ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA E TELECOMUNICAZIONI PER L'ENERGIA

MILLIMETER WAVE RADAR FOR SLAM APPLICATIONS

Elaborato in
Sistemi di Telecomunicazione LM

Relatore
Chiar.mo Prof. Ing.
DAVIDE DARDARI

Correlatori
Prof. Ing.
GIANNI PASOLINI
Dott. Ing.
FRANCESCO GUIDI

Presentata da
ELIA FARNETI

Sessione III

Anno Accademico 2019-2020

Acronyms

FMCW Frequency Modulated Continuos Wave

SLAM Simultaneous Localization and Mapping

CFAR Constant False Alarm Rate

RCS Radar Cross Section

MAP Maximum a Posteriori

MMSE Minimum Mean Square Error

KF Kalman Filter

EKF Extended Kalman Filter

FOV Field of View

FFT Fast Fourier Transform

DFT Discrete Fourier Transform

IDFT Inverse Discrete Fourier Transform

ADC Analog-Digital Converter

AF Array Factor

Contents

Introduction

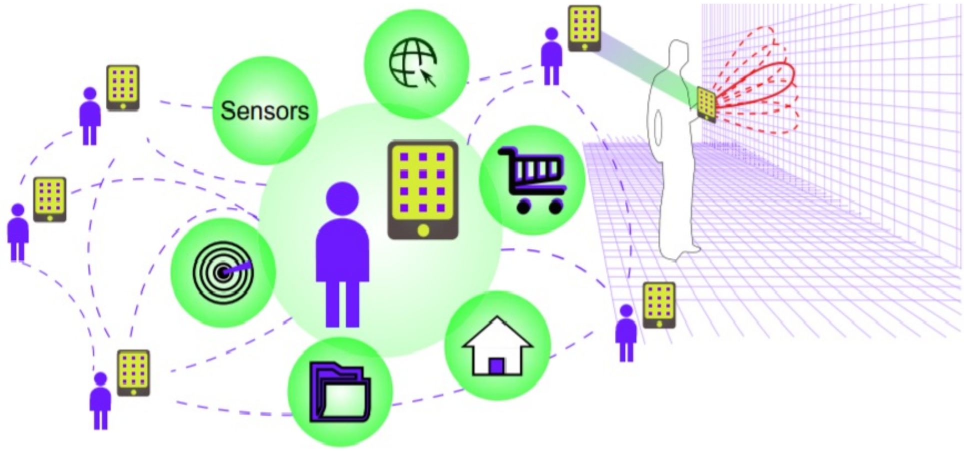

Figure 1: Example of a personal radar [1].

The purpose of this thesis is to introduce the millimeter wave radar technology applied to the mapping and localization tasks in order to investigate the feasibility of the recently proposed ”personal radar” concept.
More specifically, it has to be intended as a future feature of mobile devices for estimating the maps of the indoor enviroments while exploiting the technology already in place for wireless communications, allowing the possibility to realize infrastructure-free localization applications, such as guidance to visually impaired people, etc. [1].
The personal radar operates as follows. It automatically scans the surrounding enviroment and with the usage of radiating elements it collects the backscattered response at each scan direction. After that by analysing the collected data, the personal radar is able to infer the map of the environment. An example of personal radar application, is reported in figure 1.
Currently, the most adopted technologies are lidar (i.e., laser-based radar) or camera-based solutions but usually these technologies might be expensive, requiring mechanical steering and perfect visibility of the enviroment. For

these reasons it becomes interesting to explore the millimeter wave radar technology in order to overcome the aforementioned limitations. In fact, the millimeter wave radar technology offers the possibility to pack a large number of antennas into a small space and hence to realize narrow beam radiation diagrams at the expense of some performance degradation with respect to lidars.

For these reasons we decided to organize this thesis into the following chapters.

In the first chapter a brief introduction to the theory of radar is reported, with particular attention to the specific technology that is used in this thesis that is the FMCW radar. For this family of radars an analysis of the main parameters and how it works is reported. Finally a brief overview of the Constant False Alarm Rate (CFAR) theory is given.

Then a summary on the SLAM theory, is desribed in chapter 2, with some insights on its mathematical formulation and on examples of algorithms. In particular more emphasis is put on the theory of Graph-based SLAM, which will be exploited in the adopted Matlab algorithm.

After that an overview of the specific device that is used in this thesis, that is the Texas Instuments IWR1443, is given in chapter 3, with a particular attention on the data path chain and on how the device is able to determine the (x,y,z) coordinates of the objects detected in an unknown enviroment.

In chapter 4, some cleaning algorithms are investigated in order to perform the de-embedding of the radar's antenna effect from the backscattered measurements. At the end some numerical results that show merits and flaws of each algorithm are reported as well as a discussion about their adoption in practical applications.

Finally the numerical results obtained from dedicated experiments carried out in real environments are illustrated and discussed in chapter 5 which is then followed by conclusions and future investigations.

Chapter 1

RADAR

1.1 Fundamentals of Radar

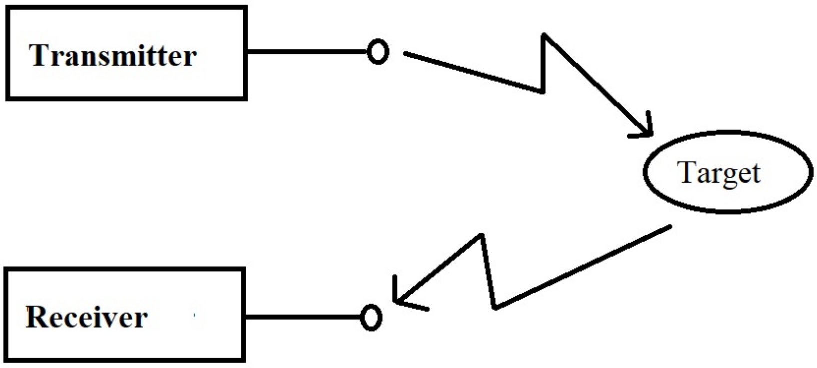

Figure 1.1: Radar schematic block

The acronym radar [2] means ”Radio Detection and Ranging” and identifies any system designed to detect the presence of an obstacle using radio technology and measure its distance. Figure 1.1 shows the basic block diagram of a generic radar in which the transmitter has the function to generate and irradiate with a suitable waveform.
The receveir has the function to reveal and elaborate the returning echo from an obstacle nearby the device. Apart from this, the device allows the estimation of many other parameters such as position and velocity.
The main idea is to transmit a continuos wave, in the form

$$x(t) = A\cos(2\pi ft) \tag{1.1}$$

and to receive a signal that can be written as

$$r(t) = B\cos(2\pi ft - \phi(t)). \tag{1.2}$$

It is demonstrated that working in this way it is hard to recognize an echo due to the fact that it might be overlapped to the transmitted signal that has a higher power than the echo and hence it is usually adopted to detect moving obstacles by exploiting the Doppler effect.
The previous descripted devices are part of the family of "Continuos Wave Radar".
To avoid the discussed problem it is possibile to transmit a modulated signal. One option is using impulse-based radar, where a very short pulse is transmitted in such a way the returning echo results to be not overlapped to the transmitted signal. By measuring the round-trip time it is possible to infer the distance of the target. Another option is to consider frequency modulated carriers, as it will be described in the next section, from which both the distance and the velocity of the target can be estimated.
In this sense, the radar equation helps at describing the behaviour of the radar technology, thanks to its dependence on the target distance, the device technology (i.e., antennas, etc) and on the propagation model of the considered enviroment. Consider the power density of the transmitted signal at distance R, that can be written as

$$S = \frac{P_T G_T}{4\pi R^2} \tag{1.3}$$

where P_T is the transmitted power and G_T is the transmitter antenna gain. The power that the obstacle backscatters to the radar depends on the Radar Cross Section (RCS) of it. The radar cross section, σ, gives an idea of how much an obstacle is detectable by a radar and it is measured in square meters. The received power density is given by

$$S_R = \frac{P_T G_T}{4\pi R^2}\frac{\sigma}{4\pi R^2}. \tag{1.4}$$

From the above relations it is possible to determine the received power at the radar:

$$P_R = \frac{P_T G_T}{(4\pi)^3 R^2}\frac{\sigma}{4\pi R^2} A_R \tag{1.5}$$

where $A_R = \frac{G_R \lambda^2}{4\pi}$ is the effective area of the receiving antenna, G_R is the receiver antenna gain and λ is the wavelength of the wave. Therefore, the received power can be written as:

$$P_R = \frac{P_T G_T G_R \lambda^2 \sigma}{(4\pi)^3 R^4} \tag{1.6}$$

It is important to notice that the received power depends on the inverse of fourth power of the distance and on the second power of the wavelength (this is one of the reasons that explains why the frequencies typically used are of the order of GHz).

1.2 FMCW Radar theory

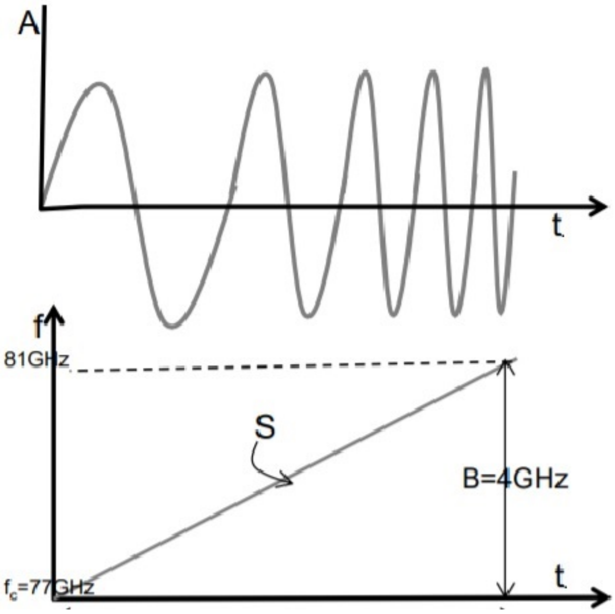

Figure 1.2: Example of a chirp [3].

The FMCW radar is very common in automotive and industrial applications and it usually measures range, velocity and angle of arrival of the obstacles in front of it [3].
The basic signal of this kind of radar is called "chirp", that represents a sinusoidal wave whose frequency is linearly increased within a time window of duration T_c. Figure 1.2-top shows an example of chirp, whereas Figure 1.2-bottom puts in relevance its slope S, which is a straight line. The slope

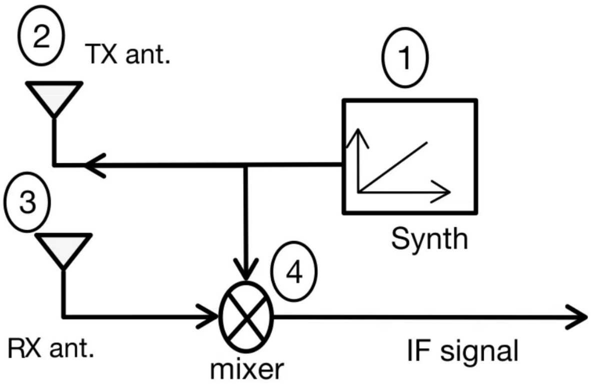

Figure 1.3: FMCW block diagram [3].

of the chirp S is an important parameter that defines the rate of increase of the frequency and in the following it will be used to determine the range resolution.

In figure 1.3 an example of a block diagram of a FMCW radar is reported. In particular, the system synthetizes a chirp and transmits it over the air, then an obstacole will reflect this chirp back to the system. After that there is a mixer that multiplies the transmitted chirp and the received chirp. This operation corresponds to the difference in the frequency domain thus generating a signal at intermediate frequency (IF). If in front of the radar there is only one obstacole then the IF signal will be only one. The discussed principle is reported in figure 1.4.

The main feature of this IF signal is that it has a constant frequency that depends on the slope of the chirp and on the round-trip-time τ of the reflected signal, where $\tau = \frac{2R}{c}$, being c the speed of light, and it is given by

$$IF = S\tau. \tag{1.7}$$

The higher is the number of obstacles in the enviroment, the higher is the number of IF signals generated along their respective frequencies.

As discussed before, the radar can estimate some different parameters about an obstacle, such as the range resolution [3], that is the capability to distinguish two closely spaced obstacles. A way to distinguish two obstacles could be the increase of the observation window, at the prize of a larger occupied bandwidth.

More precisely, let us focus on the relation between the resolution and the bandwidth, which is given by

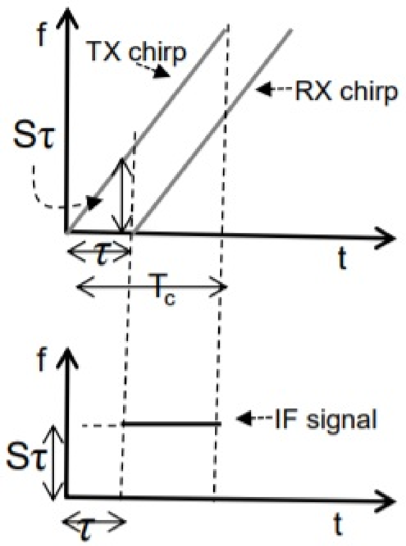

Figure 1.4: One obstacle detection [3].

$$\Delta f = \frac{S2\Delta d}{c}. \tag{1.8}$$

Due to the fact that $\Delta f = \frac{1}{T_c}$ and $B = ST_c$, equation (1.8) can be writtem as follows:

$$\Delta d > \frac{c}{2B}. \tag{1.9}$$

Equation (1.9) gives the range resolution as a function of the bandwidth so that increasing the bandwidth improves the range resolution.
Another parameter that can be of interest for some applications is the velocity, so that is important to understand what is the principle followed for the velocity estimation [4].
The idea is to transmit two or more chirps with a time separation of T_c and then to capture the returning chirps coming from the obstacle. The result on the receiver will be a sinusoidal tone at the same IF frequency but with different phase for each transmitted chirp. This phase difference can be used to estimate the velocity of a single obstacle.
The difference of the two phases, ω, corresponds to a precise spatial movement of the obstacle in the radial direction of vT_c:

$$\omega = \frac{4\pi v T_c}{\lambda}. \tag{1.10}$$

So from equation (1.10) it is possible to evaluate the velocity of the obstacle as

$$v = \frac{\lambda\omega}{4\pi T_c}. \tag{1.11}$$

Since the phase measurement is periodic of 2π, there could be an ambiguity in velocity estimation meaning that the radar is not capable to understand the obstacles's velocity, since the constraint $|\omega| < \pi$ is not respected. From the above explantion it is possible to set an upper bound to the velocity, indeed if the phase difference is equal to π then the maximum velocity that the radar can manage is:

$$v_{\max} = \frac{\lambda}{4T_c}. \tag{1.12}$$

It is clear that to increase the velocity bound it is necessary to have more closely chirps, i.e., T_c should be decreased.
The previous equations take into account only a single obstacle, so that is important to have just the idea of what the radar does if the obstacles are more than one.
The principle is to transmit a window of N chirps, that is called frame window, and the final result is very similar to an increase of the observation window. An efficient way to processs the signal is by performing a Fast Fourier Transform (FFT) along the various chirp signals thus obtaining the so-called Doppler FFT. Obstacles at different distances/velocities will appear as separated peaks in the Doppler FFT if the resolution is respected.
Also this kind of estimation has a resolution, that represents the minimum velocity differerence of two objects that the radar can sense and it is given by:

$$\Delta v = \frac{\lambda}{2NT_c}. \tag{1.13}$$

Since NT_c represents the frame time T_f, the final velocity resolution is:

$$\Delta v = \frac{\lambda}{2T_f}. \tag{1.14}$$

The last kind of information that this radar can manage is the estimation of the angle of arrival [5] of a returning signal, with the knowledge that this parameter can be estimated if multiple antennas are present (MIMO Radar) only.
First of all let us consider the case in which there is only a single obstacle in front of the radar. For this estimation at least two receiving antennas are

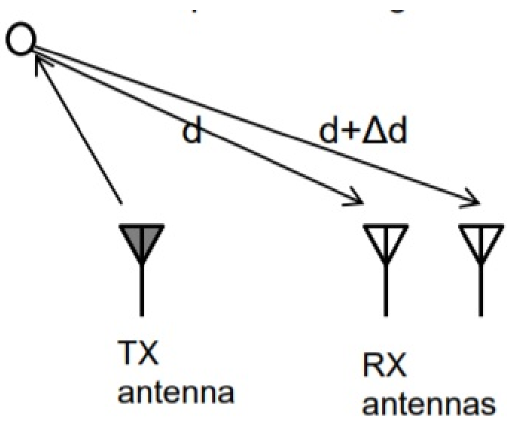

Figure 1.5: One obstacle detection [5].

required so that it is possible to determine the angle of arrival using the phase difference information of the receiving antennas. In few words the different distance of the obstacle from the receiving antennas results in a phase change ω_{a} which is given by:

$$\omega_{\mathrm{a}} = \frac{2\pi\Delta d}{\lambda} \tag{1.15}$$

where Δd is the difference between the distances of two consecutive transmitting elements and λ is the wavelength of the signal. The transmitting antenna transmits a frame of chirps and then at the receiving antennas the 2D-FFT, done accross the antennas, will show the peaks of the signal in the same location but with different phase. The measured phase difference ω can be used to estimate the angle of arrival. Due to that the phase difference will be:

$$\omega = \frac{2\pi d \sin\theta}{\lambda} \tag{1.16}$$

and from the above equation it is possible to obtain the angle:

$$\theta = \sin^{\text{-1}} \frac{\lambda\omega}{2\pi d}. \tag{1.17}$$

Unlike the previous estimates (range and velocity), now the relationship is non linear and so is the accuracy which is lower for large angles.
Also this time there is an upper bound on the estimation to avoid the ambiguity, which is given by:

$$\theta_{\max} = sin^{\text{-1}} \frac{\lambda}{2d}. \tag{1.18}$$

Notably, if more obstacles are present, antenna arrays are required to discriminate them in the angular domain. The higher is the resolution required, the higher is the number of needed antennas.
Regarding the angle resolution it is possible to define it as the minimum angle of separation for the obstacles to appear as separated peaks in the angular domain, and it is given by:

$$\theta_{\text{res}} = \frac{\lambda}{Nd\cos\theta}. \tag{1.19}$$

In chapter 3, more details will be provided on how to process the received signal usign FFT operations.

1.3 Constant False Alarm Rate (CFAR) theory

The device that is used in this thesis is based on what has been discussed in the previous sections and also on the CFAR theory.
This is a techinique with which a radar can decide whether the received signal is the result of a signal backscattered by an obstacle.
CFAR means "Constant False alarm Rate" and the purpose of this technique is to determine the power threshold above which any returning echo can be considered coming from a target. The value of this threshold determines the number of obstacles that can be detected, indeed if the threshold is low a larger number of obstacles can be detected, whereas, a high value of the threshold reduces the number of detectable obstacles. In CFAR-based radars, this value is choosen in order to achieve a specific false alarm rate. The false alarm ratecan represents a projects requirement so the application needs to satisfy a certain false alarm probability, so the number of false positive needs to be less than the rate choose, in order to maintain a good quality of its performances.
The value of the threshold depends on the receiver noise level as well as on the clutter generated by the environment. For example if the enviroment presents constant conditions then a constant threshold can be chosen. Conversely a variable threshold is needed. The adaptation of the threshold will mantain the false alarm rate constant.

Chapter 2

SLAM theory

2.1 Fundamentals of SLAM theory

After a brief introduction to radar theory it is now possible to introduce the next step of this Thesis.
One of the key methodologies necessary to enable the personal radar application is SLAM.
SLAM [6] stands for "Simultaneous Localisation And Mapping" and its purpose is the concurrent construction of a model of the enviroment (the map), and the estimation of the state of the robot or person moving within it. Before introducing the mathematical formulation of SLAM, we briefly describe the current mapping methods and for which scenario they are suitable.
In the following, two kind of SLAM algorithms are introduced:

- **Feature-Based:** These kind of algorithms are used every time the enviroment can be described by landmarks distinguishable from each other. They are often used with some accurate sensing system such as laser.

- **Grid-Based:** These kind of algortihms are used when there is the necessity to handle with some less accurate sensor data or in general, no well-defined landmarks are identifiable.

The most common formulation of a SLAM problem makes use of the probability theory and in particular formulates it as a maximum a-posteriori estimation problem [6].
Let X be the random vector that groups the trajectory of the device (as will be seen later the trajectory is described by a succession of poses) and the position of landmarks in the enviroment i.e., the state of the system. The

random process gives also a set of measurements $z_{1:T}$ in which a general form for the measurement at time instant k is:

$$z_k = h_k(x_k) + \epsilon_k \tag{2.1}$$

in which $h(x_k)$ represents the measurement model and ϵ_k is the random measurement noise.
The purpose of a SLAM algorithm is to estimate X, i.e., the position of the landmarks (i.e., the map) and the trajectory of the mobile node, starting from the collected mesurement Z. Within a Bayesian framework, this can be carried out by computing the Maximum a Posteriori (MAP) estimate, namely the belief, given by:

$$X^* = \underset{x}{\text{argmax}}[p(X|Z)] = \underset{x}{\text{argmax}}[p(Z|X)p(X)] \tag{2.2}$$

where $p(Z|X)$ represents the likelihood of the measurements Z starting from the assignement X and $p(X)$ is the prior probability which includes any prior information about the problem.
Assuming that the measurements in Z are indipendent, the formulation of the problem can be modified by factorizing the probability as follows

$$\begin{aligned} X^* &= \underset{x}{\text{argmax}}[p(X)\prod_{k=1}^{T} p(z_k|X)] \\ &= \underset{x}{\text{argmax}}[p(X)\prod_{k=1}^{T} p(z_k|x_k)]. \end{aligned} \tag{2.3}$$

Assuming now that ϵ_k is a zero-mean Gaussian noise with information matrix Ω_k (inverse of the covariance matrix) and assuming that the prior $p(X) \propto \exp^{-\frac{1}{2}||h_0(X)-z_0||^2}$ it is possible to rewrite equation (2.3) as:

$$\begin{aligned} X^* &= \underset{x}{\text{argmin}} - \log[(p(X)\prod_{k=1}^{T} p(z_k|x_k))] \\ &= \underset{x}{\text{argmin}} \sum_{k=0}^{T} ||h_k(x_k) - z_k||^2. \end{aligned} \tag{2.4}$$

Equations (2.3) and (2.4) refer to the same problem because maximizing the posterior probability is the same as minimizing the negative log posterior.

Due to the fact that equation (2.4) is not simple from the computational point of view because one should check all possible combinations of the sequence of states, To counteract this effect, other algorithms under the family of Bayesian filters have been proposed, that make use of the following Markovian[7] properties:

- The state at step k depends only on the state at step $k-1$, that is,

  $$p(x_k|x_{0:k-1}, z_{1:k-1}) = p(x_k|x_{k-1}) \tag{2.5}$$

  where $x_{0:k} = [x_0, x_1, ..., x_k]$ is the set of system states in which the positions of the landmarks are also considered in order to realize the SLAM application and $z_{1:k} = [z_1, z_2, ..., z_k]$ is the set of noisy measurements.

- A current measurement z_k is conditionally independent on the past measurements and states

  $$p(z_k|x_{0:k-1}, z_{1:k-1}) = p(z_k|x_k) \tag{2.6}$$

- In most practical cases, the measurements are also conditionally indipendent.

There are some kind of approaches that can be followed:

- **Filtering approach:**real time estimate of the future position of the device using the current measurement available.
- **Smoothing approach:** computation of the entire device trajectory from the availablle complete set of measurements.

These two approaches try to simplify the SLAM problem with a recursive solution as follows.
First of all the posteriori marginal at time step 0 is set to the prior then the algorithm computes those two following operations recursevely:

- **Prediction step**

  $$\int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\, dx_{k-1} \tag{2.7}$$

- **Update step**

  $$\frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{\int p(z_k|x_k)p(x_k|z_{1:k-1})\, dx_k}. \tag{2.8}$$

At this point it is possibe to compute the estimation following a Minimum Mean Square Error (MMSE) approach or a MAP approach:

$$x_k^{MMSE} = \int x_k p(x_k|z_{1:k}) \, d_{k_n} \tag{2.9}$$

$$x_k^{MAP} = \underset{x_k}{\operatorname{argmax}} \, p(x_k|z_{1:k-1}) \tag{2.10}$$

where if the a-posteriori distribution is Gaussian, equations (2.9) and (2.10) coincide.
To keep the complexity affordable, there is a number of available algorithms such as the Kalman Filter (KF) or Extended Kalman Filter (EKF) and others, exist.

2.2 Graph-Based SLAM

This family of SLAM algorithms [8] exploits the graph theory in order to build the map of the enviroment, representing the position of the device at different points in time with nodes and the constraints between the poses with edges. Such an information is obtained from observations of the enviroment. Those informations create a graph that is the grapichal representation of the evolution of the poses of the device.
Once a graph is constructed, the map can be computed by finding the spatial configuration of the nodes that is mostly consistent with the measurements modeled by the edges. This representation goes under "Graph Based" or "Network Based".
In Graph Based SLAM, the mobile device poses are described with some nodes in the graph that are labeled with their position in the enviroment. The information about the physical limits of the enviroment are carried by the edges between the nodes of the graph.
So it is clear that the SLAM problem is divided in two main tasks:

- **Graph construction:** Starting from the raw sensor data the algorithm constructs the graph that models the problem.
- **Graph optimization:** After the first step the algorithm optimizes the graph by computing the most likely configuration of the poses based on the edges of the graph.

The first part of this process is sensor dependent so it means that changing the features of the sensor could give different graph representations of the

same enviroment.
In the following a general formulation of the graph based problem is given[8].
Let $x = (x_1, ..., x_T)^T$ be a vector in which each element describes the pose of node i and let z_{ij} and Ω_{ij} be the real measurement and the information matrix of a measurement between the node i and the node j.
Let $\hat{z}_{ij}(x_{ij})$ be the prediction of a measurement based on a configuration of the nodes i and j. At this point it is possible to introduce $e(x_i, x_j, z_{ij})$ as the difference between the expected value of $\hat{z}_{ij}$ and the actual measurement z_{ij} provided by the robot. Just to simplify the notation the indices i and j are connected to the error function:

$$e_{\mathrm{ij}}(x_i, x_j) = z_{ij} - \hat{z}_{ij} \tag{2.11}$$

The purpose of this approach is to find the configuration of the nodes x^* that minimizes the likelihood of all observations:

$$F(x) = \sum_{<i,j> \in C} {e_{ij}}^T \Omega_{ij} e_{ij} \tag{2.12}$$

where C is the set of pairs of indices for which a observation z exists.
Finally, the problem becomes the evaluation of the following equation:

$$x^* = \underset{x}{\operatorname{argmin}} F(x) \tag{2.13}$$

that can be solved in many different ways [8].

2.3 Matlab SLAM algorithm

The algorithm that is used in this project comes from Matlab libraries [9] and it is an algorithm built for lidar's applications. In the following a short explanation of what the algorithm does regarding the graph-based theory is proposed.
To build the map of the enviroment, the SLAM algorithm processes the output of the scan operation i.e., the set of measurements related to an angle scan of the lidar, and builds a pose-graph that links the scan.
First of all it is necessary to import the data coming from the lidar, after some processing, to the matlab enviroment.
After that, the algorithm creates an object, called lidarSLAM, used as input. Inside this object it is possible to set some parameters of the object such as the map resolution and the maximum range that can be considered. It is also possible to add an extra parameter that regards the maximum number of scans that the algorithm can manage.

At this point the algorithm inserts scans until the maximum number of scans specified is reached.
An important task that this algorithm has to resolve is the so called *loop closure* which has the purpose to recognize if a pose is already present and then it tries to update it. The purpose is to iteratively improve the scan to give a more precise map of the enviroment.
The last task is to contruct the occupancy map: in few words the algorithm reads the produced data and the associated poses and builds a grid [x,y,θ], where x and y represents the position and θ is the orientation of the scan, and use it to build the occupancy map.
The matlab code of this algorithm is reported in the appendix A of this thesis.

Chapter 3

mmWave Radar Equipment

3.1 The Teaxs Instruments radar IWR1443

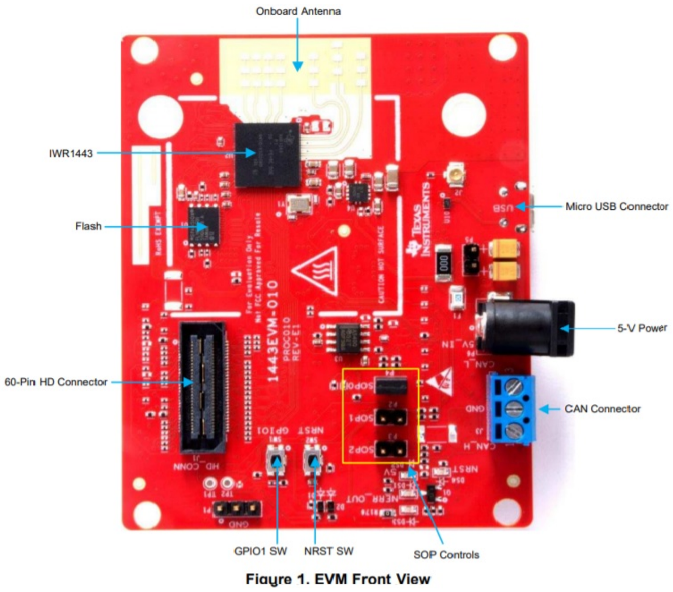

Figure 3.1: IWR1443

The IWR1443[10] [11] device is a millimeter wave sensor based on FMCW technology that works in a frequency range of 77-81 GHz with 4 GHz of bandwidth.
The device is equipped with 4 receiving antennas and 3 transmitting antennas that allows to realize a virtual array with 1 TX and 12 antennas, respectively. According to the considered arrays, 3D beamforming operations can be performed. The device has a DSP mounted on the same PCB

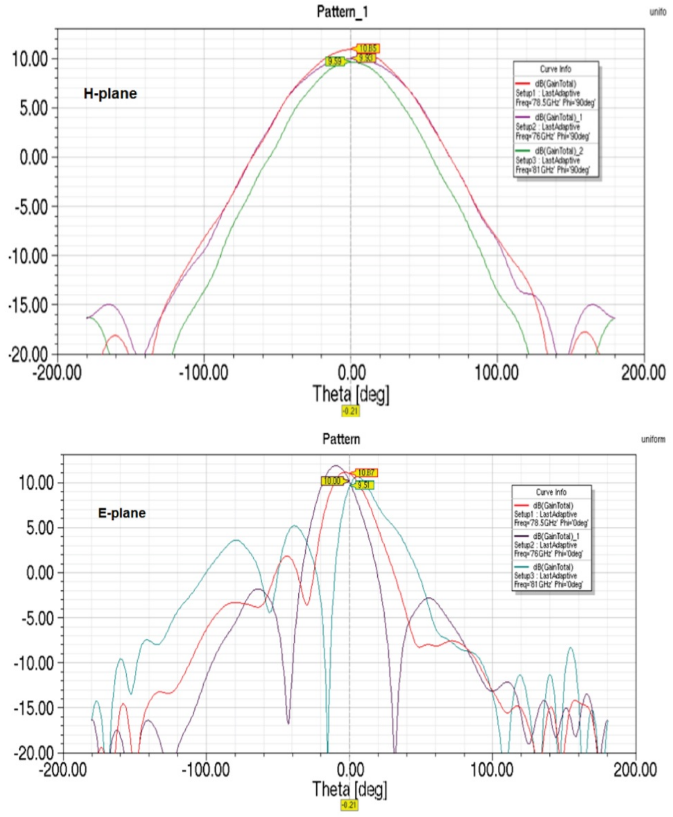

Figure 3.2: E-plane and H-plane of the antenna pattern [10].

that allows a first raw elaboration of the data in order to obtain the main information.
The main features of this device are:

- **Azimuth range:** a single radiating element of the azimuth plane of the device presents a beamwidth of 110 degrees with a resolution of 15 degrees on the azimuth plane.
- **Elevation range:** Due to the fact that the device works with 3D coordinates it is useful to consider also the elevation range, that for a single antenna of the elevation plane of this device is in the order of 30 degrees with a resolution of 57 degrees.

Just for information in figure 3.2 is shown the antenna radiating diagram both for the azimuth and for the elevation plane and it is possible to recognize the beamwidth reached from the single antenna. These two angles form the so called Field of View (FOV)of the radar, and represents the area where the

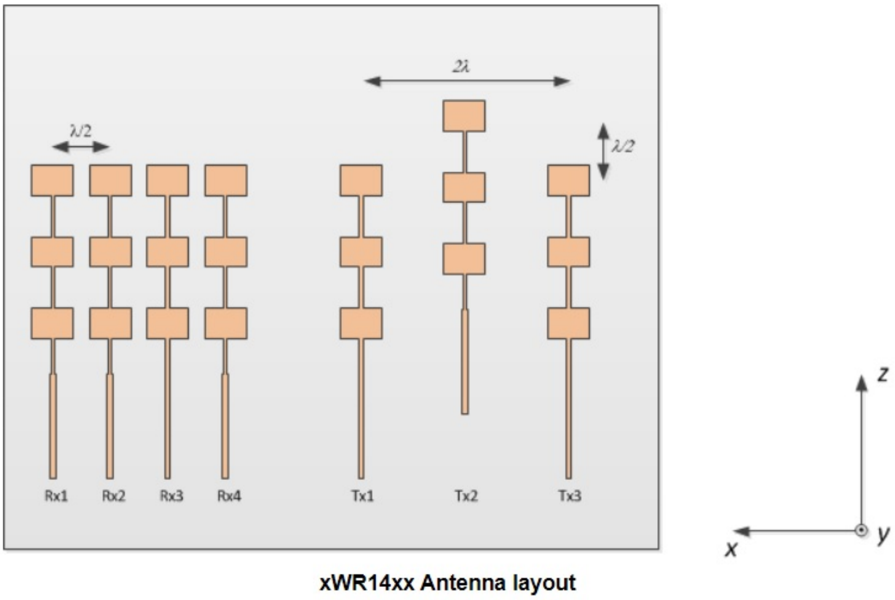

Figure 3.3: Antennas configuration [10].

radar is capable to detect objects.
To understand the angle and coordinates estimation it is useful to focus on the configuration of the antenna.
For this device it is possible to choose between the 2 and 3 transmitters configuration, but in the following just for simplicity only the 3 transmitters configuration is reported.

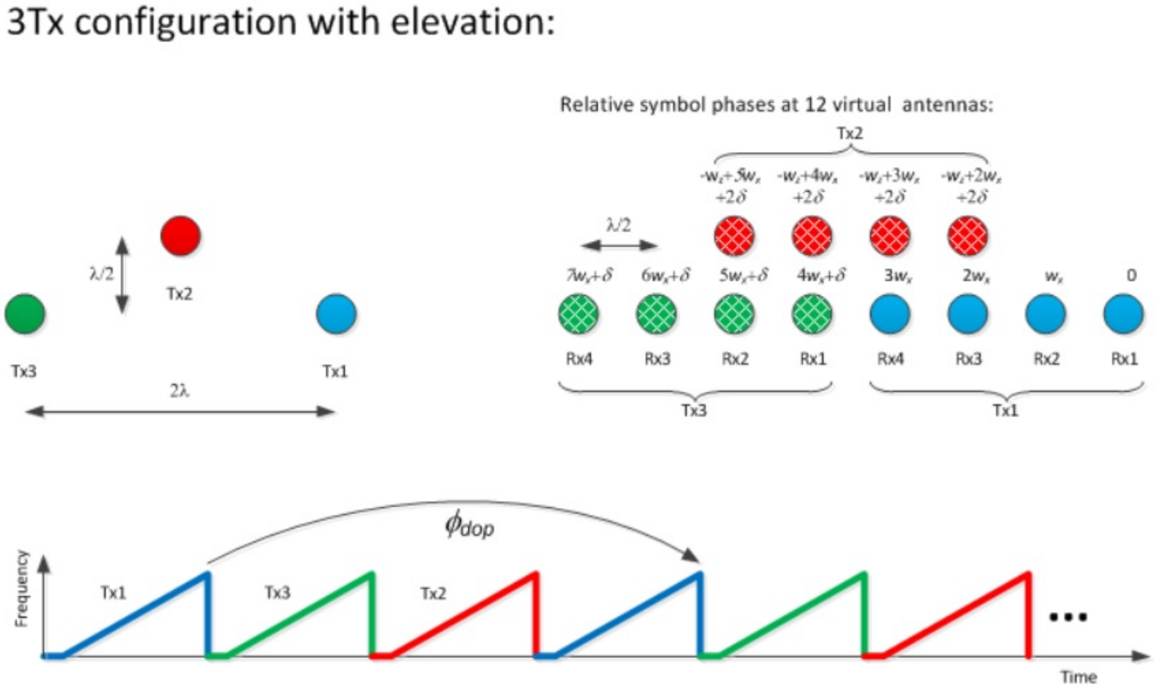

Figure 3.4: Antennas working principle [11]

In the previous figure ω_x represents the phase difference between consecutive receiver azimuth antennas , ω_z is the phase difference between azimuth and the corresponding elevation antenna above the azimuth antenna and δ represents the contribution of the Doppler effect.

From figure 3.3 and figure 3.4 it is possible to see that the receiving antennas are placed at a distance of $\frac{\lambda}{2}$, where λ is the wavelength of the signal, from each other and are placed in a horizontal plane while the transmitting antennas are placed at a distance of λ but they are not on the same plane. In particular transmitter number two is shifted up of a quantity of $\frac{\lambda}{2}$. This allows the device to give the user also 3D information.

The 3 TX antennas are activated in sequence at each chirp trasmission as shown in figure 3.4. Indeed, from the received symbols on the virtual array it is evidenced that the receiver indexed with number 1 on the virtual array is the reference one, and the relative phase shifts between antennas can be extracted.

Regarding the Detection and Localization chain it is important to pay attention on the data path chain and how the device works and gives to the user the coordinates of the obstacles in the enviroment.

As it is shown in figure 3.5 the data processing is divided in some phases:

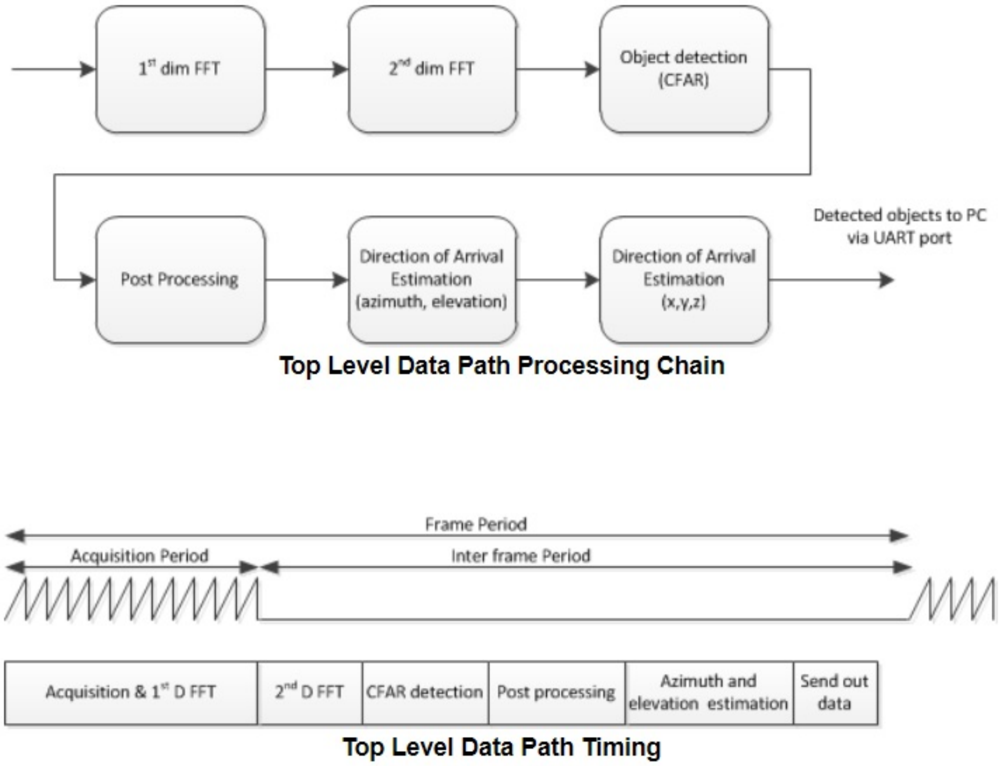
1st dim FFT
2nd dim FFT
Object detection (CFAR)
Post Processing
Direction of Arrival Estimation (azimuth, elevation)
Direction of Arrival Estimation (x,y,z)
Detected objects to PC via UART port
Top Level Data Path Processing Chain
Frame Period
Acquisition Period
Inter frame Period
Acquisition & 1st D FFT
2nd D FFT
CFAR detection
Post processing
Azimuth and elevation estimation
Send out data
Top Level Data Path Timing

Figure 3.5: Data path processing chain [11].

- **1st Dimension FFT Processing:** This first FFT block elaborates range data from raw data coming from the Analog-Digital Converter (ADC) after a Blackman windowing operation. Every time the ADC has a data stream it sends it to the datapath that processes it.

- **2nd Dimension FFT Processing:** Successively, another FFT operation is performed, but this time in two dimensions with the goal to get the Doppler informations. It is a trade-off between precision and memory storage.

- **CFAR Detection:** At this stage, a CFAR-based procedure is performed, and its output is the number of detected obstacles. Indeed the higher is the threshold the lower is the number of detected obstacles and vice-versa. The advantage of this technique is that the threshold value is set, and its value can be variable in space and/or in time, in such a way that the false alarm remains constant.

- **Post processing:** After CFAR processing it is possibile to do peak

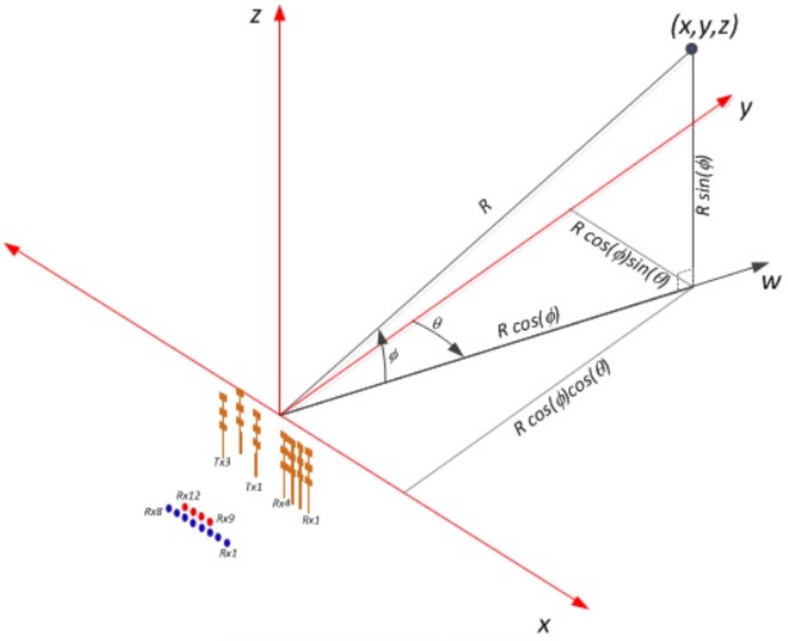

Figure A: Coordinate Geometry

Figure 3.6: Coordinates geometry [11].

grouping and Doppler phase shift compensation. For example the detected obstacles that are outside of the range specified by the configuration are discarded.

- **Direction of Arrival Estimation:**

  This is the crucial phase of data processing because here the goal is to estimate the position (i.e., the (x, y, z) coordinates) of each detected obstacle.
  From figure 3.6, ϕ represents the elevation angle while θ represents the azimuth angle of an obstacle.
  Considering a single obstacle, the signal received by the 8 azimuth receiving antennas (there are 4 receiving antennas and 2 azimuth transmitting antennas so we have 8 virtual antennas) is:

$$\mathbf{r} = A_1 e^{j\psi} \cdot \left[1 e^{j\omega_x} e^{j2\omega_x} e^{j3\omega_x} e^{j4\omega_x} e^{j5\omega_x} e^{j6\omega_x} e^{j7\omega_x} \right] \tag{3.1}$$

  where A_1 and ψ represent the amplitude and the initial phase at first antenna, respectively and ω_x is the phase shift of the signal between two consecutive antennas due to the angular position of the obstacle (ϕ,θ). To be more clear ω_x, $\omega_x = \pi \sin(\theta)\cos(\phi)$, represents the phase difference between consecutive receiver azimuth antennas of the 2D FFT.

After a FFT of vector (3.1) a peak P_1, present at $\omega = \omega_x$ with phase ψ, is given by:

$$P_1 = A_1 \cdot e^{j\psi}. \tag{3.2}$$

Defining K_{MAX} as the index associated to the FFT peak, the phase ω_x will be:

$$\omega_x = \frac{2\pi}{N} k_{MAX}. \tag{3.3}$$

Considering again a single obstacle the signal received by the elevation antennas (there are 4 receiving antennas and 1 elevation transmitting antenna so we have 4 virtual antennas) is:

$$\mathbf{r} = A_2 e^{j(\psi + 2\omega_x - \omega_z)} \cdot \left[1 e^{j\omega_x} e^{j2\omega_x} e^{j3\omega_x}\right] \tag{3.4}$$

Similar to what was done previously, by computing a FFT of the signal (3.4) will get a peak P_2 at ω_x, as it follows:

$$P_2 = A_2 \cdot e^{j(\psi + 2\omega_x - \omega_z)} \tag{3.5}$$

Now, by multiplying P_1 and P_2 we get:

$$P_1 \cdot P_2^* = A_1 \cdot A_2 \cdot e^{j(\psi - (\psi + 2\omega_x - \omega_z))} \tag{3.6}$$

From the previous equation ω_z can be computed as follows:

$$\omega_z = \angle(P_1 \cdot P_2^* \cdot e^{j2\omega_x}) \tag{3.7}$$

where ω_z represents the phase difference between azimuth and the corresponding elevation antenna above the azimuth antenna.

It is now possible to estimate the range, i.e., the distance of the obstacle from the radar as it follows:

$$R = k_r \frac{c \cdot F_{SAMP}}{2 \cdot S \cdot N_{FFT}}. \tag{3.8}$$

In the previous equation c stands for speed of light, k_r is the range index , F_{SAMP} is the sampling frequency, S is the chirp slope and N_{FFT} stands for the FFT order. From the previous equation it is finally possible to determine the (x,y,z) coordinates of the deteceted obstacles as it follows:

$$x = R \cos\phi \sin\theta = R\frac{\omega_x}{\pi} \tag{3.9}$$

$$z = R \sin\phi = R\frac{\omega_z}{\pi} \tag{3.10}$$

$$y = \sqrt{R^2 - x^2 - z^2}. \tag{3.11}$$

- **Information sent to the Host:** Finally the packets containing the data from the radar are built and sent to the host by UART protocol. In figure 3.7 data structure sent to the host is shown. It is composed of a Header which groups:
  - **Platform:** Indicates SDK version over which the device is working.
  - **Frame Number:** Indicates the number of frames that are collected during the acquisition phase.
  - **Number of Detected Obstacles**
  - **Number of TLVs: Indicates the number of the groups of data.**

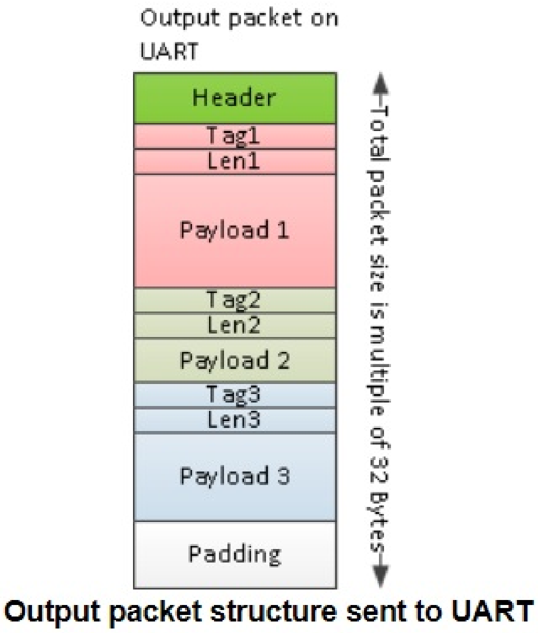

Figure 3.7: Packet format [11].

After the header there is the TLV field which discriminates the type of data that will carry the payload. For example:

- **Tag 1:** contains obstacle information, i.e., the position, the Doppler in case of movement etc..
- **Tag 2:** This tag contains the range profile information. The range profile is a graph which relates the range with the relative power, in dB, of detected obstacles. It is the graphical result of the detection processing from which the user can see the peaks caused by the obstacles in the received signal. It is an information that does not consider the Doppler as it refers only to zero doppler case.
- **Tag 3:** This tag keeps a similar information as tag 2, but this time it refers to the noise profile of the enviroment.
- **Tag 4:** This tag refers to the static heatmap information, so it keeps a matrix organized with antennas by row and range by column and in which in every single cell there is an energy information of the enviroment. Notably, there are two matrices of this kind, one for each plane.

- **Tag 5:** This tag refers to the range doppler heatmap. The concept is the same as before, with the construction of a matrix with the range information on the row, but this time columns contain the Doppler information.
- **Tag 6:** This tag contains only some statistic information about the device such as the DSP usage.

After each payload, a padding block used to adjust the entire lenght of the packet to multiple of 32 bytes is included.

3.2 mmWave Demo Visualizer

The presented device has the possibility to be used through web-based interface provided by Texas Instruments that is called mmWave Demo visualizer[12], with which it is possible to configure and change radar parameters in runtime and then visualize the data provided by the radar.
The interesting thing is that the user can select the particular device with the appropriate version of the SDK and also he/she can select the number of transmitting antennas depending on the feature of the devices.
During the work it is also possible to modify in real time some parameters and see how they affect the final scan of the surrounding of the device. For example it is possible to modify the CFAR threshold or enable/disable the range grouping or the doppler grouping or it is also possible to enable/disable the static clutter removal algorithm.
Another important feature of this visualizer is that it is possible to record all the data that the radar sends to the host which can be successively processed by other software such as Matlab. So for this reason the mmWave Demo visualizer played an important role during this thesis.
At last this visualizer is used as a comparator to make sure that the raw data are elaborated in the right way by the Matlab software we developed in this thesis that is described in the next chapter.

Chapter 4

Measurement results

In this chapter we first show the design of ad-hoc algorithms to detect and localize extended objects, as the previously described approach allows only to retrieve the peaks of the signals backscattered by targets. Then, we report the conducted measurement campaign and the attained results.

4.1 Extraction of angular information

In the following, we describe how the raw data coming from the radar have been processed in order to obtain the angle-range matrix necessary to apply a custom object detection algorithm. The raw data can be extracted from the TLV tag 4.

Inside this tag there is the energy matrix that includes the I/Q samples of the signal at different time bin and for each of the 12 virtual antennas. In the following this energy matrix will be identified as a_{rn} where $r = 1, 2, ..N_{FFT}$ refers to the indices of the ranges, with NFFT being the order of FFT made, and n is column index spanning from 1 to 12. According to (3.8), from r it is possible to extrapolate the range information.

As it is shown in figure 3.4 each antenna transmits a chirp separately and then the four receivers get the relative symbols.

From figure 3.4, ω_x represents the phase shift between the horizontal antennas, ω_z is the phase shift between the elevation antennas and δ is defined as the doppler effect that for the moment will not be considered.

First of all the azimuth angle, θ, is divided into M portions, where M is the number of angle positions. A high value of this parameter means a very smooth representation of the energy matrix.

The elevation angle is, for the moment, set in each simulation to zero but it is also possible to create some matrix with different elevation information.

The main part of this process is represented by equation (4.1) that reprents a Discrete Fourier Transform (DFT) of the received signal.

$$
\begin{aligned}
y_{\mathrm{rm}} = & \sum_{n=0}^{3} a_{\mathrm{rn}} \cdot e^{-j\pi n sin\theta_m cos\phi} \\
& + \sum_{n=4}^{7} a_{\mathrm{rn}} \cdot e^{-j\pi(n-2) sin\theta_m cos\phi - j\pi sin\phi} \\
& + \sum_{n=8}^{11} a_{\mathrm{rn}} \cdot e^{-j\pi(n-4) sin\theta_m cos\phi}
\end{aligned} \tag{4.1}
$$

with $r = 1, 2, ...N_{FFT}$, $m = 1, 2..M$ and $\theta_m = [-\frac{\pi}{2}, \frac{\pi}{2}]$ is the sweep of the azimuth angle.

This DFT permits to obtain the angular information of the incoming signal components starting from the signal received by each virtual antenna. In few words the process executes a kind of rephasing in order to discover if in a precise angle of arrival it could be present an obstacle. The final result is a range-angle map in which obstacles determine the presence of peaks in the corresponding angle-range in the azimuth plane.

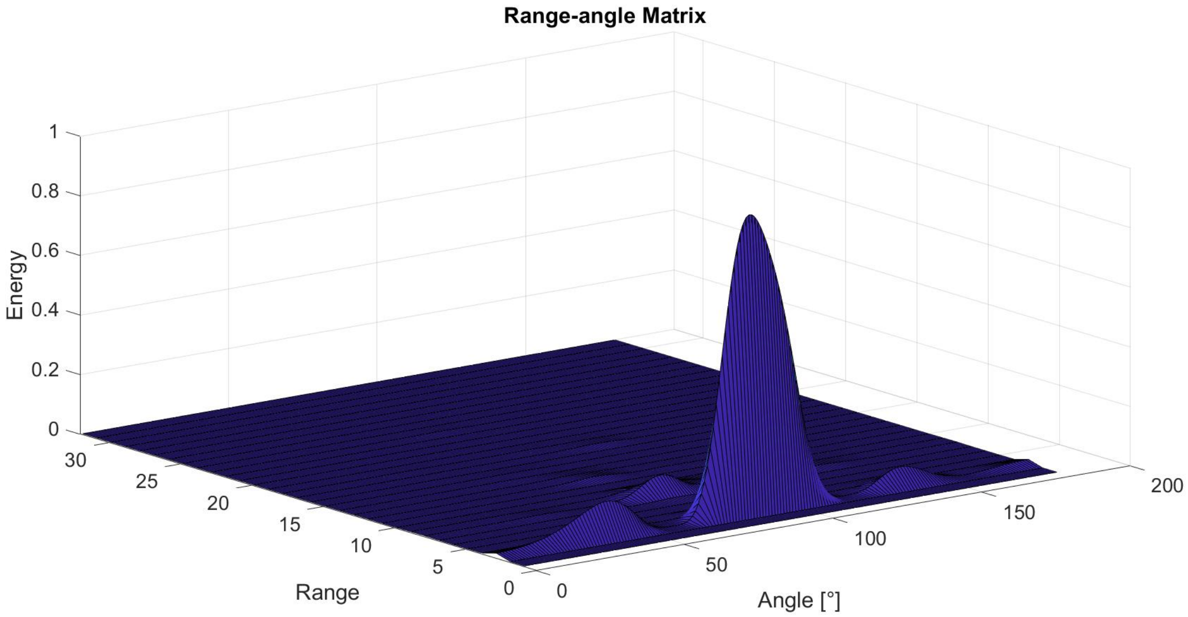

Figure 4.1: Range-angle map in the presence of an obstacle close to the radar.

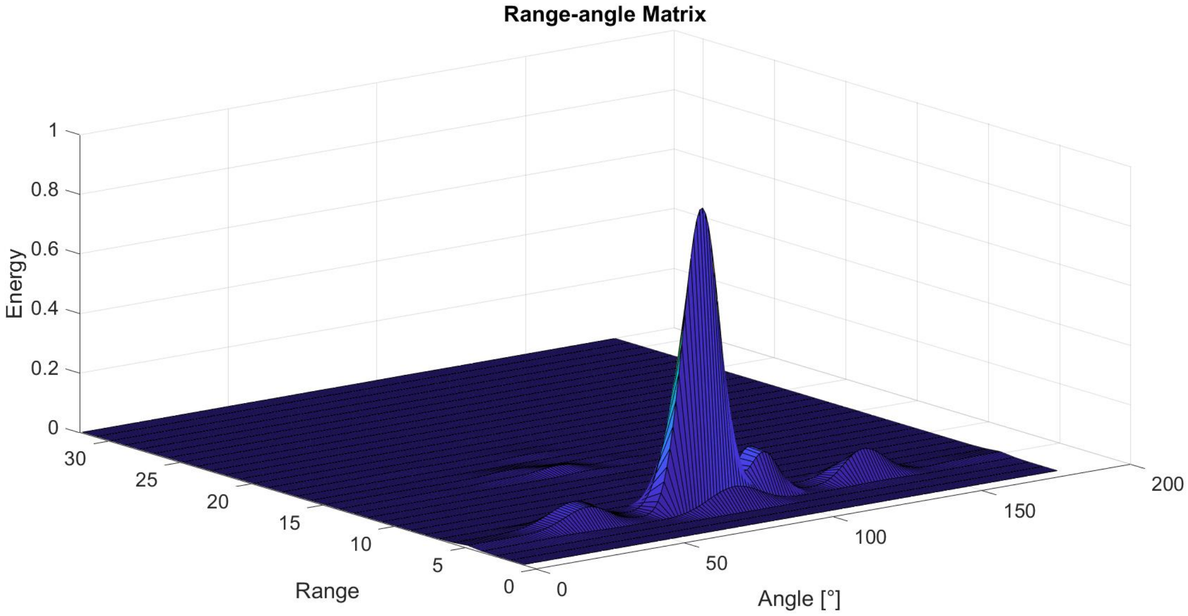

Figure 4.2: Range-angle map in the presence of an obstacle at a larger distance from the radar.

Figure 4.2 reports an example of the map $|y_{rm}|$. It can be noticed that an obstacle is near to the device in the front direction, while figure 4.3 shows the same obstacle at a slightly larger distance with respect to figure 4.2. The difference in the distance is highlighted by the different indices of the range in the matrix.
The small peaks around the obstacle's peak are responsible for a bad detection , because they are the result of the side lobes that are a source of interference. So for this we decided to develop some cleaning algorithm in order to erase their effects.

4.2 Cleaning Algorithms

In order to reconstruct an electromagnetic map of the environment, measurements should be properly post-processed in order to remove the effect of the array side-lobes. This process is necessary due to the fact that the main lobe of the antenna pattern is very wide, so when the radar "point" a particular direction it is possible that the system receives also the echoes coming from some obstacles near the direction pointed.
So in the following the cleaning algorithms that are used are described.

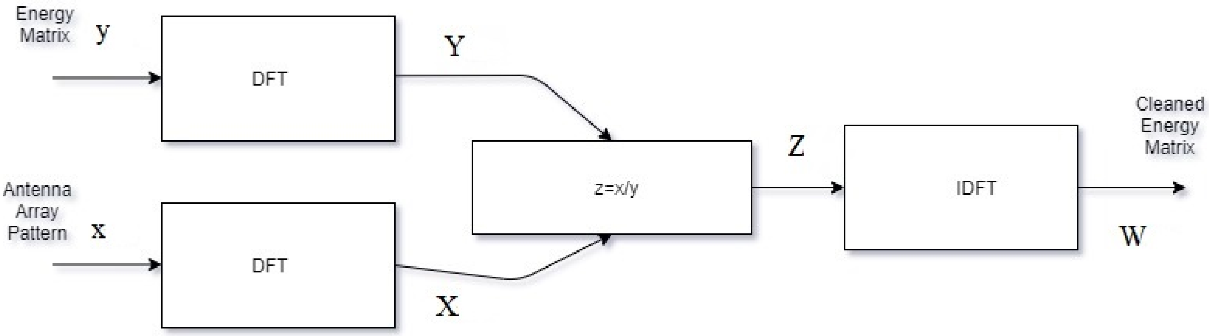

Figure 4.3: Cleaning block diagram.

- **Clean 1$^{\text{st}}$ approach:** The first algorithm, whose block diagram is reported in figure 4.3, uses the deconvolution process carried out in the frequency domain in order to try to clean the map from the antenna response.

  The algorithm applies the following operation to each row of the energy matrix y_{rm}:

$$Y_k = \sum_{m=0}^{M-1} y_{\mathrm{m}} \cdot e^{-j2\pi k \frac{m}{M}} \qquad \text{k=0,..,M-1} \tag{4.2}$$

  where Y_k represents a generic sample of the DFT of a row of the energy matrix and y_m is a sample of the matrix y_{rm}.
  At the same time the same mathematical approach is applied to the radiation pattern of the whole array x_{rm}, previously obtained starting from the radiation pattern of a single radiating element as it is shown in figure 3.2. Just to be clear the whole radiation pattern is obtained from the radiation pattern of a single element using the theory of the Array Factor [13] defined as it follows:

$$AF = \sum_{n=1}^{N} e^{j(n-1)(kd\cos\theta+\beta)} \tag{4.3}$$

  where $k = \frac{2\pi}{\lambda}$ is the wave vector, θ represents the vector that includes the angle positions from $[-\frac{\pi}{2}, \frac{\pi}{2}]$, $\beta = -kd\sin(\theta_0)$ is the initial phase and θ_0 is the steering direction, d represents the distance between the radiating elements and n is the index of the radiating elements. Then

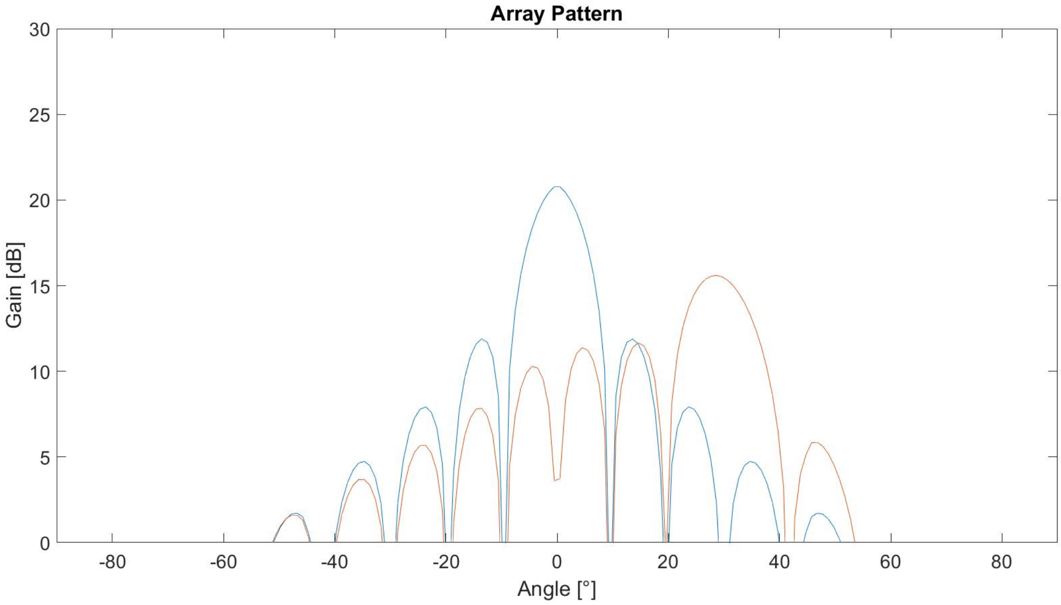

Figure 4.4: Array pattern for $\theta = 0$, $\theta = -30$ and $\theta = 30$.

the Array Factor (AF) is multiplied by the gain of the radiating element showed in figure 3.2 in order to obtain the whole array pattern diagram.

Figure 4.4 shows the array pattern for two different steering directions and it shows correctly that the main lobe moves to the steering direction and that outside the front direction the gain of the whole array is lower that the front direction gain.
So the DFT of the whole antenna pattern is given by:

$$X_k = \sum_{m=0}^{M-1} x_{\mathrm{m}} \cdot e^{-j2\pi k \frac{m}{M}} \qquad \text{k=0,..,M-1} \tag{4.4}$$

After that the folowing matrix-vector operation is computed:

$$Z = Y \cdot \frac{1}{X}. \tag{4.5}$$

The idea is to de-embed the contribution of the antenna array from the received signal so that it will depend only on the actual obstacle-channel response. The only thing that has to be checked is that the DFT of the radiation pattern does not present any zero in order to avoid noise enhancement.

Finally a Inverse Discrete Fourier Transform (IDFT) is applied on equation (4.5) as it follows:

$$w_m = \frac{1}{M}\sum_{k=0}^{M-1} Z_{\mathrm{k}} \exp^{j2\pi k \frac{m}{M}} \qquad \text{m=0,..,M-1} \tag{4.6}$$

where w_m ia generic element of the vector W that is the inverse DFT of the equation (4.5).

- **Clean 2nd approach:** This clean algorithm works row by row in order to clean the effects of the side lobes in a similar way as in [14].
  First of all the algorithm applies a constant noise threshold, ξ^z to the range-angle matrix in order to remove noisy components. Then the algorithm for each row insert all non-zero elements into a temporary vector. If the temporary vector is not empty, it is sorted from the highest to the lowest value defining the vector $E_r = [e_{r1}..., e_{rm}]$ in which r stands for the row index and the elements of the vector are the energy value sorted.
  Then, for each distance we can assume that if at least one bin, one sample of the matrix, is above the noise threshold ξ^z, the current bin likely correspond to the steering direction where the target is intercepted and thus the corresponding energy level will be considered at the output. Let also define the vector N as the difference between the energy bin and the threshdol ξ^z previously set. So at this point the algorithm uses the following energy test:

$$e_{rm} \gtrless_{D_0}^{D_1} \xi \tag{4.7}$$

  where ξ is the threshold set in this phase.
  When D_1 is satisfied, the corresponding energy bin e_{rm} is considered at the outuput of the algorithm in addition to the energy bins that are above the noise threshold ξ^z. The previous test is performed in order to check if a component has been intercepted in a side lobe direction and, consequently, it derives from a higher energy bin. Mathematically the threshold ξ is given by:

$$\xi = \mathrm{argmax}(\nu \cdot N_{rm}) + \xi^z \tag{4.8}$$

  where N_{rm} is an element of the vector N and ν is a scaling factor which puts in relation the antenna gain in the specific steering direction in the form:

$$\nu = \frac{G_k}{G_{max}} \tag{4.9}$$

where G_k is the gain in the steering direction and G_{max} is maximum array gain. The result of this process is a cleaned matrix for which the numerical results are showed in the following.

- **Clean 3$^{\text{rd}}$ approach:** This third approach is a little bit different from the previous one even if the main idea is the same. Even this algorithm tries to clean the map using the maximum values of it but in a different way.

  After the building of the energy matrix, in the same way as the previous cleaning approach, it applies a constant noise threshold in order to erase the elements that, with high probability, are not something of interest because due to noise and then starts to clean the matrix.

  Then the algorithm finds the sample of the matrix with the maximum energy and saves corresponding row (range) and column (angle) indexes. Let us introuce the vector that will contain all of the maximum values that the algorithm finds as L. Then it simulate the angular effect of the whole array pattern as it follows:

$$P = \frac{L_i}{\max x(\theta_0)} \tag{4.10}$$

$$P = P \cdot x(\theta_0) \tag{4.11}$$

where L_i is the current maximum value of the energy matrix included in the vector L, x is the array pattern in a specific steering direction and P is the power level of the maximum value cleaned from the angular effect of the radiating pattern.

After that the algorithms substracts P from the sample of the row in which is present the current maximum value. Clearly with this elaboration in correspondence of the maximum one gets a zero. Then the algorithm evaluates the entire energy of the matrix, E_{mp}, and compare it with L_i.

$$\frac{\sum_i L_i}{\sum_i L_i + E_{\text{mp}}} > 0.9 \tag{4.12}$$

If the inequality in (4.12) is satisfied, then the algorithm stops and the matrix is finally cleaned otherwise, the algorithm continues in the same way until the comparison is succesfull or the matrix contains zero elements only.

The Matlab code related to the 3 algorithms described is reported in Appendix A.

4.3 Threshold design

After the claning processing, regardless to which approach it is used, in order to reconstruct the coordinates of the obstacles or in order to use the ranges of them a threshold approach is adotped. In few word it is possible to choose between two kind of threshold regardig to what is necessary to do.
A variable threshold, with respect to the range, is useful in order to permit to the system to dectect objects that present the same RCS regardless the distance from the radar. So to compute a threshold of this kind it is necessary to know first of all the propagating law of the enviroment and then consider the radar equation given in (1.6).
Due to the fact that some parameters are not under control we decided to modify equation (1.6) as it follows:

$$P = \frac{k}{R^4} \tag{4.13}$$

where k contains every parameter of the radar equation. Then we decided to fix a distance and try to find the value for k for which the obstacle at a known distance is visible. The founded value it is used as a starting point for the threshold and in the following will be $\tilde{k}$. So the power level for this particular value will be $\tilde{P}$ and will be:

$$\tilde{P} = \frac{\tilde{k}}{R^4}. \tag{4.14}$$

From the previous equation, considering $S = \tilde{P}$ as the threshold value, it is possible to evaluate the threshold as it follows:

$$S = \frac{\tilde{k}}{R^4}. \tag{4.15}$$

so the threshold will be:

$$S[dB] = \tilde{k}[dB] - 4 \cdot 10 \cdot \log R. \tag{4.16}$$

It is also necessary to set a saturation level to the threshold.
There is another kind of threshold that can be used and this is a constant

threshold value that allows the system to perform CFAR detection. So as briefly discussed in the prevous sections a constant threshold value permit to maintain the false alarm rate constant, so in order to choose the right value for this threshold it is necessary first to model the enviroment and the noise and then choose a level of false alarm rate that need to be satisfied, as a project requirement, and then on those information it is possible to choose a contant threshold value.

4.4 Preliminary Measurements in Controlled Environment

In order to assess the validity of the cleaning algorithms explained before, we performed ad hoc measurement campaigns in some reference scenarios.

4.4.1 First experiment

The first experiment shows how a peak of the energy matrix moves when a high reflecting obstacle is moved along a circumference. In few words the system collects the measurements for some positions (i.e. the front direction and -30 degrees and 30 degrees) but with the attention to maintain the same distance from the radar that is of the order of 2.5 meters. What is expected is that the peak moves along the same row of the matrix due to the fact that the distance between the radar and the obstacle is always the same.
From figure 4.5 it is possible to see how the radiation pattern affects the energy matrix. In this example there is a single high reflecting obstacle in front of the radar but due to the radiation pattern of the antenna and due to the offline beamsteering some replicas of the obstacle are present also in some positions in which actually it is not present any obstacle. The purpose of the cleaning algorithms is to erase those replicas. Figure 4.6 shows the cleaned image after the adoption of the Approach 1. Notably, several artifacts, that are ascribed to the presence of side-lobes, are now removed. Another important aspect is that the peak due to obstacle reflection is less spread after deconvolution, since the patern effect is filtered out. Figures 4.5 and 4.7 show the comparison between the matrix before any processing and after the adoption of the Approach 2. From figure 4.5 and figure 4.8 it is possible to understand how the third algorithm works and it is possible to appreciate the accuracy in determining the object position. Each one of the 3 algorithms works well and the expected behaviour of the peak of the matrix is demonstrated from figure figure 4.5 to figure 4.8 in which the row

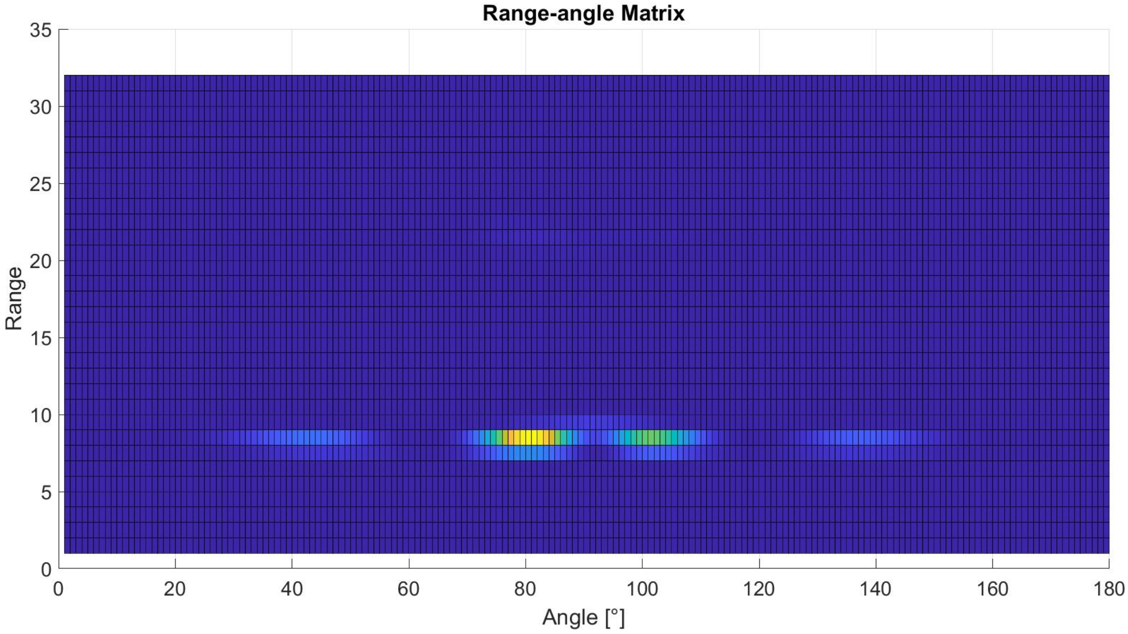

Figure 4.5: Energy matrix before any processing.

of the peak is always the same. Approach 1 and approach 3 seems to be the ones that fits better this experiment.

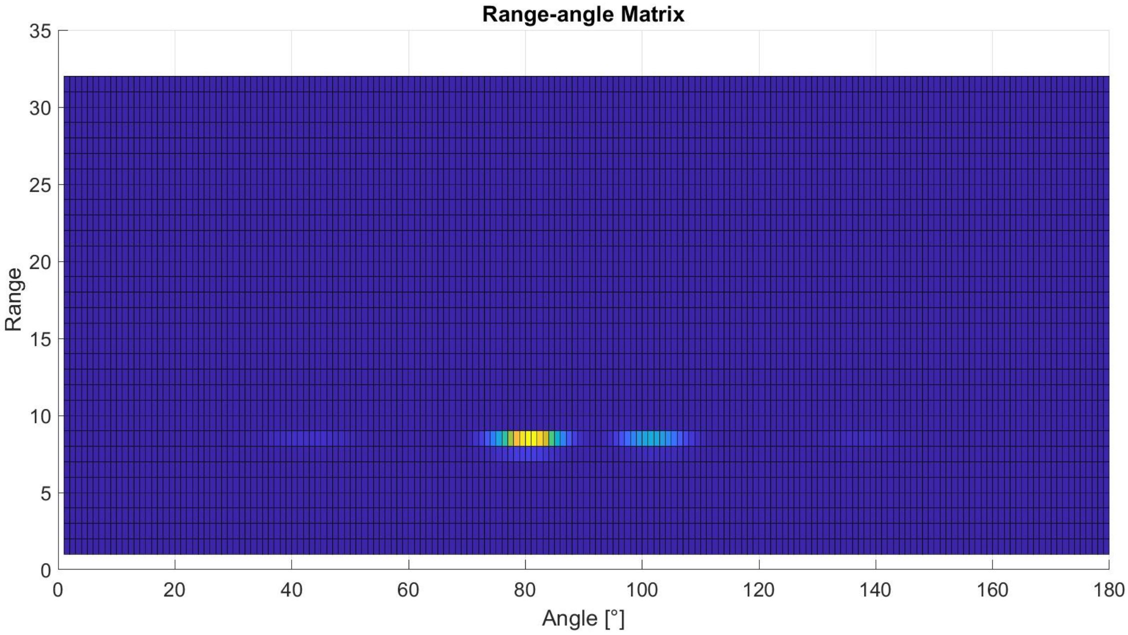

Figure 4.6: Energy matrix after the application of Approach 1.

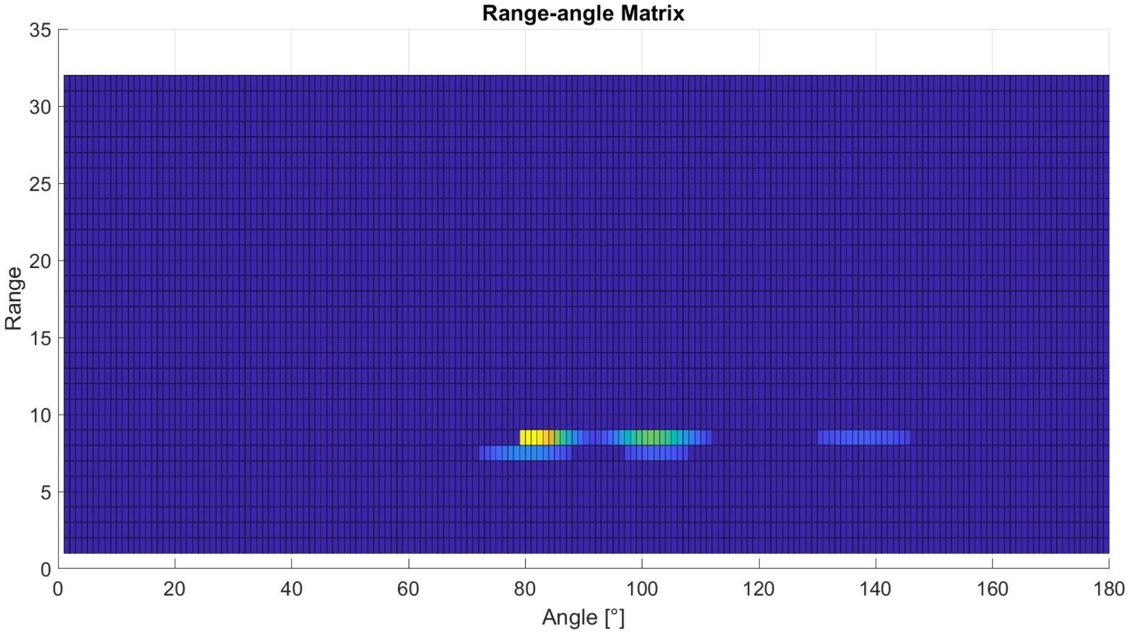

Figure 4.7: Energy matrix after the application of Approach 2.

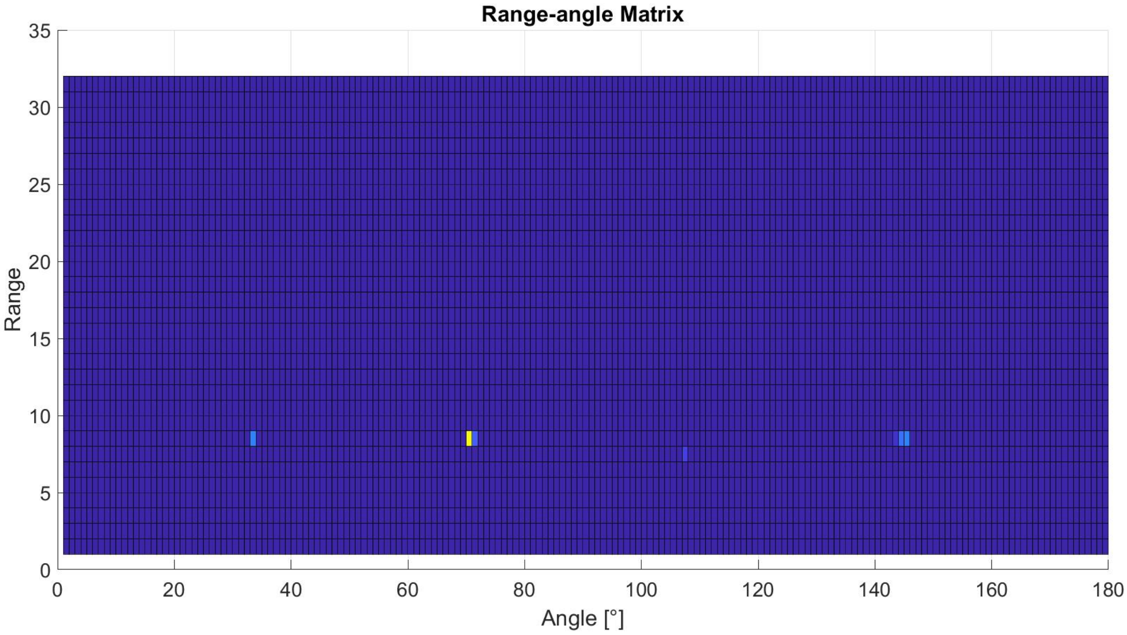

Figure 4.8: Energy matrix after the application of Approach 3.

4.4.2 Second experiment

The last experiment introduces two high reflecting obstacles in order to demonstrate that it is possible to distinguish their peaks in the matrix. The obsatcles are placed in two different angular position (i.e. -30 degress and 30 degrees) and they are placed more or less 2.5 meters far from the radar. Figure 4.9 shows the scenario before any cleaning processing. From figures 4.10,4.11 and 4.12 it is possible to see the different effetcts of the 3 algorithms on the same scenario in which are present two different reflecting obstacle. Indeed, obstacles can be well discriminated when the three approaches are used.

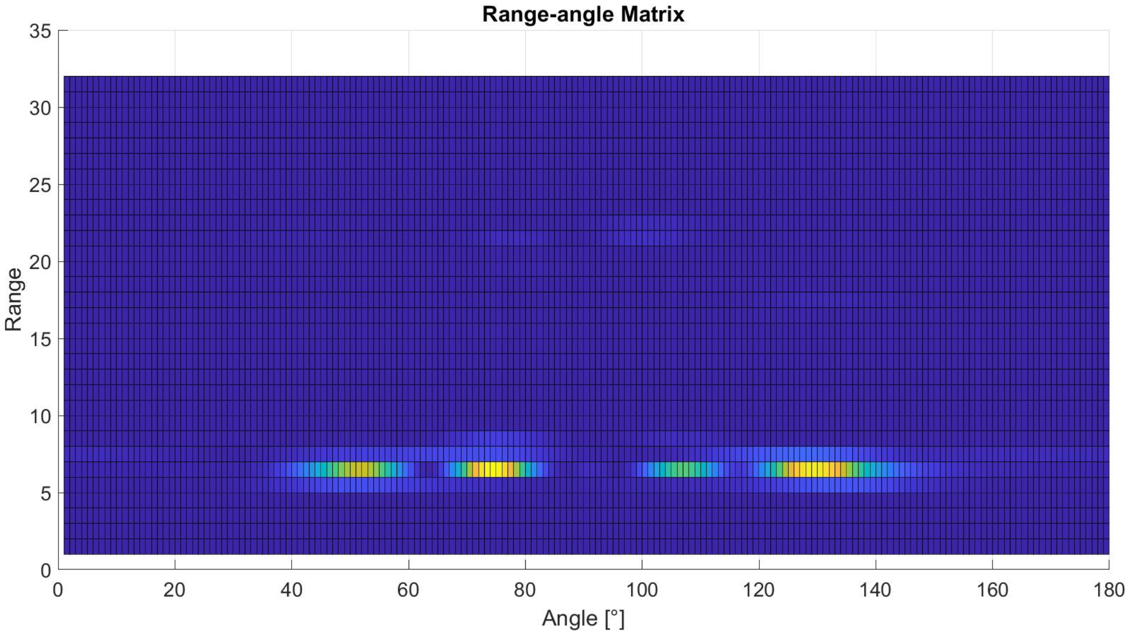

Figure 4.9: Energy matrix before any processing.

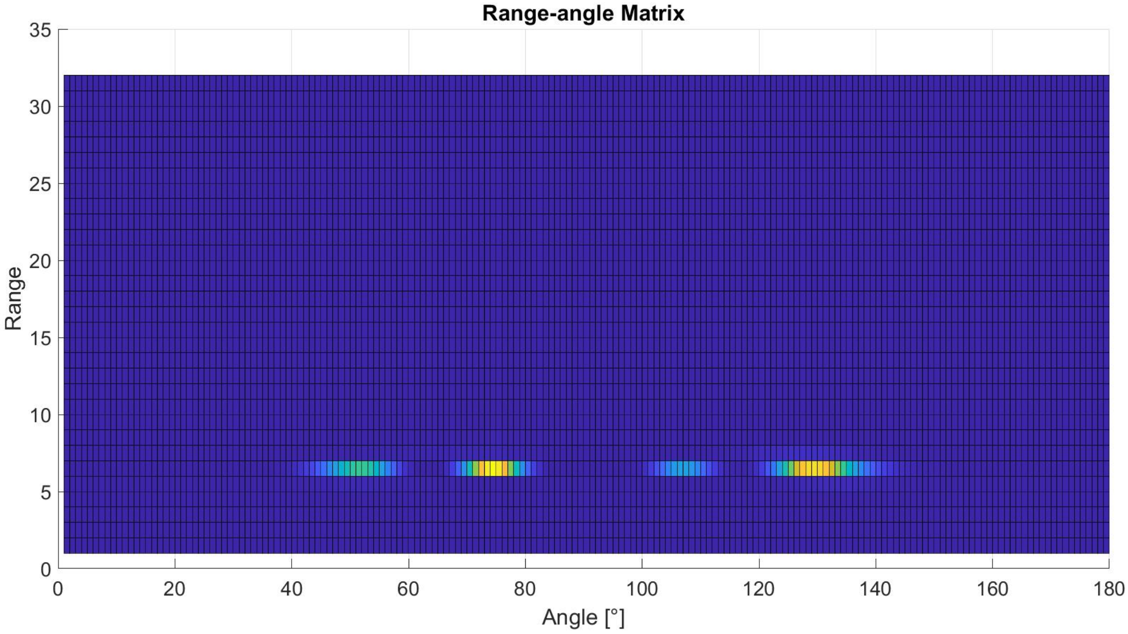

Figure 4.10: Energy matrix after the application of Approach 1.

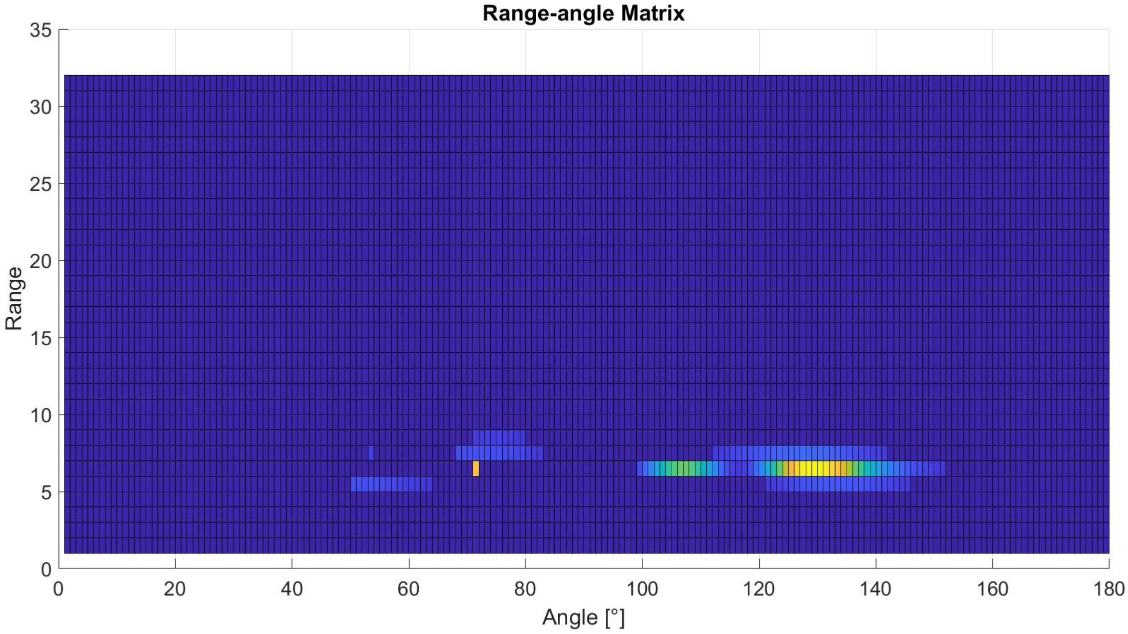

Figure 4.11: Energy matrix after the application of Approach 2.

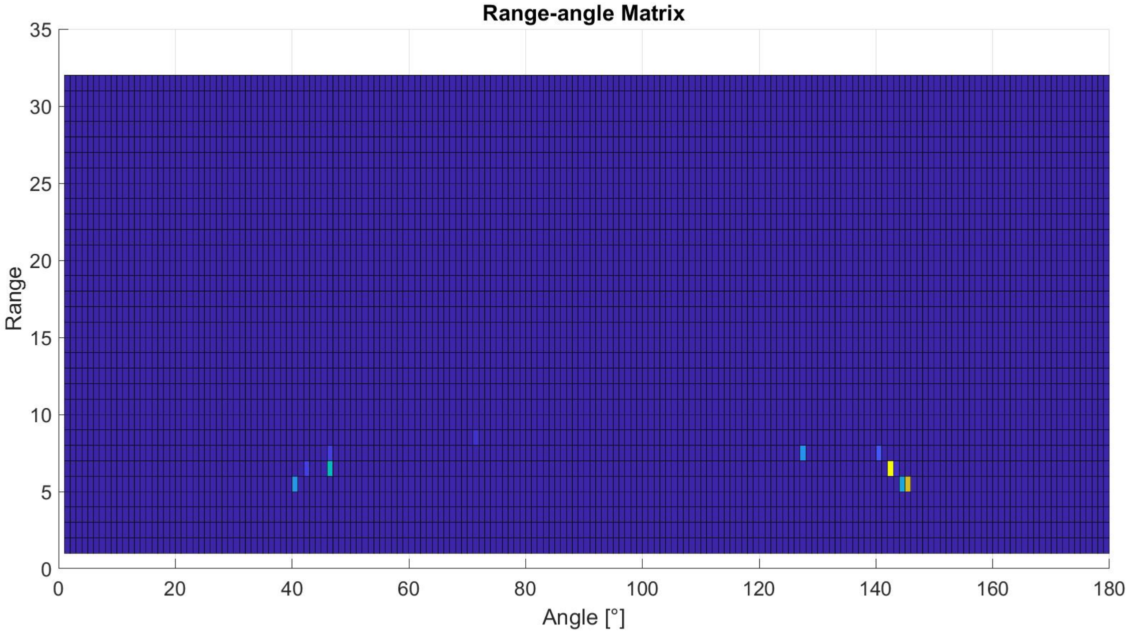

Figure 4.12: Energy matrix after the application of Approach 3.

Finally it is possible to say that approach 1 it is maybe the easiest one because it uses a standard operation, but it has the drawback that assumes that the array radiating diagram is the same for each steering directions, while it is demonstrated that the radiation pattern changes along the steering directions. The other two approaches are a little more complicated because they require some steps before obtaining the map, but they have the advantage that they consider the changes of the radiaton pattern along all the steering directions.

Chapter 5

SLAM Results

The goal of this chapter is to exploit measurement data in order to test the possibility to realize a SLAM application.
If otherwise indicated, the following parameters have been considered:

- **Frame per second (fps):** 1
- **Maximum Unambigous Range:** 5 m
- **Range Resolution:** 0.098 m
- **Maximum Radial Velocity:** 3 m/s
- **Radial Velocity Resolution:** 0.38 m/s
- **Sampling Frequency:** 4.166 MHz
- **Chirp Slope:** 100 MHzμ s
- **Chirps x Frame:** 48
- **Samples x Chirp:** 64

The maximum unambiguos range set the maximum distance beyond which a far obstacle could be confused for a near obsatcle due to the time of arrival of the echos referred to the transmission time bin. The range resolution set the minimum distance between two obsatcle for which the device is able to distinguish them.

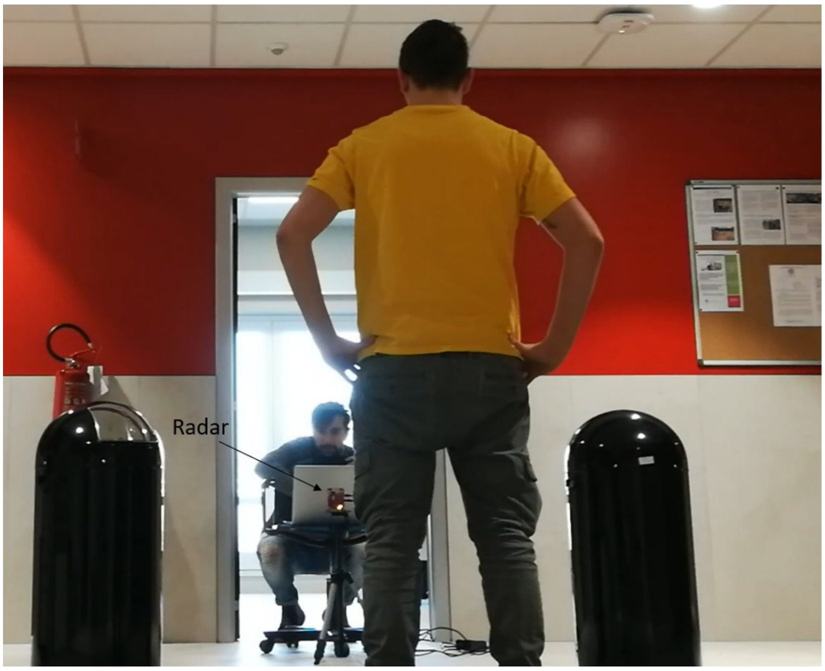

Figure 5.1: Real scenario with two high reflecting obstacles and one human obstacle.

5.1 Static Mapping

This case presents 2 high reflecting obstacles positioned approximately a meter far from the radar and 1 human obstacle positioned between the two high reflecting obstacles a little farther than the two obstacles. The scenario is shown in figure 5.1, and the purpose of this example is to verify that the system is able to recognize and discriminate the various obstacles. From figures 5.3, 5.4 and 5.5 it is possible to see the comparison of the results of the three algorithms on the same scenario. In yellow it is highlighted the remaining side lobes contribution that is significantly reduced after the three cleaning approaches. Each algorithm works well, in particular approach 3 is able to distinguish with high accuracy the three obstacles and it reduces a lot of the side lobes, approach 1 works similarly to approach 3. Approach 2 instead shows in figure 5.4 a less significant reduction of the side lobes than the other two algorithms. Just for simplicity in the following, results will refer only the adoption of the Approach 1.

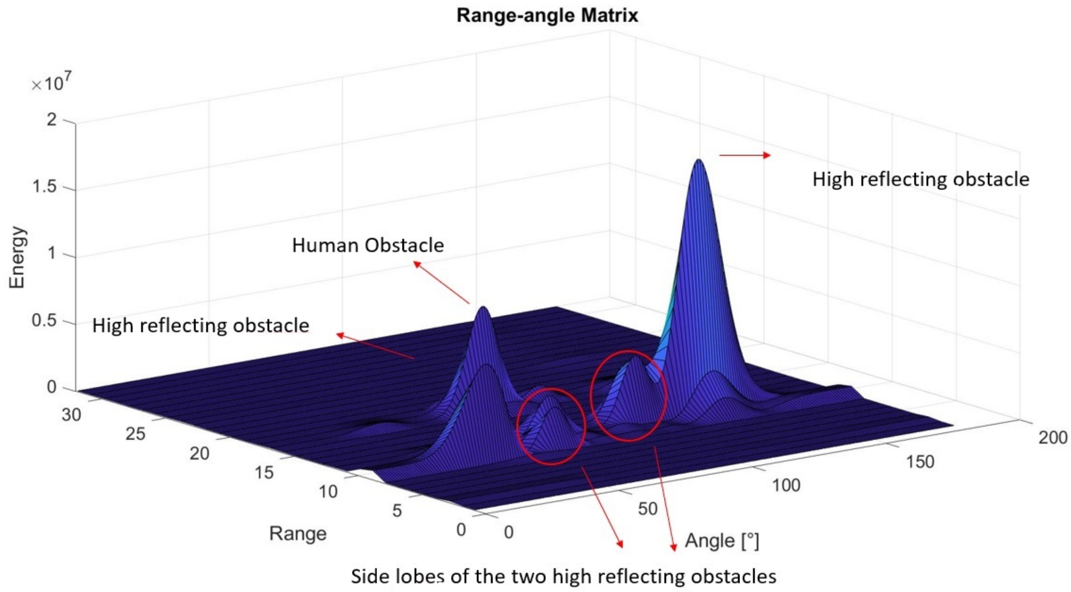

Figure 5.2: Static scenario before any processing.

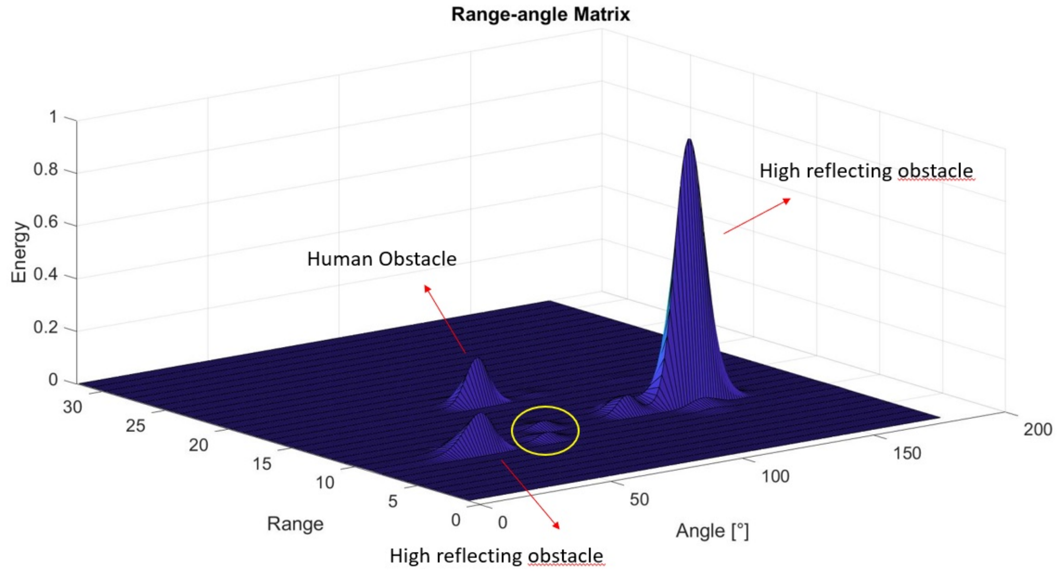

Figure 5.3: Static scenario after approach 1.

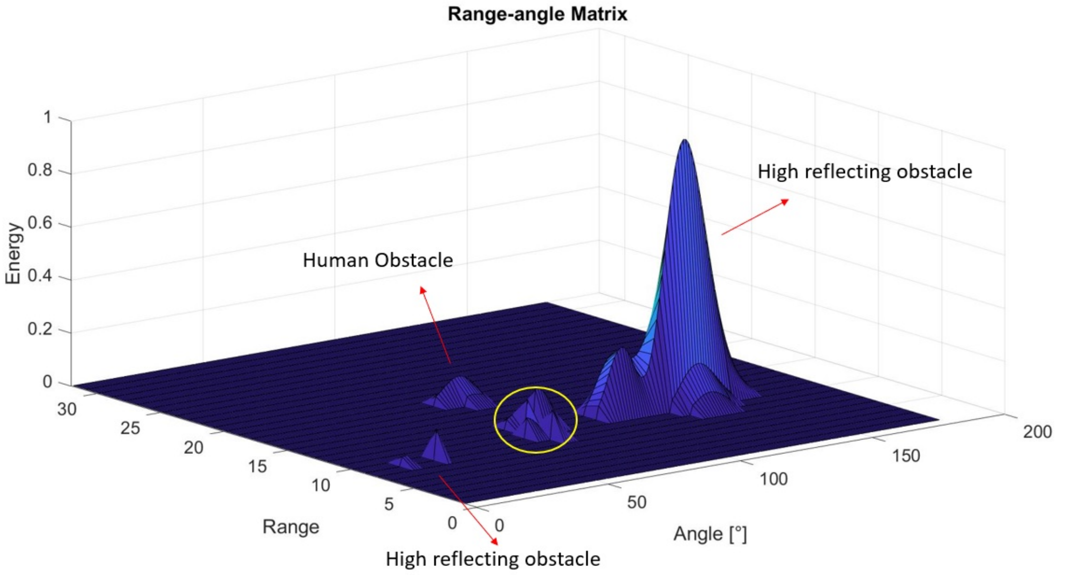

Figure 5.4: Static scenario after approach 2.

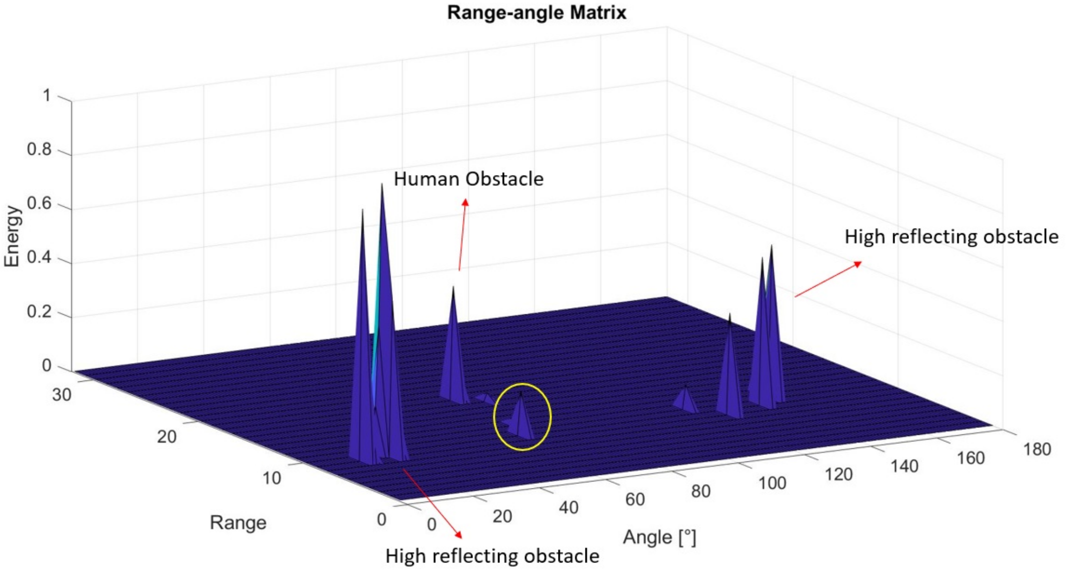

Figure 5.5: Static scenario after approach 3.

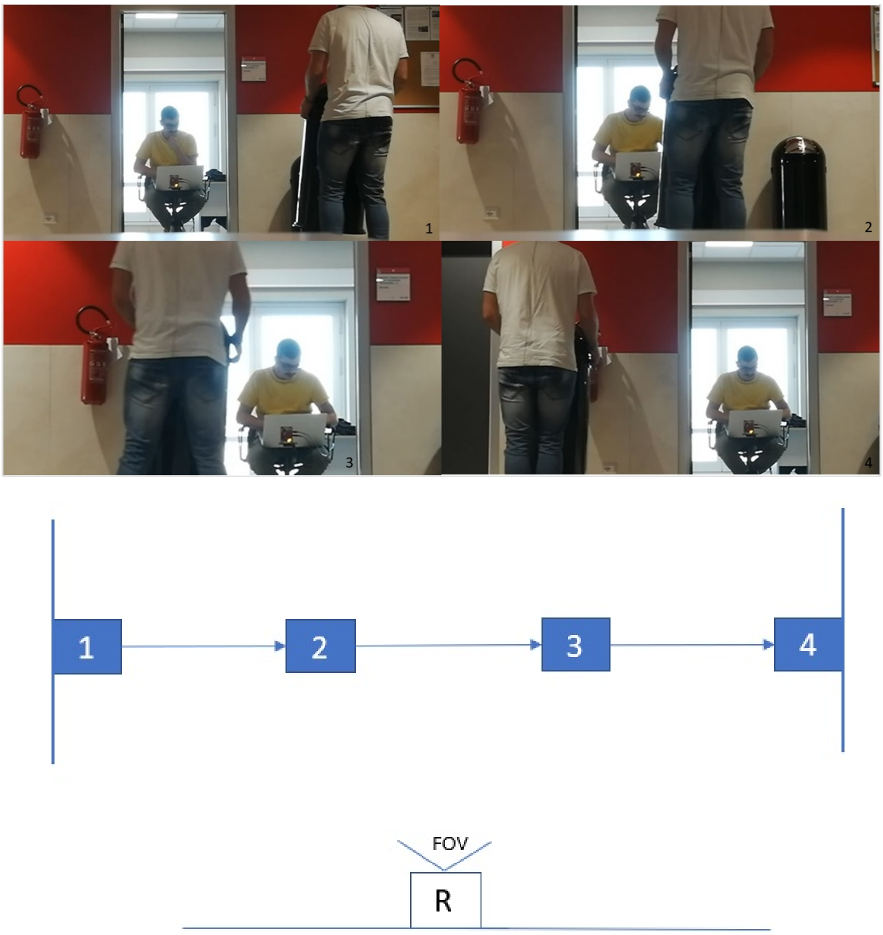

Figure 5.6: Real scenario.

5.2 Single Obstacle Movement-First scenario

In this experiment the same high reflecting obstacle used for the previous scenario is used this time in movement. In few words the obstacle moves along a straight line in front of the radar at a distance of nearly 3 meters from the radar. The purpose of this experiment is to demonstrate that the radar is able to track a moving obstacle. Figure 5.6 shows the evolution of the real experiment in which an obstacle moves in front of the radar following a straigth line, simulating a real obstacle, for example a moving person, in a possible real case.
Figures 5.7, 5.8, 5.9 and 5.10 show that the main peak follows the movement of the object, in accordance with the considered scenario.

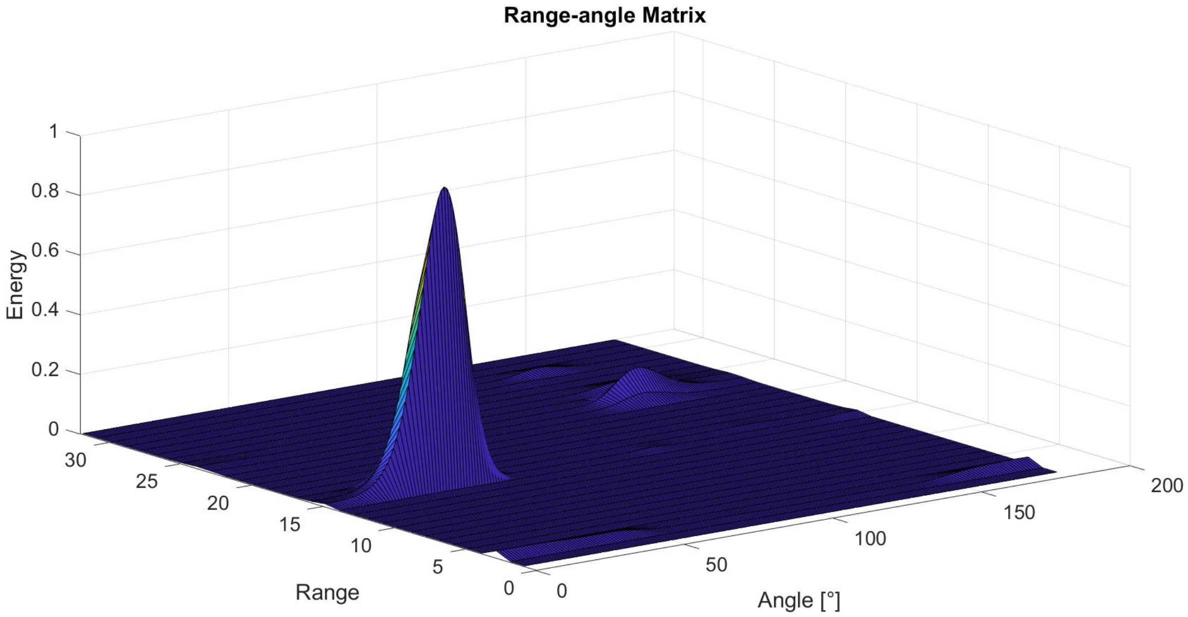

Figure 5.7: Evolution of the scenario after processing.

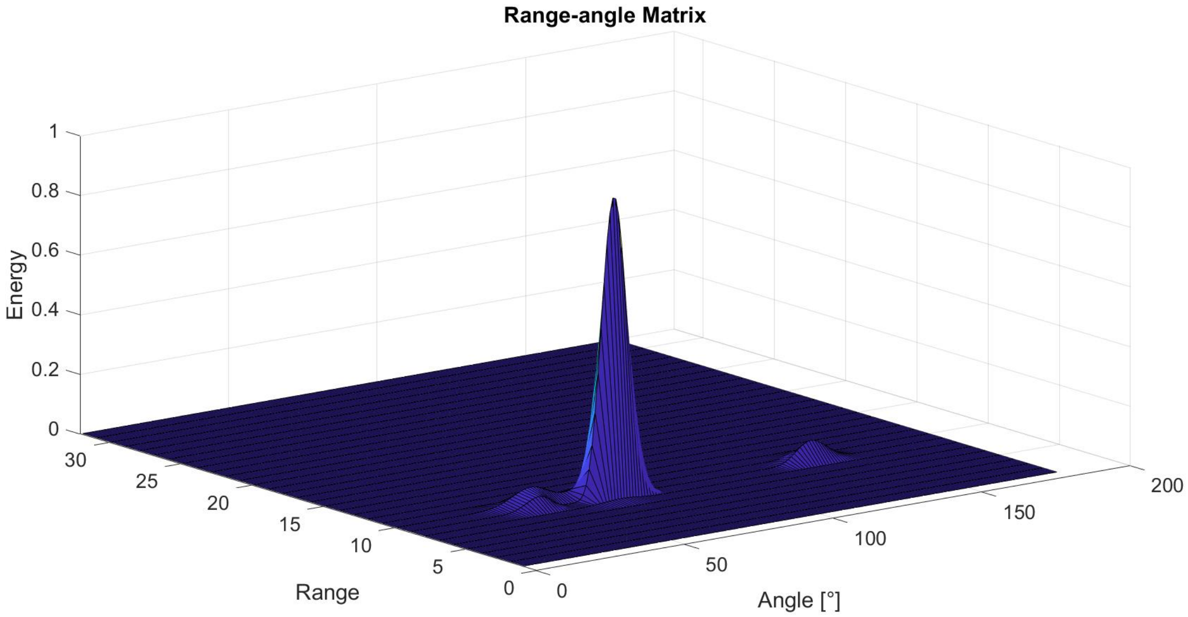

Figure 5.8: Evolution of the scenario after processing.

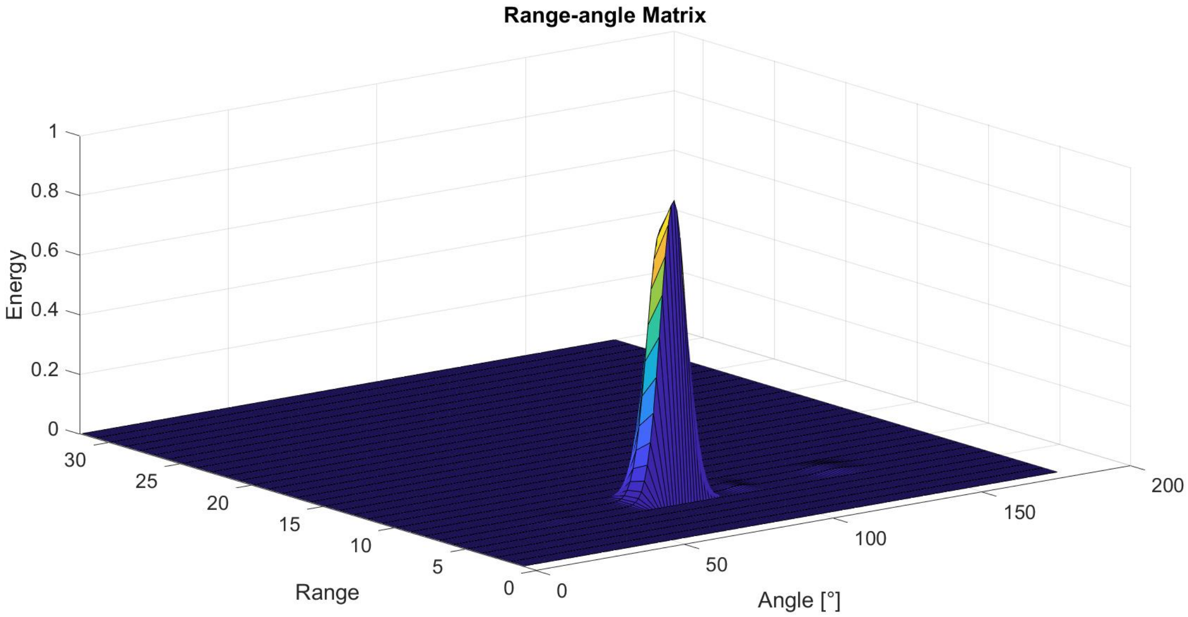

Figure 5.9: Evolution of the scenario after processing.

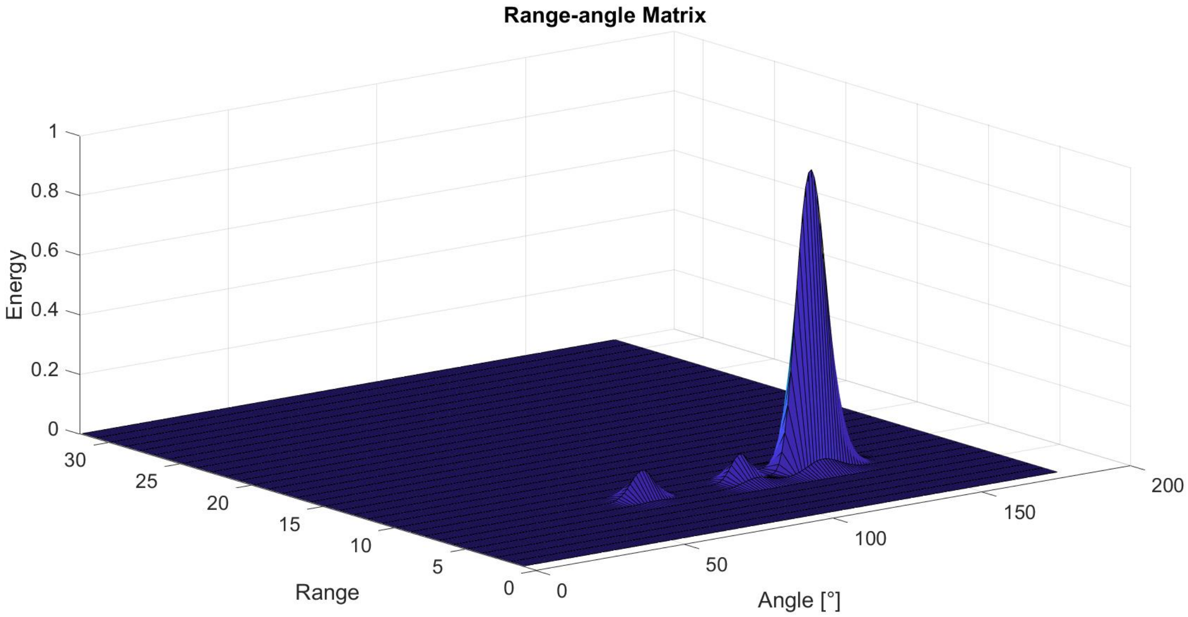

Figure 5.10: Evolution of the scenario after processing.

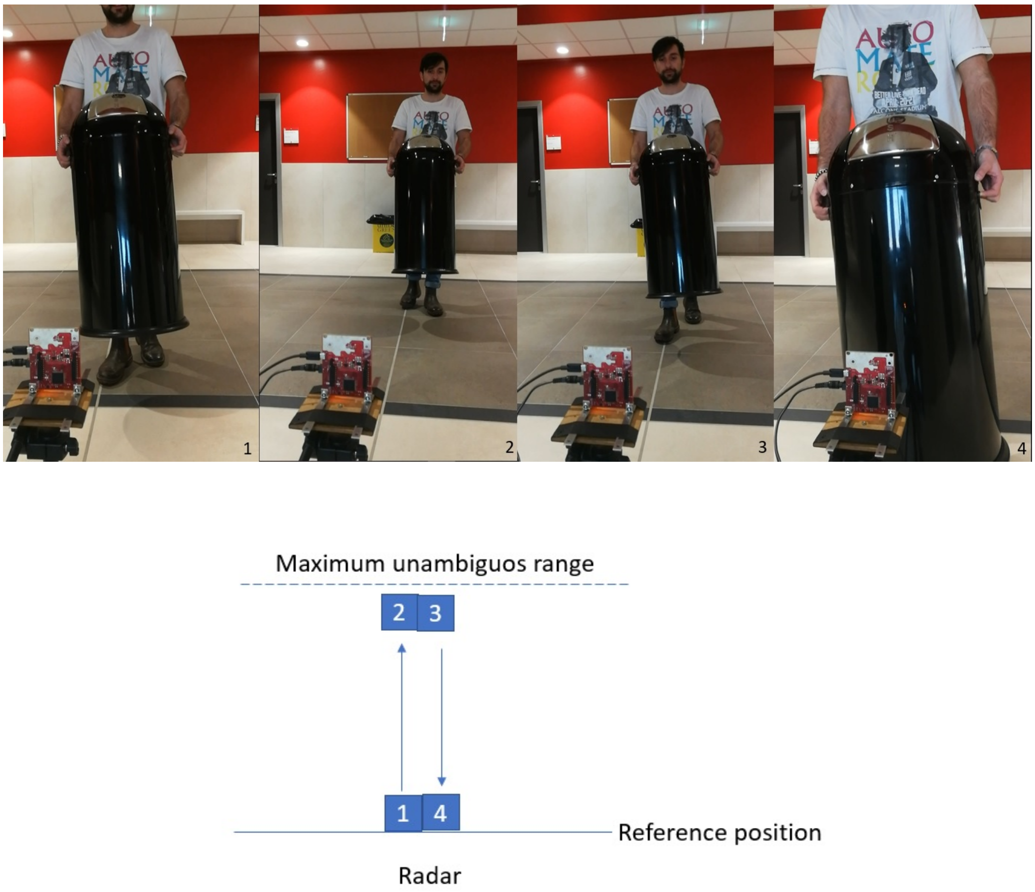

Figure 5.11: Real scenario and example of the trajectory.

5.3 Single Obstacle Movement-Second scenario

For this experiment we used the same high reflecting obstacle and this time we decided to move it in front of the radar from distance zero until the maximum unambiguos range set for the experiment that is 5 meters. In particular, the obstacle is always located within the radar reading range, so that it is always detectable. The goal of this experiment is the same of the previous one but this time we want to highlight that the obstacle moves in an another direction but the radar is able to see it even if it is moving far away.

From figure 5.12 it is possible to see in which way the peak that refers to the obstacle moves backward until reaching more or less the unambiguos range (more or less near 5 meters) and then the obstacle starts moving toward the radar.

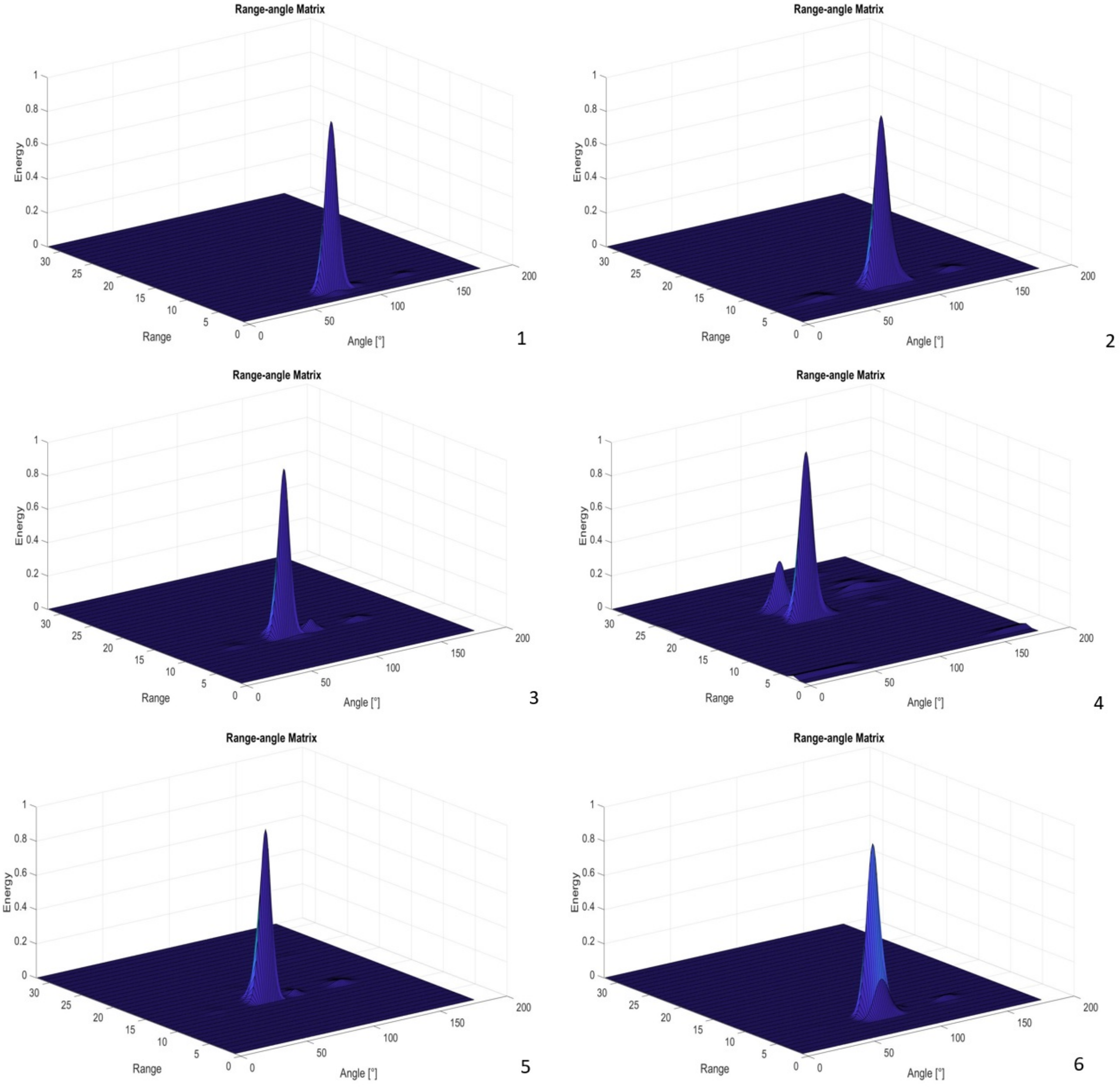

Figure 5.12: Evolution of the scenario after the first approach.

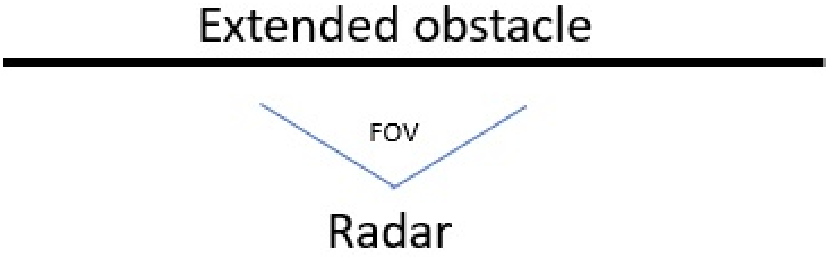

Figure 5.13: Example of the first experiment.

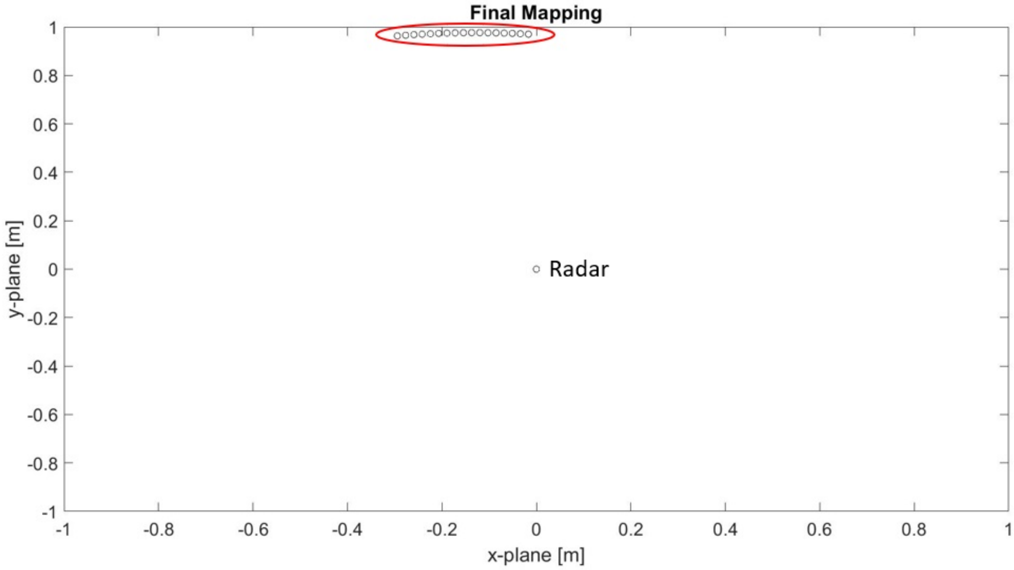

Figure 5.14: Mapping of a single wall.

5.4 MAPPING scenario

This experiment has the will to show that with this system is possible to resolve some mapping problem, without any SLAM request.
For the first scenario we decided to try to map a single wall placing the radar in front of it as close as possible. The distance that is considered stand under the meter. Due to the fact that the wall is straight and it does not presents any curvature we expect that the numerical result presents some sample placed in the same way as the real obstacle. An example of what discussed is reported in figure 5.14 in which a simply representation of the scenario is given.

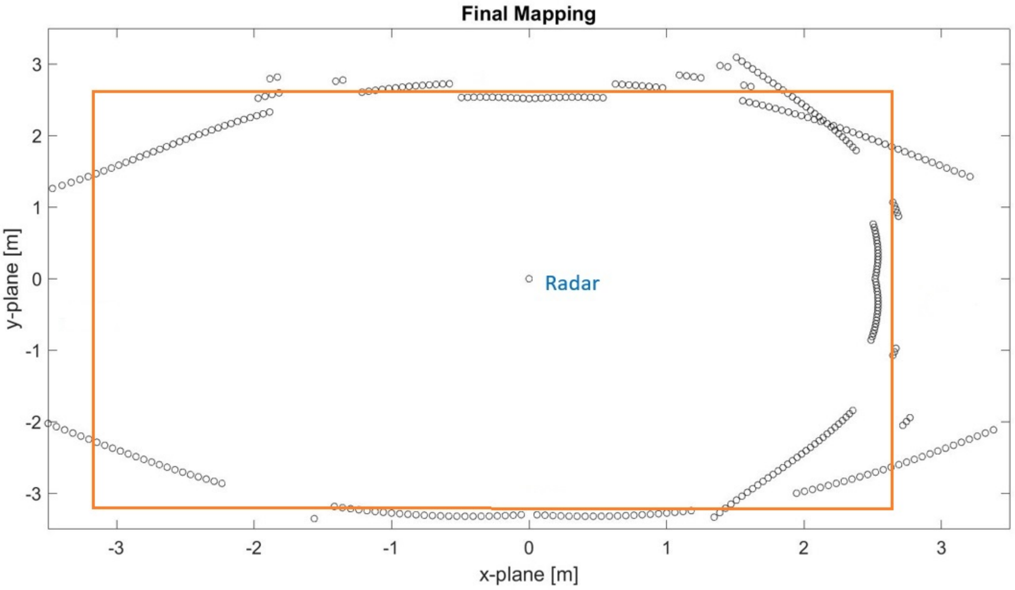

Figure 5.15: Mapping of a room.

Figure 5.14 shows the result of the mapping of a simple extended obstacle in the scenario explained even if the distances are not so similar between the real scenario and the representatio after the cleaning process.
Due to the fact that a simple mapping problem gives an acceptable result we decided to try with an entire room in order to understand if the system is able to map an enviroment with bigger dimension than the simple case considered before. So we placed the radar more or less in the center of the room and collects the measurements of the four walls. Then on matlab with a simple rotation matrix and with some multiplication between the wall measurements and the cited rotation matrix we adjust the four measurements in order to create the walls of the room. This just because our system does not rotate automatically so we decided to separate the four measurements.
What we expect is that the system is able to reconstruct each wall of the room representing them with some straight lines and we expect that also the distance between the real scenario and the numerical result corresponde.
Unfortunately referring to figure 5.15 the mapping process does not work well and the final representation of a square room is circular. This phenomenon is probably the effect of the fact that the number of radiating element is not high and so the antenna pattern diagram is not so tight. So for this reason even if in the previous sections we have demonstrated that the system works in this case the system fails and it is necessary as a future development to fix this problem.

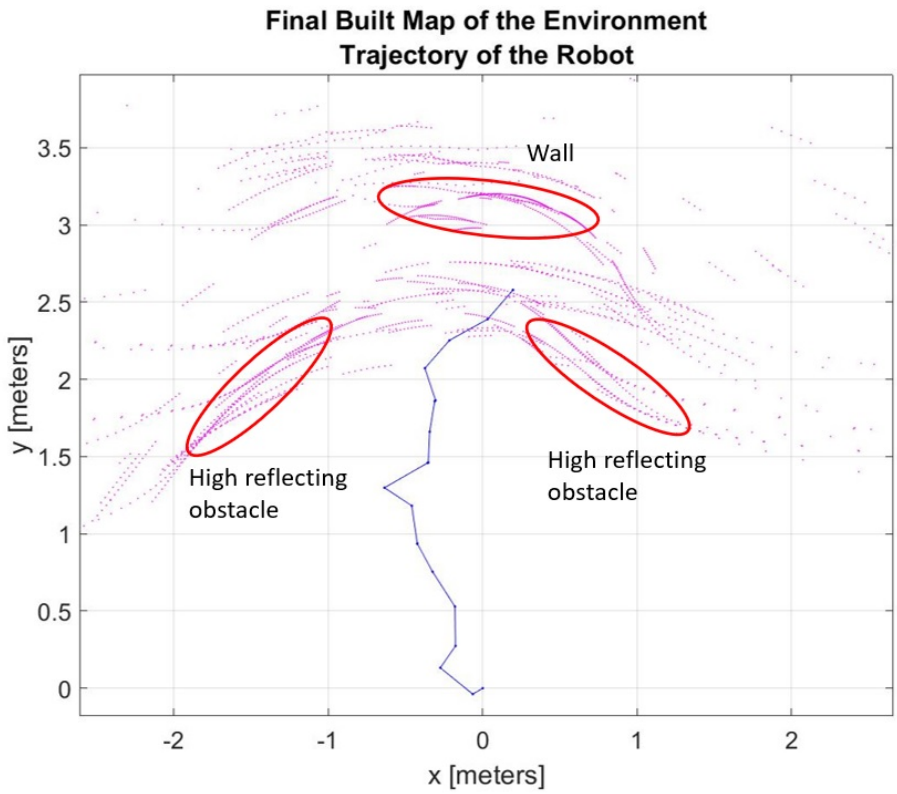

Figure 5.16: Evolution of the scenario after processing.

5.5 SLAM scenario

This scenario is similar to the one proposed for the static mapping, indeed there are two high reflecting obstacles and instead of a human obstacle there is a wall. In this case the radar moves along a straigth line directed to the wall and passing through the two high reflecting obstacles. The purpose of this experiment is to show that the system is able to track the position of the radar and simultaneously create a map of the enviroment and place the obstacles in it.

From figure 5.16 it is possible to have a look of the simple considered scenario, that has been put in place in order to preliminary test the SLAM performance. In particular the trajectory of the radar is not reported correctly and so the results are not consistent. This issue could come from the fact that the radar has a limited number of radiating elements and the first option is to enlarge the number of employed antennas, as done in [1]. Another possible source of issues is the fact that for simplicity in this thesis the changings of the pattern along the steering direction are not considered so the final result will report this problem. This is the first characterization of a personal radar using a commercial millimeter wave radar system and due to that the performances are not as good as the performances of more expensive measurement systems.

Conclusions

The purpose of this thesis is to introduce the millimeter wave technology applied to SLAM and mapping problems and to show the principal results of the proposed algorithms. In addition, we report few considerations about the difficulties arisen during the conducted activity, and we draw some possible future direction.

For this thesis work we, first of all, studied the feature of the radar and then fixed its set-up, then we made a measurement campaign that allowed us to develop some cleaning algorithms and then we applied some mapping SLAM algorithms.

According to section 5.4 and figure 5.15 it is clear that the mapping process can be improved since the reconstruction of a square room presents some inaccuracies (e.g., curve perimeter in some parts, etc). A possible reason to explain this effect is that the radar is equipped with an array with a limited number of antennas, and thus it does not allow to reach long distances and presents side-lobes that might include unwanted signal components.

Refering to section 5.5 it has emerged that the system is able to compute a simple SLAM problem obtaining interesting results. For simple SLAM problem we mean that the radar follows a straigth line trajectory and it never changes its trajectory. But if the scenario becomes more complictated or the trajectory of the radar considers a random movement the application fails doing the tracking and the localization. Again, we ascribed this effect to the limited number of involved antennas in the SLAM procedure. In a more complicated case it is also important to test an enviroment with a several number of high reflecting obstacles in order to help the SLAM algorithm to work properly.

For those particular experiments it is also importan to initially test controlled environments, where it is possible to well understand the quality of the performance.

It is necessary to say that all the measurements and the processings done do not take into account the Doppler phenomenon. This is of course a strong approximation, especially in all those environments characterized by moving

objects.
For a future development of the activity it could be useful to study the Doppler effect and then compensate it.
For all the reasons discussed before the main future development that can be done could be the usage of a different millimeter wave device with more radiating elements in order to improve the antenna array pattern and thus the quality of the measurements. In fact, despite we proposed algorithms accounting for the side-lobes effect, their impact might be still appreciable, thus affecting the overall SLAM performance.
Another possible development could be doing an ad hoc measurement campaign, maybe keeping under control all the parameters discussed in this thesis, and maybe in an enviroment in which there are not interference sources.

Appendix A

Matlab Code

A.1 Matlab SLAM algorithm

```
%load()
[scans]=DatiRadar_IWR1443();

maxLidarRange = 10;
mapResolution = 20;

slamAlg = lidarSLAM(mapResolution, maxLidarRange);

slamAlg.LoopClosureThreshold = 200;
slamAlg.LoopClosureSearchRadius =8;

firstTimeLCDetected = false;

figure ();

for i=1:length(scans)
[isScanAccepted, loopClosureInfo, optimizationInfo]=addScan(slamAlg, scans{i});
if ˜isScanAccepted
continue;
end

show(slamAlg, 'Poses', 'off');
hold on;
show(slamAlg.PoseGraph);
hold off;
firstTimeLCDetected = true;
drawnow
```

```
end
title('First loop closure');
xlabel('x [meters]')
ylabel('y [meters]')

figure ()
show(slamAlg);
title({'Final Built Map of the Environment', 'Trajectory of the Robot'});
xlabel('x [meters]')
ylabel('y [meters]')

[scans, optimizedPoses]  = scansAndPoses(slamAlg);

map=buildMap(scans, optimizedPoses, mapResolution, maxLidarRange);

figure ();
show(map);
hold on
show(slamAlg.PoseGraph, 'IDs', 'off');
hold off
title('Occupancy Grid Map Built Using Lidar SLAM');
xlabel('x [meters]')
ylabel('y [meters]')
```

A.2 Deconvolution Algorithm

```
theta_0=0;
array_pattern = ArrayPattern(theta_0);

range_angolo_dft=zeros(range_bins_num,M);

for i=1:range_bins_num
k=1;
 for k=1:M
  for j=1:M
   range_angolo_dft(i,k)=range_angolo_dft(i,k)...
                        +range_angolo_medio(i,j)*exp(-1i*k*2*pi*(j/M));
  end
 end
end

array_pattern_dft=zeros(1,M);

for i=1:length(array_pattern)
for j=1:M
array_pattern_dft(i)=array_pattern_dft(i)+array_pattern(j)*exp(-1i*k*2*pi*(j/M));
end
```

```
end

prod_trasformate=range_angolo_dft./array_pattern_dft;
Deconvolution=zeros(range_bins_num,M);

for i=1:range_bins_num
k=1;
for k=1:M
 for j=1:M
  Deconvolution(i,k)=Deconvolution(i,k)
                    +prod_trasformate(i,j)*exp(1i*k*2*pi*(j/M));
  end
 end
end

Deconvolution=abs(Deconvolution./M).^2;
range_angolo_medio=Deconvolution;
```

A.3 Clean 1^{st} approach:

```
c=3e8;
Fsamp=4.166e6;
lambda=c/Fsamp;
d=lambda/2;

gain_direction1=linspace(-10*log10(15.84893192),10*log10(1.584893),45);
gain_direction2=linspace(10*log10(1.584893),10*log10(10),45);
gain_direction3=linspace(10*log10(10),10*log10(1.584893),45);
gain_direction4=linspace(10*log10(1.584893),-10*log10(15.84893192),45);
gain_direction=[gain_direction1 gain_direction2 gain_direction3 gain_direction4];

N=12;

k=(2*pi)/lambda;
theta=linspace(-90,90,180).*pi/180;
theta_0=linspace(-90,90,180).*pi/180;
AF=zeros(length(theta_0),length(theta));

for l=1:length(theta_0)
beta=-k*d*sin(theta_0(l));
    for i=1:length(theta)
        for n=1:N
            AF(l,i)=AF(l,i)+exp(1i*(n-1)*(k*d*sin(theta(i))+beta));
        end
    end
end

for l=1:length(theta_0)
    for i=1:length(theta)
        array_pattern_matrix(l,i)=abs(AF(l,i))*10^(gain_direction(i)/10);
    end
end

Rx_gain=10*log10(63095.73445);

i_start=5;
i_stop=20;
i_self=3;

for i=1:range_bins_num
    for j=1:M
        if(10*log10(range_angolo_medio(i,j)) < sat_level | i < i_self)
            range_angolo_medio(i,j)=0;
        end
    end
end

temp=[];
temp_index=[];
max_value=[];
max_index=[];
k=1;

for i=i_start:i_stop
    for j=1:M
        if(range_angolo_medio(i,j)~=0)
            temp(k)=range_angolo_medio(i,j);
            temp_index(k)=j;
            k=k+1;
        end
     end

cont=length(temp);
l=1;
    while (cont>0)
        [max_value(l),max_index(l)]=max(temp);
         temp(max_index(l))=0;
         cont=cont-1;
         l=l+1;
    end

    if (length(max_index~=0))
        P=max(max_value)/max(array_pattern_matrix(max_index(1),:));
        P=P*array_pattern_matrix(max_index(1),:);

    for m=2:length(max_value)
        G_SL=array_pattern_matrix(max_index(m),:);
        P_SL=P.*(G_SL);%+M;
            if(max_value(m) < P_SL(max_index(m)))
                max_value(m)=0;
                range_angolo_medio(i,max_index(m))=0;
            end
    end
        k=1;
    end
end
```

A.4 Clean 2$^{\text{nd}}$ approach:

```
c=3e8;
Fsamp=4.166e6;
lambda=c/Fsamp;
d=lambda/2;

gain_direction1=linspace(-10*log10(15.84893192),10*log10(1.584893),45);
gain_direction2=linspace(10*log10(1.584893),10*log10(10),45);
gain_direction3=linspace(10*log10(10),10*log10(1.584893),45);
gain_direction4=linspace(10*log10(1.584893),-10*log10(15.84893192),45);
gain_direction=[gain_direction1 gain_direction2 gain_direction3 gain_direction4];

N=12;

k=(2*pi)/lambda;
theta=linspace(-90,90,180).*pi/180;
theta_0=linspace(-90,90,180).*pi/180;
AF=zeros(length(theta_0),length(theta));

for l=1:length(theta_0)
beta=-k*d*sin(theta_0(l));
    for i=1:length(theta)
        for n=1:N
            AF(l,i)=AF(l,i)+exp(1i*(n-1)*(k*d*sin(theta(i))+beta));
        end
    end
end

for l=1:length(theta_0)
    for i=1:length(theta)
        array_pattern_matrix(l,i)=abs(AF(l,i))*10^(gain_direction(i)/10);
    end
end

i_start=5;
i_stop=20;
i_self=3;

for i=1:range_bins_num
    for j=1:M
        if(10*log10(range_angolo_medio(i,j)) < sat_level | i < i_self)
            range_angolo_medio(i,j)=0;
        end
    end
end
```

```
fine_ciclo=0;
M_i=zeros(1,M);
pos_i=zeros(1,M);
col_i=zeros(1,M);
k=1;

while(fine_ciclo==0)
[M_i(k),col_i(k)]=max(max(range_angolo_medio));
[M_i(k),pos_i(k)]=max(range_angolo_medio(:,col_i(k)));

P=M_i(k)/max(array_pattern_matrix(col_i(k),:));
P=P.*array_pattern_matrix(col_i(k),:);

    for i=1:M
        range_angolo_medio(pos_i(k),i)=range_angolo_medio(pos_i(k),i)-P(i);
        if (range_angolo_medio(pos_i(k),i)<0)
            range_angolo_medio(pos_i(k),i)=0;
        end
    end
    E_mp=sum(sum(range_angolo_medio));

    if(sum(M_i)/(sum(M_i)+E_mp) > 0.99)
        fine_ciclo=1;
    else
        k=k+1;
    end
end
```

Ringraziamenti

Con questo elaborato si conclude la mia carriera universitaria. Un percorso lungo e non sempre facile ma che mi ha regalato tante soddisfazioni e mi ha fatto incontrare professori da cui prendere ispirazione e amici da portarmi avanti nella mia vita. Se potessi tornare indietro rifarei tutto quello che ho fatto, con il senno di poi migliorando alcune mie scelte, ma di certo ripercorrerei tutta la strada fatta.

A questo punto non posso fare altro che ringraziare il Professor Davide Dardari che mi ha dato la possibilità di lavorare a questo progetto di tesi e che mi ha supportato in ogni momento anche quando gli impegni era tanti e tutti concomitanti. Sono contento di aver collaborato con lei e spero che anche per lei possa valere la stessa cosa. Ringrazio anche il dottor Francesco Guidi che seppur nella fase finale del progetto mi ha dato una grandissima mano e dandomi sempre le giuste indicazioni. Ringrazio anche il professor Gianni Pasolini che ha contribuito anche lui alla sviluppo di questo progetto.

Desidero poi ringraziare i miei genitori perchè hanno fatto tanti sacrifici per farmi finire questo percorso e hanno sempre cercato di aiutarmi quando magari le cose non andavano come speravo. Grazie a mia sorella che si è dimostrata la fan numero uno durante il mio percorso e che anche lei ha contribuito a darmi una mano per superare tutte le prove.

Grazie a Lorenzo, che anni fa mi ha consigliato questa strada e che per fortuna ho scoperto essere la strada giusta, se non me ne avessi parlato tu forse a quest'ora non starei scrivendo queste cose.

Grazie agli amici che hanno condiviso una carriera universitaria e hanno sempre capito i miei momenti di nervoso e difficoltà. So di essere stato difficile da sopportare specialmente in questo ultimo periodo, ma vi ringrazierò sempre per la comprensione.

Infine desidero ringraziare la persona più importante della mia vita, la persona che mi sta dietro tutti i giorni e che mi sprona ad andare avanti per incominciare a creare un nostro futuro insieme. Grazie perchè non è facile sopportami ma tu lo fai anche quando non dovresti per forza. Grazie Valentina!

Buona vita a tutti!

List of Figures

Bibliography

[1] F. Guidi, A. Guerra, and D. Dardari. “Personal Mobile Radars with Millimeter-Wave Massive Arrays for Indoor Mapping”. In: *IEEE Transactions on Mobile Computing* 15.6 (2016), pp. 1471–1484.

[2] Fabrizio Berizzi. *I sistemi di Telerilevamento Radar*. 2015th ed.

[3] Texas Instruments. *Range-Estimaton*. URL: `https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-1-range-estimation`.

[4] Texas Instruments. *Velocity-Estimation*. URL: `https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-3-velocity-estimation`.

[5] Texas Instruments. *Angle-Estimation*. URL: `https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-5-angle-estimation`.

[6] Cesar Cadena et al. “Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age”. In: *IEEE Transactions on Robotics* 32.6 (2016).

[7] P. Closas D. Dardari and P. M. Djurić. “Indoor Tracking: Theory, Methods, and Technologies”. In: *IEEE* 64.Transactions on Vehicular Technology (2015).

[8] Giorgio Grisetti et al. “A tutorial on graph-based SLAM”. In: *IEEE Intelligent Transportation Systems Magazine* 2.4 (2010).

[9] Mathworks. *Matlab SLAM Algorithm*. URL: `https://it.mathworks.com/help/nav/ug/implement-simultaneous-localization-and-mapping-with-lidar-scans.html`.

[10] Texas Instruments. *IWR1443-Datasheet*. URL: `http://www.ti.com/lit/ds/symlink/iwr1443.pdf`.

[11] Texas Instruments. *IWR1443-Doxygen*. URL: `file:///C:/ti/mmwave/sdk/02/01/00/04/packages/ti/demo/xwr14xx/mmw/docs/doxygen/html/index.html`.

[12] Texas Instruments. *mmWave Demo Visualizer*. URL: `https://dev.ti.com/gallery/view/mmwave/mmWave/Demo/Visualizer/ver/2.1.0/`.

[13] Constantine A. Balanis. *Antenna Theory: Analysis and Design, 3rd Edition*. 2005.

[14] F. Guidi et al. "Indoor Environment-Adaptive Mapping With Beamsteering Massive Arrays". In: *IEEE Transactions on Vehicular Technology* 67.10 (2018).