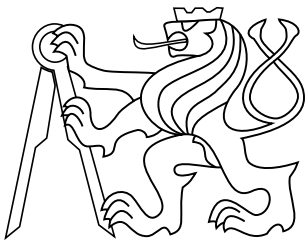
bachelor's thesis

Environment mapping using radar

Martin Burian



May 21, 2014

Ing. Jan Chudoba

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics

Acknowledgement

I would like to thank my supervisor Jan Chudoba for restraining my hasty deductions and giving advice when necessary. Many thanks also belong to my friend Zuzka Wangle, for her help with the more intriguing language aspects of this thesis. And finally there is my family and friends, who have enabled me to complete this work. Thank you, all of you.

Prohlášení / Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic thesis.

V Praze dne ....

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Martin B u r i a n

Study programme: Open Informatics

Specialisation: Computer and Information Science

Title of Bachelor Project: Environment Mapping Using Radar

Guidelines:

1. Familiarize yourself with radar module IGEP Lambda, its functions and application possibilities.
2. Study the commonly used environment mapping methods, applicable to radar data processing.
3. Design and implement IGEP Lambda radar data processing method in form of 2D environment map.
4. Perform experiments in real environment with radar mounted on mobile robot. Evaluate radar module functionality and implemented mapping method from the human point of view, as well as according to map utilization by robot.

Bibliography/Sources:

[1] Thrun S., Burgard W., Fox D. Probabilistic Robotics, The MIT Press, Cambridge, Massachusetts 02142, 2005.
[2] Integration Software & Electronics Engineering, IGEP RADAR LAMBDA Hardware Reference Manual (rev. 20130731), Barcelona, Spain, 2013.

Bachelor Project Supervisor: Ing. Jan Chudoba

Valid until: the end of the summer semester of academic year 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 10, 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Martin B u r i a n

Studijní program: Otevřená informatika (bakalářský)

Obor: Informatika a počítačové vědy

Název tématu: Mapování prostředí radarem

Pokyny pro vypracování:

1. Seznamte se s radarovým modulem IGEP Lambda, jeho funkcemi a možnostmi použití.
2. Prostudujte používané metody mapování prostoru mobilním robotem vhodné pro zpracování dat z radaru.
3. Navrhněte a implementujte metodu zpracování dat z radaru IGEP Lambda ve formě 2D mapy prostředí.
4. Proveďte praktické experimenty s radarem umístěným na robotu a vyhodnoťte funkci radarového modulu a implementované mapovací metody z pohledu využití mapy robotem i člověkem.

Seznam odborné literatury:

[1] Thrun S., Burgard W., Fox D. Probabilistic Robotics, The MIT Press, Cambridge, Massachusetts 02142, 2005.
[2] Integration Software & Electronics Engineering, IGEP RADAR LAMBDA Hardware Reference Manual (rev. 20130731), Barcelona, Spain, 2013.

Vedoucí bakalářské práce: Ing. Jan Chudoba

Platnost zadání: do konce letního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2014

Abstract

Technological advances have enabled manufacturing of radar devices small enough to be carried by mobile robots. One of such devices is IGEP Radar Lambda manufactured by a Spanish company ISEE. We have evaluated the prospects of using the Lambda sensor in mobile robotics. The Lambda radar operates on 24 GHz ISM band in FMCW mode. It provides range measurements at ranges 0.9–25 m with beam width 26° and standard error 19 cm.

We have designed an algorithm based on Bayes filter to reconstruct environment maps from radar data. The algorithm has been tested in indoor and outdoor environment and yielded satisfactory results.

The sensor results are promising. The sensor provides false measurements under certain conditions as of now, but we believe that substantial improvements can be achieved by better data processing and sensor utilization.

Keywords

radar, environment mapping, Bayes filter, occupancy grid

Abstrakt

Technologické pokroky umožňují výrobu radarových senzorů dostatečně malých k využití na mobilním robotu. Jedním z takových senzorů je IGEP Radar Lambda španělské firmy ISEE. Vyhodnotili jsme možnosti využití senzoru Lambda v mobilní robotice. Radar Lambda využívá 24 GHz ISM frekvence a funguje v FMCW módu. Poskytuje měření vzdáleností v rozsahu 0.9–25 m se šířkou paprsku 26° a standardní chybou 19 cm.

Navrhli jsme algoritmus k rekonstrukci mapy prostředí z radarových dat založený na Bayesových filtrech. Algoritmus byl otestován na datech z vnitřního i venkovního prostředí a poskytuje uspokojivé výsledky.

Výsledky senzoru jsou slibné. Senzor má v této chvíli za určitých podmínek poskytuje chybná měření, ale věříme, že existuje možnost dalšího zlepšení ve zpracování dat a využití senzoru.

Klíčová slova

radar, mapování prostředí, Bayesův filtr, mřížka obsazenosti

Contents

List of Figures

List of Tables

Abbreviations

ISM	Industrial, Scientific, Medical bands
FMCW	Frequency Modulated Continuous Wave
SLAM	Simultaneous Localization And Mapping
RTT	Round-Trip Time
RCS	Radar Cross-Section
IF	Intermediate Frequency
BW	BandWidth
FFT	Fast Fourier Transform
DFT	Discrete Fourier Transform
ADC	Analog-Digital Converter
SPI	Serial Peripheral Interface
DDS	Direct Digital Synthetizer
PLL	Phase Locked Loop
AGC	Automatic Gain Control
API	APplication Interface
TCP	Transmission Control Protocol
DAQ	Data AcQuisition

1. Introduction

Robots have been around for quite a long time. Oxford dictionary defines a robot as *„A machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer“*[1] At first however, robots were not robots at all. The term 'robot' was invented in Karel Čapek's play R.U.R. 'Rossum's Universal Robots' in 1920. Early robots were purely mechanical automata created to amuse rather than to do actual work. There were automatic musicians like drummers and flute players, automatic puppets for theater and so on. [2]

Later, as the electricity took over the industry and the society, electric robots were built. Electricity made robot design much easier. The robots were controlled by remotely switching their components on and off. But the complexity of controlling the robot's movement was immense and had to wait for electronics to develop. With electronic circuits, scientist were able to simulate simple biological processes as phototaxis [2] and first somewhat autonomous robots emerged. They were capable of perceiving the environment they move in, analysing the measurements and taking action based on the analysis results.

Then the digital era came, with its computers. Computers allowed us to simulate complex processes and to give the robots true autonomy and "free will". Computer-driven robots can perform much deeper analysis of the measured data. Nearly all imaginable sensors have been used, including tactile sensors (bumpers), sonars, laser range-finders, cameras, … The collected data is then used either to take action based on the immediate measurements, or to construct some sort of an internal environment model and decide based on this model.

With today's advancing technology, new opportunities open in the sensor field. Radar systems have been extensively used since the World War II in the military and civilian sector. They were massive devices using kilowatts of power to detect aircrafts and ships hundreds of kilometers away. Today's miniaturized technology allowed development of small radar devices, operating at powers low enough to be powered by batteries and capable of measuring ranges low enough to be applicable to local measurements. Radars, or the radio waves they use, exhibit many properties desirable for a sensor used by a robot. They are robust to the environment as fog, rain and dust don't affect them. They can penetrate some materials, mainly dielectrics, and are thus capable of imaging objects behind for example a closed door.

An example of such miniature radar device is the IGEP Radar Lambda. It is a small radar device operating on 24 GHz ISM band capable of both pulsed and CW operation, combined with an embedded computer.

The aim of this thesis is to test and evaluate possibilities of IGEP Radar Lambda for use in mobile robotics. We will thoroughly explain the principles behind the radar sensing

and the mapping techniques used. We will test the sensor properties by experiments and deliver a comprehensive description of the results.

1.1. Related work

There have been several works regarding use of radar units in mobile robotics, but using hardware very different to the Lambda radar.

Foessel [3] has successfully used FMCW radar in evidence grid framework. He used 77 GHz radar device with a pencil beam to construct 3D map of the environment. He has later published a comprehensive study of radar sensor properties for mobile robotics [4].

Reina [5] has used a FMCW radar device operating at 95 GHz for ground segmentation. The radar had a pencil beam and used mechanical scanning.

Both the devices were physically larger than the Lambda radar, using antenna apertures of about 20 cm while IGEP Lambda sensor is only 5×10 cm small. That allowed the narrow beams and high angular resolution.

1.2. Outline of the thesis

- **Chapter 1 — Introduction** gives a mild introduction into the topic and states the aims of this thesis
- **Chapter 2 — Robotic mapping** examines several algorithms used in mobile robot mapping
- **Chapter 3 — Radar theory** introduces the reader into basics of radar technology
- **Chapter 4 — IGEP Lambda sensor** describes the radar sensor and its properties
- **Chapter 5 — Mapping algorithm** describes the algorithm designed to interpret the radar measurements
- **Chapter 6 — Experiments** describes the experiments conducted and analyses their results in detail
- **Chapter 7 — Conclusions** concludes the results of our work

2. Robotic mapping

To be able to move around the environment, the robot needs to know where it is and what the environment looks like. This is trivial for humans, but rather complicated for a robot.

When we know for sure where the robot is going to operate, we can provide it with a map of the environment, a building floor for example. The plan can, over time, become inaccurate as furniture gets moved around. Furthermore the robot is limited to operation only on the particular floor of the particular building it has a plan of.

To make the robot more flexible, to allow it to operate on another floor or anywhere else, the robot needs to create its own picture of its surroundings and act according to them. This means the robot usually remembers the places it has visited and builds a map. The map contains information important for the robot's task. It may be a map describing obstacles like a floor plan to navigate the environment, it may be a map describing positions of certain objects in the environment.

Location (*pose*) estimation and environment mapping are deeply connected. You need to know where you are to build a map, but you need a map to know where you are. This chicken-egg problem is referred to as *Simultaneous Localization And Mapping* or *SLAM*. In this thesis, we will focus on the mapping part and take the assumption that the robot pose is known. The assumption can be easily fulfilled by trusting odometry or running a SLAM algorithm with another sensor while collecting radar data.

2.1. Map representation

The robot needs to represent the environment map in such way that it can use it to perform its task — usually navigate the environment. Several types of representation emerged as the subject was studied in the 80's and 90's. The mapping field has namely split into *metric* and *topological* maps [6, sec. 2].

2.1.1. Metric maps

Metric maps describe the environment in a metric framework, a position- and distance-centric approach. Metric maps concentrate on describing the position and shape of obstacles. There are two main methods used to represent the environment: occupancy grids and geometric models.

Occupancy grids

As the name suggests, occupancy grid approach represents the map in a grid. The environment is divided into equally sized cells. Each cell is represented in the robot's memory and carries information about the portion of the environment it represents.

Occupancy grids are nowadays usually used with probabilistic maps, where each cell is represented as the probability that it is occupied — there is an obstacle [7, chap. 9]. Grid maps can however hold different quantities connected to the environment than the occupancy probability, for example the robot's confidence that the information it has about this cell is correct.

The cell size needs to be chosen carefully, as large cells cannot hold enough details, but small cells increase the memory requirements considerably. Map size grows with square inverse cell size and large maps can grow out of memory[1]. Small cells also imply higher computation cost of all operations, particularly of updating the map. As a measurement is taken, the robot needs to update the map to reflect the measurement. The measurement carries information about a portion of the environment, and the smaller the cells, the more of them lie in the imaged region and need to be updated.

Geometric maps

The other popular approach to map representation are geometric maps. In a geometric map, the environment is represented as a list of primitive objects, like lines, arcs and other geometrical shapes, their placement and relations between them.

The geometric representation is generally more memory-efficient than occupancy grids. A wall spanning the whole width of the map can be represented by a rectangle, that means four numbers instead of the hundreds or thousands of cells in a grid map.

On the other hand, updating the geometric map based on a measurement is usually not as straightforward as with occupancy grids. The new measurement needs to be fused with the existing geometrical data, which may include complex computation.

2.1.2. Topological maps

Topological maps describe the environment in a topological framework, concentrating on places and relationships between them. The map is a graph where nodes represent different places and and the edges represent paths between them. The paths are described by information like distance or navigation commands. This allows the robot to navigate between the listed places.

Topological maps are generally a higher level of abstraction. They are harder to obtain from crude sensor readings and dominate when mapping vast environments.

[1] Today, this is still an issue with 3D occupancy grids where memory requirements grow with third power of resolution

2.2. Probabilistic framework

All sensors provide readings with a certain error. The robot therefore needs not only to know what is around, but also how certain it is about this information. Probability theory comes handy in this task as it provides us with formal apparatus to deal with uncertainty.

The environment is represented as a set of probabilities. The probability of a piece of information is called the robot's *belief* in the information. In case of occupancy grids these are probabilities that the cell is occupied in the environment.

A cell C can be either Occupied ($O(C)$) or Free ($F(C)$). As these states are complementary, $p(O(C)) = 1 - p(F(C))$ and the belief about the cell state can be represented as only $p(O(C))$.

As new measurements come in, the robot updates the probability values in his world model. A group of algorithms called filters is used to incorporate this new piece of information into the environment model. *Bayes filters* are used in discrete cases like occupancy grids. *Kalman filters* can accommodate continuous cases like continuous pose estimation. Other techniques like *particle filters* use different tricks to represent the uncertainty [7].

2.3. Bayes filter

Bayes filter is an algorithm to update the environment model based on the Bayes rule. The Bayes rule (2.1) states that the probability of O when we know that M has occurred is related to the probability of M occurring if O is known.

$$p(O|M) = \frac{p(M|O)p(O)}{p(M)} \tag{2.1}$$

When we write down the Bayes rule for occupied and free cells, we obtain equation (2.3). The desired result is $p(O(C)|M)$, that is probability of cell C being occupied when we registered measurement M

$$p(O(C)|M) = \frac{p(M|O(C)p(O(C))}{p(M|O(C))p(O(C)) + p(M|F(C))p(F(C))} \tag{2.2}$$

$$= \frac{p(M|O(C)p(O(C))}{p(M|O(C))p(O(C)) + p(M|F(C))(1 - p(O(C))} \tag{2.3}$$

There are three terms in the equation. Term $p(O(C))$ represents the current belief that cell C is occupied. Terms $p(M|O(C))$ and $p(M|F(C))$ are probabilistic models of the sensor. They tell us what the probability of measurement M would be if cell C was occupied or free respectively.

The probabilistic sensor model describes the sensor behavior, the probability of overlooking an obstacle or of false detection.

3. Radar theory

Radar, an acronym for RAdio Detection And Ranging, uses radio waves to detect objects and measure their distance from the sensor. The radar device transmits radio waves and waits for an echo created by reflection from an obstacle. Measuring the delay from transmission to echo return yields distance from the transmitter to the target.

3.1. Radar basics

Since radar science has been around for a long time and has been an area of intensive study and technological advances, there are numerous traditional terms and equations used.

3.1.1. Range and round trip time

The signal travels to the target at a *range* R and back over time T_r. T_r is called *round-trip time* (RTT). As the signal travels to the target and back, covering $2R$ meters at the speed of light $c \approx 3 \times 10^8$ m/s, R can be calculated as

$$R = \frac{1}{2} c T_r \tag{3.1}$$

3.1.2. The radar equation

The transmitted signal deteriorates with the distance it travels and is reflected in different ways from different objects. The signal behavior is described by the so-called *radar equation* (3.3) [8]

$$P_r = P_t G_t \frac{1}{4\pi R^2} \sigma \frac{1}{4\pi R^2} A_{eff} = \tag{3.2}$$

$$= \frac{P_t G_t A_{eff}}{16\pi^2} \frac{\sigma}{R^4} \tag{3.3}$$

In the slightly expanded form of the equation (3.2), it is clear what happens to the signal. P_t [W] is the power transmitted by the antenna.

Radars use directional antennas to be able to determine the direction to the target. The transmitted power is thus multiplied by the antenna gain G_t [dB] that specifies how many times the output power would need to be higher if we were using an isotropic

antenna instead of the directional one and we wanted to get the same output power at the target.

The transmitted power that can now be considered as coming from an isotropic antenna is distributed over the sphere created by the signal propagating isotropically. At a distance R, the sphere has a surface of $4\pi R^2$ and so the *power density* at range R is the transmitted isotropic power over the sphere surface.

Then the signal arrives to the target, which reflects a part of the power it absorbs. The reflected power depends on the *radar cross-section* (or *RCS*) σ [m^2] which specifies the equivalent surface area of the target.

The dimension of the formula so far is $Wm^{-2}m^2 = W$, which gives the power reflected from the target.

The reflected power is again distributed over the sphere resulting in lower power density back at the receiver. The echo signal is picked up by a receiver antenna with an *effective area* A_{eff} [m^2]. The effective area of the antenna collects the signal with the power density given by the previous terms and receives power P_r [W].

When we rearrange the equation (3.2) to separate hardware constants from variables, we obtain equation (3.3). We can see that the received power is proportional to the RCS of the target and inversely proportional to the fourth power of range.

3.2. Signal propagation

The signal is transmitted by an antenna, it propagates through the environment, reflects from an obstacle and then propagates back. All the steps in signal path affect the resulting signal received back at the radar device.

3.2.1. Antenna

The transmission antenna usually directs the signal to illuminate only a limited space. The antenna usually directs the signal by a reflector. The waves reflected on the far end of the antenna interfere with those reflected closer to the emitter. These inhomogeneities in the antenna beam can cause variations of echo power, as the interference can dampen the signal even for large targets.

This phenomenon occurs in small ranges called the *near field*. Its opposite, the *far field*, is defined as the region where radiation intensity is identical throughout the beam and decreases with square distance. The near field ends and the far field begins at range $R_{\text{nf}} = D^2/\lambda$ [8, p.229] where D is the antenna aperture, that is the largest physical dimension of the antenna, and λ is the signal wavelength. Return intensity of targets closer than R_{nf} is largely dependent on target position in the cone and these targets are thus not really characterised by the return. Radar units should operate for targets in the far field region.

3.2.2. Environment propagation

The signal propagates through the environment at the speed of light. The environment has major influence on signal damping. At 24 GHz, water molecules in the environment resonate with the signal, absorbing considerable amount of its power. This is the reason why large radar devices don't operate at 24 GHz, as water vapor in the air makes detection at large distances very difficult [8]. Short range operations don't suffer from this damping as severely.

3.2.3. Reflection

When the signal hits an obstacle, a part of the signal penetrates it and a second part is reflected in all the directions. The reflected energy is divided between specular and diffuse reflection. In case of specular reflection the signal moves according to Snell's law, it reflects at the angle equal to the incidence angle of the original beam. The rest of the incoming energy is diffused and reflected in other directions.

The amount of diffusion is a material property. Some materials like metals reflect virtually all of the energy in specular way, other exhibit certain amount of diffuse reflections.

Experiment 6.2.2 has shown that diffuse reflections are negligible.

3.3. Radar technologies

There are two major approaches to measuring the time interval from transmission to echo registration.

First approach is the pulsed radar that transmits a short pulse of radio waves and then measures the time until the echo or echoes arrive. There are many issues with the measurement errors like range resolution problems originating from pulse duration, but since radar has been a strategic military technology since the World War II, the issues have been addressed successfully and pulsed radar is nowadays a very precise instrument.

However, some of the pulsed radar properties are rendering it hardly usable for small range operation like mobile robotics. One of the limitations is time measurement precision. Since the transmitted waves travel at the speed of light, the delays are extremely short at a small range (about 60 ns at 10 m range). This would make precise measurements at a short range require very precise (and thus costly) equipment.

Second approach is the FMCW (Frequency Modulated Continuous Wave) radar which transmits the signal continuously. The signal frequency is modulated over time and echo range is determined by comparing the frequency of the received signal with the frequency of the signal currently transmitted. With the knowledge of the modulation pattern, one can then calculate the round-trip time and thus the target range.

This approach avoids any kind of time measurement making short-range measurements at reasonable precision much easier. Longer transmission times also imply lower necessary transmission power making the equipment smaller and more energy efficient.

3.4. FMCW radar

A FMCW radar continuously transmits the modulated signal and simultaneously registers the return signal mixing it with the signal being transmitted. The result is a signal with frequency equal to the difference between the frequencies received and transmitted. This signal is called the *intermediate frequency* or IF signal. Knowing the modulation shape, one can then infer the time elapsed from transmitting the signal which is the RTT.

3.4.1. Modulation

The modulation pattern needs to allow calculation of time elapsed between transmitting two frequencies. The most popular modulation patterns are linear based saw and triangle patterns which are very easy to use, but others like sine modulation can be used too.

One linear modulation period is called a *ramp* or a *sweep*. One triangle period is actually two sweeps, one increasing and one decreasing. A sweep has a duration T and a *bandwidth* BW, that is the frequency range over which the transmission frequency is changed.

An example of a modulation pattern is presented in Figure 3.1. The solid line represents the transmitted frequency, the dashed line is the received echo signal frequency. The frequency difference Δf corresponds to RTT Δt.

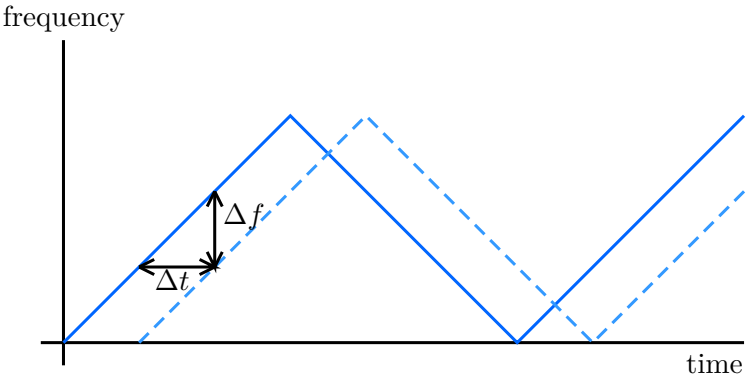


Fig. 3.1. Frequency modulation principle

IF frequency meaning is illustrated in Figure 3.2. The solid and dashed grey lines represent the transmitted and received signal respectively. The solid blue line represents the IF signal, that is the difference between the frequencies transmitted and received.

As the modulation is periodic, there are periods of time when signals from a sweep are being received when the next sweep is already being transmitted. These measurements are invalid and need to be left out from the following processing. These periods are apparent in Figure 3.2 as the periods when the IF frequency is not constant.

It can also happen that a target is so far that its returns will always end up in the next sweep. This target will register as very close instead of very distant. The range

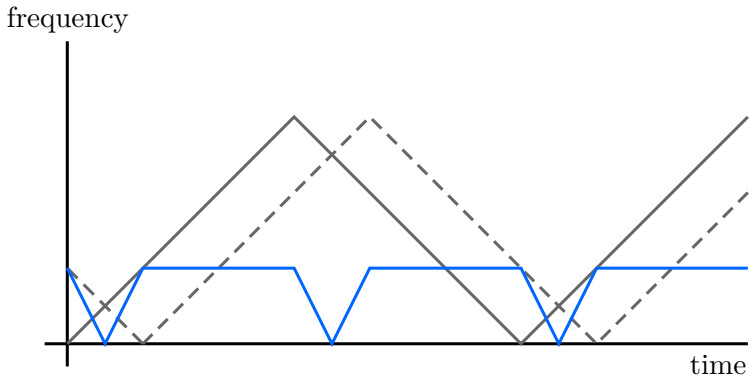


Fig. 3.2. IF signal

R_{max} at which this happens depends on the sweep duration T and is called *maximum unambiguous range*. It can be calculated as

$$R_{max} = \frac{1}{2}cT \tag{3.4}$$

3.4.2. Frequency/range conversion

The linear patterns allow very simple frequency/range conversions. IF frequency of BW Hz would mean RTT of T s, so 1 Hz of IF frequency corresponds to RTT $\frac{T}{BW}$ s or range $\frac{cT}{2BW}$ m. Given the IF frequency f Hz, the corresponding range is

$$R = f\frac{cT}{2BW}\ [\text{m}] \tag{3.5}$$

3.4.3. Doppler effect

Since the range measurement depends on the returned frequency, issues arise with the Doppler effect. When the transmitted signal reflects from a moving target, the reflection's frequency is altered by the Doppler effect.

Consider a target moving towards the sensor at a speed of v_t[1]. The sensor transmits at frequency f_0. Then the echo arrives. Meanwhile, the transmitted frequency has changed to f_t. The received echo frequency will be

$$f_e = \left(1 + \frac{v_t}{c}\right) f_0 = f_0 + \frac{f_0 v_t}{c} = f_0 + f_D \tag{3.6}$$

The frequency shift, called the *Doppler frequency*, affects the range calculation. Instead of the frequency difference $f = f_t - f_0$ that would yield the correct range R, frequency difference of

[1] This means that a target moving towards the sensor has a positive speed, while a target moving in the opposite direction has a negative speed

$$f' = f_t - f_e = f_t - f_0 - f_D = f - f_D \tag{3.7}$$

is registered, resulting in a different range solution R_D:

$$\begin{aligned} R_D &= (f - f_D)\frac{cT}{2BW} \\ &= R - f_D\frac{cT}{2BW} \end{aligned} \tag{3.8}$$

When we consider a linear increasing sweep, a target moving towards the sensor increases the return frequency making the target seem closer. However, if the sweep was decreasing, lower return frequency would make the target seem farther.

This is the reason why triangle modulation is often used. The target is detected on the increasing and decreasing sweeps independently. Then the actual range is the mean of the two ranges and the Doppler frequency is half the difference between increasing and decreasing sweeps.

3.5. IF signal processing

The IF signal carries the range information in its frequency. To isolate the frequency components, or the targets at different ranges, so-called filter banks were used in the early days of radar technology. The measurable range was divided into *range bins*, where each bin was assigned an interval of ranges, and hence frequencies. There were banks of parallel band-pass filters the IF signal was fed to. The frequency components belonging to each bin were filtered out and responses for the range bins were isolated [8].

Fortunately, today is the era of digital signal processing. When we digitise the signal, we can examine the frequency spectrum using the Fast Fourier transform.

3.5.1. FFT

There are some properties of the *Fast Fourier transform* (FFT) that are key to understanding the performance limits of the sensor.

Periodicity assumption

FFT assumes the processed signal is periodical. That means that the samples of the signal are assumed to be periodically repeating themselves from minus infinity to infinity. Breaking this assumption leads to artifacts in the FFT. Fortunately, due to the periodic nature of the modulation, we can easily fulfill this assumption by taking FFT of whole modulation periods.

Nyquist frequency

According to the Nyquist theorem [9], the maximum frequency detectable by FFT is $^1/_2$ of the sampling frequency called the *Nyquist frequency*. For example, with data sampled at 500 ksps (500 kHz), the highest frequency that will be detected by FFT (the Nyquist frequency) is $f_n = 250\,\text{kHz}$.

Frequency resolution

Since FFT is a discrete operation, it outputs discrete frequency components. *Bin size* is the difference between frequencies represented by two consecutive bins (frequency results). The *frequency resolution* of the FFT is inverse of the bin size. The smaller the bin size, the finer the results and the higher the resolution.

Let us assume N real samples have been measured at a sampling rate of f ksps. FFT of such data will be $N-1$ real numbers and the output will be symmetrical. The first $N/2$ numbers are the sought result. They represent amplitudes of the frequency components from 0 Hz (zeroeth bin, the DC component) to the Nyquist frequency of $f_n = f/2$. The $N/2$ frequency bins will be evenly distributed in this range, where i-th bin will represent frequency

$$f_i = \frac{f_n/2}{N/2} i \tag{3.9}$$

$$= \frac{f_n}{N} i \text{ [kHz]} \tag{3.10}$$

Zero padding

FFT is an algorithm to compute the DFT, *Discrete Fourier Transform*. DFT allows computation of more frequency domain results than there are time domain samples. FFT can achieve this with *zero padding*. The signal is extended by a number of zeros. The result of a FFT of such signal is the result of a DFT of the original signal with as many frequency results added as zeros appended to the signal. Zero padding thus allows us to increase the frequency resolution. It however doesn't carry any new information and so needs to be used carefully not to waste computation time on unnoticeable improvements.

3.5.2. FFT of the IF signal

The FFT result gives us amplitudes of the frequency components in the IF signal. Then ranges are assigned to the isolated frequencies and the FFT result gives us response coming from the corresponding range bin. Typical FFT outcome can be seen in Figure 3.3

Presence of an obstacle is indicated by a peak. This is a particularly sharp peak, many times the target shows as a broad hill rather than a peak.

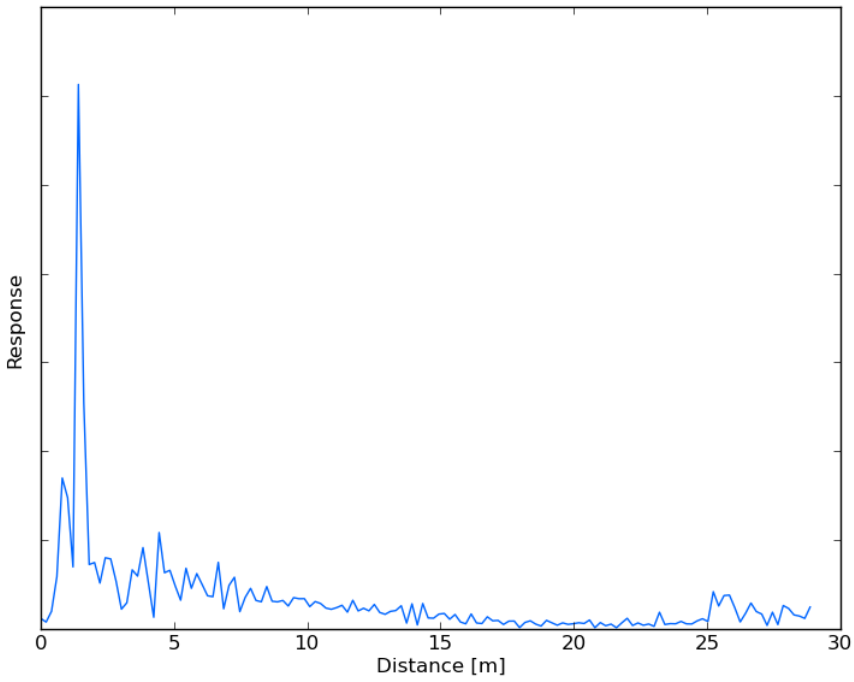


Fig. 3.3. Typical FFT result. Note the target at 1.3 m

3.5.3. Processing parameters

The IF processing has several parameters that need to be set to define the process. It is the sampling frequency for the ADC, the number of samples and the padding factor.

The sampling frequency needs to be chosen high enough to satisfy the Nyquist criterion, that is twice the highest detectable frequency. There are no other constraints on the sampling frequency.

The number of samples has major influence on the resulting resolution. If not enough samples are used, the FFT results may miss the target frequency whatever the padding. Too many samples mean low measurement rate as they will take more time to measure.

The padding can improve the results drastically, mainly in multi-target resolution [3, sec. 3.2]. It can however also dramatically increase the computation time.

The effect of padding is illustrated in Figure 3.4. The original signal has 288 samples (3 ramps). 3.4b and 3.4c are padded with 288 and 566 zeros respectively. We can see the smoother character of the padded results, useless to computers. There is however an interesting phenomenon called sidelobe. In 3.4b, immediately to the right of the main peak, there is a new peak that was not present in the intrinsic results. It is an analogy of a shadow of the main peak. Sidelobes appear in every FFT and are caused by using finite time domain input, but the padded FFT has enough samples to actually show them. Sidelobes could register as false targets and should be avoided.

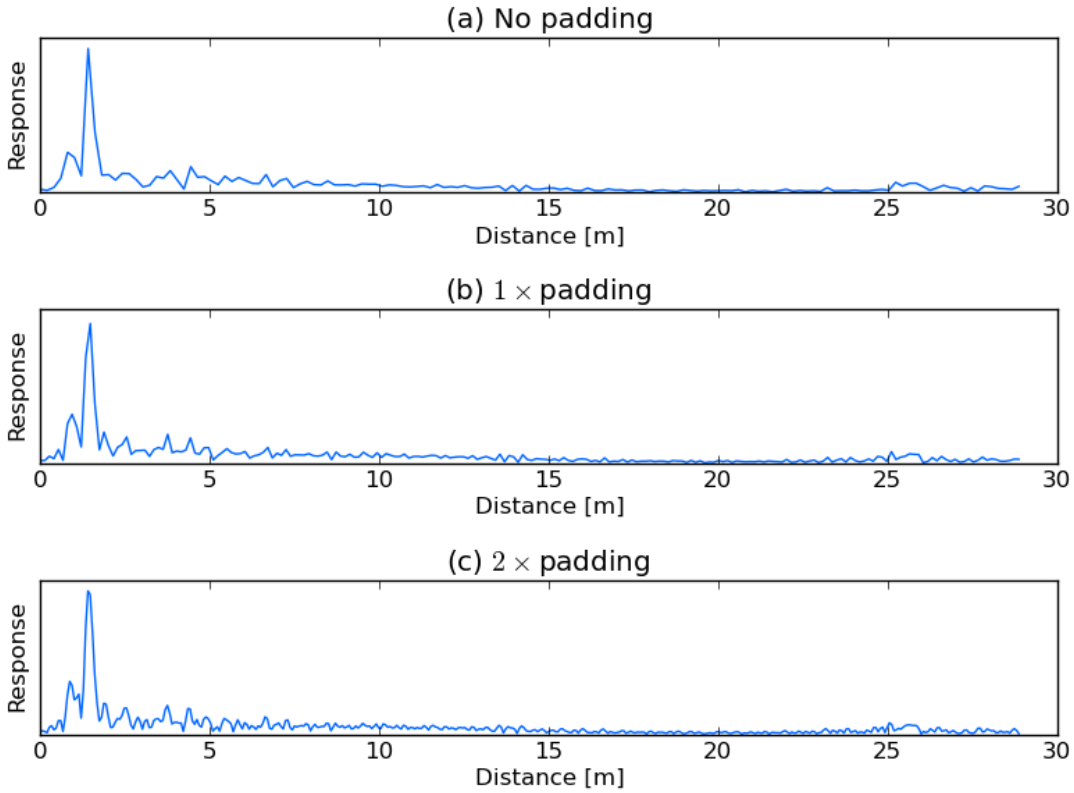


Fig. 3.4. Effects of zero padding on IF signal

3.5.4. Optimal processing parameters

To ensure optimum performance of the sensor, we need to set the parameters carefully.

Frequency resolution should be as high as possible, because frequency resolution means range resolution with FMCW radar. According to formula (3.10), bin size is $^{N}/_{f_n}$ and thus frequency resolution is $^{f_n}/_{N}$. Therefore two things can be done to increase the resolution: use a lower sampling frequency, or use more samples.

There is a lower bound on the sampling frequency. We want to detect frequencies up to a certain frequency that represents the maximum measurable distance. Thus we cannot use a sampling frequency of less than twice this maximum frequency. But we want to use as low sampling rate as possible.

The upper bound on number of samples is a soft one. Arbitrarily long measurements can be taken (apart from hardware limitations, of course), but there is a tradeoff between measurement quality and quantity. This tradeoff needs to be found empirically as number of sweeps that should be processed. Moreover, if too many sweeps are processed, the results become interleaved with zero or nearly zero responses, making the zeroed range bins obsolete and decreasing the actual resolution. This effect can be seen in Figure 6.4 as the empty columns.

Padding adds frequency results, increasing sample count and thus the resolution. Sidelobes can appear in the results and the computation cost grows, so employing padding should be carefully considered.

4. IGEP Lambda sensor

The Lambda radar sensor is a radar sensor module from a Spanish company ISEE. It is presented as an evaluation kit to test ISEE radar technology and ISEE provides possibilities to manufacture custom designs.

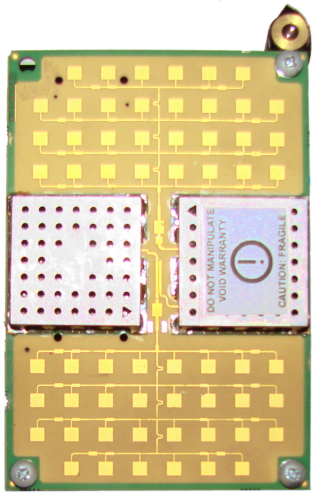


Fig. 4.1. The Lambda radar module. This sensor orientation will be referred to as vertical.

The sensor module consists of the radar sensor itself, equipped with a SPI interface, and an IGEPv2 embedded computer.

A block schematic of the Lambda module hardware is presented in Figure 4.2

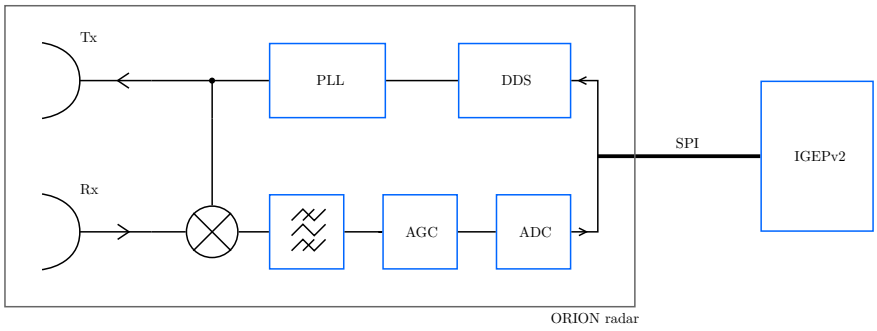


Fig. 4.2. Block diagram of the sensor hardware

4.1. Orion radar

The radar sensor itself is called the Orion sensor. It is a compact radar unit operating at 24 GHz. The 24–24.25 GHz band is public worldwide, rendering the sensor fit for usage anywhere. The sensor contains all the signal processing components and provides SPI interface to configure the sensor and read out the measurements.

4.1.1. Signal synthesis

The signal to be transmitted is generated by a *direct digital synthesizer* (DDS) and a *phase locked loop* (PLL). The DDS generates a low frequency signal with the desired modulation. The signal is then multiplied and amplified in the PLL.

The DDS can generate both saw and triangle modulation patterns. It is configured by a SPI interface. The configuration is expressed by several parameters:

FSTART sets the base frequency
STEP sets frequency increment
NINCR sets the number of frequency increments in one sweep
SLOPE sets the increment duration

The parameters are not the physical values, but rather an internal representation convertible to real values by simple formulae:

$$F_{start} = \frac{50 \times 10^6 FSTART}{2^{13}} = 6103.5156\ FSTART\ \ [\text{Hz}] \tag{4.1}$$

$$\Delta F = \frac{50 \times 10^6 STEP}{2^{13}} = 6103.5156\ STEP\ \ [\text{Hz}] \tag{4.2}$$

$$N_{incr} = NINCR \tag{4.3}$$

$$T_{base} = 20\ SLOPE\ \ [\text{ns}] \tag{4.4}$$

The resulting sweep will have duration $T = N_{incr} T_{base}$ and bandwidth of $BW = N_{incr} \Delta F$.

Once the modulated signal is generated, it is fed to the PLL where it is multiplied by a fixed ratio of 2048.

4.1.2. Received signal processing

The signal being received is mixed with the transmitted signal resulting in IF signal. The IF is amplified first.

A bandpass filter with cutoff frequencies 1[1] – 32 kHz is applied to the IF signal. This affects the range of measurable distances, as targets at ranges corresponding to IF frequencies outside this frequency range will be registered as much smaller or will not be registered at all.

[1] The value is ambiguous in the datasheet, values 1 kHz or 3 kHz appear [10, p. 27, 31]. The true value has been confirmed by experiment described in 6.1.2.

The IF signal is then passed through an *automatic gain control* (AGC) module. It corrects the IF signal amplitude. The AGC has an adjustable gain of 40dB, so it can adjust the signal intensity by factor of 10^4. The compensation results in for example targets at different ranges returning equal intensities.

Finally the IF signal is sampled by a 12bit *AD converter* (ADC) with an SPI interface. The sampling is timed by the SPI clock, so the sampling frequency can be adjusted. The converter supports sampling frequencies 500 – 1021 ksps [11].

4.2. IGEPv2 computer

The IGEPv2 computer is an embedded computer system based on a TI ARM processor. The ARM core supports NEON SIMD instructions and has access to an external DSP also present on the board.

The computer runs a Linux distribution for embedded systems.

4.3. Lambda radar module

When combined, the Orion sensor and IGEPv2 computer are called *Lambda* module. The SPI interface to the DDS and ADC on the ORION sensor are connected to HW SPI interface of the IGEPv2 board, which enables fast communication.

4.3.1. Module control

The module is controlled by a program called *server*. The server drives the Orion HW directly through the SPI interface, performs the necessary basic signal processing and provides a TCP API to allow other computers (*clients*) to control and use the sensor over LAN.

Clients can set modulation parameters, read out single measurements or run continuous measurement.

4.3.2. Setting DDS parameters

The modulation parameters can be set via the TCP interface. Parameters that can be programmed are modulation pattern, FSTART, SLOPE, NINCR and BW. STEP will be calculated from BW and NINCR by the server as BW is a much more important and relevant number than STEP.

The configuration changes are applied immediately. Changing configuration while in continuous measurement mode can have undefined results.

4.3.3. Setting ADC parameters

The ADC parameters can also be programmed through the TCP interface. The programmable parameters are sampling frequency and number of samples to be captured in a single measurement.

The ADC sampling frequencies are limited to frequencies supported by the processor SPI module. Configurable frequencies are 1021, 590, 324, 171 and 85 ksps. The ADC, however, supports only 500 ksps or more [11]. A software solution has been devised. The data are sampled at 590 ksps and then decimated by a user-set integer divisor. Maximum divisor value is 512.

The number of samples to capture is limited to 16384 at full speed. Since the decimation is carried out post-hoc, the number of samples captured is the requested number of samples times the divisor. So with divisor set to 8, the sensor can capture at most 2048 samples. It means at most about 20ms of data can be acquired whatever the sampling rate.

4.3.4. Taking measurement

The server is capable of providing clients with either raw IF signal or processed range data. The client can request a number of measurements or enter continuous measurement mode.

Each measurement starts its own modulation cycle. The previous transmission is canceled and a fresh cycle is started. This allows for precise synchronization of the sampling with the modulation, allowing users to sample for example precisely one ramp.

The measurements ignore the Doppler shift. The frequency shift results in range shift of about 12 cm per m/s of relative speed. As robots usually drive at low speeds and static environment is assumed by the mapping algorithm, the Doppler shift can be neglected.

Continuous measurements will be sent as fast as they can be acquired by the module. That usually means as fast as possible, because the optional processing typically takes less time than raw data acquisition.

4.3.5. Measurement algorithm

The server runs in three threads: data acquisition (DAQ) thread, processing thread and API thread.

The API thread is responsible for performing the operations requested by the client. That is for example altering the DDS/ADC settings or initiating measurements.

The measurements themselves operate on a two-stage pipeline. The pipeline works with two buffers used to store raw data that are operated as a degenerated round buffer.

Stage one gets raw data as acquired by the ADC. The data are read and stored in the buffer by the DAQ thread.

In stage two, the raw data are processed by FFT routine and sent to the client in the processing thread.

A batch of data can be processed while the next batch is being measured. This ensures the best measurement performance possible, because the processing and data transfer take typically much less time than it took to acquire the data.

4.4. Sensor properties

Since the sensor datasheet is unclear about basic sensor properties as radiation characteristics, we have performed a series of experiments to determine the actual values and effects of the following properties. The experiments are thoroughly described in the Experiments chapter in section 6.1.

4.4.1. Transciever pattern

The sensor detects targets in a cone-like area of the environment. The shape of this area is defined by the transciever characteristic (described in 6.1.1).

As we focus on 2D mapping, the important dimension of the cone is the beam width. We want the beam to be as narrow as possible to obtain maximum angular resolution. Beam width is defined as angle between directions in which the registered intensity is half the maximum registered intensity.

The Lambda sensor has beam width of 26°

The beam height is also important as it tells us how low/high placed targets will be registered. As opposed to the beam width, we want the height to be relatively high to register low targets close to the robot like rocks. The Lambda sensor has a high beam, as the main beam is accompanied by two so-called sidelobes. The combined beam height of the beam and the sidelobes is about 60°. There are however blind spots between the main lobe and the sidelobes (plotted in Figure 6.3a in detail).

The near-field limit for the sensor is 28.8 cm. That is comfortably within the non-measurable range and will not cause any trouble.

4.4.2. Range accuracy

The sensor returns a list of range-response pairs. This is different than common sensors like laser range-finders or sonars that return only one distance reading meaning the distance to the obstacle. The interpretation of the values is left to the user here. To measure distance to a target, one needs to know which response corresponds to a target.

Different target detection schemes are possible, the easiest of which are thresholding and maximization. In a single target scenario, when only one target is present in the field of view of the sensor, the target can be identified as the maximum echo value. If we allow multiple targets, we can mark a range bin as occupied when the echo exceeds some threshold intensity.

With the maximum-response target detector employed and with parameter settings as in Table 4.1 in a single-target scenario, the sensor shows error distribution in Figure 6.6 with mean +0.03 m and standard deviation 0.19 m. These values were obtained in experiment 6.1.2.

More complex detection schemes can be based on noise distribution estimation [3, sec. 3.3] or fitting response shapes to the range readings [12]. These techniques can largely improve the sensor performance in both single- and multi-target scenarios.

There is also the possibility of better performance with different module parameters.

4.4.3. Environment responses

The radio waves emitted by the sensor interact differently with different materials. All the materials reflect the signal predominantly in specular way, diffuse reflection being very weak (6.2.2). This results in the sensor ignoring surfaces viewed from high angles.

Common wall building materials like bricks and porous concrete (YTONG and others) produce strong reflections and occlude the rest of the signal so that obstacles behind the wall cannot be registered. Glass has the interesting property that it both returns a well detectable echo and permits enough of the signal through to let obstacles behind it to be detected.

Multipath reflections and false reflections have been registered during the experiments (6.3). These can bring problems into sensor applications and need to be taken into account.

4.4.4. Empirically optimal parameters

After experimenting with the settings, we have settled on a sub-optimal set of parameters.

parameter	*value*
SLOPE	128
NINCR	512
BW	250 MHz
sampling frequency divider	8
number of samples	288 (3 ramps)
padding	1×

Tab. 4.1. Best parameter combination found

5. Mapping algorithm

We will be reconstructing the map using occupancy grids with Bayes filter employed to incorporate the new measurement into the map.

Occupancy grids give us a way to represent the map in computer memory. Bayes filter provides the framework in to fill in the map. The sensor however still has some properties that need to be taken into account.

First is the sensor directionality. When a cell seems empty from one direction, it might be because of viewing angle with low RCS, like a wall viewed at large incident angle. The algorithm needs to compensate that a cell scanned from one direction may well return very different reading when measured from another angle.

Second is the low range and azimuth resolution. The range resolution of the experimental data is 30 cm. That is much compared to for example LIDARs. The azimuth resolution can make a small strong target fill the entire corridor in the map.

We have devised a modification to the Bayesian mapping algorithm to suppress these effects.

5.1. New approaches

5.1.1. Normals estimation

The problem with directionality is that the sensor looking along a wall sees empty space, even if it has seen a wall there before. Enough of these false measurements can override the previous obstacle registered and persuade the robot that the space is obstacle-free. We eliminate this behavior by estimating the surface normal in each cell. If we take the assumption of polygonal environment, not that far from the truth in urban and indoor environments, the assumption that every obstacle has a clearly defined normal is correct.

The normal is estimated as the direction of maximum registered return. The normal estimates are updated as the robot moves through the environment. The deviation from estimated normal is then used to weight the updates so that measurements along the normal have maximum weight, while measurements perpendicular to the estimated normal have no effect at all.

5.1.2. Confidence limiting

Another problem is that pure Bayesian algorithm was getting overconfident about free space. Too many measurements taken along a wall made the robot confident nothing is there. Then, when the wall was scanned perpendicularly, the new measurements were ignored. We have solved this by introducing an upper bound on the robot's confidence

in free space. A cell can be considered truly free only if it has been scanned from all directions and no obstacle has been registered.

The algorithm therefore keeps track of the directions in which each cell has been scanned. The directions are divided into discrete bins matching the wall registration angle as determined in experiment 6.2.2. Each cell has an upper bound on the confidence in free space (that is a lower bound on the occupancy probability) dependent on from how many of the direction bins the cell has been scanned.

5.1.3. Range attenuation

The measurements degrade with distance. The beam energy drops with range and noise is more likely to bias the measurement.

The beam is also wide at far ranges and a point target can seem much larger. The measurement contribution is thus weighted proportionally to the measurement range as to distribute the weight over the width of the beam. The erroneous wide target can then be corrected more easily due to lower weight of the wrong measurements.

5.2. Algorithm overview

The mapping algorithm reads a measurement consisting of pose information (x, y, z, θ) and a range-response vector $r_i : a_i$. The range-response vector is interpreted into Bayesian sensor quantities $p(M|O(r_i))$ and $p(M|F(r_i))$ for each range bin r_i. A linear interpolation step is then employed to improve the resolution. Then, for each interpolated range bin, the corresponding cells are iterated and their occupancy values updated. The update is weighted to eliminate the problems with directionality and range degradation. Finally the confidence bound is applied to eliminate false confidence in empty space (5.1.2).

5.3. Map representation

The map is represented as a set of two-dimensional arrays of variable size. The physical dimensions of the represented environment are independent on the array size. Six arrays are used to accommodate the map and metadata used by the algorithm.

One array contains the occupancy grid itself. The cells contain $p(O(C))$ at the given time. The probabilities are represented directly. Numerical stability is not an issue thanks to confidence limiting (5.1.2).

Two arrays contain the x and y components of the estimated normal for each cell C. The normal vector is represented as a unit vector $\mathbf{n_C}$. One array contains the maximum return associated with the estimate a^C_{max}.

One array is used to keep track of the directions a cell has been scanned from. The integers in the array are used as bit arrays where each bit is associated with one scanning direction.

The last array holds the lower bounds for the empty space confidence p^C_{min}. This array is used to avoid computing cell Hamming weights every time and increase the computational efficiency.

5.4. Range-response vector interpretation

Foessel has developed radar sensor model for Bayesian framework in [3]. We have adopted a simplified version of the model.

First the vector a_i is sharpened by subtracting local mean from each a_i.

$$a'_i = 1/N \sum_{j=i-N/2}^{i+N/2} a_j \tag{5.1}$$

The border conditions are resolved by clipping the window and taking average only from valid values. N being a small number, the bordering N/2 measurements are always invalid due to the band-pass filter anyway. This step reduces peak width, making the obstacles in the signal better defined.

The vector $r_i : a'_i$ is then interpreted according to heuristic rules. Obstacles occlude the signal and what lies behind them, even though some allow the signal through and let obstacles behind them to show. The more obstacles lie between a response value and the sensor, the less likely the value is to be correct. If another target is not registered through an obstacle, the low return behind the obstacle does not necessarily mean empty space. Low return between two targets however means high confidence for empty space.

Obstacles are identified by a simple threshold T. This is a rather weak classifier, so the $p(M|O(r))$ and $p(M|F(r))$ are only slightly offset from $1/2$. As a cell is measured repeatedly, the weak values integrate into a strong classification of the cell.

Let us have N targets in the measured vector. Let n obstacles be present between the sensor and the interpreted range bin r_t. Then the update values are

$$p(M|O(r_t)) = \begin{cases} 1/2 + 0.3\frac{N-n}{N} & \text{if } a'_t > T \\ 1/2 - 0.1\frac{N-n}{N} & \text{if } a'_t < T \end{cases} \tag{5.2}$$

$$p(M|F(r_t)) = \begin{cases} 1/2 - 0.15\frac{N-n}{N} & \text{if } a'_t > T \\ 1/2 + 0.05\frac{N-n}{N} & \text{if } a'_t < T \end{cases} \tag{5.3}$$

The constants for offset from $1/2$ have been chosen empirically based on the heuristic that if a target is detected, something must have caused the reflection. When a target is not detected, on the other hand, it could mean it has only been scanned from a wrong direction.

Formulae (5.2) and (5.3) have the desired properties. Behind the last obstacle, $n = N$ and both the terms are equal to $1/2$ meaning no information. The less obstacles in front of processed bin, the more influence the classification has.

The range, amplitude and measurement probability values are then linearly interpolated to improve the resolution of the generated map. The measurements, spaced at 20 cm or more, are sparse and can be used to build very low-resolution maps. As the range measurement standard error is about one range bin, the interpolation does not improve the map resolution for machine interpretation. The interpolated map is however more comfortable to view for humans.

5.5. Map update

The map update step receives robot pose and the interpreted range-response vector, that is (x, y, z, θ) and tuples $(r_i, a_i, p(M|O(r_i)), p(M|F(r_i)))$ for interpolated range bins i. All these data, when put together, form the measurement M referred in (2.3).

5.5.1. Cell iteration

The algorithm needs to update all cells that lie in the measured area. The sensor has a fan beam with horizontal beamwidth of 26 °. When projected to the ground plane, the beam has a circular sector shape, with points equidistant from the sensor forming arcs. In each step, we need to iterate cells in the arc that corresponds to range r_i and update them.

To simplify the algorithm and reduce computational cost, the arcs are approximated by straight lines with least square distance from the arc. This line is iterated for each interpolated range bin. At maximum measurable range, the maximum error of this approximation is 21 cm.

The algorithm uses Bayes update rule to incorporate new measurement into the current grid map. The Bayesian update is however weighted with two factors characteristic for radar. One is the directionality, the other is range signal attenuation.

5.5.2. Update weighting

When incorporating a single measurement into the grid, the grid contains the priors $p(O(C))$. The update as in (2.3) therefore consists of multiplying the prior $p(O(C)$ by update factor

$$upd(C) = \frac{p(M|O(C))}{p(M|O(C))p(O(C)) + p(M|F(C))(1 - p(O(C))} \tag{5.4}$$

The update weight is calculated as

$$w = |\cos(\phi)|^{10} \left(\frac{r_{max} - r}{r_{max}} \frac{r_{fallof} - 1}{r_{fallof}} + \frac{1}{r_{fallof}} \right) \tag{5.5}$$

where ϕ is the deviation of measurement vector from the estimated normal $\mathbf{n_C}$, r_{max} is the maximum measurable range and r_{fallof} is range falloff parameter.

The first term takes the estimated normal into account. The exponent 10 was determined to reduce weight to approximately $^1/_3$ at 26 °, the beamwidth. Cosine has been chosen for being the simplest form of computing angle between two unit vectors as a simple dot product.

The second term represents linear weight falloff with range. The function yields 1 at zero range, $^1/_{r_{falloff}}$ at range r_{max} and decreases linearly with range.

Alltogether, w assumes values $[0, 1]$, zero meaning no confidence in the measurement, one total confidence.

The update factor is then weighted by w and merged into the grid

$$p(O|M) = p(O)(upd(C))^w \tag{5.6}$$

By exponentiating $upd(C)^w$, zero weight results in multiplication by one and no update, while unit weight doesn't change the update value at all.

6. Experiments

6.1. Sensor properties

The datasheet [10] provides a very poor specification of the sensor's properties. We have conducted a number of experiments to determine the properties of the sensor in order to assess the possible applications.

6.1.1. Radiation and receiving characteristic

Radiation characteristic of an antenna specifies the intensity of signal emitted in a given direction. It allows us to specify for example the direction (angle) at which the target receives enough power for the reflection to return. Another property closely related to the radiation characteristic is the *receiving characteristic*, which specifies what intensity will be registered for signal coming from a particular direction.

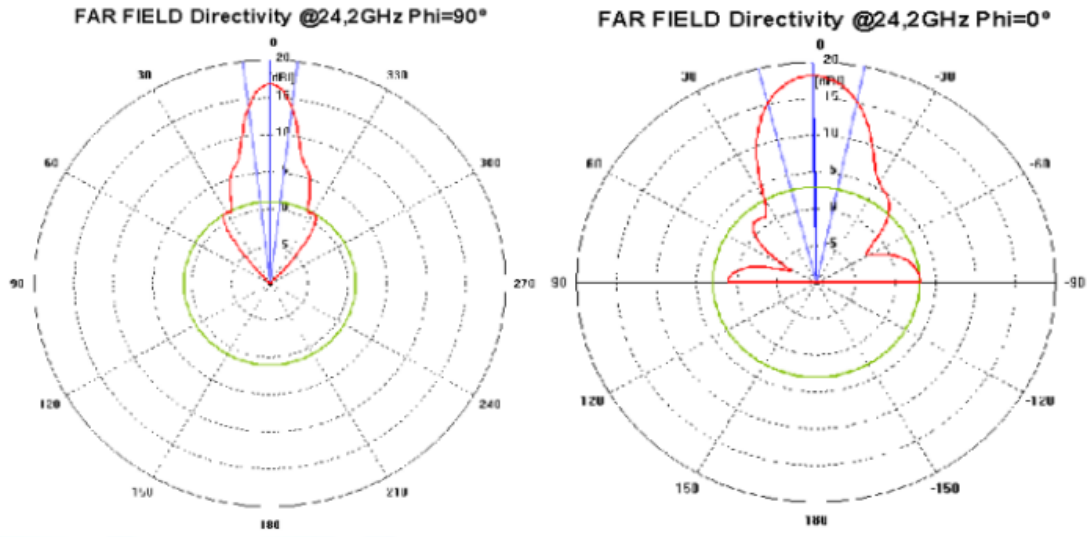


Fig. 6.1. Radiation characteristic as presented in the datasheet [10]

A combination of these two is of particular interest for application of the sensor as a monostatic radar. Since the transmitting and receiving antennas are identically oriented, the transmission angle to a target is equal to the reception angle for the echo. The result is the *transciever characteristic* specifying registered intensity for a target illuminated by the sensor itself.

If the radiation and receiving characteristics are known, the transciever characteristic is a simple product of the two. Since antenna gain is given in dB traditionally, the product becomes a sum.

The datasheet [10] specifies the radiation characteristic with a poor-resolution image of two cross-sections of the characteristic 6.1. One is supposedly a vertical cross-section while the other should be a horizontal one — but which is which is not stated.

We have set up an experiment to determine the transciever characteristic of the sensor. The sensor has been mounted on a pan-tilt actuator unit on a tripod. A reference corner reflector was placed 4m from the sensor in the same height of 1.3 m. The sensor and reflector were aligned using a laser pointer to ensure precise orientation.

The experiment has been conducted on a narrow roof to ensure as low clutter as possible. Therefore all the measured positions were unobstructed, i.e. no objects other than the reflector were present in the main lobe.

The radar configuration was as follows: BW=250 MHz, SLOPE=128, NINCR=512, 480 samples (5 ramps).

The pan-tilt unit was then used to scan the different angles of radiation of the sensor. The scanning mesh used spacing of 1 °. 10 measurements were taken at each mesh node. An average was taken from measured values for each node.

The pan-tilt unit operation range is limited in the tilt axis. To accommodate for this and to avoid pointing the radar to the ground, we have changed the sensor orientation and scanned the target four times alltogether. That allowed us to always point the radar to the free space to eliminate ground reflections.

Figure 6.2 shows cartehsian plot of the combined results. There are surprisingly intensive sidelobes in the vertical plane, their intensity is nearly the same as of the main lobe. Second sidelobes are also present in this plane. In the horizontal plane, on the other hand, the beam is relatively narrow and no sidelobes appear. The interference lines come from local traffic on the 24 GHz band used for Internet connection.

Vertical and horizontal cross-sections of the characteristic are presented in Figure 6.3. In the vertical cross-section 6.3a, the side lobes are clearly visible. The intensity of the first side lobes is at about -1 dB relative to the main lobe, the second side lobes are at $-4 - -5$ dB.

The horizontal cross-section shows a single lobe. The single narrow lobe implies that optimal placement of the sensor is vertical to ensure maximum angular resolution. The beam width, defined as the angle between half-power (≈ -3 dB) intensity, is 26°.

6.1.2. Range measurement accuracy

The discrete nature of FFT used to calculate distance data from the received signal implies some sort of binning. Only certain frequencies are recognised by the FFT. What if the target is directly between two range bins?

The datasheet [10] simply states that measurement accuracy is ± 1 mm|100 mm|300 mm (minimum | typical | maximum) with *„Data considering 200MHz Modulation Bandwidth, single point target scenario, 75dBsm RCS target placed at 5m distance and adequate modulation and IF signal processing is applied.“*[10] — that is under optimal conditions.

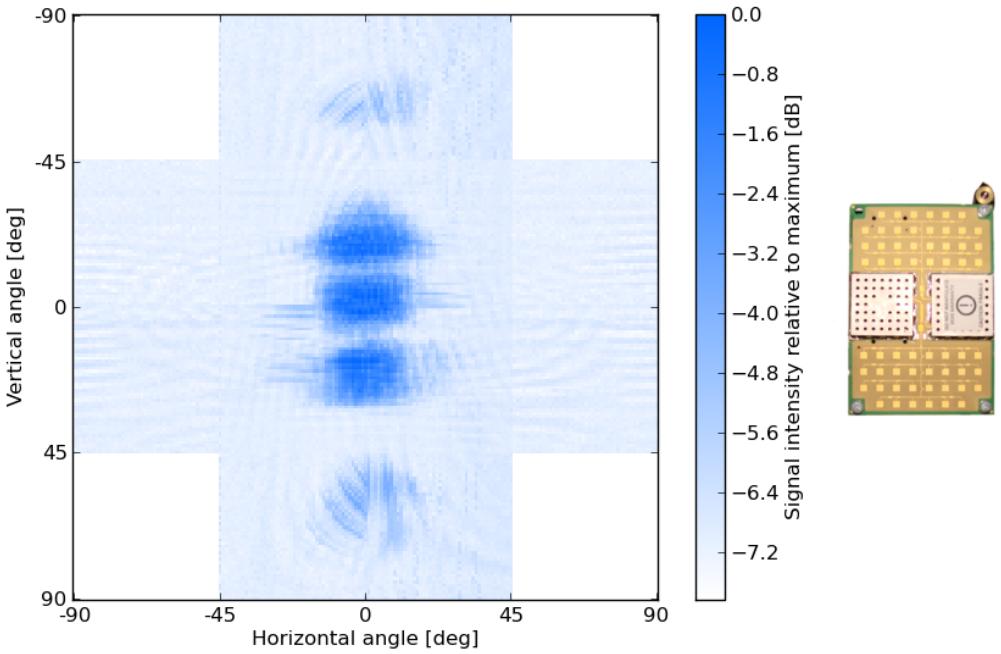
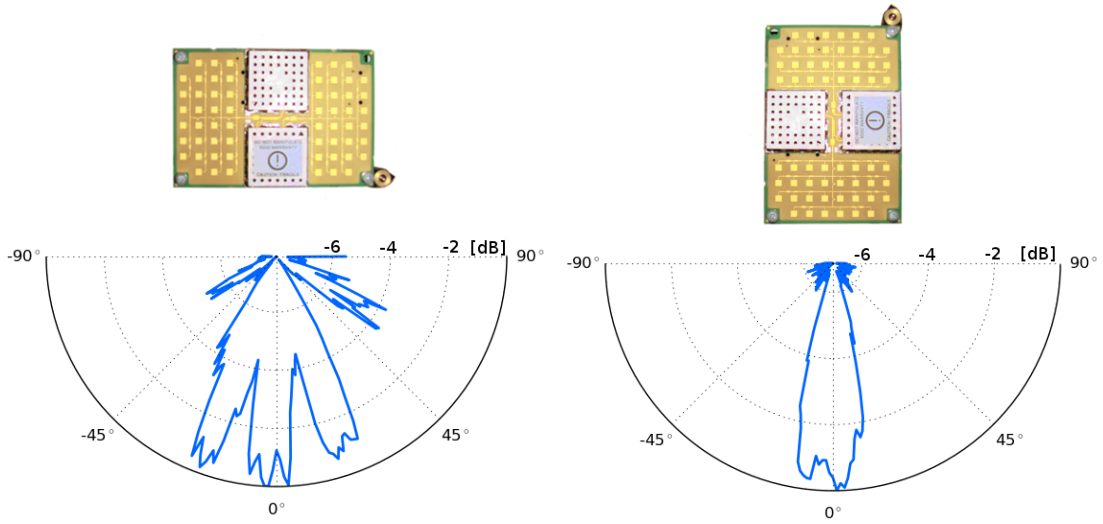


Fig. 6.2. Measured transciever characteristic in a carthesian plot. The associated sensor orientation is shown on the right.



(a) Vertical cross-section **(b)** Horizontal cross-section

Fig. 6.3. Measured transciever characteristics in polar plots. The associated sensor orientations are depicted above the plot.

We have performed an experiment to determine the range accuracy of the sensor. A strong target (a 1mm aluminum sheet) was placed in front of the sensor. High placement precision was necessary because of extremely high reflectivity and low diffusion of aluminum, even the slightest deviations caused the sheet to "disappear" for the radar as all the transmitted signal was reflected away from the receiver.

Then the sensor was moved away along a line, taking 10 measurements every 1 cm. The actuation was performed by hand, precise alignment was again achieved by a laser pointer.

The sensor configuration was BW=250 MHz, SLOPE=128, NINCR=512, 384 samples (4 ramps), no padding.

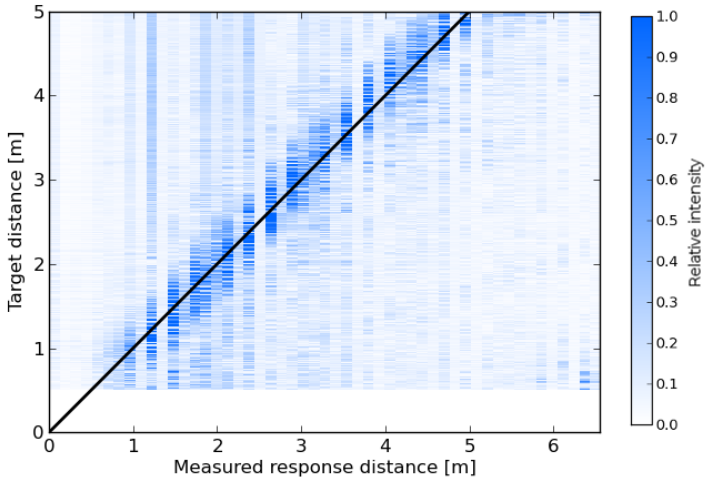


Fig. 6.4. Responses measured for target at known distance (384 samples per measurement)

The raw results are shown in Figure 6.4. Every horizontal line represents mean of the 10 measurements taken at the particular distance from the target. The color represents the echo intensity in the particular range bin normalized by the highest registered echo. The black line shows the true range.

Based on the data we can confidently state that the bandpass filter of the sensor has low cutoff frequency of 1 kHz. 1 kHz IF signal corresponds to 0.9 m range, which is where the data start to appear correct.

We can note the "stripey" nature of the data. Some range bins are never used, or always have a very low return. This is caused by the effect of processing too many modulation ramps effectively repeating the signal as described in 3.5.4. The empty bins are useless, so we have reduced the sample number for the following measurements to 288 (3 ramps). These results are plotted in Figure 6.5.

Because the experiment contains a single target, we have chosen a simple target detection scheme of maximum return. The measured target distance, as referred to in the following analysis, is the range corresponding to the range bin containing the highest response.

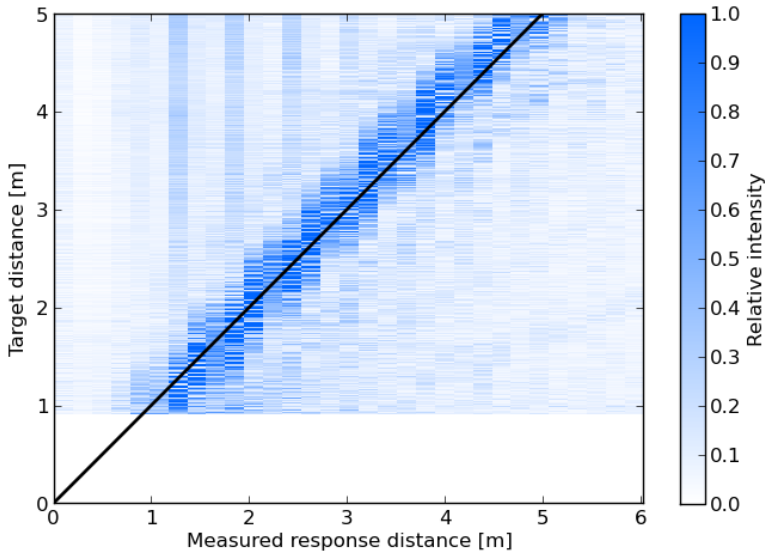


Fig. 6.5. Responses measured for target at known distance (288 samples per measurement)

Figure 6.6 shows the distribution of measurement error, the difference between the measured target range and the known true range. Only measurements with true range in the measurable range, i.e. farther than the range corresponding to low cutoff frequency of 1 kHZ (which is 0.9 m with the radar configuration used), are considered.

The error distribution has practically zero mean (+0.035 m) and standard deviation 0.194 m.

In Figure 6.7 the estimated ranges are plotted against the known values. The grid shows the range bins on both axes, the line shows the true relation. Some bins are skipped as in Figure 6.4, but this is caused by the simple distance estimator. A more advanced technique of target identification could improve the results drastically.

With the configuration used in this experiment, the radar has range resolution of 20 cm. The resolution can be improved by FFT padding as described in 3.5.4. The resolution improvement however doesn't improve the accuracy.

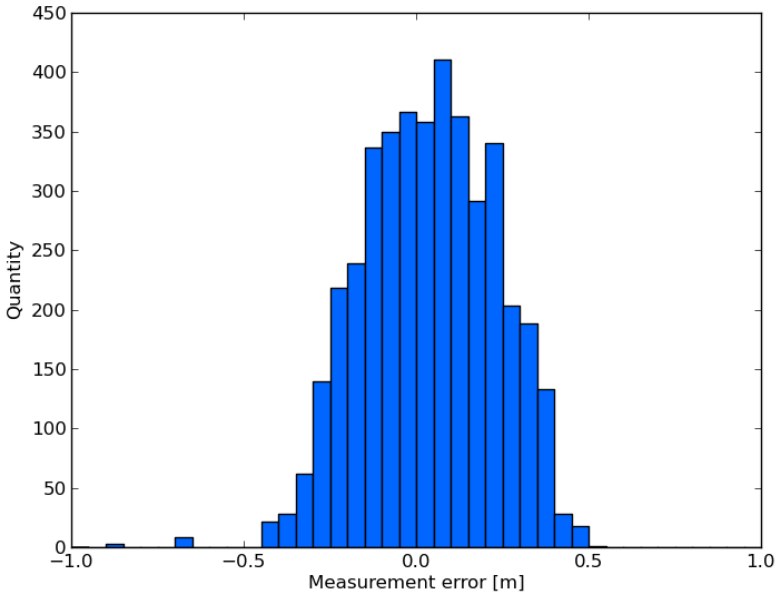


Fig. 6.6. Measurement errors over the whole experiment

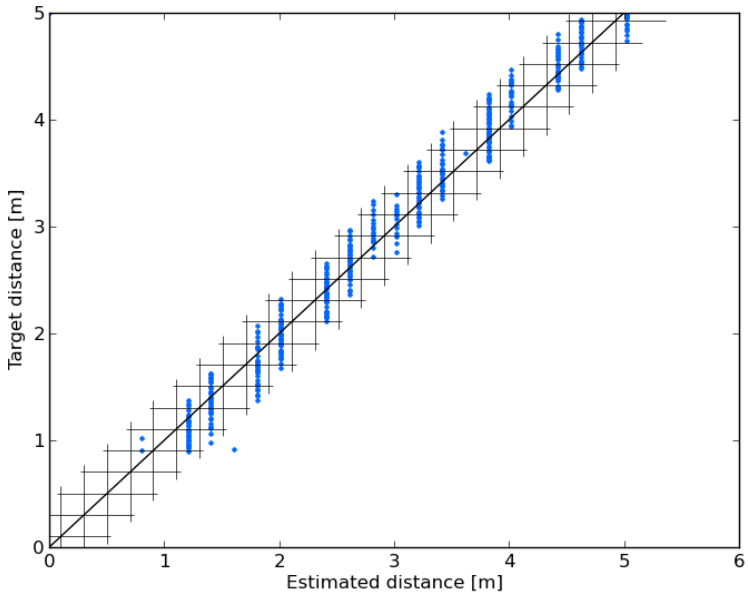


Fig. 6.7. Estimated range vs. true range. The grid represents the range bins on both axes

6.2. Material properties

Radio signal interacts differently with different materials. Materials show different echo amplitudes and amounts of diffuse return, some materials even let the signal through, allowing the radar to "see through" them. We have performed an experiment to charasterise some materials commonly found around.

6.2.1. Reflectivity and transmittance

The sensor was aimed straight at a reference corner reflector 5 m away. Material samples large enough to fill the whole beam width and effective height (the angle of view of the reflector) were inserted into the path of the beam 3 m from the sensor. The walls were measured separately by placing the reflector behind the wall. With each sample obstacle in place, 10 measurements were taken.

Average of the 10 measurements is shown in figs. 6.9a to 6.9j.

An intuitive look on the results divides the materials into three classes: *transparent*, *semi-transparent* and *opaque* materials. Transparent materials like paper and polystyrene barely show in the results, allowing nearly all the signal power through. Semi-transparent materials as glass show up in the results, but let some of the signal through (and back). Opaque materials like aluminum reflect all the power back and don't let nearly anything through.

obstacle material	transmittance [dB]	reflectivity [dB]
Air (reference)	0.000	-5.963
Paper (250 g/m^2)	-0.139	-5.742
Extruded PS (6 mm)	-0.124	-5.993
Cotton cloth	-1.001	-4.331
Cardboard (2 mm)	-2.779	-2.875
Window glass (1 mm)	-2.688	-2.768
Spruce (18 mm)	-3.893	-5.073
Al sheet (1 mm)	-6.370	-1.703
YTONG wall (10cm)	-6.530	-0.371
Brick wall (60cm)	-8.347	-0.558

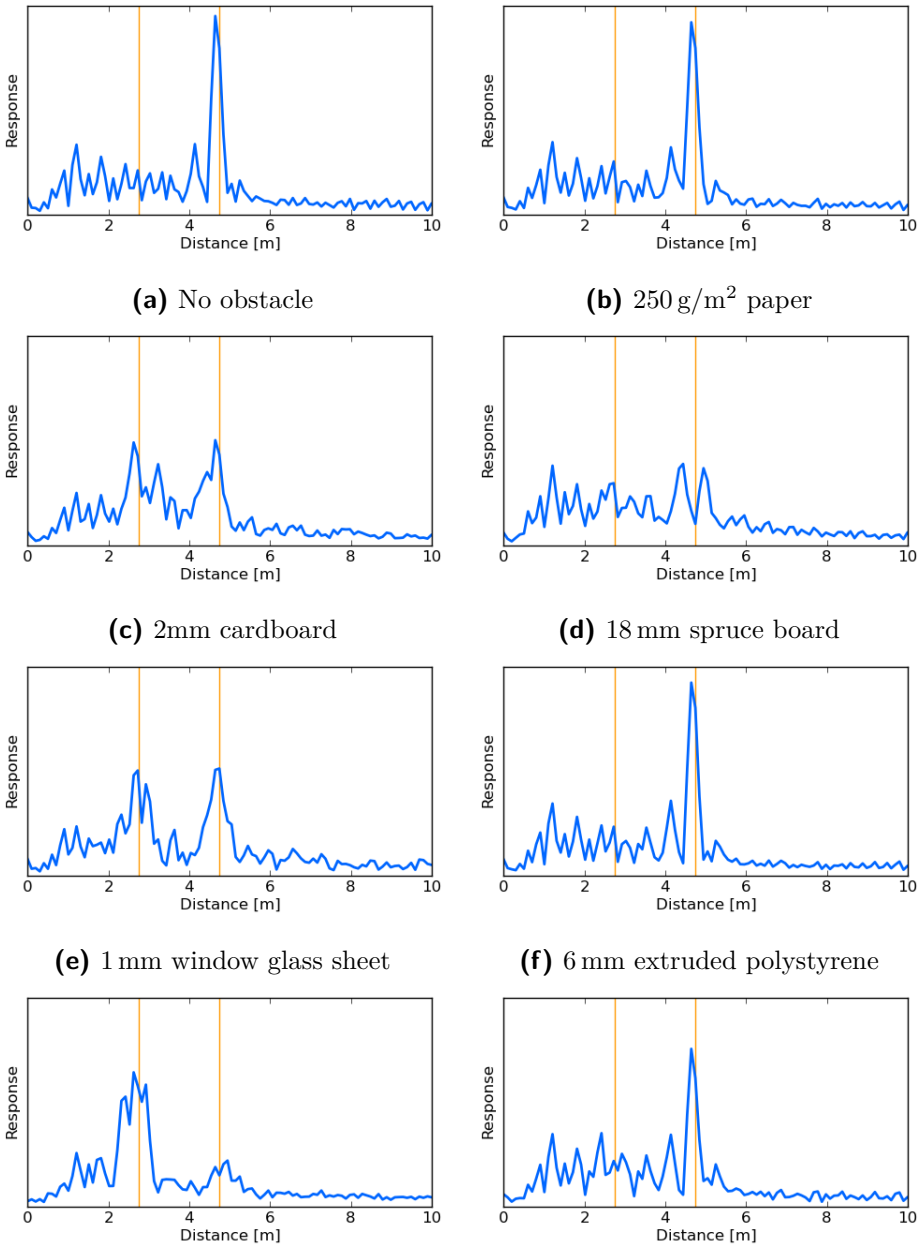
Tab. 6.1. Material reflectivity and transmittance. Transmittance is in dB relative to air (background noise). Reflectivity is in dB relative to the reference reflector.

Table 6.1 summarizes the results for material reflectivity and transmittance. The most surprising result is for glass. We expected glass to be highly transmissive, it however also reflects a considerable amount of the signal.

6.2.2. Diffuse and specular reflections

Initial experiments showed that flat surfaces as walls are ignored by the sensor when viewed from higher angles. We have preformed an experiment to verify and study this

Fig. 6.8. Reflection and transmittance measurements of common materials. The graphs are equally scaled. The orange lines show the position of the targets.



(a) No obstacle

(b) 250 g/m^2 paper

(c) 2mm cardboard

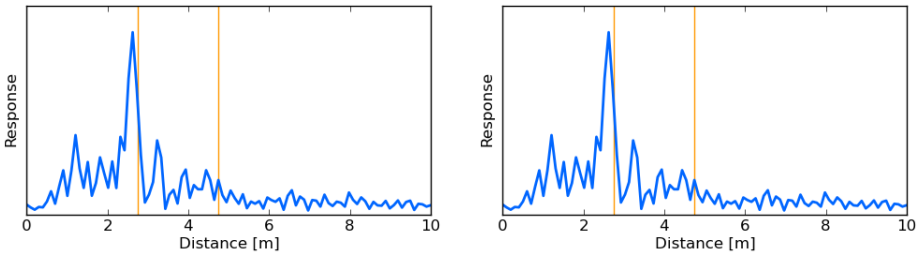
(d) 18 mm spruce board

(e) 1 mm window glass sheet

(f) 6 mm extruded polystyrene

(g) 1 mm aluminum sheet

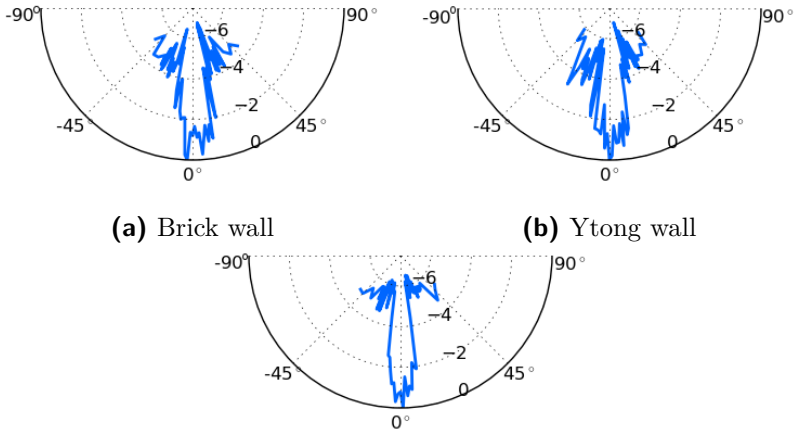
(h) Cotton cloth



(i) 60 cm full brick wall **(j)** 10cm YTONG wall

phenomenon on common materials used for walls in indoor and urban environment.

The sensor was aimed at a brick wall, an ytong wall and a sheet of window glass 2 m away from angles −50°–50°. The angle was changed by 1 degree in the interval −30°–30° and by 5 degrees elsewhere. The sensor was always aimed to the same spot, this alignment was ensured by a laser pointer. 10 measurements were taken at each angle, the mean of these measurements is used.



(a) Brick wall **(b)** Ytong wall

(c) Glass

Fig. 6.10. Reflection characteristics for common building materials. The value is in dB relative to the maximum return.

The measured values are shown in Figure 6.10. We can see certain amount of diffuse reflection from the walls. The walls have a rough surface and thus some diffusion occurs. The diffuse reflections are however very weak at −4 dB. The visibility angle for both the walls, if defined the same way as beamwidth as the angle of half-power return (−3 dB) is identical to the beamwidth of the sensor, 26°.

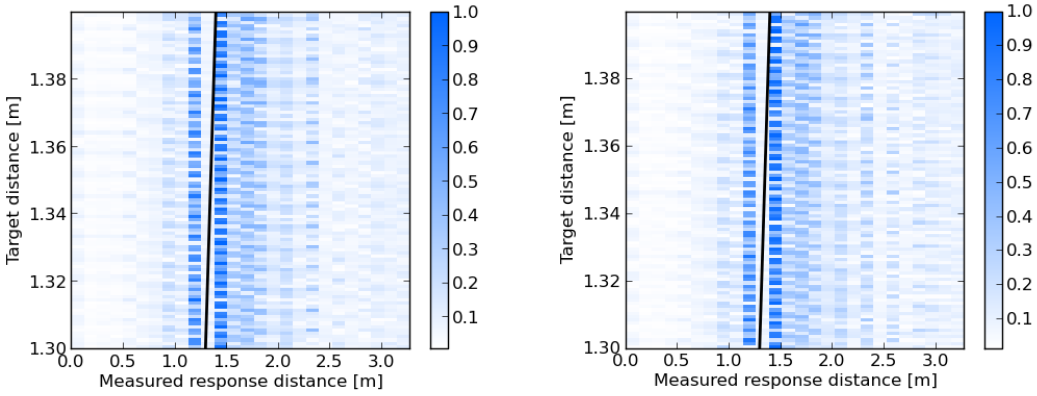
With glass, the visibility angle is 18°, even smaller than the beamwidth. Diffuse reflections from glass ale virtually non-existent. Glass has a very smooth surface that reflects all the signal in a specular way.

6.3. Sensor behavior in the environment

6.3.1. "Chaotic" measurements

When manipulating the radar by hand, we noticed that even the slightest shifts have a dramatic impact on the registered signal. We have carried out an experiment hoping to explain this phenomenon.

The setup was identical to the experiment described in the above section 6.1.2. We moved the radar between the 1.3 m and 1.4 m marks in 1 mm increments, taking 10 measurement in each step. The results are plotted in Figure 6.11.



(a) Responses in environment 1 **(b)** responses in environment 2

Fig. 6.11. Responses measured for target at known distance with very fine range resolution

The periodic nature of the measurements is evident. The period of the measurements is 6-7 mm, which is $^1/_2$ the wavelength of the radar signal. We have repeated the experiment in the same environment with identical results, which rules out random noise.

Further experiments in different location show very similar results. However, when compared carefully, the images are distinct.

We believe that the changes are caused by interference of signals bouncing around the environment. As of now, this phenomenon can cause trouble changing the response completely after just a milimetric shift. However, it also presents an opportunity to find something more about the environment.

6.3.2. Multipath reflections

We have confirmed occurrence of multipath and false measurements with the sensor. A scenario as in Figure 6.12 occurred, when two returns came from a reference corner reflector.

False reflections also occurred. The sensor was aimed on reinforced concrete ceiling. A corner reflector placed next to the sensor was registered as path sensor – ceiling – reflector – ceiling – sensor. The respective measurements can be found in Figure 6.13.

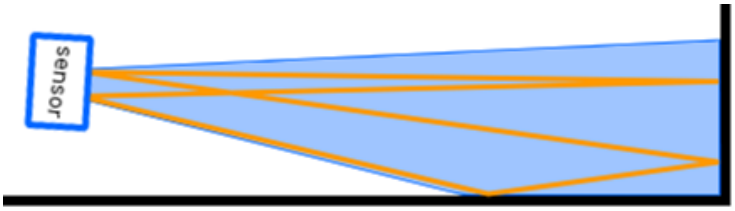


Fig. 6.12. Example of multipath reflection of radar waves in the environment. The floor appears as an obstacle behind the front wall.

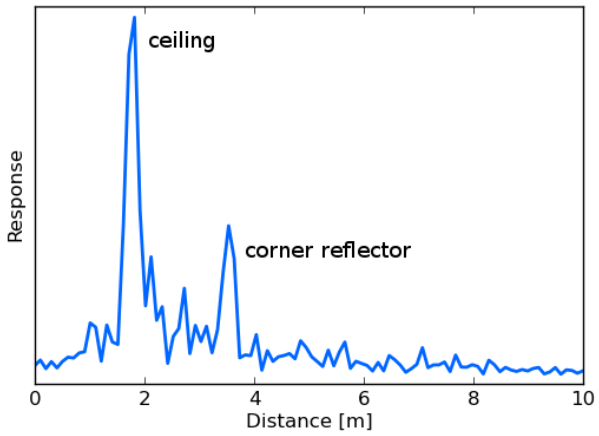


Fig. 6.13. Measurement containing a true target (a reinforced concrete ceiling) and a false target (a reference corner reflector placed next to the sensor)

The primary and the secondary reflector need to be highly reflective for the false reflection to occur. Multipath measurements and false reflections can cause trouble when building a map.

6.4. Mapping experiments

We have performed a series of experiments with the sensor mounted on a mobile robotic platform. The platform for indoor experiments was driven by two stepper motors. The steppers enabled precise steering and thus collection of high-precision odometry data. This allowed us to rely solely on odometry data on shorter distances. For longer drives, the robot has been equipped with a SICK PLS laser rangefinder to correct the odometry with a SLAM localization algorithm.

The outdoor platform moved on tracks, making usage of SLAM algorithm for localization. It was also equipped with SICK PLS laser rangefinder for this task.

The SLAM algorithm used in both cases is a particle-filter based algorithm from the GMapping library [13].

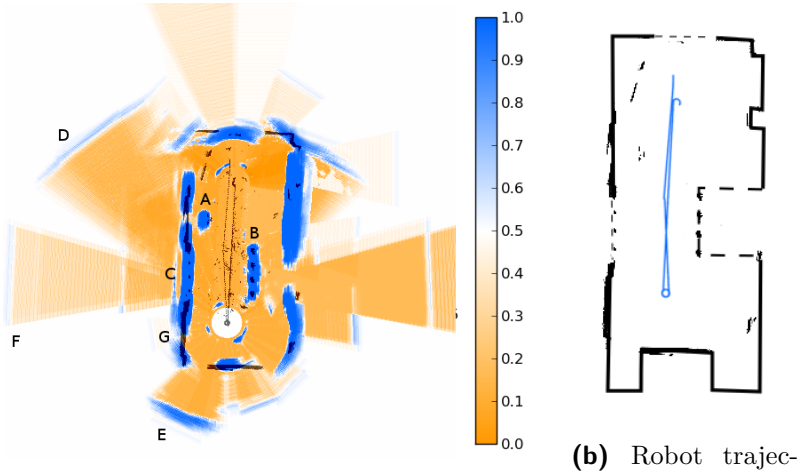
The Lambda sensor assembly was identical in both cases. The Lambda sensor was

mounted in vertical orientation on a sweeper platform driven by a stepper motor. The sweeper platform enabled us to scan the environment by a sweeping motion when necessary. Under the radar there was a small Hokuyo laser rangefinder. The rangefinder remained stationary even when the radar sensor was swept. Data from the rangefinder helped with orientation in the map and provided "ground truth" for interpreting the map. In case of laser-unfriendly materials like glass, the laser data weren't that true. The whole assembly was mounted side-facing on the robot.

The reconstructed maps are overlaid with laser scans. Large images are included in appendix A. All the experiments were performed in the CTU complex on Karlovo náměstí.

6.4.1. Building A entrance hall

The robot was driven around in the entrance hall in building A. The floorplan and the trajectory are depicted in Figure 6.14b. The robot started on the upper end and continued downward. On the lower end of the trajectory, the robot performed a 540° turn and returned back. The trajectory was short enough to allow us to rely solely on odometry for robot localization.



(a) Reconstructed map

(b) Robot trajectory

Fig. 6.14. Experiment conducted in the building A entrance hall. The floor plan sketch (b) is built from laser data by hand. The vacancies in the left and upper wall are glass-filled doors. The dashed line represents the tourniquets.

The reconstructed map 6.14a resembles the environment to human eye. The walls registered as strong and wide returns. This is however due to the fact that the robot was driven on a optimal trajectory parallel to the walls. The points of the walls that have been scanned only during rotations at higher incidence angles, e.g. point G, appear empty. This behavior is expected and has been confirmed by experiment 6.2.2.

Large metallic targets as the steam generator A and the tourniquets B exhibit the expected strong returns. The glass-filled door C are registered as well as the brick walls.

As glass is quite transmissive for the radar signal, the echo F showed through the door. The distance to the echo does not offer any false echo explanation, but corresponds well to a tram driving past the building.

Another distant echo D is registered partly through the glass door, partly through the thick outer wall. The low transmittance of the wall together with the echo strength indicate D being a false reflection. It is indeed perfectly symmetrical to the upper right corner of the room.

Echo E should correspond to the lower left corner of the room, but is about 2 m further than the actual corner. That suggests multipath reflections occurring in the narrow space between the wall and the doorkeeper's booth.

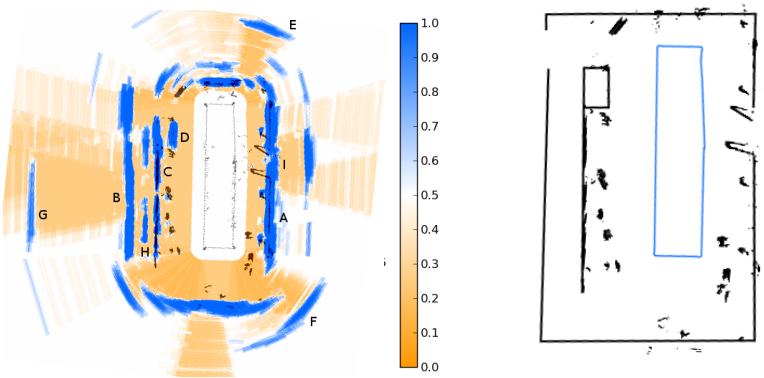
The effect of confidence limiting is clearly visible in the map. Deeper orange areas have been scanned from more directions, allowing the algorithm to become more confident about their emptiness. The area in the upper right corner is considerably better scanned than the rest of the room, resulting in lower occupancy values.

6.4.2. Building A atrium

The robot was driven around the glass atrium between building A and the inner yard. The trajectory is depicted in Figure 6.15b. The robot started in front of the door on the right side and then drove around the room clockwise. The localization relied solely on odometry again.

In the first round, the sensor was stationary. In a second and third run, the sensor was scanning the environment by rotating ±70° from the position perpendicular to the driving direction.

Stationary sensor



(a) Reconstructed map **(b)** Robot trajectory

Fig. 6.15. Experiment performed in the glass atrium, first pass. The sensor was stationary. The floor plan sketch (b) is built on top of laser data by hand.

Map reconstructed from the first run is presented in Figure 6.15a. Both the glass (A) and the concrete (B) walls return a strong echo. Even a low concrete railing of the same height as the sensor placement (C) is well visible.

A glass exhibition box containing a massive steel generator D shows returns from both the front and the rear face. The signal must have traveled around the generator, as signal penetration is impossible.

Echos E and F are well outside the glass walls and could be attributed to the other buildings around. Echo G is however registered behind a thick outer building wall and is thus probably a false target. Echo H registered as a third wall between railing C and wall B is most curious. We have no explanation as to why it occurred.

The door I point out the poor cross-range resolution of the sensor. The door opening is 1.3 m wide and yet it seems filled to the sensor. The next door further on the right show through however.

Scanning sensor

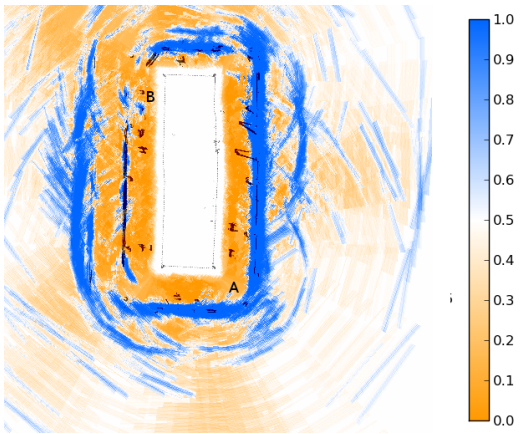


Fig. 6.16. Experiment performed in the glass atrium, second pass. The sensor was scanning the left side as the robot was driving. The floor plan and the trajectory are identical to 6.15b

Figure 6.16 presents the map reconstructed from the second pass. The robot trajectory was identical to the first run.

The important features like walls appear on both maps. In the scanned run, the corners as A could be reconstructed more precisely due to more view angles. Small features like the generator B have however been suppressed by the algorithm.

The features in the map generally appear thicker than they are. It is caused by integrating measurements of the wide beam from multiple angles.

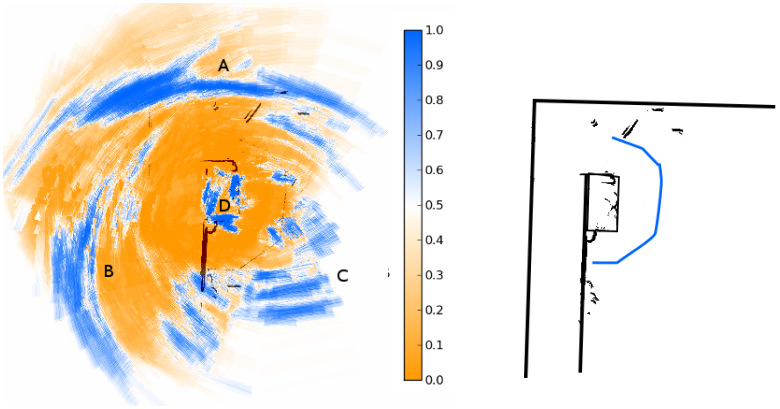
Around the generator

We have driven the robot around the exhibited generator in the third drive. The experiment was meant to test the algorithm in conditions where scans from many directions

need to be fused. The reconstructed map in Figure 6.17a exceeds the expectations.

The upper wall A is registered strongly. The side wall B appears to be 2 m further than in reality. It is probably caused by a multipath reflection. The returns around the room C are caused by spectators and people moving around the room.

The generator itself D is well defined with front, rear and left walls registered. The right wall was only scanned at an angle and so was not registered.



(a) Reconstructed map **(b)** Robot trajectory

Fig. 6.17. Experiment performed around the exhibited generator. The sensor was scanning as the robot drove. The floor plan sketch (b) is built on top of laser data by hand.

6.4.3. Building E first floor

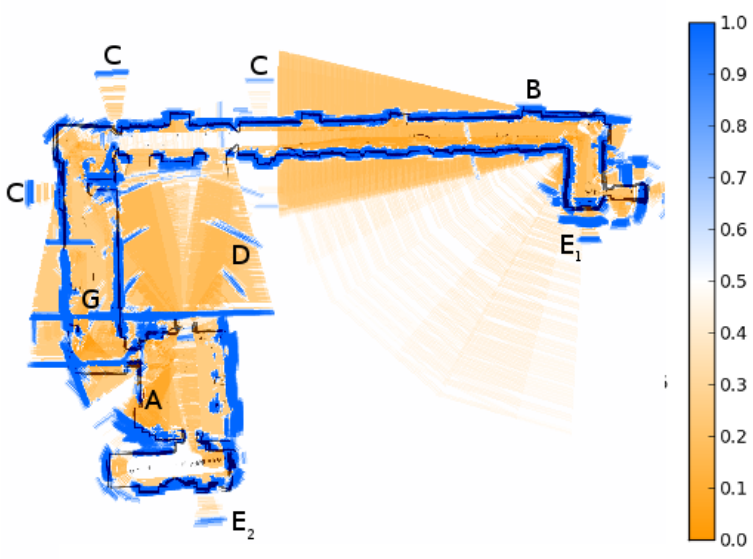
The robot was driven around the first floor of building E. The trajectory is depicted in Figure 6.18b. The robot drove along the walls with the sensor assembly facing left. The trajectory is long enough for the odometry error to take effect, so it was necessary to correct the localization by SLAM. Even with the correction, the localization was imperfect mainly in the bottom part.

The reconstructed map is presented in Figure 6.18a. The walls show well when scanned perpendicularly. When scanned from a higher angle as in A where the robot was driven at an angle to the wall, the wall is registered weakly or not at all.

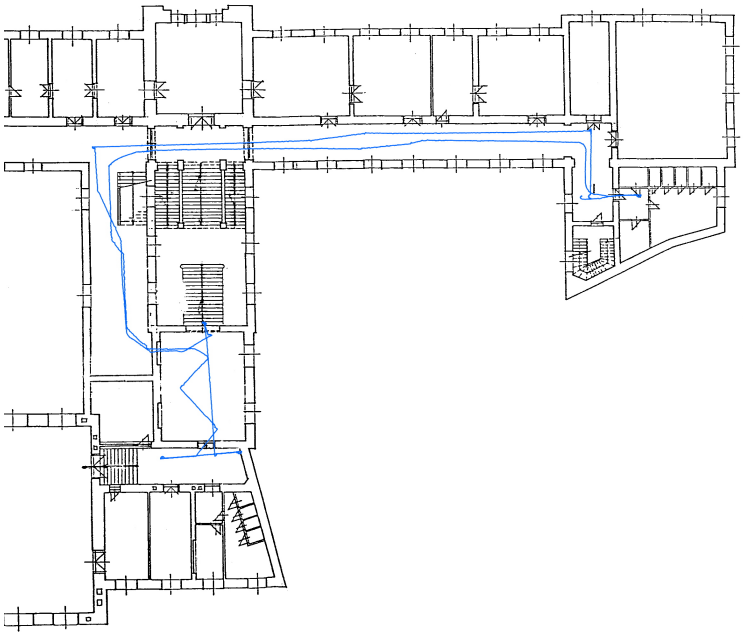
The range resolution is good enough to recognize the inset door B along the corridor. The door inset is 30 cm deep, only 1 range bin, but the sensor still picks up the change.

There are typical false reflections C in the map. Reflections D are more interesting. These are reflections coming from corners in front of false columns in the room facade. The false columns are only about 15 cm thick and yet return such distinguishable echos.

Echos E may look like false reflections, but are in fact detections through closed door. Echos around E_1 come from a narrow stairwell. The strongest echos behind the door come from a metal elevator shaft, a very strong target. Echo E_2 however comes from a common wall.



(a) Reconstructed map



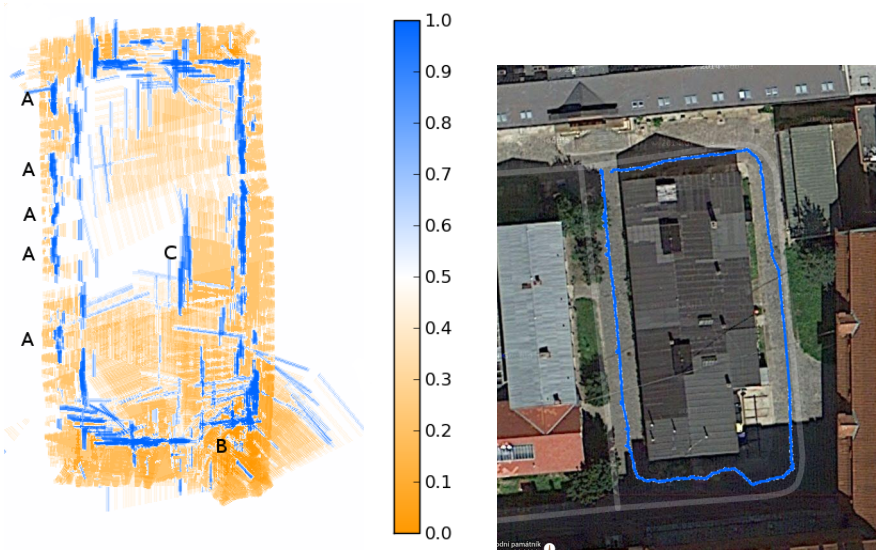
(b) Robot trajectory

Fig. 6.18. Experiment performed on the first floor of building E on Karlovo náměstí.

The long echo G that seems to block the entire hallway is caused by long-range returns from a wall in the hall to the right and from a metal-plated fridge on the left side of the hallway. There haven't been enough other scans to correct the map despite the range weighting.

6.4.4. The inner yard

An outdoor test was performed in the inner yard. The robot carrying the sensor followed the trajectory as presented in Figure 6.19b. The robot was driven around one of the buildings in the yard, following the walls counter-clockwise with the sensor assembly facing left. In this case it was necessary to use the Gmapping library to localize the robot based on the SICK PLS data.



(a) Reconstructed map **(b)** Robot trajectory

Fig. 6.19. Outdoor experiment conducted in the yard of the CTU complex on Karlovo náměstí

The reconstructed map is presented in Figure 6.19a. The building is clearly visible in the map. The irregularities A on the left wall are cars parked along the wall. The building has stairs in the lower right corner. The stair in the same height as the sensor is already indented into the building outline and produces the inner corner B.

The strong echo C inside the building is necessarily a false reflection. We however have no explanation for its source.

7. Conclusions

7.1. Sensor properties

The properties of the IGEP Lambda sensor, as tested in the experiments we have conducted, speak in favor of its usage for mobile robotics.

The sensor detects walls made of common materials like brick and porous concrete. Other materials like glass and wood are also detected, although as weaker targets. The walls are however detected only when scanned near perpendicularly. At bigger angles they are not detected at all.

The signal produces false reflections under certain conditions. Highly reflective surface can redirect the beam and an echo originating from the redirected beam can be registered. Multipath reflections also occur when scanning environment with multiple obstacles oriented at different angles. These effects cause errors in the reconstructed maps.

The range measurement precision of the sensor with the particular parameter settings used is approximately $\pm 19\,\mathrm{cm}$ or 1 range bin at ranges up to $5\,\mathrm{m}$. The accuracy of the sensor could be improved by employing different target detection techniques.

The angular resolution of the sensor is rather poor at beam width $26°$. The angular resolution might be improved by employing SAR processing techniques [8, sec. 14.1] that combine measurements taken from different places to improve angular resolution. We have however not tested the techniques.

7.2. Map reconstruction

We have designed an algorithm to reconstruct a map based on data from the IGEP Lambda module. The algorithm based on Bayes filter uses knowledge of the sensor properties to compensate for its shortcomings as the directionality.

The algorithm has reconstructed maps that represented the environment well to a human eye. False reflections and clutter were a minor issue and generally did not spoil the whole map. A bigger issue were the walls the radar failed to detect due to high viewing angle. A human with prior knowledge of the environment can guess the wall is continuous, a path planning robot cannot.

The virtual holes in the walls could prove fatal to a robot navigating the environment. It is necessary to scan a point from as many directions as possible before classifying it as empty. The best way to ensure as many scanning angles as possible is to continuously change the sensor orientation as in experiment 6.4.2.

7.3. Application possibilities

The sensor could prove useful in robotic applications. Its robustness would be desirable in outdoor applications and the ability to penetrate materials like wood and glass could be exploited even in indoor environments. The target detection and mapping algorithms would need to be improved for the radar to provide absolutely reliable data.

7.4. Future work

The key to improve the wall detection would be scanning the environment with the sensor instead of manipulating it only by movement of the whole robot. The algorithm would need to be further improved by implementing the techniques mentioned in 7.1. The improved target detection schemes and a SAR-like algorithm could both increase the map resolution and suppress the side effects of a scanning sensor.

The directionality of the sensor also suggests possibilities of building geometric maps from the data. The surface normal estimate inherent to the sensor could be exploited when constructing a geometric map.

If reasonable precision data can be obtained from the sensor, it might be possible to employ a SLAM algorithm to localize the robot. A localization method with the robustness radar offers would mean a huge advantage for robots operating in harsh environments with fog or dust.

Appendix A.

Maps reconstructed from the experiments

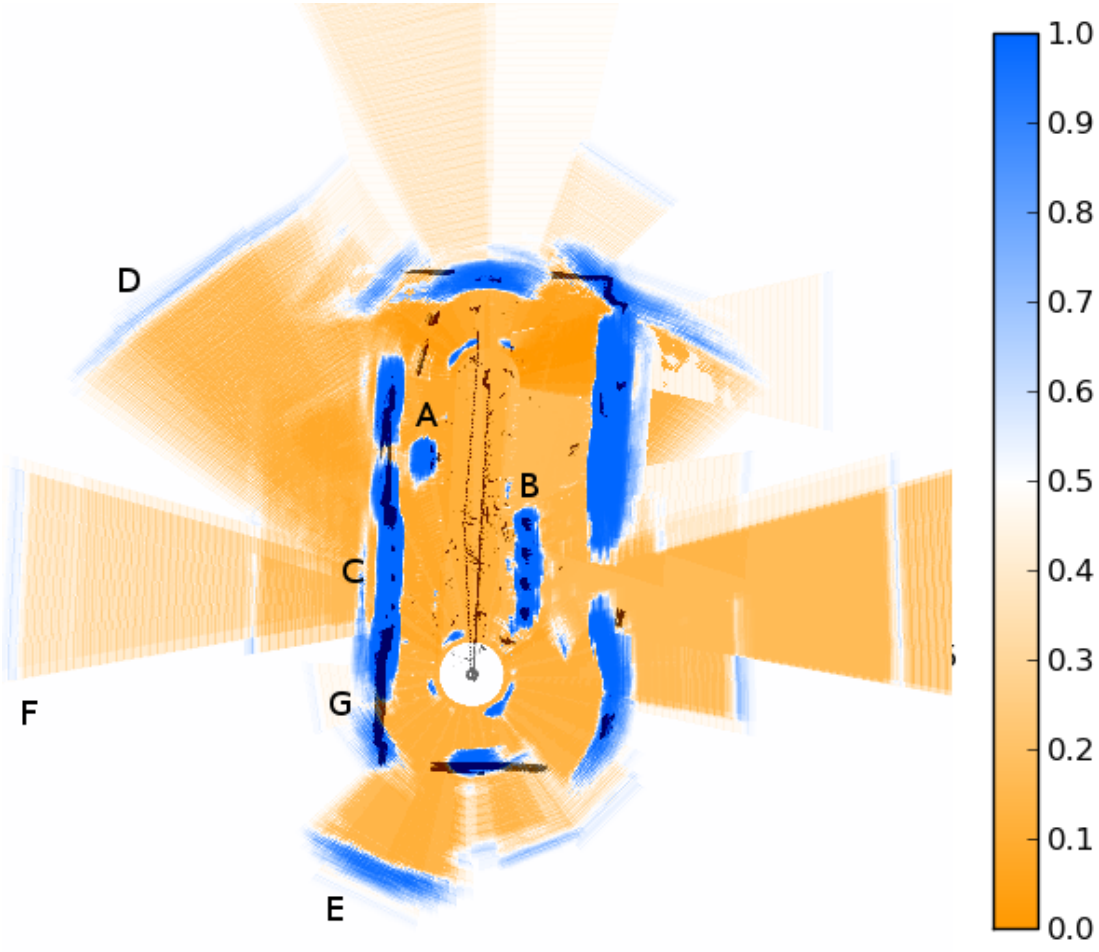


Fig. A.1. Reconstructed map from experiment 6.4.1

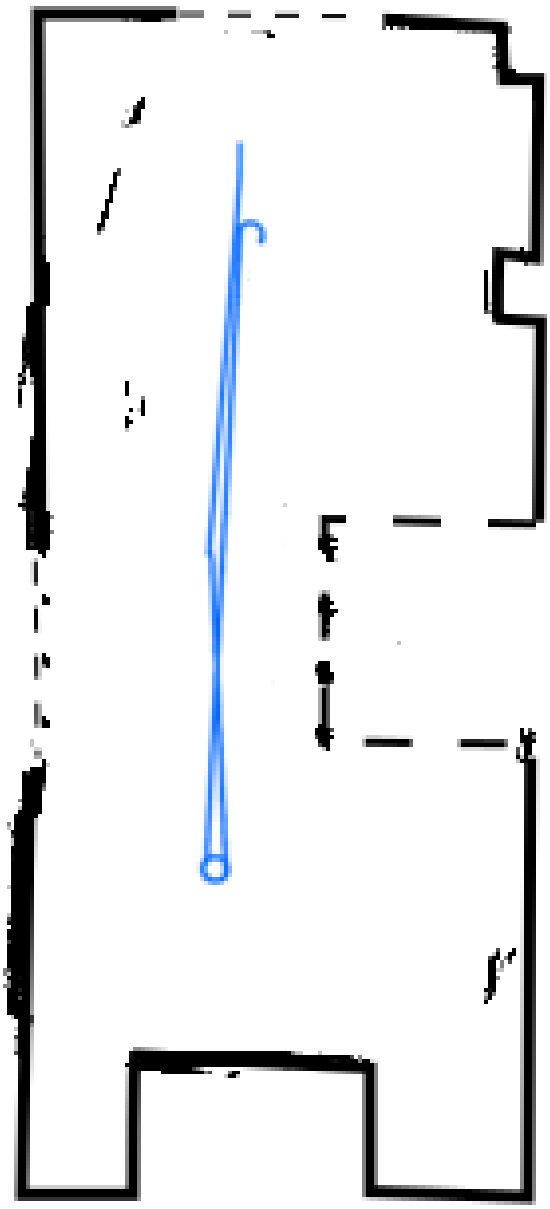


Fig. A.2. Robot trajectory from experiment 6.4.1

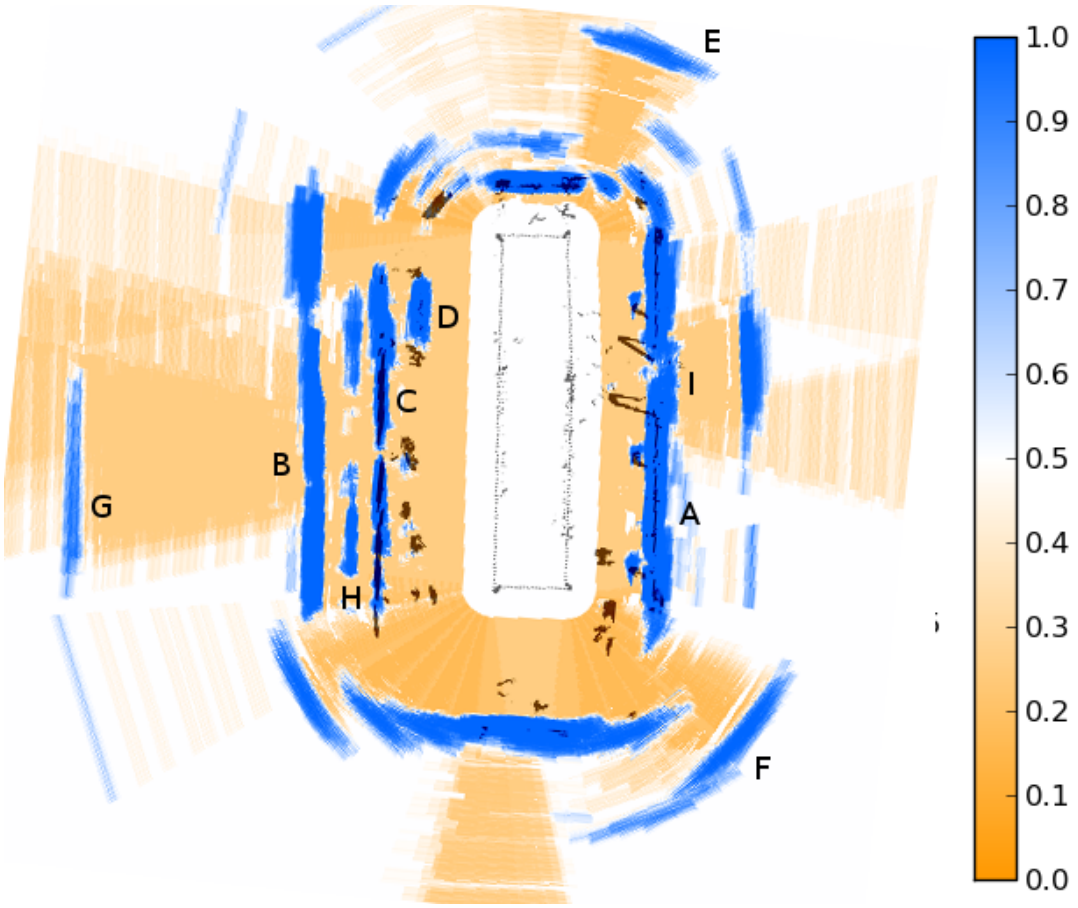


Fig. A.3. Reconstructed map from experiment 6.4.2

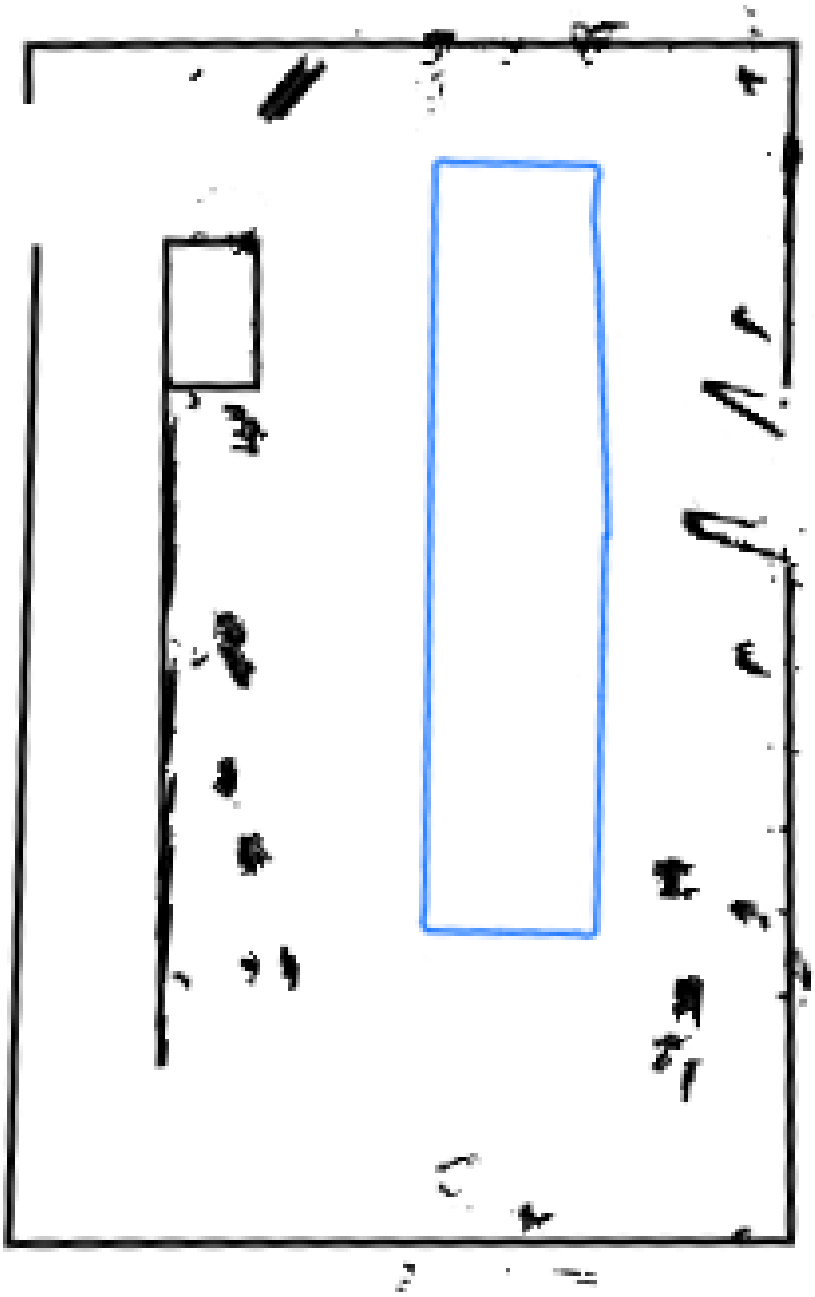


Fig. A.4. Robot trajectory from experiment 6.4.2

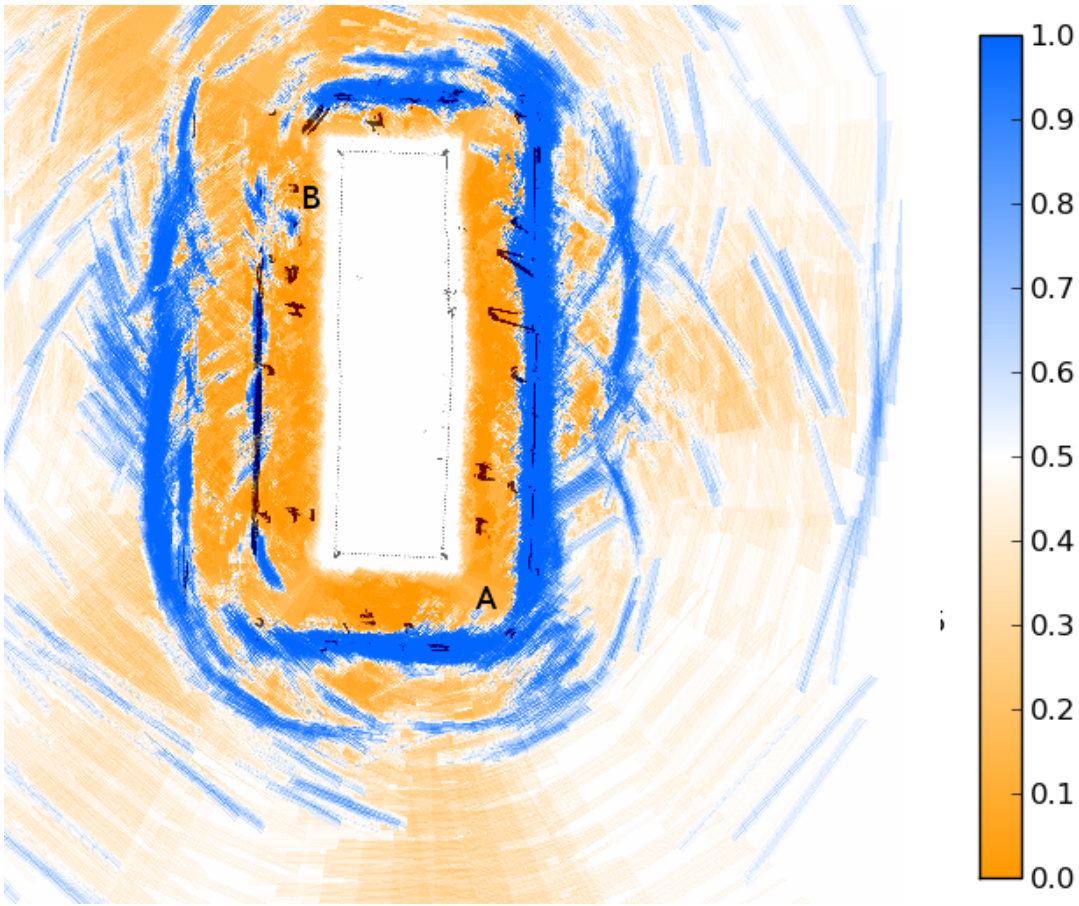


Fig. A.5. Reconstructed map from experiment 6.4.2

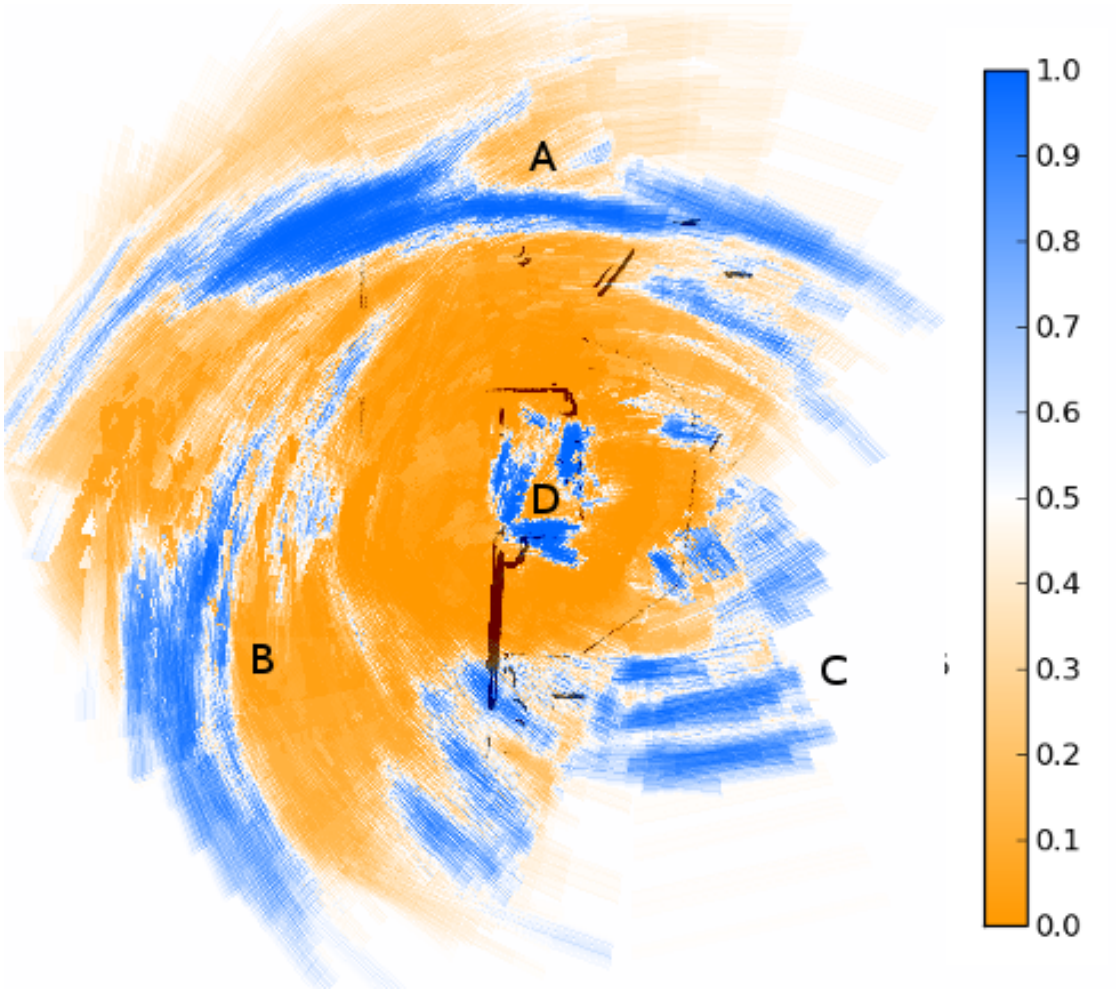


Fig. A.6. Reconstructed map from experiment 6.4.2

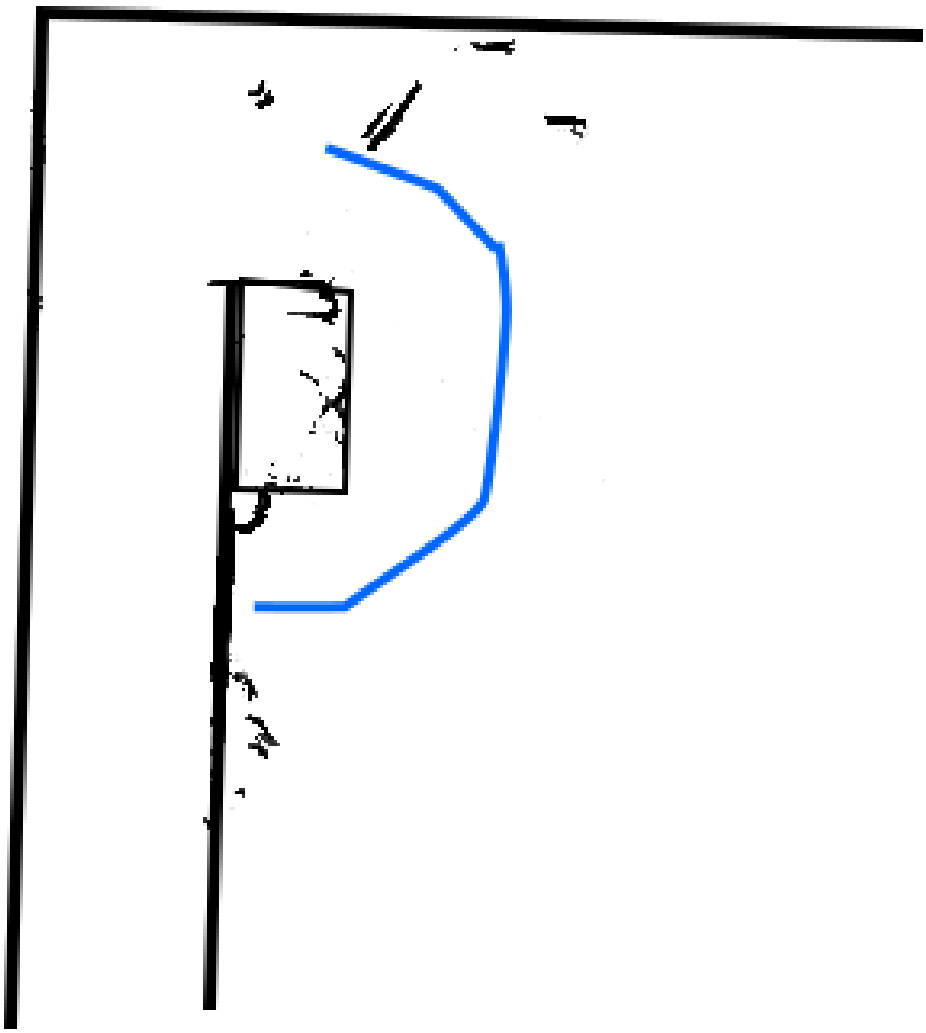


Fig. A.7. Robot trajectory from experiment 6.4.2

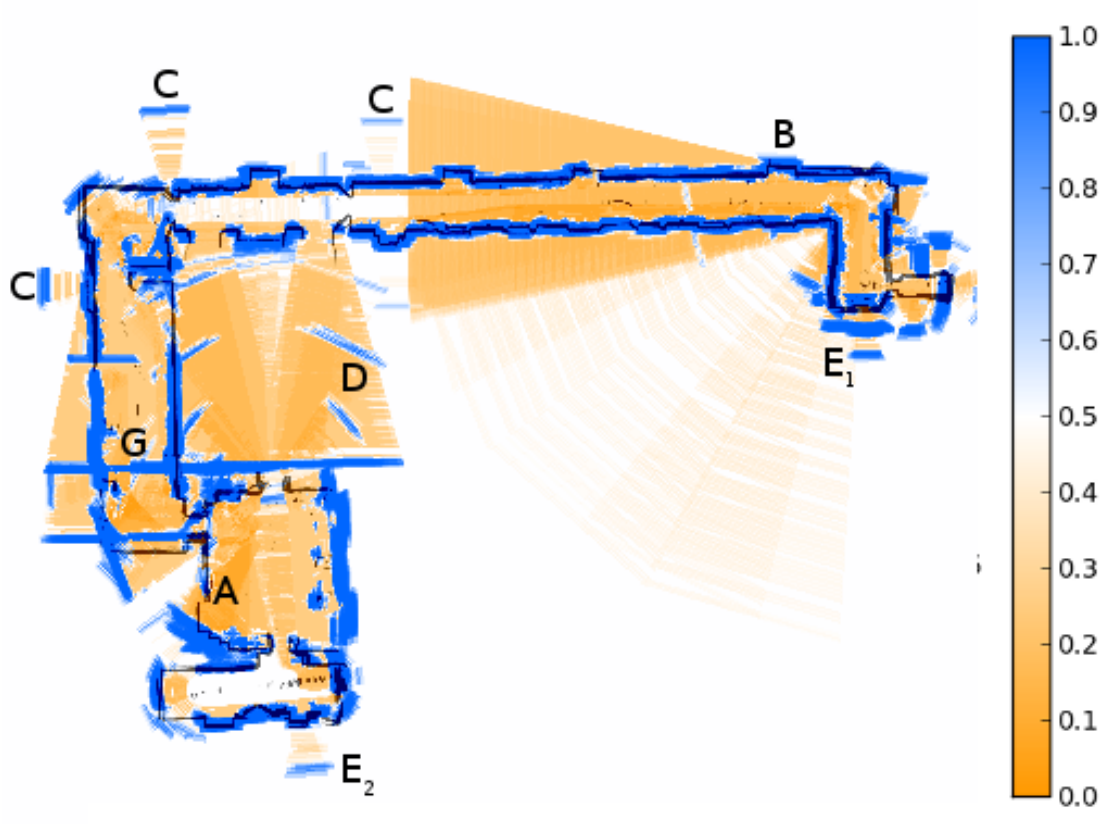


Fig. A.8. Reconstructed map from experiment 6.4.3

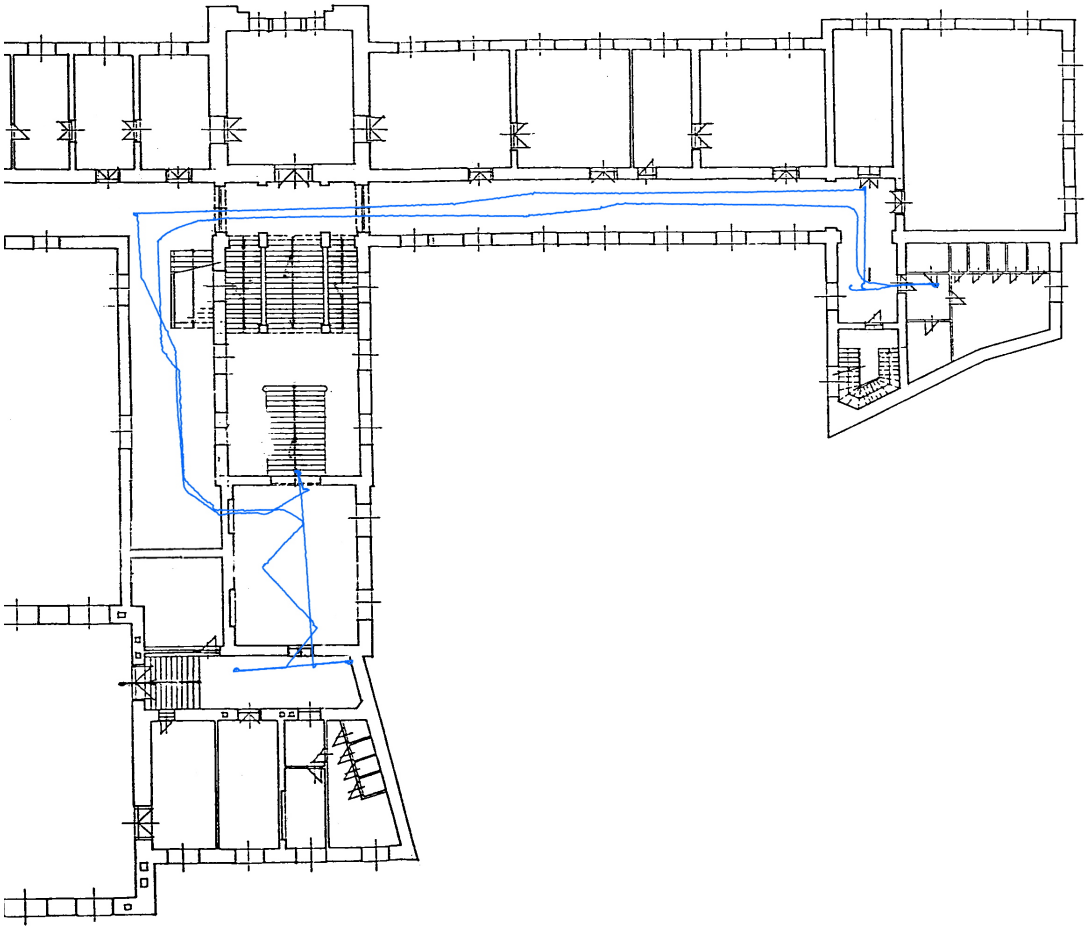


Fig. A.9. Robot trajectory from experiment 6.4.3

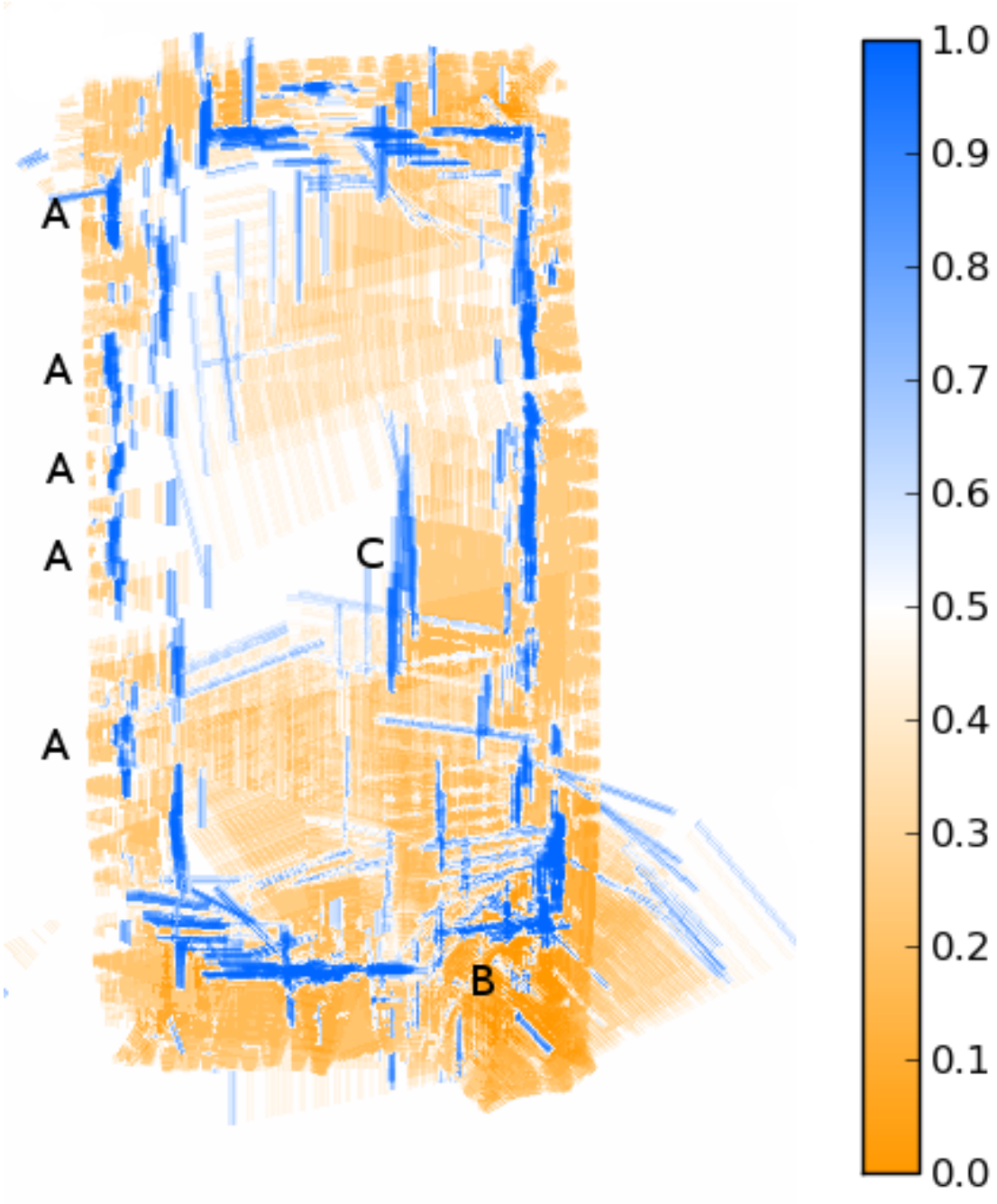


Fig. A.10. Reconstructed map from experiment 6.4.4

Fig. A.11. Robot trajectory from experiment 6.4.4

Appendix B.

CD contents

The CD contains the data collected and software created over the course of writing this thesis. It also contains a digital copy of this thesis.

data directory contains the raw radar and laser data logs used to reconstruct the maps. Raw map images are also included.

software directory contains the software created or modified. It contains the sensor driver (server), a client library and a GUI frontend, and a map reconstruction application.

documents directory contains a digital copy of this text.

The file listing follows.

```
/
├── data
├── software
├── documents
└── README
```

```
data
├── entrance.log
├── laser-entrance.log
├── map-entrance.png
├── atrium.log
├── laser-atrium.log
├── map-atrium.png
├── atrium_sweep.log
├── laser-atrium_sweep.log
├── map-atrium_sweep.png
├── generator.log
├── laser-generator.log
├── map-generator.png
├── first_floor.log
├── laser-first_floor.log
├── map-first_floor.png
├── yard.log
├── map-yard.png
└── README
```

```
software
├── README
├── radar_server
│   ├── radar_server.pro
│   ├── radar_server
│   ├── radar_dds.c
│   ├── status.h
│   ├── radar_config.c
│   ├── server.h
│   ├── main.c
│   ├── radar_fft.h
│   ├── radar.c
│   ├── radar_dds.h
│   ├── protocol.txt
│   ├── local.mk
│   ├── config.h
│   ├── server.c
│   ├── radar_server.pro.user
│   ├── radar_config.h
│   ├── radar_adc.h
│   ├── Makefile
│   ├── radar_adc.c
│   ├── radar_fft.c
│   ├── status.c
│   ├── radar.h
│   └── README
└── pyradar
    ├── viewer.py
    ├── mapper.py
    ├── radar.py
    ├── params.py
    ├── pantilt.py
    ├── client.py
    ├── laser.py
    ├── utils.py
    └── README
```

```
documents
└── thesis.pdf
```

Bibliography

[1] *Oxford Dictionaries*. URL: http://www.oxforddictionaries.com/.

[2] Ronald C. Arkin. *Behavior-based Robotics*. Bradford book. MIT Press, 1998.

[3] Alex Foessel. "Radar sensor model for three-dimensional map building". In: *Intelligent Systems and Smart Manufacturing*. International Society for Optics and Photonics. 2001, pp. 127–138.

[4] Alex Foessel and William L Adviser-Whittaker. "Scene modeling from motion-free radar sensing". In: (2002).

[5] Giulio Reina, Annalisa Milella, and James Underwood. "Self-learning classification of radar features for scene understanding". In: *Robotics and Autonomous Systems* 60.11 (2012), pp. 1377–1388.

[6] Sebastian Thrun et al. "Robotic mapping: A survey". In: *Exploring artificial intelligence in the new millennium* (2002), pp. 1–35.

[7] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[8] Merril I. Skolnik. *Introduction to Radar Systems*. 2nd ed. New York: McGraw Hill Book Co., 1980.

[9] Steven W. Smith. *Digital signal processing: a practical guide for engineers and scientists*. Newnes, 2003.

[10] Integration Software & Electronics Engineering. *IGEP RADAR LAMBDA Hardware Reference Manual (rev. 20130731)*. 2013. URL: https://www.isee.biz/support/downloads/item/igep-radar-lambda-hardware-reference-manual.

[11] Texas Instruments. *ADC121S101/ADC121S101–Q1 Single Channel, 0.5 to 1 Msps, 12-Bit A/D Converter datasheet*. URL: http://www.ti.com/lit/ds/symlink/adc121s101.pdf.

[12] A. E. Carr, L. G. Cuthbert, and A. D. Olver. "Digital signal processing for target detection FMCW radar". In: *IEE Proceedings F (Communications, Radar and Signal Processing)*. Vol. 128. 5. IET. 1981, pp. 331–336.

[13] Cyrill Grisetti Giorgio; Stachniss and Wolfram Burgard. *GMapping SLAM library*. URL: http://openslam.org/gmapping.html.