881182.

[8] F. Guidi, A. Guerra, D. Dardari, A. Clemente, and R. d'Errico, "Environment mapping with millimeter-wave massive arrays system design and performance", Washington, DC, United States: IEEE, 2016-12. DOI: 10.1109/GLOCOMW.2016.7848895.

[9] S. Dogru and L. Marques, "Through-wall mapping using a short range radar", 2018-12. DOI: 10.1109/ICSENS.2018.8589866.

[10] A. Yassin, Y. Nasser, A. Y. Al-Dubai, and M. Awad, "Mosaic: Simultaneous localization and environment mapping using mmwave without a-priori knowledge", vol. 6, IEEE, 2018-11, pp. 68 932–68 947. DOI: 0.1109/ACCESS.2018.2879436.

[11] R. Mendrzik, H. Wymeersch, and G. Bauch, "Joint localization and mapping through millimeter wave MIMO in 5g systems - extended version", *CoRR*, vol. abs/1804.04417, 2018. arXiv: 1804.04417. [Online]. Available: http://arxiv.org/abs/1804.04417.

[12] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization", vol. 14, IEEE, 2014-02, pp. 444–457. DOI: 10.1109/TMC.2014.2320254.

[13] P. Fritsche and B. Wagner, "Comparison of two radar-based scanning-techniques for the use in robotic mapping", in *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, vol. 01, 2015-07, pp. 365–372. [Online]. Available: https://ieeexplore.ieee.org/document/7350495.

[14] A. Olivier, G. Bielsa, I. Tejado, M. Zorzi, J. Wildmer, and P. casari, "Lightweight indoor localization for 60-ghz millimeter wave systems", London, UK: IEEE, 2016-06. DOI: 10.1109/SAHCN.2016.7732999.

[15] M. Aladsani, A. Alkhateeb, and G. C. Trichopoulos, "Leveraging mmwave imaging and communications for simultaneous localization and mapping", *CoRR*, vol. abs/1811.07097, 2018. arXiv: 1811.07097. [Online]. Available: http://arxiv.org/abs/1811.07097.

[16] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2019.

[17] *Robot mapping*. [Online]. Available: http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/ (visited on 2019-05-01).

[18] *Bno055 datasheet*. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/BST_BNO055_DS000_12.pdf (visited on 2019-05-04).

[19] M. I. Skolnik, *Introduction to Radar Systems*. McGraw-Hill Book Company, 1962.

[20] C. Iovescu and S. Rao, *The fundamentals of millimeter wave sensors*. [Online]. Available: http://www.ti.com/lit/wp/spyy005/spyy005.pdf (visited on 2019-05-04).

[21] *Intro to mmwave sensing : Fmcw radars - module 1 : Range estimation*. [Online]. Available: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-1-range-estimation (visited on 2019-05-04).

[22] *Intro to mmwave sensing : Fmcw radars - module 2 : The phase of the if signal*. [Online]. Available: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-2-phase-if-signal (visited on 2019-05-04).

[23] *Intro to mmwave sensing : Fmcw radars - module 3 : Velocity estimation*. [Online]. Available: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-3-velocity-estimation (visited on 2019-05-04).

[24] *Intro to mmwave sensing : Fmcw radars - module 5 : Angle estimation*. [Online]. Available: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-5-angle-estimation (visited on 2019-05-04).

[25] *Mimo radar*. [Online]. Available: http://www.ti.com/lit/an/swra554a/swra554a.pdf (visited on 2019-05-04).

[26] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005, ISBN: 0262201623.

[27] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. California technical Publishing, 1999.

[28] R. R. L. Jr, *Kalman and Bayesian Filters in Python*. 2018.

[29] *Introduction to mobile robotics*. [Online]. Available: http://ais.informatik.uni-freiburg.de/teaching/ss18/robotics/ (visited on 2019-05-01).

[30] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation", *Artificial Intelligence*, vol. 99, pp. 21–71, 1998-02.

[31] C. Yi, S. Jeong, and J. Cho, "Map representation for robots", *Smart Computing Review*, vol. 2, pp. 18–27, 2012-02. DOI: 10.6029/smartcr.2012.01.002.

[32] J. O. Wallgrün, "Robot mapping", in *Hierarchical Voronoi Graphs: Spatial Representation and Reasoning for Mobile Robots*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 11–43, ISBN: 978-3-642-10345-2. DOI: 10.1007/978-3-642-10345-2_2. [Online]. Available: https://doi.org/10.1007/978-3-642-10345-2_2.

[33] A. Elfes, "Occupancy grids: A probabilistic framework for robot perception and navigation", AAI9006205, PhD thesis, Pittsburgh, PA, USA, 1989.

[34] S. Thrun, "Robotic mapping: A survey", 2002-03.

[35] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees", *Autonomous Robots*, 2013, Software available at http://octomap.github.com. DOI: 10.1007/s10514-012-9321-0. [Online]. Available: http://octomap.github.com.

[36] S. Thrun, "Learning occupancy grid maps with forward sensor models", *Autonom. Rob.*, vol. 15, 2003-11. DOI: 10.1023/A:1025584807625.

[37] C. G. D. P. FERNÁNDEZ, "Grid-based multi-sensor fusion for on-road obstacle detection: Application to autonomous driving", Stockholm, Sweden, 2015.

[38] *Parallax continuous rotation servo*. [Online]. Available: https://www.parallax.com/sites/default/files/downloads/900-00008-Continuous-Rotation-Servo-Documentation-v2.2.pdf (visited on 2019-05-04).

[39] *Awr1843 evaluation module (awr1843boost) single-chip mmwave sensing solution*. [Online]. Available: http://www.ti.com/lit/ug/spruim4/spruim4.pdf (visited on 2019-05-04).

[40] *Automated parking system reference design using 77- ghz mmwave sensor*. [Online]. Available: http://www.ti.com/lit/ug/tidueo9/tidueo9.pdf (visited on 2019-05-04).

[41] J. M. O'Kane, *A Gentle Introduction to ROS*. 2014, ISBN: 978-14-92143-23-9.

[42] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: An open-source robot operating system", in *ICRA Workshop on Open Source Software*, 2009.

[43] *Robotic operating system: Concepts*. [Online]. Available: http://wiki.ros.org/ROS/Concepts (visited on 2019-04-24).

[44] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system", in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*, Springer, 2014-07. [Online]. Available: http://docs.ros.org/melodic/api/robot_localization/html/_downloads/robot_localization_ias13_revised.pdf (visited on 2019-04-24).

[45] *Point cloud library (pcl)*. [Online]. Available: http://docs.pointclouds.org/1.8.1/group__octree.html (visited on 2019-05-04).

[46] *Ros enhancement proposal 103: Standard units of measure and coordinate conventions*. [Online]. Available: http://www.ros.org/reps/rep-0103.html (visited on 2019-04-24).

[47] *Ros enhancement proposal 105: Coordinate frames for mobile platforms*. [Online]. Available: https://www.ros.org/reps/rep-0105.html (visited on 2019-04-24).

[48] *Ros driver package ($ti_m mwave_r ospkg$)*. [Online]. Available: http://dev.ti.com/tirex/explore/node?node=AMOLXoQh4Kh8-a1Bfh8YZg__VLyFKFf__LATEST (visited on 2019-05-04).

[49] *Adafruit python bno055*. [Online]. Available: https://github.com/adafruit/Adafruit_Python_BNO055 (visited on 2019-05-04).

[50] *Mmwave software development kit*. [Online]. Available: http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/latest/index_FDS.html (visited on 2019-05-04).

[51] *Mmwave demo visualizer*. [Online]. Available: https://dev.ti.com/gallery/view/mmwave/mmWave_Demo_Visualizer/ver/3.1.0/ (visited on 2019-05-04).

[52] *Mmwave demo visualizer user guide*. [Online]. Available: http://www.ti.com/lit/ug/swru529b/swru529b.pdf (visited on 2019-05-04).

[53] *Detecting walls of different materials*. [Online]. Available: http://dev.ti.com/tirex/explore/node?node=AGcMw5shW3sPnCKkNyypXg__VLyFKFf__LATEST (visited on 2019-05-04).

[54] *People counting throught plastic and drywall and wood*. [Online]. Available: http://dev.ti.com/tirex/explore/node?node=AB64dFbicd2BPc23jvVjcg__VLyFKFf__LATEST (visited on 2019-05-04).

# Appendix 1 – Octomap_server configuration octomap_mapping.launch

```
<launch>
        <node pkg="octomap_server" type="octomap_server_node"
         name="octomap_server">
                <param name="resolution" value="0.05" />
                <param name="frame_id" type="string" value="map" />
                <param name="sensor_model/max_range" value="1" />
                <remap from="cloud_in" to="/mmWaveDataHdl1/RScan" />
        </node>
</launch>
```

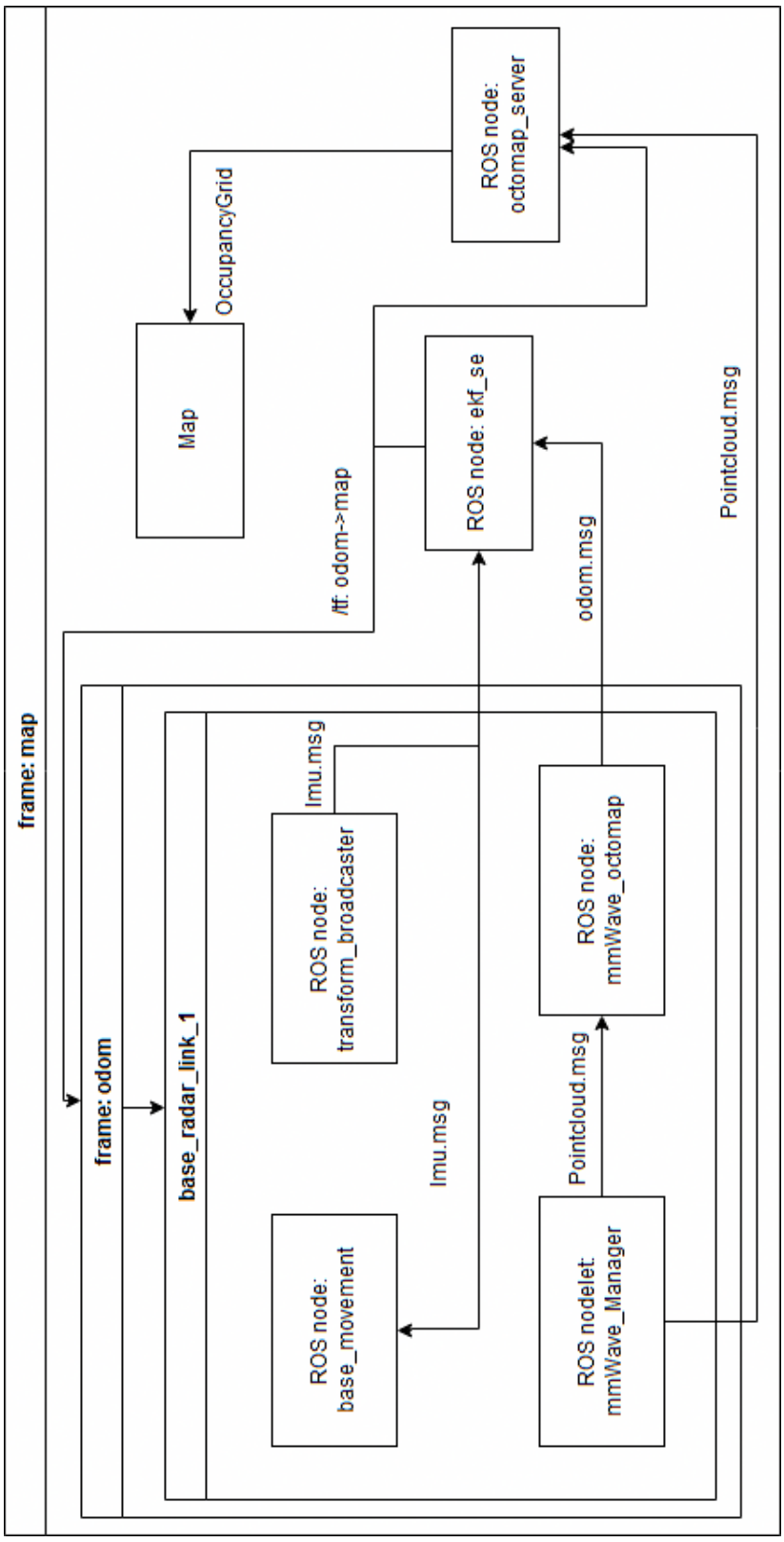
# Appendix 2 – Robot_localization configuration file ekf.launch

```
<launch>
  <node pkg="robot_localization"
  type="ekf_localization_node" name="ekf_se" clear_params="true">
        <param name ="frequency" value="30"/>
        <param name ="two_d_mode" value="true"/>
        <param name = "odom0" value="/odom0"/>
        <rosparam param="odom0_config">[false ,false ,false ,
                                        false ,false ,false ,
                                        true ,false ,false ,
                                        false ,false ,false ,
                                        false ,false ,false]</rosparam>
        <param name ="imu0" value="/imu0"/>
        <rosparam param="imu0_config">[false ,false ,false ,
                                        true ,true ,true ,
                                        false ,false ,false ,
                                        true ,true ,false ,
                                        false ,false ,false]</rosparam>
        <param name = "imu0_differential" value="false"/>
        <param name ="imu0_relative" value="false"/>
        <param name = "imu0_remove_gravitational_acceleration"
         value="true"/>
        <param name ="map_frame" value="map"/>
        <param name ="odom_frame" value="odom"/>
        <param name ="base_link_frame"
        value="base_radar_link_1"/>
        <param name ="world_frame" value="map"/>
  </node>
</launch>
```

# Appendix 3 – AWR1843 configuration file 1843_3d.cfg

```
% ***************************************************************
% Created for SDK ver:03.01
% Created using Visualizer ver:3.1.0.1
% Frequency:77
% Platform:xWR18xx
% Scene Classifier:best\_range\_res
% Azimuth Resolution(deg):14.5
% Range Resolution(m):0.039
% Maximum unambiguous Range(m):16.16
% Maximum Radial Velocity(m/s):2.35
% Radial velocity resolution(m/s):0.15
% Frame Duration(msec):66.667
% Range Detection Threshold (dB):x
% Doppler Detection Threshold (dB):x
% Range Peak Grouping:enabled
% Doppler Peak Grouping:enabled
% Static clutter removal:disabled
% Angle of Arrival FoV: Full FoV
% Range FoV: Full FoV
% Doppler FoV: Full FoV
% ***************************************************************
sensorStop
flushCfg
dfeDataOutputMode 1
channelCfg 15 5 0
adcCfg 2 1
adcbufCfg -1 0 1 1 1
profileCfg 0 77 7 7 200 0 0 20 1 512 2666 0 0 30
chirpCfg 0 0 0 0 0 0 0 1
chirpCfg 1 1 0 0 0 0 0 4
frameCfg 0 1 32 0 66.667 1 0
lowPower 0 0
guiMonitor -1 1 1 0 0 0 1
cfarCfg -1 0 0 8 4 4 0 800 1
cfarCfg -1 1 0 8 4 4 1 800 1
multiObjBeamForming -1 1 0.5
clutterRemoval -1 1
calibDcRangeSig -1 0 -5 8 256
compRangeBiasAndRxChanPhase 0.0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
measureRangeBiasAndRxChanPhase 0 1.5 0.2
aoaFovCfg -1 -90 90 -90 90
cfarFovCfg -1 0 0 15.99
cfarFovCfg -1 1 -2.35 2.35
sensorStart
```

# Appendix 4 – Full software architecture

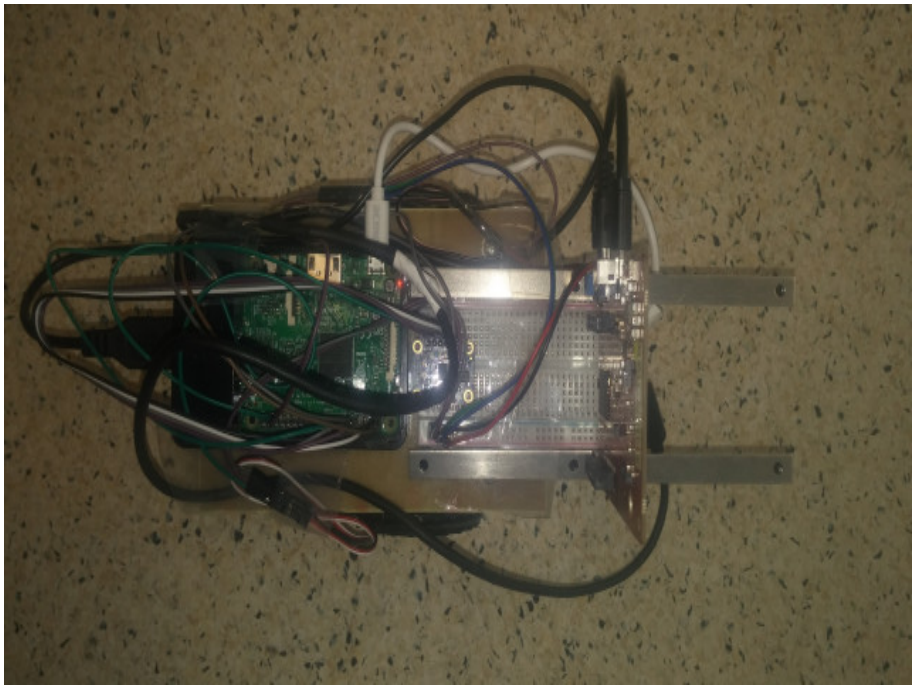


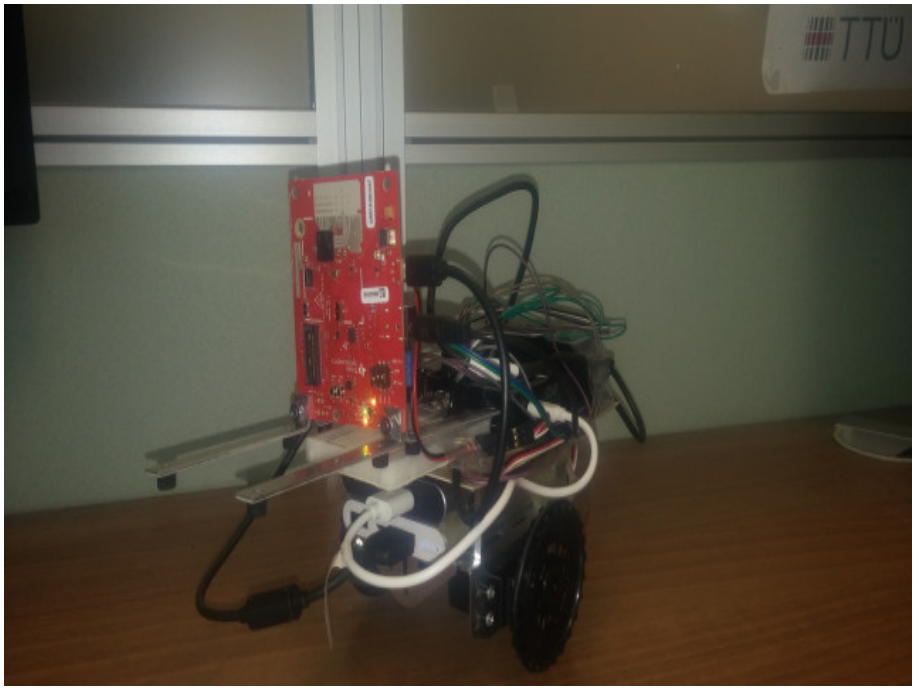
# Appendix 5 – Source code

Git repository:

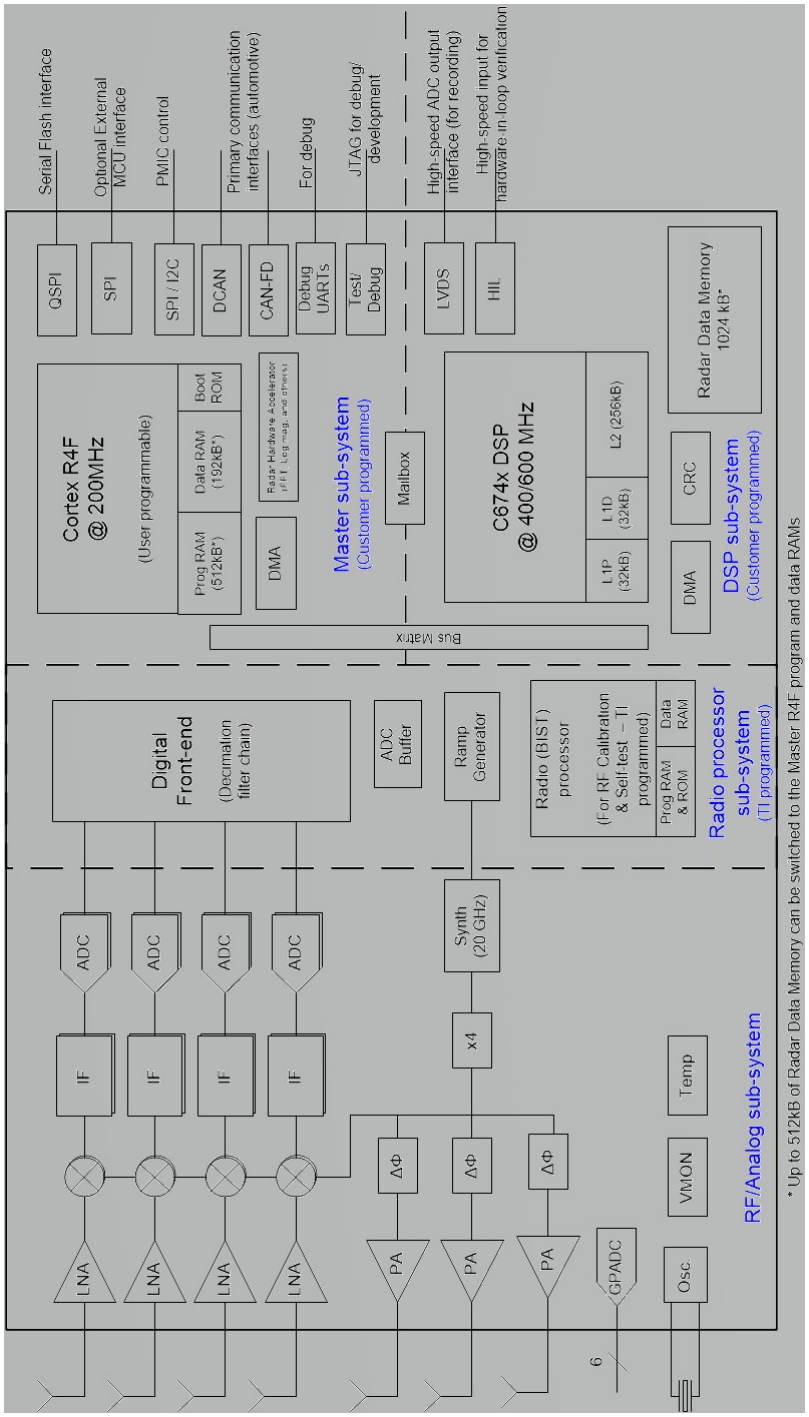
`https://github.com/umblauka/mmWave_mapping`

# Appendix 6 – Final physical form of the solution





# Appendix 7 – AWR1843 functional block diagram



# Appendix 8 – BNO055 operating modes [18]

| Operating mode | Available sensor signals | | | Fusion Data | |
|---|---|---|---|---|---|
| | Accel | Mag | Gyro | Relative orientation | Absolute orientation |
| CONFIGMODE | | | | | |
| ACCONLY | X | | | | |
| MAGONLY | | X | | | |
| GYROONLY | | | X | | |
| ACCMAG | X | X | | | |
| ACCGYRO | X | | X | | |
| MAGGYRO | | X | X | | |
| AMG | X | X | X | | |
| IMU | X | | X | X | |
| COMPASS | X | X | | | X |
| M4G | X | X | | X | |
| NDOF_FMC_OFF | X | X | X | | X |
| NDOF | X | X | X | | X |