



**Bruno Miguel
Silva Santos**

**Contagem e Rastreamento de Pessoas com Radar
FMCW para Sistemas de Vídeo Vigilância
Aumentada**

**People Counting and Tracking with FMCW Radar for
Augmented Video Surveillance Systems**



Universidade de Aveiro
2021

**Bruno Miguel
Silva Santos**

**Contagem e Rastreamento de Pessoas com Radar
FMCW para Sistemas de Vídeo Vigilância
Aumentada**

**People Counting and Tracking with FMCW Radar for
Augmented Video Surveillance Systems**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eléctronica e Telecomunicações, realizada sob a orientação científica do Doutor Arnaldo da Silva Rodrigues de Oliveira, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Pedro Cruz, Arquitecto de Sistemas na Bosch Security Systems, S.A.

o júri / the jury

presidente / president

Professor Doutor Paulo Miguel de Jesus Dias
Professor Auxiliar da Universidade de Aveiro

vogais / examiners committee

Professor Doutor Rafael Ferreira da Silva Caldeirinha
Professor Coordenador C/Agregação do Instituto Politécnico de Leiria

Professor Doutor Arnaldo Silva Rodrigues de Oliveira
Professor Auxiliar da Universidade de Aveiro

agradecimentos / acknowledgements

Em primeiro lugar queria agradecer a Bosch Security Systems, S.A. Ovar por me ter dado esta oportunidade e me ter recebido de forma fantástica nas suas instalações. Agradeço aos membros da equipa de Ovar que me acolheram como se fosse um deles. Um agradecimento especial ao Doutor Pedro Cruz pelo apoio e por ter acreditado em mim.

Ao Professor Doutor Arnaldo Oliveira quero deixar a minha gratidão, pela compreensão e disponibilidade para me ajudar, dentro e fora do mundo académico.

Às amizades que se fizeram durante este percurso académico, um obrigado por tudo e por todos os momentos que passamos ao longo destes anos e por terem feito com que o meu dia fosse sempre melhor.

À Sara, um obrigado especial, por me ter acompanhado e apoiado durante todos os momentos.

Por último lugar, mas não menos importante, aos meus pais e avós, agradeço por me terem suportado e me terem dado a oportunidade de seguir o meu percurso académico. Sem o apoio deles, nada disto seria possível.

Palavras Chave

Radar, Detecção de Pessoas, Fusão Sensorial, Rastreamento.

Resumo

Esta dissertação descreve como um Radar de onda contínua modulada em frequência (FMCW) pode ser utilizado na contagem e localização de pessoas dentro de uma sala. Para além da contagem de pessoas, também é realizado o rastreamento espacial da pessoa através de métodos de processamento da nuvem de pontos provenientes do Radar.

É feito um estudo teórico, detalhando como o Radar é capaz de calcular parâmetros fundamentais como a distância, velocidade e ângulo de chegada de um objeto presente no campo de visão (FoV), assim como os cálculos de processamento de sinal efetuados com o kit de avaliação do Radar selecionado. A aplicação desenvolvida no âmbito da dissertação contempla tópicos como, técnicas de redução de ruído, algoritmos de segmentação de dados e associação destes a pessoas e/ou objetos, utilizando algoritmos de processamento digital de sinal (DSP) capazes de predição de movimento em ambientes ruidosos com dados extraídos do Radar, abordando ainda a projeção de coordenadas cartesianas numa câmara de videovigilância localizada num ponto distinto da sala.

Este projeto foi pensado para sistemas de segurança e proteção, onde o Radar seria integrado numa câmara de videovigilância, complementado as capacidades de deteção de pessoas obtidas por meio de algoritmos de processamento de imagem, os quais tendencialmente falham em determinadas condições atmosféricas ou de iluminação enquanto o Radar é menos afetado.

A aplicação de interface gráfica de utilizador (GUI) desenvolvida nesta dissertação, foi testada em cenários onde as pessoas percorrem movimentos normais como caminhar ou estão sentadas no seu local de trabalho e construído de forma a ser capaz de funcionar em tempo real sem grandes perdas na aquisição de vídeo. São apresentados alguns resultados práticos focando diversos casos de uso de deteção e rastreamento de pessoas, atingindo um desempenho interessante.

Keywords

Radar, People Detection, Sensor Fusion, Tracking

Abstract

The present dissertation describes how a frequency-modulated continuous-wave (FMCW) Radar can be employed to count and locate persons inside a room. Apart from counting, it is also performed the spatial person tracking through processing of the Radar point cloud.

A theoretical study is done, detailing how the Radar can compute fundamental parameters like distance, velocity, and angle of arrival of an object present in the field of view (FoV) and the signal processing computations done with the selected Radar evaluation kit. The application developed in the scope of this dissertation contemplates topics such as, noise mitigation, clustering algorithms and association with detected persons and/or objects. It also uses digital signal processing (DSP) algorithms capable of movement prediction in noisy environments with data extracted from the Radar and approaches the projection of Cartesian coordinates into a video surveillance camera located at a different place.

This project was intended for security and safety systems, wherein the Radar would be integrated in a video surveillance camera, complementing people detection capabilities achieved via image processing algorithms, which fail in certain atmospheric or lightning conditions while the Radar is less affected.

The graphical user interface (GUI) application developed in this dissertation was tested in scenarios where people would walk in regular patterns or are seated in their workplace and constructed in a way that allows it to work in real-time without major losses on the video acquisition performance. Practical results of diverse person detection and tracking use cases are demonstrated with an interesting performance.

Contents

Contents	i
List of Figures	v
List of Tables	ix
Acronyms	xi
1 Introduction	1
1.1 Context	1
1.2 Motivation	2
1.3 Objectives	3
1.4 Contributions	3
1.5 Dissertation Structure	3
2 Fundamental Radar Technology	5
2.1 Introduction to Radar Technology	5
2.1.1 Basic Radar Architecture	6
2.1.2 Radar Concepts and Parameters	7
2.2 FMCW Radar	8
2.2.1 Range Estimation	9
2.2.2 Range Resolution	11
2.2.3 Maximum Measurable Distance	12
2.2.4 Velocity Estimation	12
2.2.5 Velocity Resolution	16
2.2.6 Maximum Measurable Velocity	16
2.2.7 Angle Estimation	17
2.2.8 Angle Resolution	18
2.2.9 Angular Field of View	18
2.2.10 Summary of Radar Parameters	19

2.3	MIMO Radar	19
2.3.1	MIMO Model	19
2.3.2	Bartlett Beamforming	21
2.3.3	Capon Beamforming	21
2.4	Signal Characterization	23
2.4.1	SNR	23
2.4.2	Radar Cross Section	24
2.5	Constant False Alarm Rate	24
2.5.1	Cell Averaging CFAR	25
2.5.2	Smallest Of Cell Averaging CFAR	26
3	Texas Instruments IWR6843ISK-ODS Kit	27
3.1	Hardware Architecture	27
3.1.1	Antennas	30
3.1.2	Transmitter and Receiver System	30
3.1.3	Chirp Generator	32
3.1.4	DSP and MSS	33
3.2	People Counting Application Processing Chain	34
3.2.1	Range Processing and Clutter Removal	36
3.2.2	Spatial Covariance Matrix Estimation and Capon Beamforming	36
3.2.3	CFAR Detection Algorithm	37
3.2.4	Elevation Beamforming	40
3.2.5	Doppler Estimation	40
3.3	Application Processing Interface	41
3.3.1	Serial Communication Setup	41
3.3.2	TLV Data Format	43
4	People Counting and Tracking	45
4.1	Application Flowchart	45
4.2	Point Cloud Processing	46
4.2.1	Clustering with DBSCAN	46
4.2.1.1	Choosing Parameters Value	47
4.2.1.2	Calculating Centroids of the Cluster	48
4.2.2	Point Cloud Filtering	48
4.2.2.1	Two-Step Identification	49
4.2.2.2	SNR Evaluation	49
4.2.3	Tracking With Kalman Filter	49
4.2.3.1	Predict Stage	50

4.2.3.2	Update Stage	50
4.2.3.3	Predict Target Movement	51
4.2.3.4	Allocate to the Closest Target	51
4.2.3.5	Update Target Position	51
4.2.4	Sensor Fusion with Camera	54
4.2.4.1	Cartesian Coordinates System into Image Pixels	54
4.2.4.2	Image Enhancement with Radar Information	55
4.3	People Counting GUI	55
4.3.1	3D Target Visualizer	56
4.3.1.1	Live Statistics	57
5	Results	59
5.1	Chirp Parameters	59
5.2	People Counting	60
5.2.1	Scenario Setup	60
5.2.2	Tests Performed	60
5.2.3	People Counting Accuracy	61
5.3	Real Time People Tracking	61
5.3.1	Scenario Setup	61
5.3.2	Tests Results	62
5.3.3	Tracking Analysis	65
5.4	Sensor Fusion	66
5.4.1	Image with Radar Data	66
5.4.2	Extending the FoV	66
5.4.3	Tracking in Low Visibility Conditions	67
5.5	Analysis of the Results Obtained	68
6	Conclusion and Future Work	69
6.1	Conclusion	69
6.2	Future Work	70
	Bibliography	71
	Appendix	73
	Abstract Presented at Conference URSI GASS 2021	73

List of Figures

2.1	Basic Radar principle.	5
2.2	An FMCW radar schematic.	6
2.3	3D geometry of the radar [9].	7
2.4	Transmitted signal in the time domain.	8
2.5	Chirp signal.	9
2.6	Transmitted and received signals with respective IF signals for a single target (left side) and multiple targets (right side).	10
2.7	The 1D-FFT or range-FFT.	11
2.8	The amplitude-time plot (left) of the IF signals of two targets at very close distance. At the right plot is the FFT of the IF signals with a small observation time window with just N samples. Adapted from [10].	12
2.9	The amplitude-time plot (top) of the IF signals at very close distance where the frequency difference is really low. The right side is the FFT of the IF signals, but this time, with a large observation time window that allows $2N$ samples. Adapted from [10].	12
2.10	The amplitude-time plots of different signals from two time distinct time slots. Adapted from [10].	13
2.11	The FFT of the IF signals. Adapted from [10].	14
2.12	At the left side is phase difference in the exponential form at multiple chirp indexes and in the right, the 2D-FFT across the phase differences with peak at the at the angular velocity ω_1 . Adapted from [10].	14
2.13	Velocity scenario of two targets at the same distance from the radar with different velocities. Adapted from [10].	15
2.14	The results of the 2D-FFT where we can separate both targets in the range and velocity domain. Each colored square represent the objects from 2.13. Adapted from [10].	16
2.15	A single target AoA estimation with the trigonometry relation between the two received waves. The reflection is originated from a target at θ degrees, and the antennas are separated by a distance d	17
2.16	The 3D-FFT across the phase difference from different receiving antennas to resolve targets in the angular domain.	18

2.17	TX and RX antennas of a MIMO Radar.	20
2.18	Capon (orange) and Bartlett (blue) beamforming with three close targets at [-10,0,10] degrees. The plot is the power-angle spectrum Capon and Bartlett beamforming, maximums of this function indicate that there is a reflection coming from that angle [18].	22
2.19	Capon (orange) and Bartlett (blue) beamforming, but this time with four targets with a large separation between them [18].	23
2.20	Architecture of the CFAR algorithm. At top there is the CFAR window where the red cell is the CUT, the "x" marked cells are the guard cells and the blue cells are the reference cells.	25
3.1	TI Mmwave FMCW Radar IWR6843ISK-ODS board. From [22].	27
3.2	TI mmwave FMCW Radar IWR6843 functional block diagram. From [23].	29
3.3	Antenna configuration and virtual array.	30
3.4	Radiations Patterns. From [22].	30
3.5	Transmitter Sub-System for each Tx antenna. From [23].	31
3.6	Receiver Sub-System for each Rx antenna. From [23].	31
3.7	ADC spectrum for real and complex-baseband architectures where each arrow represents a reflection detected. From [24].	32
3.8	Clock system and Synthesizer for chirp generation. From [23].	33
3.9	Processor system block diagram in IWR6843 devices. Left side is the DSP Sub-System and right side is the Main Sub-System. From [23].	34
3.10	People Counting application block diagram of the processing chain. Converts sampled data into point clouds. From [25].	35
3.11	Block diagram Range-FFT until Range-Azimuth Heatmap. From [25].	36
3.12	CFAR Detection and elevation estimation. From [25].	37
3.13	2D SOCA-CFAR flowchart. Adapted from [9].	38
3.14	2D CFAR Input. Range-Angle heatmap where the represented cells are a Range-FFT at a θ angle. Adapted from [9].	39
3.15	Doppler estimation and combination of results with detected points. From [25].	40
3.16	IWR6843 UART communication setup with PC. Adapted from [25]	41
3.17	Timing diagram of the task activity done between PC and Radar. From [9]	42
3.18	Data Packet structure sent to PC with multiple TLV blocks from a single packet. From [9].	43
3.19	The frame header structure. From [9].	43
3.20	Point Cloud unit structure. From [9].	43
3.21	Point detected Structure. From [9].	43
4.1	Developed application flowchart.	46
4.2	Clustering in 3D. Each color represents a label, the yellow color is the noise which are points with no associated cluster.	47

4.3	Centroid (Green dot) of the cluster (Red).	48
4.4	Diagram of how the algorithm predict, allocate and update the calculated cluster centroid into existent targets.	51
4.5	Target Detected (blue bounding box) and tracking (blue line) with movement prediction (red arrow) in a circular movement (white dots).	52
4.6	Plot of the top view of the target tracking. The red line is the ground truth, the green markers are the calculated cluster centroids and the blue line is the updated state vector.	52
4.7	The RMSE plot. Difference between Kalman and ground truth in meters across 150 steps.	53
4.8	Top-view plot of a non-linear movement and the output of the tracking stage.	53
4.9	Respective RMSE from figure 4.8.	54
4.10	Perspective projection of a point located at (X, Y, Z) onto a plane located at distance f from the origin.	54
4.11	FoV enhancement in Azimuth FoV.	55
4.12	People Counting and Tracking GUI.	56
4.13	Multi-tracking of two targets.	56
4.14	Statistics table	57
5.2	Tests performed for People Tracking.	62
5.3	Results for perpendicular walking pattern of a person with a slow speed.	62
5.4	Results for the parallel test.	63
5.5	Results for a complex walking pattern.	63
5.6	Results for two targets moving parallel to each other in opposite directions.	64
5.7	Point cloud plus the target location projection onto the camera.	66
5.8	Target moving out of camera vision but still in radar FoV.	67
5.9	Tracking in extreme visibility conditions.	67

List of Tables

2.1	Summary of target estimation parameters.	19
5.1	Parameters used for the tests above derived from chapter 2 (table 2.1).	59
5.2	Tests performed and results obtained for people counting.	60

Acronyms

ADC	Analog-to-digital Converter
AoA	Angle of Arrival
CFAR	Constant False Alarm Rate
CUT	Cell Under Test
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DMA	Direct Memory Access
DSP	Digital Signal Processor
DSS	Digital Sub-System
EKF	Extended Kalman Filter
FFT	Fast Fourier Transform
FMCW	Frequency Modulated Continuous Wave
FoV	Field of View
GUI	Graphical User Interface
IF	Intermediate Frequency
LNA	Low Noise Amplifier
MIMO	Multiple-input Multiple-output
MSS	Master Sub-System
RCS	Radar Cross Section
RF	Radio Frequency
RMSE	Root Mean Square Error
SNR	Signal-to-noise Ratio
TDM	Time Division Multiplexing
TI	Texas Instruments
TLV	Tag Length Value
UART	Universal Asynchronous Receiver/Transmitter

Chapter 1

Introduction

RADAR stands for **R**ADio **D**etection **A**nd **R**anging and it is most of the time linked with military equipment and speed control systems used by law enforcement, with the purpose of measure distance, direction and velocity [1]. Although the technology had a great jump in development during World War II its invention is dated to the early 20th century when a German physicist developed a “Telemobiloscope” [2], a device capable of detecting distant ships through the fog. This was based on the experiments of *Heinrich Hertz* that showed that electromagnetic waves were reflected by metallic objects[3]. The first Radars were mostly used in the military in the detection of oncoming airplanes and missiles, but after the war, the technology continued to evolve into better devices. These future devices would be capable of achieving higher frequencies and had integrated hardware capable of modern digital signal processing techniques, which allowed the radar to achieve finer resolutions in distance and velocity. With the advance of production techniques, a Radar can be produced relatively cheap and small, this makes the radar a perfect integration into modern and civil applications such as smart cities, automobile navigation systems, meteorology, and also medicine [4].

1.1 Context

The applications for the radar can range from the most simple cases as automated door openers, lighting and traffic control, into more complex uses as gesture recognition and vital sensing where the radar needs to detect more information than just the presence [5]. The industry is constantly discovering new uses for the radar to help in the creation of smart factories, capable of autonomous transporting with collision avoidance, robots with enhanced accuracy, perimeter security of the facilities, people counting in a room, etc.

Recently there is a rising trend in Radar applications in the automotive industry with high-end automobiles enhanced with Radars which allows them to provide parking assistance, door opening collision detectors and lane departure warnings [6]. The new autonomous driving concept of cars requires the Radar to be more than just a passive sensor with only the ability to warn the driver. The new Radar must be capable of acting in the control of the vehicle.

The data provided by the Radar in combination with data from other sensors could

enhance the capabilities of a system, a camera with a radar providing detection and location of people could allow detection even in extreme conditions where the image processing techniques would fail to detect. This would bring more safety and reliability to the security systems by enabling sensor fusion. The process of combining data from multiple sensors is called Sensor Fusion and lessens the uncertainty of the system if these sensors would be used separately.

This dissertation was done in collaboration with Bosch Security Systems, S.A. – Ovar, Portugal, in the scope of running R&D projects at Video Systems (VS) business unit. All the material and conditions for the developments of this dissertation were made available. The BT-VS team is currently working on projects which involve the development of new technologies and solutions for safer cities and smart surveillance systems, where Radar technology is seen as a major skill to potentiate further knowledge to complement scene understanding in diverse scenarios and weather conditions.

1.2 Motivation

There are multiple devices capable of detecting the presence of a person like a sonar, laser, or lidar, but the radar is preferred over these because is less affected by diverse environmental effects such as fog, rain, darkness, heat or even a fire [7]. With the increased interest in Radar solutions, multiple companies are putting effort into creating self-contained development kits that permit fast prototyping and testing of Radar applications. The fact that this is achievable in a small and cheap board makes them an interesting solution for academic researches and integration in multi-sensory applications.

This dissertation objective includes the development of a system integrating a camera + radar to detect and track the presence of people in a room, this could be achieved with just the camera embedded with YOLOv4 which is a real-time object detection algorithm that performs well in the detection of people in an image. The problem with solutions based on machine learning is that they require images with certain conditions to achieve the expected performance, but in a real-life scenario, these conditions are not always met. If these applications have a security interest, they should not just perform well 90% of the time, so the addition of radar to the system would help complement the camera because the data extractions are unaffected by elements that usually degrade the quality of image processing algorithms.

Despite the Radar being unaffected by elements like light conditions and fog, the principle of operation of the radar is prone to generate a diversity of unwanted objects (clutter) in indoor environments which imposes a challenge into the development of processing techniques.

This dissertation aims to add robust detection and tracking of moving persons in an indoor scenario, feeding the results into a camera and doing a live projection of the radar detections on the image. This dissertation does not discuss other alternatives for peoples detections since the radar will work as a complementary instrument when other sensors would fail.

1.3 Objectives

The envisioned objective of this dissertation is the development of an application capable of detecting and tracking people based on data provided by a MIMO FMCW Radar. The application objectives to be achieved in this dissertation can be listed as the following:

1. Detect and locate the spatial position of people on the point cloud;
2. Count the number of people present in the FoV;
3. Track the movement of multiple people in the Radar FoV;
4. Project the results onto an image in conditions where image processing algorithms would fail.

The application must be capable of working in real time data in a indoor scenario.

1.4 Contributions

The main contribution of this dissertation consists in the development of a robust Python module capable of receiving point clouds through UART and perform counting and tracking of people in real-time. The information provided by this module is then fed into a live image provided by a camera for sensor fusion. An adaptation of a TI application GUI was developed to show all the relevant data in real-time.

The work developed on this dissertation resulted in an extended abstract presented at *Union Radio-Scientifique Internationale* General Assembly and Scientific Symposium (URSI GASS 2021) held in Rome, Italy between 28th September and 4th August. The title of the submitted abstract was "*FMCW Millimeter-Wave Radar-Augmented People Tracking Solution for Video Surveillance Systems*".

1.5 Dissertation Structure

The dissertation outline is structured into six chapters:

1. **Chapter 1** - The current chapter serves as an introductory chapter to the Radar by giving a brief historical description and explaining why the Radar was the choice of election amidst other possibilities for people detection and tracking.
2. **Chapter 2** - The state of the Art, the theory behind an FMCW Radar and how it can measure the distance, velocity and Angle of Arrival of a target. The concepts of MIMO Radar, SNR, Radar Cross Section and CFAR algorithms are also introduced.
3. **Chapter 3** - This chapter presents the development kit TI IWR6843ISK-ODS and its hardware tools and the software processing chain. The People Counting application is also presented with the information provided by Texas Instruments about how the Radar converts the received signals into point clouds.
4. **Chapter 4** - Describes the application developed capable of interpreting the point clouds returned by the radar and returning the counting and tracking of people detected while plotting the results on a GUI based on Texas Instruments application mmwave People Counting GUI. The targets detected are then plotted into a live camera view.

5. **Chapter 5** - This chapter provides the results of the application developed, in multiple scenarios. Separating the results into tracking, detection and image projection.
6. **Chapter 6** - The last chapter presents the conclusion of the work developed in this dissertation with emphasis on the problems faced. This chapter also mentions the future work that can be generated from this dissertation.

Chapter 2

Fundamental Radar Technology

In this chapter, we will present a typical architecture for an FMCW radar which will be our focus of study since it is a cheap and growing technology. The chapter will also give an explanation of the theory and mathematics behind the FMCW signals present in the architecture and the procedure to compute characteristics such as range, velocity and angle of a target. A MIMO FMCW Radar concept will also be introduced to show how accurate angle calculations are done in multiple targets environment. At the end of the chapter, the characterization of the oncoming signals and a CFAR algorithm will be presented to show the conditions required by a reflection received by the radar to be considered a valid target.

2.1 Introduction to Radar Technology

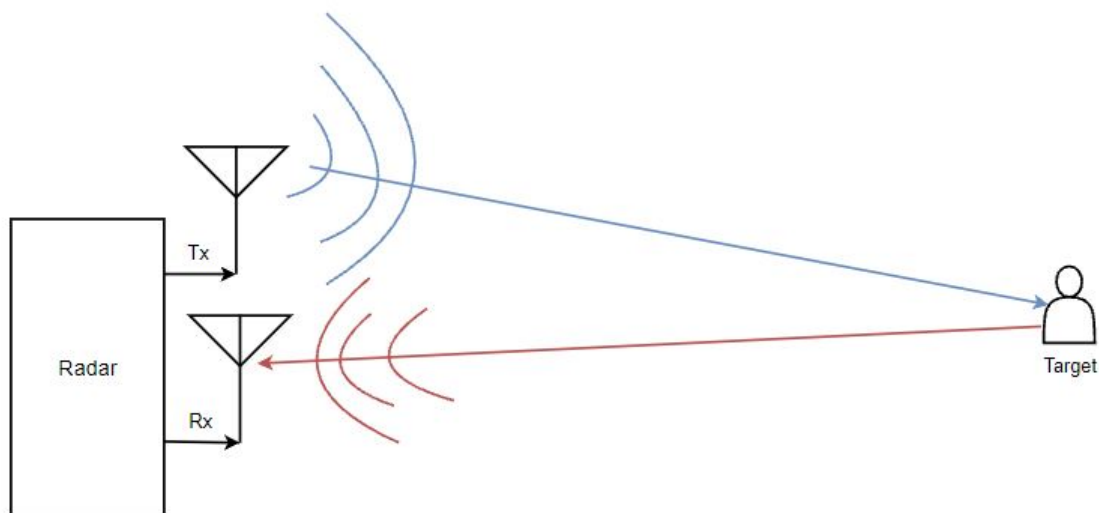


Figure 2.1: Basic Radar principle.

The principle behind an FMCW radar is to radiate a wave with linearly increasing frequency and then receive and detect the reflected waves as illustrated in figure 2.1, by comparing the transmitted signal and the respective reflection and with the help of modern

signal processing techniques it is not only possible to measure the localization of a target, but also, classify it, tracking the movement and suppressing the noise also known as clutter [1].

2.1.1 Basic Radar Architecture

As seen in figure 2.2, a typical FMCW Radar architecture consists of [8]:

- An **Waveform Generator** to generate the triangular-shaped signal paired with a Voltage Controlled Oscillator (VCO) to create a wave with linearly increasing frequency;
- Two **antennas**, a transmitter and a receiver to radiate and receive the waves generated by the waveform generator;
- **Mixer** to generate the Intermediate Frequency (IF) signal by multiplying both Tx and Rx signals;
- **Power Amplifier (PA)** to give enough power so that the radiated waves can strike the target and have enough power to be reflected;
- **Low Noise Amplifier (LNA)** to act as a filter for the unwanted components at different frequencies that interfere with the signals and also to amplify the components in interest while maintaining a good noise figure;
- **Low Pass Filter (LPF)** to remove the unwanted components generated by the mixer, typically works at baseband;
- **ADC** to convert to digital domain so that digital signal processing techniques could be applied.

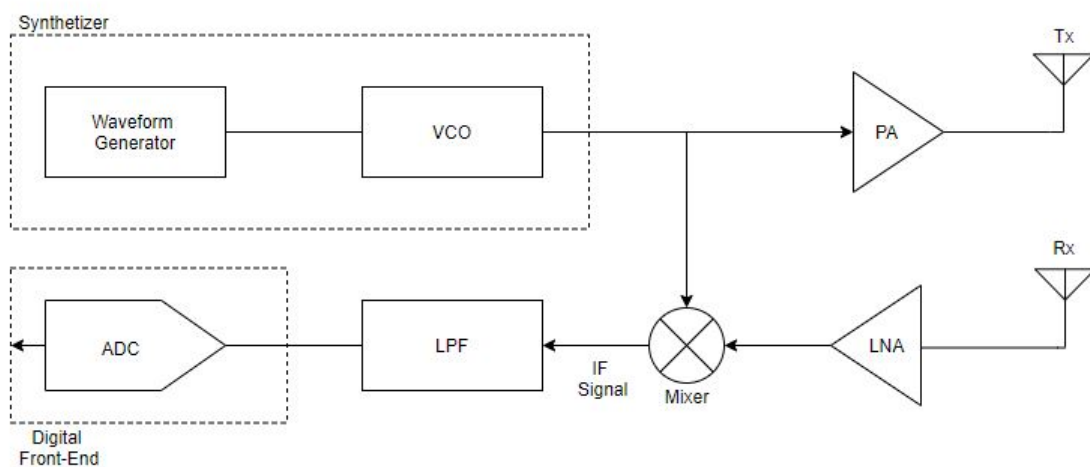


Figure 2.2: An FMCW radar schematic.

2.1.2 Radar Concepts and Parameters

To help with further reading of this dissertation a few basic concepts about radar systems are presented here.

- A **Target** is an object that can be detected by the radar system at a certain distance.
- **Speed** or **Doppler** is the scalar representation of the radial velocity. This can be obtained by computing the frequency shift caused by the moving target onto the reflected wave, also known as the Doppler effect.
- **Angle of Arrival** (AoA) is the angle from which the reflected wave arrives at the receiver antennas and the central axis.
- **Chirp** is a transmitted wave with linearly increasing frequency.
- **SNR** is the relation between the Power of a detected point and the noise power.
- A **frame** is the basic structure of data that has information about the target distance, speed and possibly angle from multiple consecutive chirps.
- **Point Cloud** is the result of the processing chain which identifies a target or multiple targets speed and distance in relation to the radar at a given instant of time.
- **Clutter** is the echo that originated from the multi-path reflection on walls, ground, etc. These can later create ghost/fake targets on the point clouds.
- The **multiple-input multiple-output** (MIMO) radar is a technology used in multiple transmission and reception systems because it adds new degrees of freedom which in the FMCW case, allows to perform beamforming techniques that enables the system to detect reflections from multiple directions.

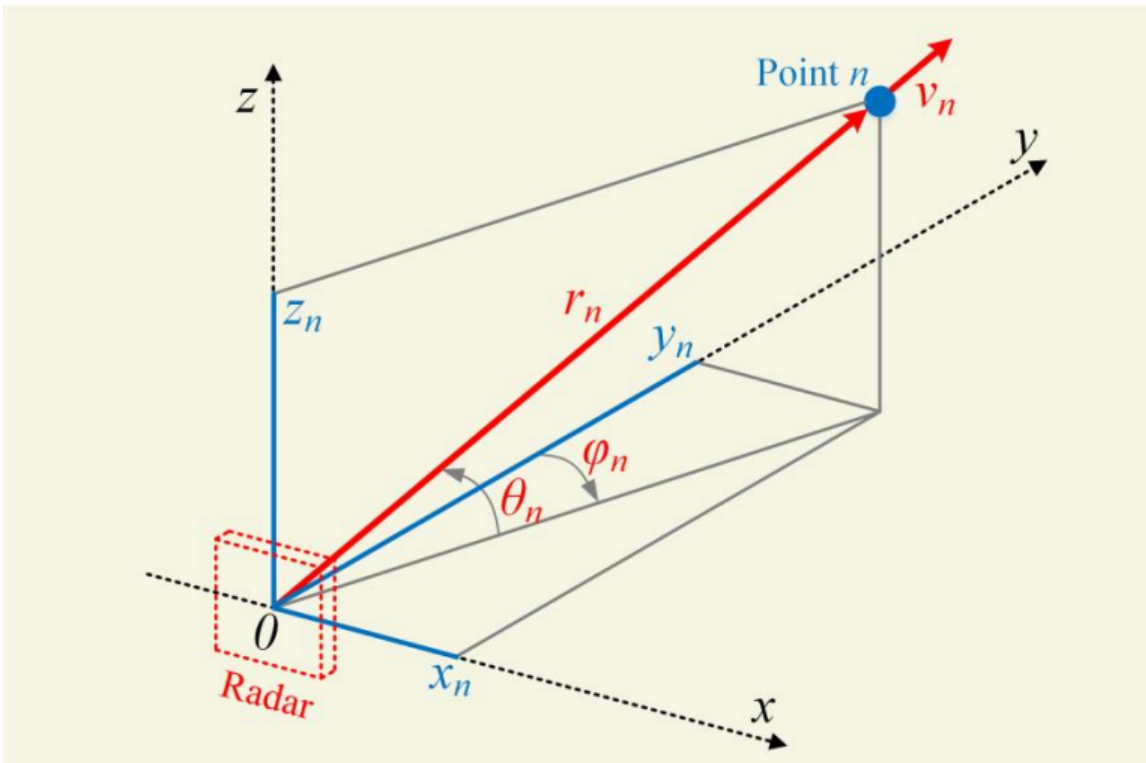


Figure 2.3: 3D geometry of the radar [9].

The radar geometry is illustrated in 2.3 where a single target can be classified with the data provided by the signal processing. Each target will have a **range**, **azimuth angle** θ (angle between x-axis and y-axis), **elevation angle** φ (angle between z-axis and y-axis), **radial velocity** v and a **SNR**. Although in the image the azimuth angle is portrayed as φ , in this dissertation work, the selected symbol for azimuth angle is θ and vice versa.

2.2 FMCW Radar

In this section, we will approach how an FMCW radar can locate and detect a target using an FMCW Radar type. FMCW stands for **Frequency Modulated Continuous Wave** and is a subset of the continuous waveform type of system. There are other types of waveforms such as FSCW (Frequency Shifting Continuous Wave) or SFCW (Stepped Frequency Continuous Wave) but in this dissertation, the focus will be only on FMCW. The FMCW Radar consists of transmitting a wave with a frequency that changes over time, in this case, we will approach using a linearly increasing frequency [10]. Figure 2.4 shows the transmitted wave in an Amplitude-Time plot where it is possible to see the frequency increasing with time and in figure 2.5 the respective frequency modulation also known as **Chirps**, where S is the rate of change or **Slope**, B is the **Bandwidth** and f_0 the initial frequency. Each signal transmitted is separated by T_C which is the time between **Chirps**. From this, it is possible to define the slope of the chirp as:

$$S = \frac{B}{T_C} \quad (2.1)$$

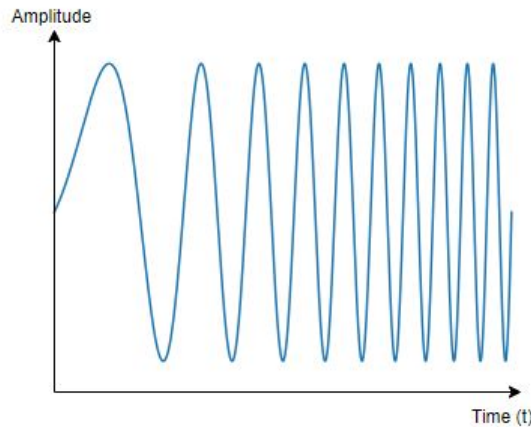


Figure 2.4: Transmitted signal in the time domain.

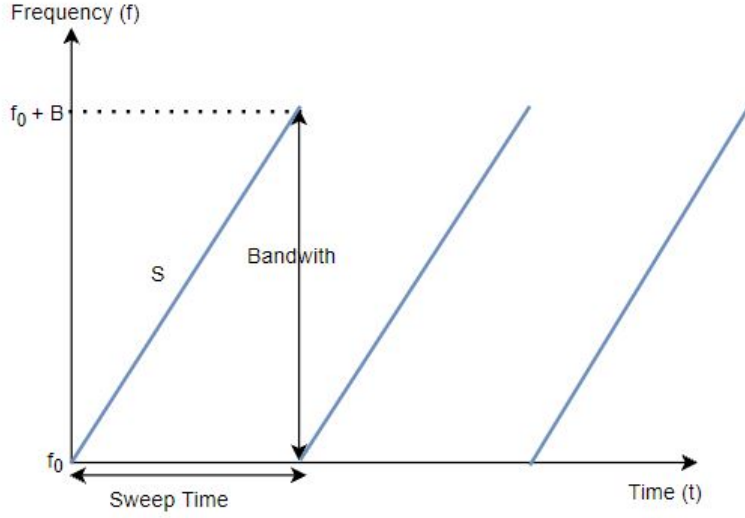


Figure 2.5: Chirp signal.

2.2.1 Range Estimation

When a transmitted wave reflects on an object the echo will eventually arrive at the radar and be captured by the receiving antennas after a short delay proportional to the traveling distance. To understand how the FMCW Radar measures the distance to the target, it is necessary to look back to 2.2 and analyze the IF signal which is the output of the mixer block. If the input signals are two sine waves $\sin(\omega_1 t + \phi_1)$ and $\sin(\omega_2 t + \phi_2)$ then the output of a mixer is:

$$IF(t) = \sin [(\omega_1 - \omega_2) t + (\phi_1 - \phi_2)] \quad (2.2)$$

This means the IF signal frequency is proportional to the frequency difference between Tx and Rx. Because there is a low pass filter after the mixer the second part of the equation will be filtered thus no high-frequency components will be considered at the signal processing level.

The figure 2.6 depicts the transmitted and received signals (top right and left) and the respective IF signals (bottom right and left) with a single reflection (left side) and multiple reflections (right side). f_b is the beat frequency, τ is the flight time, f_i is the starting frequency and f_t is the frequency after τ time. Adapted from [10].

If d is the distance between the target and the radar, the time of flight, τ , can be calculated as:

$$\tau = \frac{2d}{c} \quad (2.3)$$

where c is the wave traveling speed which can be approximated as the speed of light.

Considering the beat frequency, f_b , is the frequency difference between the frequency at τ time, f_t , and the initial frequency, f_i , the result of f_t can be calculated using the following equation:

$$f_t = f_i + S \cdot \tau \quad (2.4)$$

and thus, considering 2.3 it is possible to rewrite as:

$$f_t = f_i + S \frac{2d}{c} \quad (2.5)$$

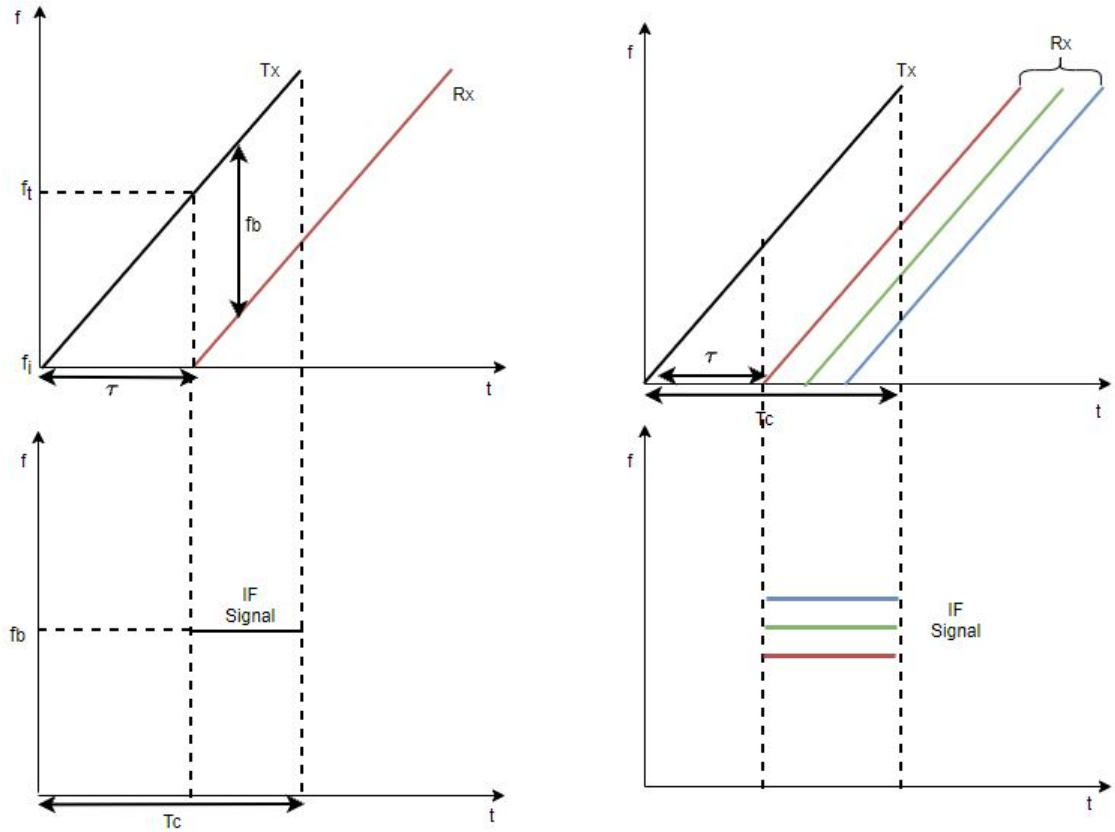


Figure 2.6: Transmitted and received signals with respective IF signals for a single target (left side) and multiple targets (right side).

The IF signal frequency beat is then:

$$f_b = f_t - f_i = S \frac{2d}{c} \quad (2.6)$$

By rearranging 2.6 we can describe the distance between the radar and target as:

$$d = f_b \frac{c}{2S} \quad (2.7)$$

The distance of a target is proportional to the beat frequencies present in the IF signal whereas the **Fourier Transform** is a mathematical transform that decomposes a time-domain function into a frequency domain function and the **Fast Fourier Transform (FFT)** is an implementation of the above [11]. In Radar literature, this is also called **1D-FFT** or **range-FFT**.

The figure 2.7 represents how the data is stored for further processing in a form of a matrix. The top image is a frequency-time plot of the transmitted signals. The bottom table is a visualization of how the data is stored, where the IF signal FFT from each chirp is stored as a column. A square from the table represents a frequency bin from the FFT which can later be converted to the range. A row represents the same FFT bin across multiple chirps. Each color is a peak in the FFT and from the analysis of the table, we can see that we have three or more targets across all the chirps. Adapted from [10].



Figure 2.7: The 1D-FFT or range-FFT.

2.2.2 Range Resolution

Range resolution is the measurement ability to separate two very close targets in the spatial domain. The 1D-FFT of two close objects will result in also very close IF frequencies causing the FFT to only present one peak if the observation window is small as seen in figure 2.8. To resolve this issue, a bigger observation time is required.

Considering the definitions of beat frequency 2.6 and the slope of the chirp 2.1, and that two frequencies can be resolved as long as their difference (Δf_c) is greater than $1/T_c$, the range resolution (Δd) can be defined as:

$$\begin{aligned}
 \Delta f_c &> \frac{1}{T_c} \Leftrightarrow \\
 \frac{2S\Delta d}{c} &> \frac{1}{T_c} \Leftrightarrow \\
 \Delta d &> \frac{c}{2ST_C} \Leftrightarrow \\
 \Delta d &= \frac{c}{2B}
 \end{aligned} \tag{2.8}$$

which is inversely proportional to the bandwidth (B) of the chirp transmitted.



Figure 2.8: The amplitude-time plot (left) of the IF signals of two targets at very close distance. At the right plot is the FFT of the IF signals with a small observation time window with just N samples. Adapted from [10].

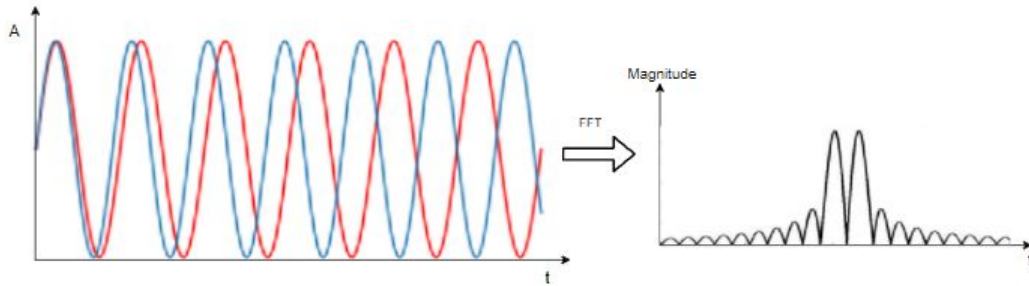


Figure 2.9: The amplitude-time plot (top) of the IF signals at very close distance where the frequency difference is really low. The right side is the FFT of the IF signals, but this time, with a large observation time window that allows $2N$ samples. Adapted from [10].

2.2.3 Maximum Measurable Distance

Since the distance of a target is directly proportional to the frequency of the IF signal, then the maximum distance possible to be detected is the maximum frequency that can be processed. The Radar system architecture has an ADC and, thus the maximum distance will be limited by the **Sampling Rate** (f_s). Then the maximum distance will be:

$$d_{max} = \frac{f_s c}{2S} \quad (2.9)$$

2.2.4 Velocity Estimation

To measure the velocity of a target it is required to have a new degree of freedom which is the time domain, by correlating the IF signals information from multiple chirps separated in time it is possible to measure displacements on the target distance. Because in a real-world scenario an FMCW would be working with time separations in transmitted chirps as small as milliseconds, a human would not have enough time to move in the range domain, so the peak frequency detected in the IF signals across two transmitted chirps would be the same, and thus, makes it difficult to measure the amount of distance variation in the designated time period.

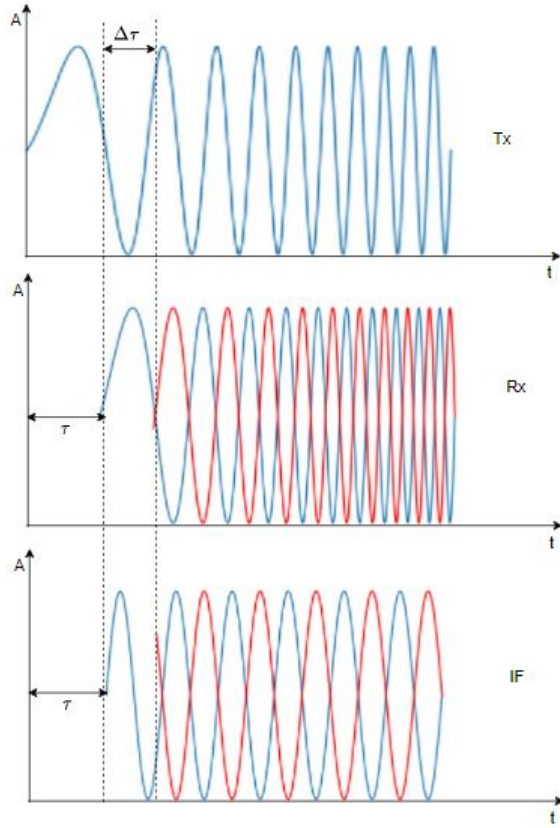


Figure 2.10: The amplitude-time plots of different signals from two time distinct time slots. Adapted from [10].

In figure 2.10, at the top is the transmitted signal, in the middle is the respective reflected signal after a flight time τ (blue) and the reflected signal from a second chirp with a small variation in the flight time $\Delta\tau$ (red). The bottom plot is the respective IF signals with the same beat frequency. If a target has a relative velocity to the radar bigger than zero, a small displacement will appear in the form of a phase difference between the first received signal and the second. Considering the small flight time displacement $\Delta\tau$, the phase difference between the IF signals from two consecutive chirps is:

$$\Delta\Phi = 2\pi f_b \Delta\tau = \frac{4\pi\Delta d}{\lambda} \quad (2.10)$$

Now, since the time separation between chirps is known as T_c and the phase difference can be used to estimate the distance variation of a target and knowing that the velocity can be calculated as $v = \frac{\Delta d}{T_c}$, by replacing in 2.10 and rearranging the expression, the velocity of the target is:

$$v = \frac{\lambda\Delta\Phi}{4\pi T_c} \quad (2.11)$$

A target at a determined distance produces an IF signal with a constant frequency and phase, but if the target is moving relative to the radar, the IF signal phase from a second chirp will be different while maintaining the same beat frequency. It is possible to represent the phase

difference in a form of a phasor in the exponential form as seen in figure 2.11 where each plot represents the FFT on the IF signal from different chirps.

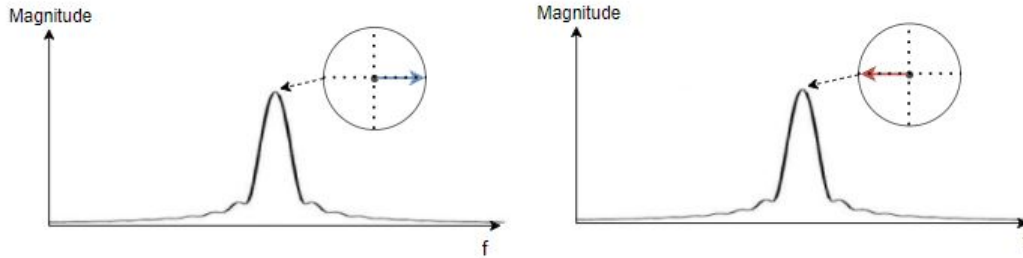


Figure 2.11: The FFT of the IF signals. Adapted from [10].

In the figure 2.11, at the left is the FFT of the blue signal from the figure 2.10 and in the right plot is the respective FFT from the red signal with a flight time variation $\Delta\tau$. The circles with the vectors are a visual representation of the phase difference between the beat signal and the transmitted signal in the exponential form $Ae^{j\omega}$.

Considering the phasor as a form of a discrete signal rotating with **Angular Speed** (ω), an FFT on these samples will show a peak at the respective ω . This is called **2D-FFT** or **Doppler-FFT** which is the FFT across the sequence of phasors. In 2.12 a target with constant velocity is considered and then, the phase difference is measured from the sequence of chirps transmitted and a 2D-FFT is applied and the result will be a constant **Angular Velocity** at ω . The angular velocity measured in radians per sample can be replaced in 2.11 and a velocity can be calculated from the results of 2D-FFT as:

$$v = \frac{\lambda\Delta\omega}{4\pi T_c} \quad (2.12)$$

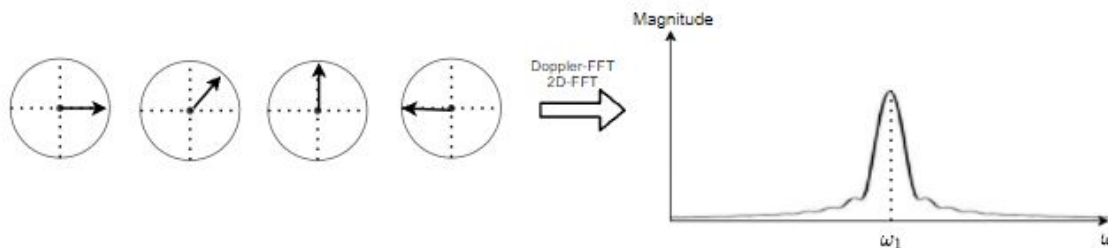


Figure 2.12: At the left side is phase difference in the exponential form at multiple chirp indexes and in the right, the 2D-FFT across the phase differences with peak at the at the angular velocity ω_1 . Adapted from [10].

To better understand how the FMCW radar measures velocity in a multiple target environment, the following scenario is purposed. Two targets at the exact same distance from the radar are moving with different velocities in relation to the radar. Since they are at the same distance the beat frequency of the IF signal will be the same, but this time, the phase

difference will be the sum of two phasors each one caused by each target. A 2D-FFT is able to resolve both target velocities by detecting a peak at two distinct angular velocities which are proportional to the target velocity. The following image (figure 2.13) depicts the above scenario explained. Figure 2.13 is a case scenario where the box on the left side is the relative

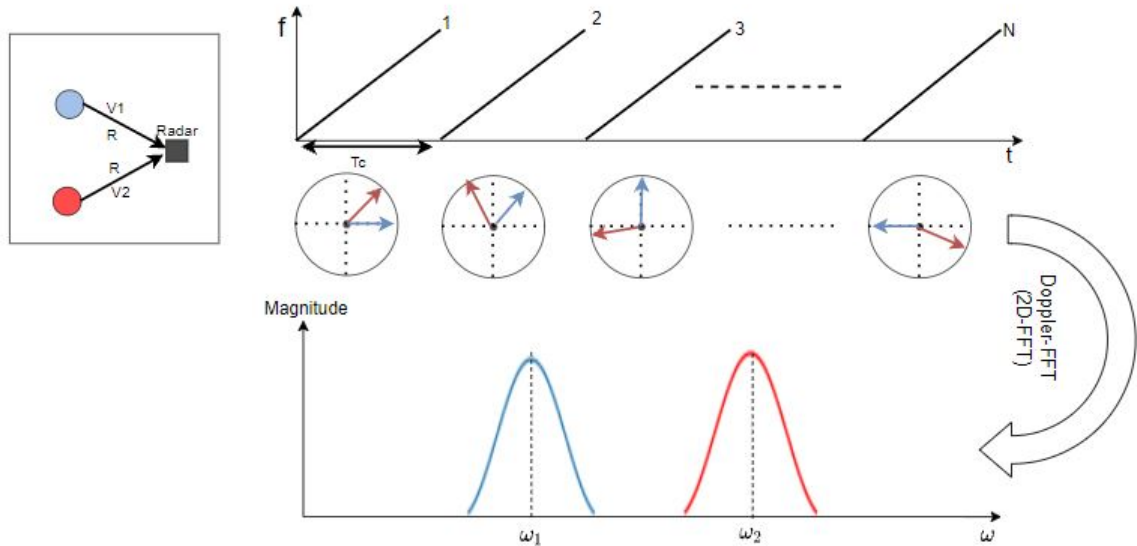


Figure 2.13: Velocity scenario of two targets at the same distance from the radar with different velocities. Adapted from [10].

position of two targets with different speeds, but at the same distance from the radar. At the top is the frequency-time plot of N transmitted chirps separated by T_C time. In the middle we have the visual representation of the phase difference in the exponential form, the blue vector corresponds to the reflection from the blue target and the red vector corresponds to the red target. The last plot is the 2D-FFT from the sum of both phasors with peaks at the respective angular velocities.

The results of the 2D-FFT will be stored as rows in the data matrix where each row represent the velocity measured across multiple transmitted chirps as seen in figure 2.14. In the literature this is called a **Range-Doppler-Heatmap**.

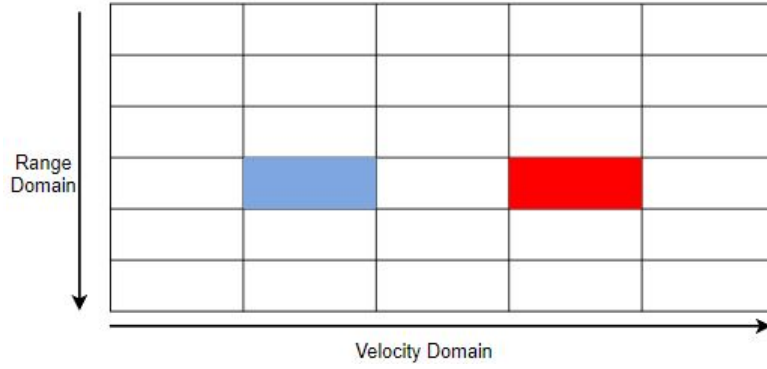


Figure 2.14: The results of the 2D-FFT where we can separate both targets in the range and velocity domain. Each colored square represent the objects from 2.13. Adapted from [10].

2.2.5 Velocity Resolution

The velocity resolution is the ability to separate different peaks at the 2D-FFT. In order to do that, the difference between the two peaks $|\omega_1 - \omega_2|$, $\Delta\omega$, must be bigger than $\frac{2\pi}{N}$. Considering the equations $\Delta\Phi = \Delta\omega T_c$, the velocity resolution can be estimated as the following:

$$\begin{aligned} \Delta\omega &= \frac{2\pi}{N} F_s \\ \Leftrightarrow \Delta\Phi &> \frac{2\pi T_c}{T_c N} \Leftrightarrow \Delta\Phi > \frac{2\pi}{N} \end{aligned} \quad (2.13)$$

Knowing that phase difference is proportional to the velocity of the target as seen in equation 2.11:

$$\frac{4\pi v T_c}{\lambda} > \frac{2\pi}{N} \Leftrightarrow \quad (2.14)$$

$$v > \frac{\lambda}{2NT_c} \Rightarrow v_{res} = \frac{\lambda}{2T_f} \quad (2.15)$$

where T_f is the frame time and is the number of chirps considered multiplied by the chirp time ($T_f = NT_c$).

2.2.6 Maximum Measurable Velocity

The maximum velocity can be extracted from the fact that if the absolute value of the phase difference is bigger than 180° (π radians) then the measurement of the velocity becomes ambiguous because the variation on the angular velocity does not give information on how the phase difference is evolving through each sample. If the limit in phase difference is π radians, then the maximum velocity can be defined as:

$$\Delta\Phi < \pi \quad (2.16)$$

$$v < \frac{\lambda}{4T_c} \Rightarrow v_{max} = \frac{\lambda}{4T_c} \quad (2.17)$$

This shows that the maximum velocity is inversely proportional to the chirp periodicity.

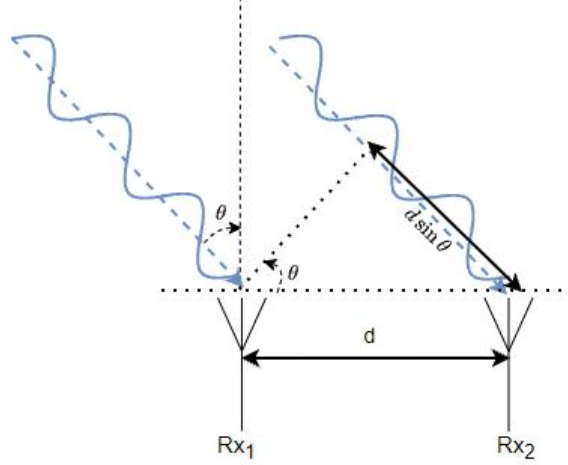


Figure 2.15: A single target AoA estimation with the trigonometry relation between the two received waves. The reflection is originated from a target at θ degrees, and the antennas are separated by a distance d .

2.2.7 Angle Estimation

To be able to discriminate two targets at the same distance and with the same relative speed to the radar, it is necessary to increase a degree of freedom in the spatial domain to be able to compare the received reflection from physically separated points. In other words, for an FMCW Radar to be able to measure the **Angle of Arrival** of a reflection it is necessary to have two or more receiving antennas. Figure 2.15 shows the difference between the received waves from two physically separated antennas originated by a single reflection. In this case, it is a **Single Input Multiple Output** system with one transmitter and two receivers. Assuming the additional distance the wave received by antenna Rx_2 has to travel in relation to the first antenna is represented by (Δd) and that its value is equal to $d \sin(\theta)$. The phase difference between the signals received by the two antennas can be expressed as:

$$\Delta\Phi = \frac{2\pi\Delta d}{\lambda} = \frac{2\pi d \sin(\theta)}{\lambda} \quad (2.18)$$

Then it is possible to rearrange to calculate the AoA value θ :

$$\theta = \sin^{-1}\left(\frac{\lambda\Delta\Phi}{2\pi d}\right) \quad (2.19)$$

This shows the direction from which the reflection originated is measured through the phase difference, just like the velocity measurement. To resolve multiple targets in the angle domain, the same approach as velocity is made, each peak detected at the 2D-FFT has a respective phasor that varies with the receiving antenna. A **3D-FFT** or **Angle-FFT** is made across the multiple receiving antennas and can calculate the AoA.

The data is stored in a form of a cube, where a 2D-heatmap (2.14) is created for each received antenna. As seen in figure 2.16.

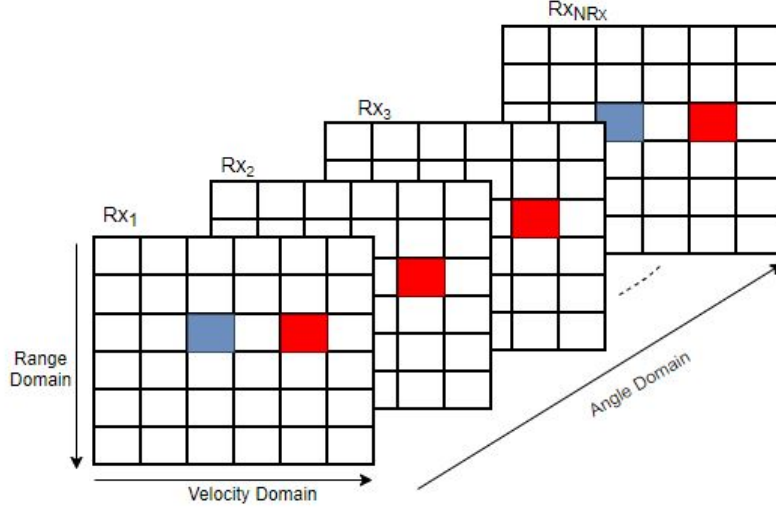


Figure 2.16: The 3D-FFT across the phase difference from different receiving antennas to resolve targets in the angular domain.

2.2.8 Angle Resolution

Angle resolution is the minimum angular separation between two targets to appear as two separate peaks in the 3D-FFT, to estimate the angular resolution the same approach of the velocity resolution is used. When the angle of two targets are θ and $\theta + \Delta\theta$ and knowing the phase difference must be bigger than $\frac{2\pi}{N_{Rx}}$, then, the angular resolution can be formulated as:

$$\Delta\Phi = \frac{2\pi d}{\lambda} \cdot (\sin(\theta + \Delta\theta) - \sin\theta) > \frac{2\pi}{N_{Rx}} \quad (2.20)$$

$$\frac{2\pi d}{\lambda} \cdot \cos(\theta) \Delta\theta > \frac{2\pi}{N_{Rx}} \quad (2.21)$$

$$\Delta\theta > \frac{\lambda}{N_{Rx} d \cos\theta} \quad (2.22)$$

$$\rightarrow \theta_{res} = \frac{\lambda}{N_{Rx} d \cos\theta} \quad (2.23)$$

Since in most FMCW Radar antenna designs the distance between antennas d is usually $\lambda/2$, the expression can also appear in the form of:

$$\theta_{res} = \frac{2}{N_{Rx} \cos\theta} \quad (2.24)$$

2.2.9 Angular Field of View

The maximum angle aperture can also be estimated by the same criteria of the maximum velocity. The phase difference must be lesser than π for the result to be unambiguous.

Therefore the θ_{max} is:

$$\Delta\Phi = \frac{2\pi d}{\lambda} \cdot (\sin(\theta + \Delta\theta) - \sin\theta) < \pi \quad (2.25)$$

$$\theta < \sin^{-1}\left(\frac{\lambda}{2d}\right) \quad (2.26)$$

$$\rightarrow \theta_{max} = \sin^{-1}\left(\frac{\lambda}{2d}\right) \quad (2.27)$$

In the same way of the angular resolution, a separation between receivers of $\lambda/2$ results in a **Field of View** of $\pm 90^\circ$.

2.2.10 Summary of Radar Parameters

The following table 2.1 represents a summary of the above calculations done previously.

Range	$d = f_b \frac{c}{2S}$	(m)
Range resolution	$\Delta d = \frac{c}{2B}$	(m)
Maximum range	$d_{max} = \frac{f_s c}{2S}$	(m)
Velocity	$v = \frac{\lambda \Delta\Phi}{4\pi T_c}$	(m/s)
Velocity resolution	$v_{res} = \frac{\lambda}{2T_f}$	(m/s)
Maximum velocity	$v_{max} = \frac{\lambda}{4T_c}$	(m/s)
Angle of Arrival	$\theta = \sin^{-1}\left(\frac{\lambda \Delta\Phi}{2\pi d_{Rx}}\right)$	(degrees)
Angle of Arrival resolution	$\theta_{res} = \frac{\lambda}{N_{Rx} d_{Rx} \cos\theta}$	(degrees)
Field of View	$\theta_{max} = \sin^{-1}\left(\frac{\lambda}{2d_{Rx}}\right)$	(degrees)

Table 2.1: Summary of target estimation parameters.

2.3 MIMO Radar

Multiple-Input Multiple-Output (MIMO) radar is a type of radar system that uses multiple transmitter and receiver antennas. Each antenna transmits independently and orthogonal signals from the other antennas, so that the receiver can separate from where the reflected signal originated. This will allow defining a virtual array which contains all possible to be created, independent channels containing information from each transmitter to each receiver, thus if the number of Tx is N_{Tx} and the number of Rx is N_{Rx} , it is possible to create $N_{Tx} \cdot N_{Rx}$ independent pairs. A MIMO radar will provide a few advantages such as decreased angular resolution, increased FoV and higher sensitivity [12][13].

2.3.1 MIMO Model

As stated before, the transmitted signals should be orthogonal. This can be obtained by using **time-division multiplexing** (TDM) or **frequency-division multiplexing** (FDM) or spatially encoded. From this point forward the TDM modulation will be considered because it is the most simple form of achieving orthogonality by allowing only one transmitter to

transmit at each time. If we consider N number of transmitting antennas and M the number of receiving antennas, the signal received at the m^{th} receiver antenna can be defined as:

$$x_m(t) = \sum_{n=1}^N s_n(t) e^{j \cdot 2 \cdot \pi \cdot \tau_n \cdot (m-1)} + n_m(t) \quad (2.28)$$

where $s_n(t)$ is the signal from the n^{th} transmitter antenna and $\tau_n = \frac{d_{Rx} \cos \theta_n}{\lambda}$ which is phase shift per antenna as seen in figure 2.15 and equation 2.10 and $n_m(t)$ are the noise terms. The antenna array in the matrix form can be seen as $X = AS + N$ with X being the matrix of the received signals, S the transmitted signals, N the noise terms and A is the steering vector matrix containing the phase delays received at Rx originated from all transmitter antennas.

$$\begin{bmatrix} x_1(t)^T \\ x_2(t)^T \\ \vdots \\ x_M(t)^T \end{bmatrix} = [a(\theta_1) \quad a(\theta_2) \quad \dots \quad a(\theta_N)] \begin{bmatrix} s_1(t)^T \\ s_2(t)^T \\ \vdots \\ s_N(t)^T \end{bmatrix} + \begin{bmatrix} n_1(t)^T \\ n_2(t)^T \\ \vdots \\ n_M(t)^T \end{bmatrix} \quad (2.29)$$

with

$$a(\theta) = [1 \quad e^{j2\pi\tau(\theta)} \quad e^{j2\pi\tau(\theta) \cdot 2} \quad \dots \quad e^{j2\pi\tau(\theta)(M-1)}]^T \quad (2.30)$$

The steering vector represents the phase delays experienced by each wave that is received in relation to the first one to the wave received by the first antenna.

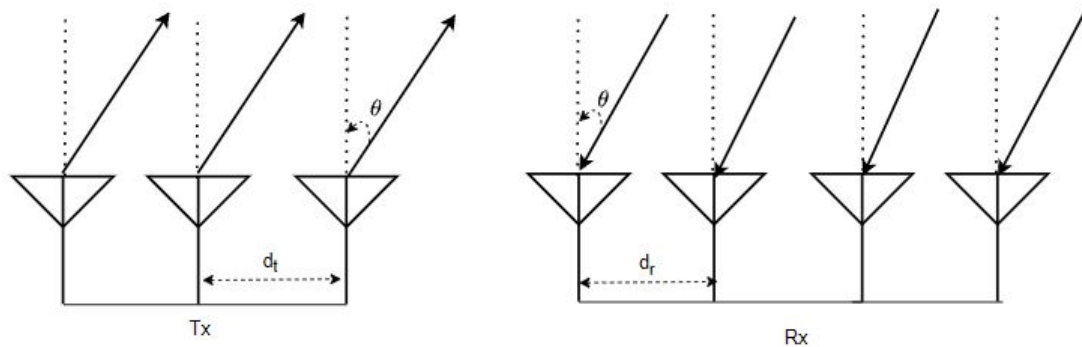


Figure 2.17: TX and RX antennas of a MIMO Radar.

To understand how the radar identifies the AoA in multiple target environments it is required to apply some beamforming techniques. These techniques could be applied on the physical domain like rotating the Radar or applying phase shifts to the transmitted signals or could be virtually applied by creating radiation patterns on the receiver antennas. These patterns could be achieved by combining the signals received with different weights, $Y = w^H X$, where this w matrix could magnify the signals in a specific direction while suppressing the other directions. The latter technique is widely used in wireless communications. A simple algorithm based on the beamforming techniques would be to adjust the weight vector to

rotate across the FoV, measure the respective power received after the weighting at each angle inside the FoV, and then plot the power of the received signal in function of the beamforming angle θ .

2.3.2 Bartlett Beamforming

The Bartlett beamforming algorithm is normally applied on radar systems where the receiver antenna works as uniform linear arrays. This algorithm essentially magnifies the signals from a certain direction by compensating the phase shift resulting in signals from a particular direction to sum constructively while from the unwanted directions to sum destructively. If a signal $s(t)$ from a target arrived with AoA θ_0 , then the signal at m^{th} antenna would be $x_m(t) = s(t) \cdot e^{j \cdot 2\pi \cdot \tau(\theta_0)(m-1)}$ and the weight vector in Bartlett beamforming for the m^{th} antenna is $e^{j \cdot 2\pi \cdot \tau(\theta)(m-1)}$. Since the received signal is the product of the transmitted signal with the weight vector, $Y = w^H X = \sum w_m^* x_m(t)^T$, when $\theta = \theta_0$ the result is maximized. For Radar systems, the weight vector is the steering vector defined in 2.30 so $w = a(\theta)$. The power received at a angle θ is then [14]:

$$P(\theta) = Y \cdot Y^H = (w^H X) \cdot (w^H X)^H = w^H X X^H w = w^H R_{xx} w = a^H(\theta) R_{XX} a(\theta) \quad (2.31)$$

with R_{XX} being the covariance matrix of the received signal which by definition is $R_{XX} = E \{x(t) \cdot x^H(t)\}$. The results of this function $P(\theta)$ will have a maximum at the direction of the AoA. This algorithm works well when there is only a target because it has very low resolution [15].

2.3.3 Capon Beamforming

Also known as **Minimum Variance Distortionless Response** (MVDR) is an upgrade from the previous algorithm proposed by Capon [16]. The key idea is to maintain the signal from the desired direction by forming a beam in the look direction and suppressing it in other directions by minimizing the power contributed by noise and signals coming from other directions [17]. The solution is to find the weighting vector that minimizes the output power while maintaining a unit gain, mathematically

$$\min \{P\} = \min \{Y \cdot Y^H\} = \min \{w^H R_{XX} w\} \quad (2.32)$$

$$\text{subject to } w^H a(\theta) = 1 \quad (2.33)$$

The solution to the above minimization problem is the weighting vector

$$w = \frac{R_{XX}^{-1} a^H(\theta)}{a(\theta) R_{XX}^{-1} a^H(\theta)} \quad (2.34)$$

Then, the received signal power at angle θ

$$P(\theta) = Y \cdot Y^H = w^H X X^H w = w^H R_{xx} w = \frac{1}{a(\theta) R_{XX}^{-1} a^H(\theta)} \quad (2.35)$$

The resolution is enhanced compared to Bartlett method.

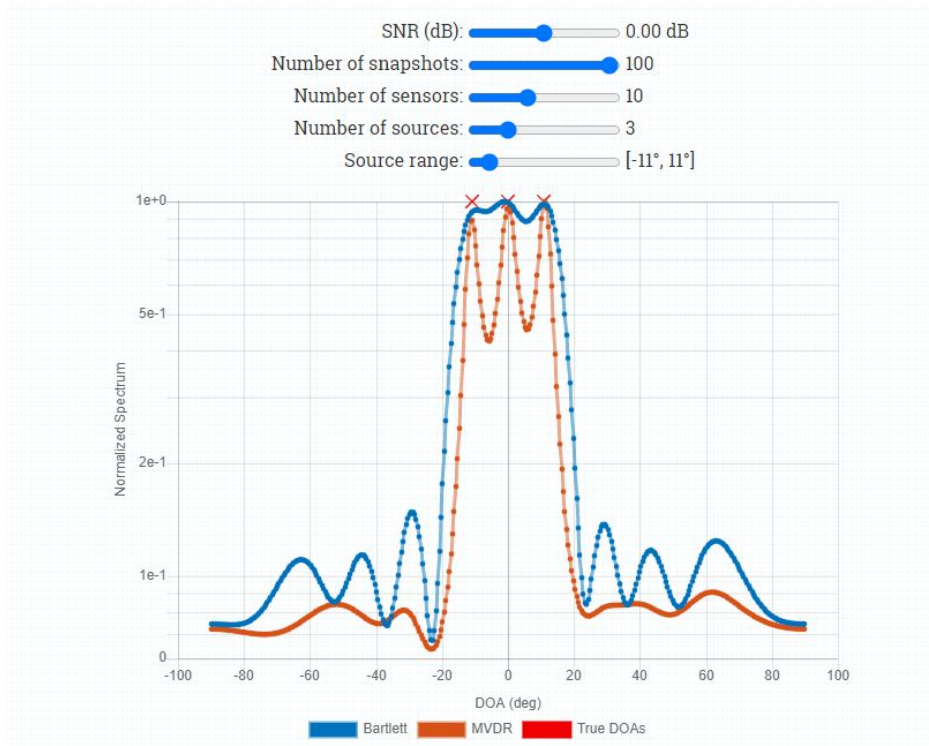


Figure 2.18: Capon (orange) and Bartlett (blue) beamforming with three close targets at $[-10,0,10]$ degrees. The plot is the power-angle spectrum Capon and Bartlett beamforming, maximums of this function indicate that there is a reflection coming from that angle [18].

In figure 2.18 the Bartlett beamforming cannot resolve the three targets because the resolution for this algorithm is high, which is why this beamforming is only suited for single target AoA estimation. On the other hand, the Capon beamforming can resolve both three targets. If the targets are separated by a large angle between them (figure 2.19), then the Bartlett beamforming is able to resolve the AoA, as well as Capon beamforming which has been seen before of being able to resolve much smaller angles between targets.

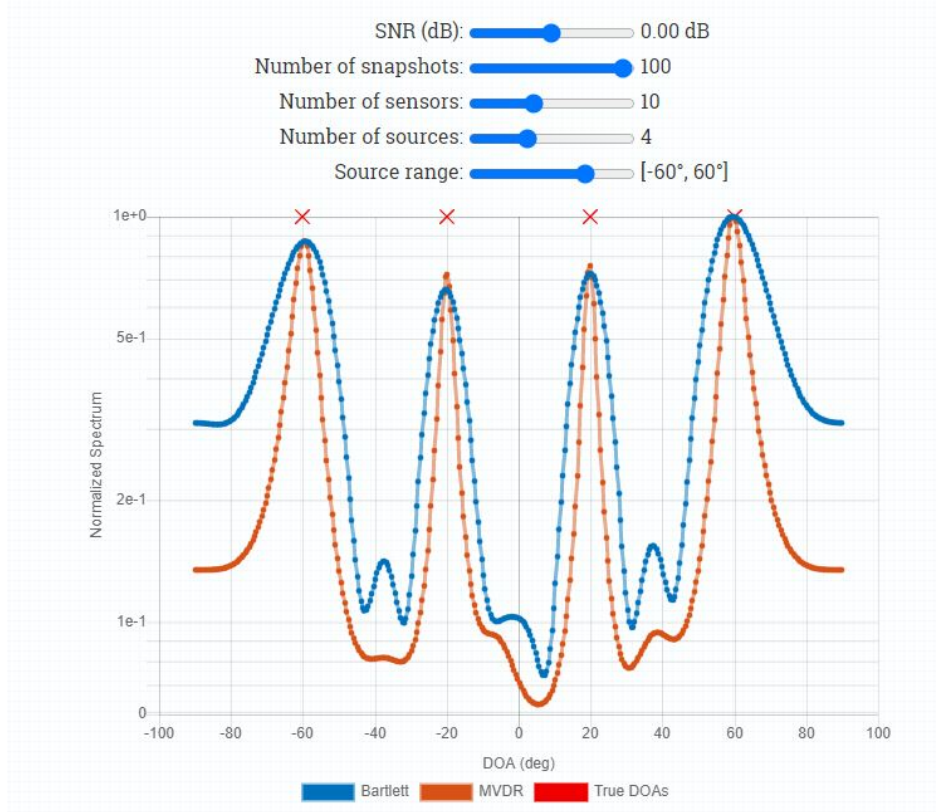


Figure 2.19: Capon (orange) and Bartlett (blue) beamforming, but this time with four targets with a large separation between them [18].

2.4 Signal Characterization

In the previous sections, we have analyzed target characteristics estimation based on the behavior of the signals present in the radar system, but in order to do so, it is required to have some metrics in regard to the quality of the signals.

2.4.1 SNR

The **Signal-to-Noise-Ratio** is a measure of confidence in the information to be considered by comparing the power of the received signal to the power noise level. This relation can be defined as

$$SNR = \frac{P_{Signal}}{P_{Noise}} \quad (2.36)$$

where the P_{Signal} is the power received by the radar from a reflection on a target and P_{Noise} is the power of the unwanted power. The unwanted power can come from various origins where the two most important ones are the clutter and the thermal noise. The thermal noise can be described as a random signal with equal intensity across all frequencies while the clutter can be divided into sidelobe clutter and mainlobe clutter [19]. Mainlobe clutter appears when a high reflectivity surface such as the ground or a metal structure is within the radar FoV and the sidelobe clutter comes from reflection outside the mainlobe and these are usually not significant because of the directivity of the antenna usually attenuates these components. To

define the SNR in a radar system it is important to consider the radar range equation. If P_t is the transmitted power and P_r the received [10]

$$\frac{P_r}{P_t} = \sigma \frac{G^2}{4\pi} \left[\frac{\lambda}{4\pi d^2} \right]^2 = \frac{\sigma G^2 \lambda^2}{(4\pi)^3 d^4} \quad (2.37)$$

then the received power can be expressed as

$$P_r = \frac{\sigma P_t G^2 \lambda^2}{(4\pi)^3 d^4} \quad (2.38)$$

where σ is the **Radar Cross Section** (RCS), G the antenna gain and d the target distance. Considering P_{Noise} is modeled as thermal noise, then $P_{Noise} = k_B T \Delta f$ with k the Boltzmann constant, T is the temperature of the antenna and Δf as bandwidth. The SNR can then be rewritten as

$$SNR = \frac{P_r}{P_{Noise}} = \frac{P_r}{P_{k_B T \Delta f}} \quad (2.39)$$

then the SNR is

$$SNR = \frac{\sigma P_t G^2 \lambda^2}{(4\pi)^3 d^4 k_B T B} \quad (2.40)$$

2.4.2 Radar Cross Section

Represented by σ , the Radar Cross Section is a target parameter that describes how the energy is distributed at the target when the wave hits. This parameter depends on many factors like the object geometry, material, angle of incidence, frequency of operation and polarization of the transmitted wave [20]. In practice, some of the energy is absorbed by the target while the rest is scattered in multiple directions in a non-uniform way which can lead to difficulties in the calculation of the value, therefore, the RCS of a complex object is usually determined by measurements. By rearranging eq.2.40, the Radar Cross Section can be expressed as

$$\sigma = \frac{SNR \cdot (4\pi)^3 \cdot d^4 \cdot k_B \cdot T \cdot B}{P_t \cdot G^2 \cdot \lambda^2} \quad (2.41)$$

Since the RCS varies with different objects this can turn into useful information to the radar and open a new door of opportunities by exploring the discretization of different types of targets and classifying them according to its RCS value, because the variables in the expression 2.41 are known constants or could be estimated.

2.5 Constant False Alarm Rate

The detection of targets in real-life scenarios is a complicated task, because of the various form of unwanted noise which comes in unknown forms. To detect objects in such environments, some digital processing techniques can be applied to reduce the impact of noise. The **Constant False Alarm Rate** is a sort adaptive algorithm to detect objects in the background with noise, clutter and interference. The CFAR role is to read the post-processed signal received with multiple interferences and decide if is detection or not, by dynamically adjusting the detection threshold as a function of the sensed environment [21].

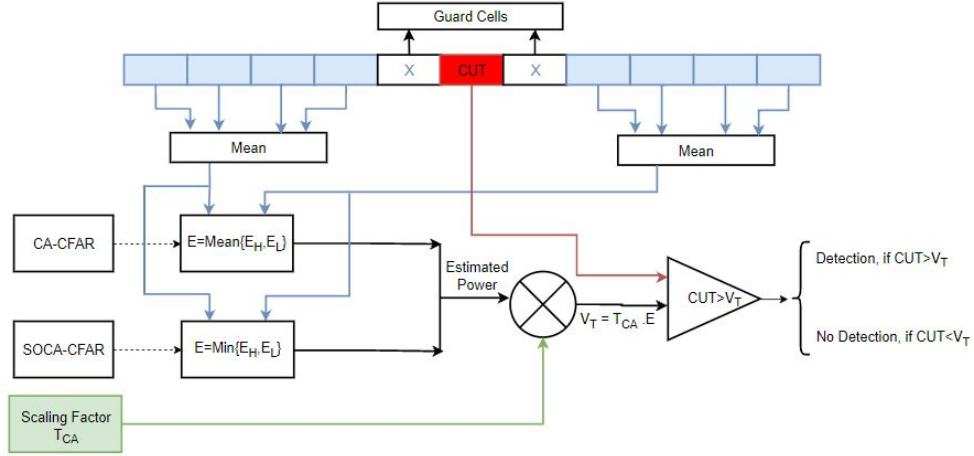


Figure 2.20: Architecture of the CFAR algorithm. At top there is the CFAR window where the red cell is the CUT, the "x" marked cells are the guard cells and the blue cells are the reference cells.

2.5.1 Cell Averaging CFAR

The **Cell Averaging CFAR** (CA-CFAR) is an implementation of CFAR where the detection threshold will be proportional to the cell average around a target. The CFAR algorithm is an iterative method that tests every cell in the range domain by looping across all the FFT bins.

As seen in figure 2.20 each iteration starts by designating a window with N cells around the **Cell Under Test** (CUT) that are included in the estimation of the interference power, these cells belonging to the window are referred to as **Reference cells** (Blue cells in image). Then, some cells that are adjacent to the CUT are discarded, these are called **Guard Cells** and are excluded from the window because, in a real-life case, a radar with a very small resolution detecting a human body would have multiple consecutive cells containing both reflected energy from the target and interference. The CA-CFAR algorithm will then compute the average power of the reference cells on the left (Lagging Window) and on the right (Lead Window) separately and then do the average of both averages done previously. To define the threshold to be compared, the arithmetic mean of reference cells is then multiplied by a scaling factor T_{CA} . This value is computed by knowing that the cells will have a random behavior and there is an analytical formula of the probability of the false alarm [21]. The T_{CA} affects the **Probability of False Alarm** (P_{FA}) as expressed in

$$P_{FA} = \left(1 + \frac{T_{CA}}{N}\right)^{-N} \quad (2.42)$$

Given a P_{FA} , the scaling factor will be:

$$T_{CA} = \left(P_{FA}^{-\frac{1}{N}} - 1\right) \quad (2.43)$$

The main limitations of the CA-CFAR are when there are multiple peaks/detections in the

reference cells where these detections will increase the threshold and therefore, the target under test might not be detected.

2.5.2 Smallest Of Cell Averaging CFAR

The limitations of CA-CFAR led to the development of new techniques that minimize the impact of these concerns. One of the developed techniques is **Smallest of Cell Averaging-CFAR** (SOCA-CFAR) where the purpose is to tackle the interference of targets close to the CUT. This approach is a small variation of CA-CFAR where the lead and lagging windows are averaged independently and then, the smallest of these averages is selected to be the threshold.

The next chapter will introduce the radar development kit used in this dissertation. A brief explanation of the hardware present and how the radar can compute in real-time some of the fundamental data, seen in this chapter. It will also be shown how the data is transferred between the radar and the PC.

Chapter 3

Texas Instruments IWR6843ISK-ODS Kit

This chapter purpose is to introduce the IWR6843 device and the application People Counting designed by Texas Instruments (TI). The IWR6843ISK-ODS (fig 3.1) is an evaluation board for the IWR6843 single-chip, Frequency Modulated Continuous Wave sensor capable of working in the 60 to 64 GHz band. This chip is built-in RFCMOS process with $45nm$ transistors which enables three transmitters and four receivers with phase shifters, as well as a programmable Digital Signal Processor (DSP) unit and a Master Sub-System (MSP)[22].

The People Counting application is a laboratory built by TI that outputs point clouds detected, with each point containing a polar coordinate, radial velocity and Signal-to-Noise-Ratio. This data is sent out through UART to PC.

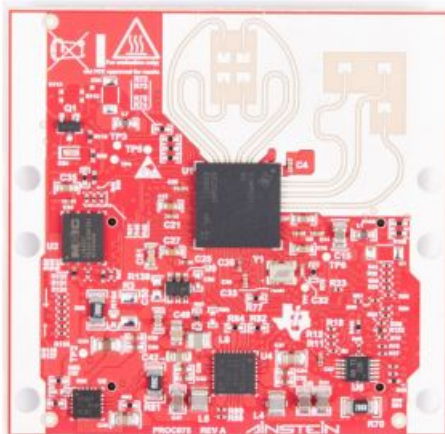


Figure 3.1: TI Mmwave FMCW Radar IWR6843ISK-ODS board. From [22].

3.1 Hardware Architecture

The RF and analog Sub-System (leftmost block in fig 3.2) includes the analog circuitry such as the transmitters and receiver antennas, a **Low-Noise Amplifier** (LNA), a **Power**

Amplifier (PA), a **Mixer**, **Phase Shifters**, a **Synthesizer**, and **ADC** capable of working in complex baseband.

The DSP Sub-System has a C674x DSP working at $600MHz$, a Hardware Accelerator module to off-load the burden of the FFT and Log-Magnitude computations, a high-bandwidth interconnect (128 bits @ 200MHz), and some peripherals such as DMA for data transfer and a memory block called radar data cube in which the FFT results and computations are stored [23].

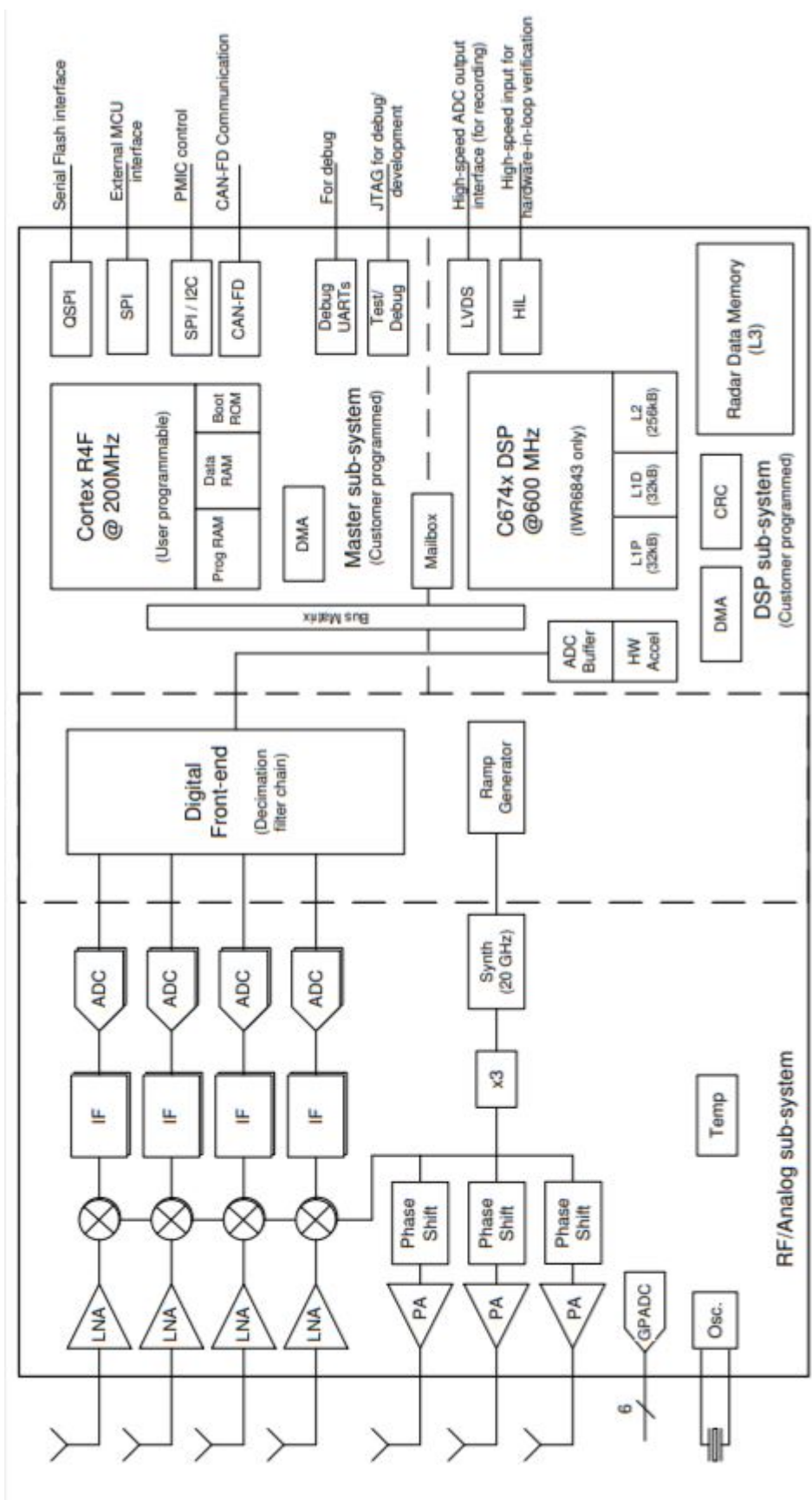


Figure 3.2: TI mmwave FMCW Radar IWR6843 functional block diagram. From [23].

3.1.1 Antennas

The IWR6843ISK-ODS has on-board short-range antennas for the four receivers and three transmitters. The arrangement of the antennas can be seen in figure 3.3a, which provides the same angular resolution in azimuth and elevation because the distance between the azimuth transmitters is the same as the elevation transmitters [22]. The figure 3.3b represents the

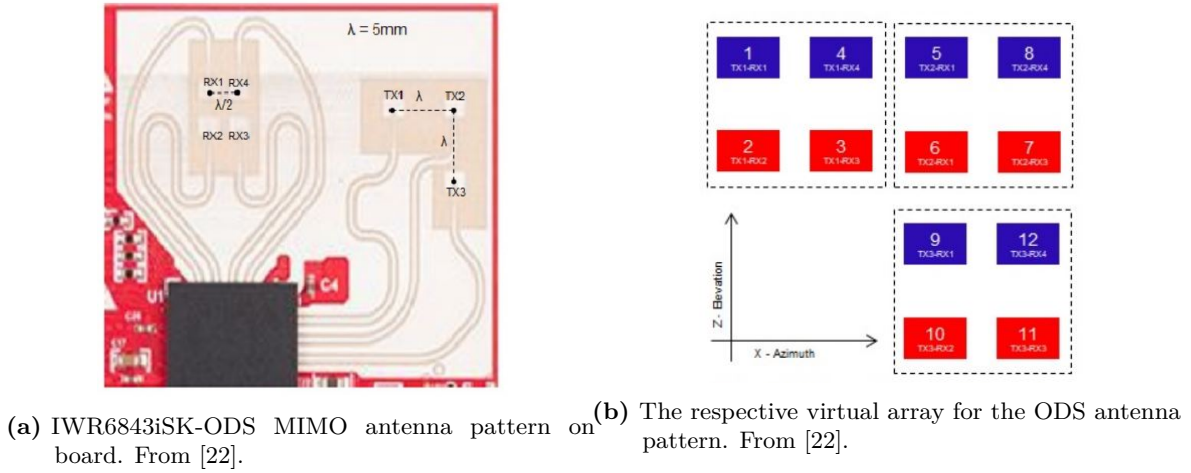


Figure 3.3: Antenna configuration and virtual array.

respective virtual array formed by the 4x3 antenna array. The blue antennas are 180° out of phase from the red ones because the microstrip lines have a longer path, so a software-induced phase shift on the receiver side must be applied.

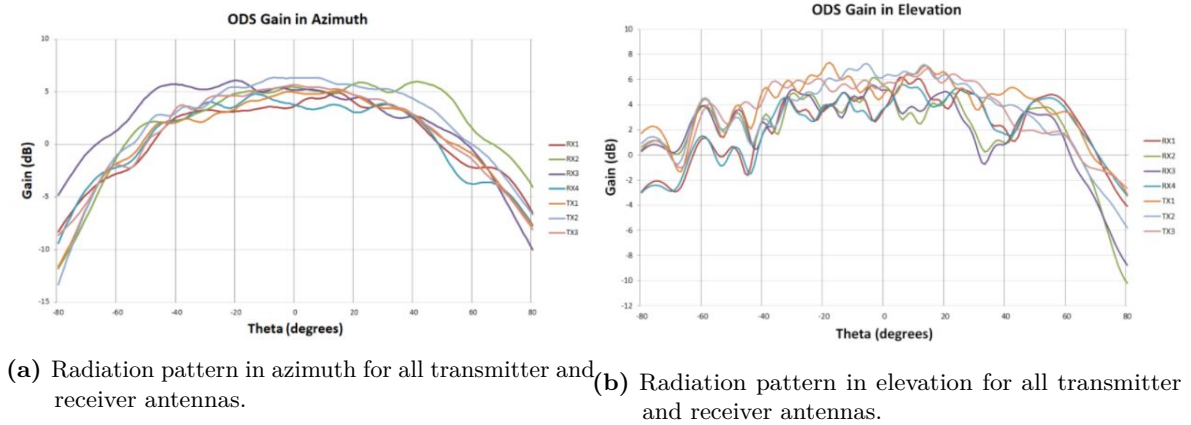


Figure 3.4: Radiations Patterns. From [22].

Figures (3.4a and 3.4b) show the respective radiation pattern measured in both azimuth and elevation plane.

3.1.2 Transmitter and Receiver System

Each transmitter channel is capable of amplitude control and has a 6-bit linear phase shifter that allows beamforming techniques. The output power of the transmission system

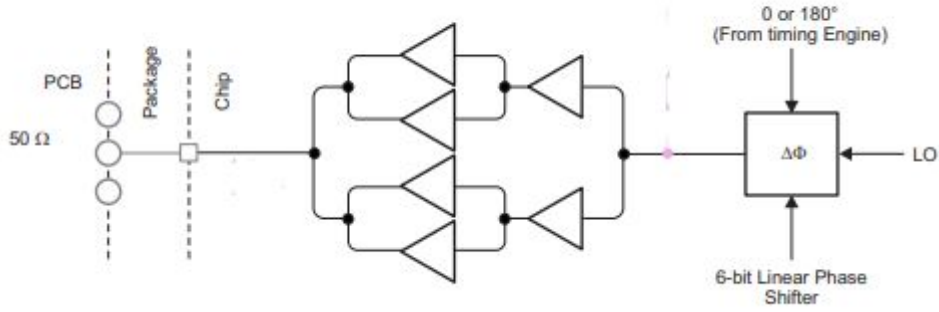


Figure 3.5: Transmitter Sub-System for each Tx antenna. From [23].

is 12dBm. The ramp generator for saw-tooth modulations is done through a block called timing engine which modulates the synthesizer to generate the custom waveforms at the local oscillator [23].

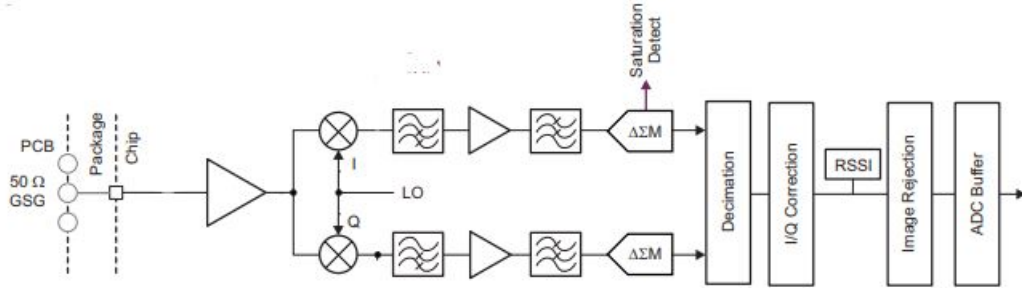


Figure 3.6: Receiver Sub-System for each Rx antenna. From [23].

The Receiver Sub-System consists of four receiver channels, where every single receiver consists of an LNA, mixer, IF filter, an ADC and a decimation filter [23]. The receiver channel has a noise figure of 12 dB between 60 to 64 GHz and a maximum gain of 48dB. The IF filter has a bandwidth of 10MHz with a user-defined lower cutoff frequency to filter the detections at a very close range and the ADC has a resolution of 12 bits with a user defined sampling rate capable of achieving 12.5 Msps in complex baseband [24]. The received signal is mixed with in-phase (I) and quadrature (Q) versions of LO, the complex-baseband architecture allows an improvement in the noise figure as much as 3dB compared to a single-sideband. The minimum sampling frequency f_s for the complex base-band ADC is $f_{b,max}$ while in real only sampling the minimum is $2f_{b,max}$ according to Nyquist theorem (fig 3.7).

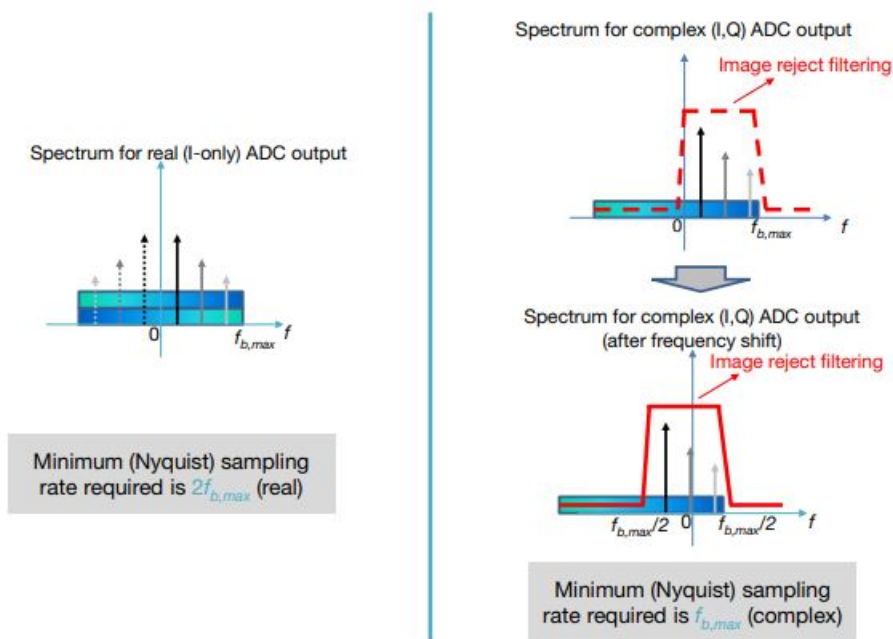


Figure 3.7: ADC spectrum for real and complex-baseband architectures where each arrow represents a reflection detected. From [24].

3.1.3 Chirp Generator

The Clock Sub-System works with an external clock of 40MHz and can generate frequencies between 60 to 64GHz [23]. It has a built-in oscillator followed by a fraction-N PLL circuit that generates the reference clock for the system and a synthesizer controlled by a timing engine that generates a 20GHz local oscillator which later on is processed by an X3 multiplier to create the required frequencies. The ramp modulation is done by the timing engine block.

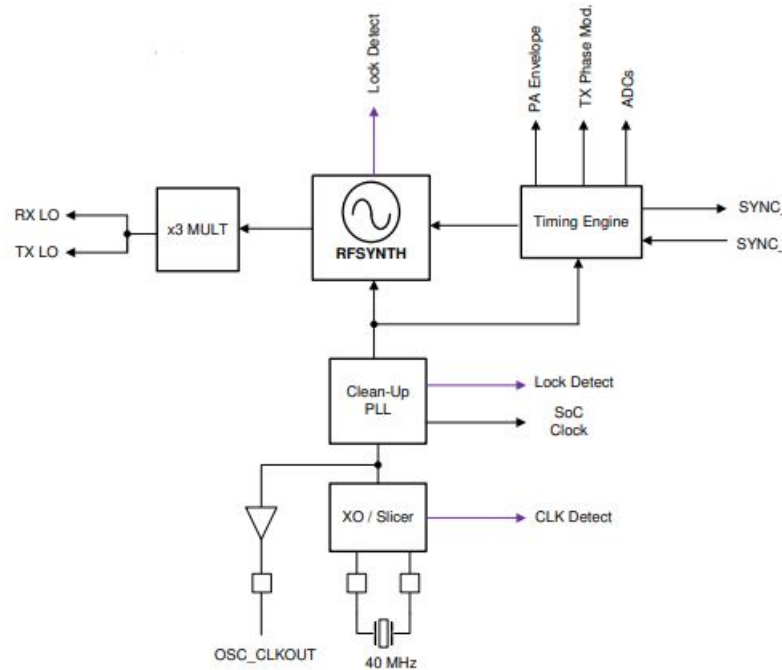


Figure 3.8: Clock system and Synthesizer for chirp generation. From [23].

3.1.4 DSP and MSS

The DSP Sub-System (left side of figure 3.9) is responsible for the computation necessary to obtain the point clouds from each frame. The Sub-System contains TI high-performance C674x DSP, a hardware accelerator, a high-speed interconnect and some peripherals such as DMAs for data transfer, ADC buffers, a Cyclic Redundant Check Engine and a data handshake memory.

The Main Sub-System (right side of figure 3.9) is the brain of the device and controls all the peripherals and activities. It contains the Cortex-R4F processor and associated peripherals such as I2C, UART, Clocking module, PWM and others connected to the Main interconnect.

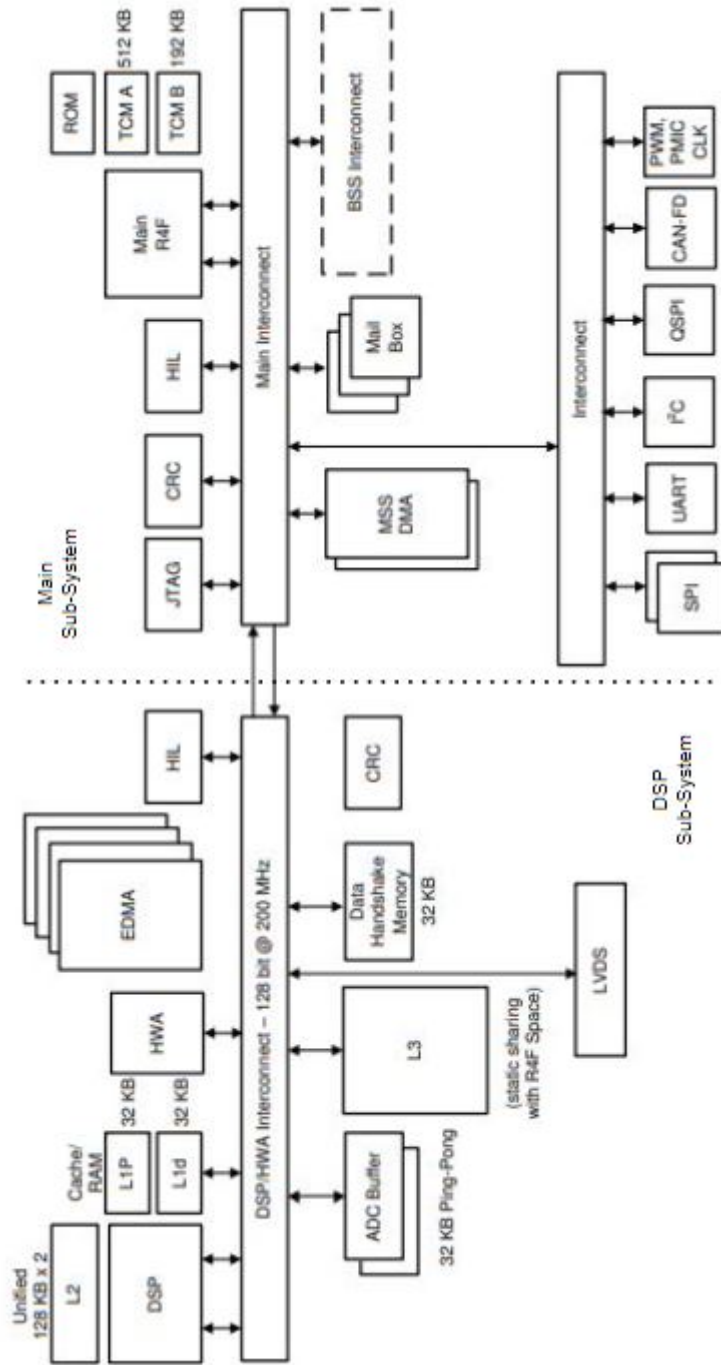


Figure 3.9: Processor system block diagram in IWR6843 devices. Left side is the DSP Sub-System and right side is the Main Sub-System. From [23].

3.2 People Counting Application Processing Chain

The people Counting application developed by TI is able to transmit a frame of chirps for each transmitter antenna, the signals received by the receiver antennas are then converted into point clouds. The processing chain of this application can be seen in figure 3.10. This chain

receives the sampled IF signals in a cube data format and outputs the detected points with Range, azimuth, elevation, SNR and radial velocity estimation [25]. Except for the range-FFT, all the blocks are implemented in the c674x DSP while the range-FFT is implemented on the hardware accelerator. In the following sections, this processing chain will be separated into smaller blocks and a brief explanation of how they operate and their purpose.

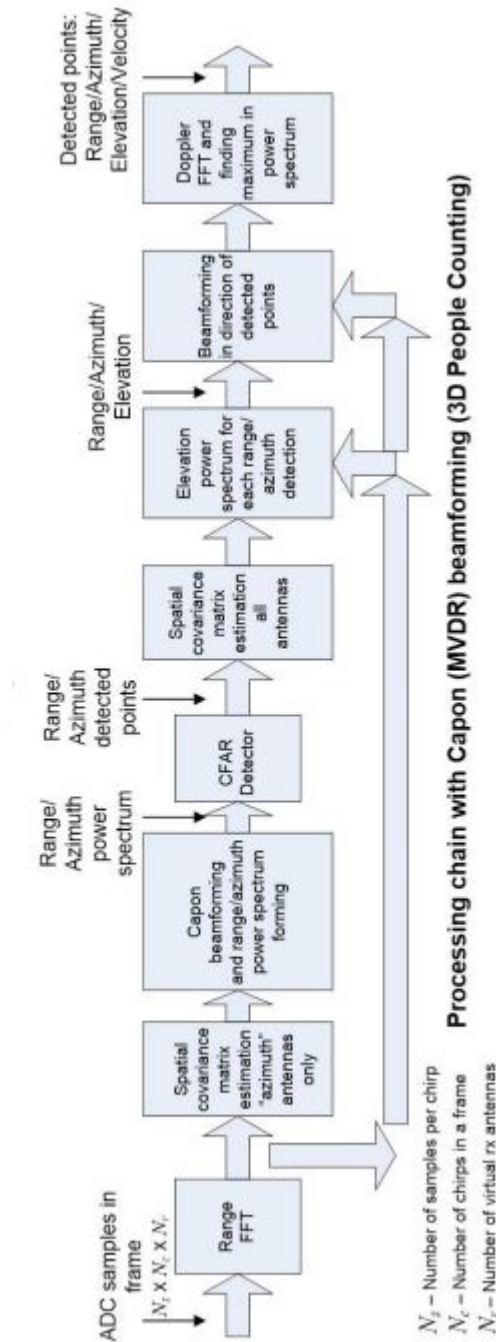


Figure 3.10: People Counting application block diagram of the processing chain. Converts sampled data into point clouds. From [25].

3.2.1 Range Processing and Clutter Removal

The first stage of point cloud generation is to read the samples given by the ADC. For each receiver antenna, the DMA moves samples from the ADC buffer into the Hardware Accelerator which computes the FFT in a 16-bit, fixed-point, 1D-FFT. After computation, the DMA moves the data into the radar data cube memory with size $N_s \times N_c \times N_R$ which is the number of samples per chirp N_s , the number of chirps N_c and number of virtual receiver antennas N_r . The Range Processing is interleaved with the transmission which means is doing inline FFT of the samples received from each transmitted chirp, this means the radar data cube is being filled while the transmission is still occurring [25].

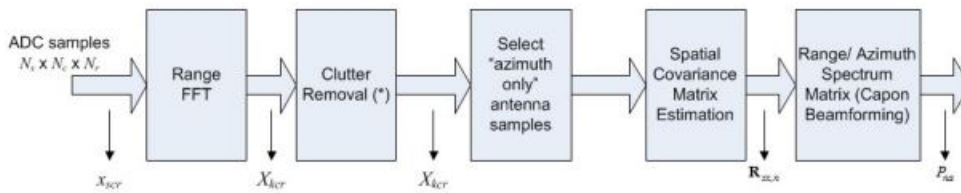


Figure 3.11: Block diagram Range-FFT until Range-Azimuth Heatmap. From [25].

Once the frame transmission and reception are complete, the RF front end goes idle for a while to give time for the DSP to calculate the point clouds. The first stage of point cloud generation is the **Static Clutter Removal**, the 1D-FFT data is averaged across all transmitted chirps for a single receive antenna. The average calculated before is then subtracted from each chirp received, this removes the static components from the signal leaving only the signals with movement associated [25]. The formula for the above formulation is

$$\begin{aligned} X_{nr} &= \frac{1}{N_c} \sum_{c=0}^{N_c-1} X_{ncr} \\ X_{ncr} &= X_{ncr} - X_{nr} \end{aligned} \quad (3.1)$$

With N_c the number of chirps transmitted, N_r the number of receive antennas, X_{nr} average for a single receive antenna across all chirps and X_{ncr} the samples of a single chirp from a receive antenna.

Now that the static components of the detections are removed, the next step is to build the Range-Azimuth heatmap.

3.2.2 Spatial Covariance Matrix Estimation and Capon Beamforming

The spatial Covariance Matrix of the received signals is important to compute the angle of arrival of an object, as seen in chapter 2. The first capon beamforming is done to detect objects in the azimuth domain so the covariance matrix is constructed with data from only the azimuth antennas [25].

$$\begin{aligned} R_{yy} &= \frac{1}{N_c} \sum_{c=0}^{N_c} Y_c Y_c^H \\ Y_c &= [Y_{c,0}, Y_{c,1}, \dots, Y_{c,N_r-1}]^T \end{aligned} \quad (3.2)$$

$Y_{c,i}$ is the array of received samples for a single antenna i from chirp c . According to literature, [26], Capon beamforming requires a good estimation of the covariance matrix. The performance of beamforming by weight calculations declines rapidly if the matrix values are inaccurate, therefore, by adding diagonal elements the inaccuracy is corrected. A diagonal loading is applied to the spatial covariance matrix to ensure stability

$$R_{yy} = R_{yy} + \gamma \frac{\text{tr}(R_{yy})}{N_r} I_{N_r} \quad (3.3)$$

With $\text{tr}(R_{yy})$ the trace of covariance matrix and γ the respective eigenvalues. After the covariance computation is completed, the azimuth steering vector a_a is defined as

$$a_a = \left[1, e^{j2\pi \frac{d}{\lambda} \sin(\theta_a)}, \dots, e^{j2\pi \frac{d}{\lambda} \sin(\theta_a)(N_r-1)} \right]^T \quad (3.4)$$

Where each θ_a is a azimuth bin. The Range-Azimuth Heatmap can be built with power-angle spectrum for each range bin.

$$P_{na} = \frac{1}{a_a^H R_{xx}^{-1} a_a} \quad (3.5)$$

The result P_{na} is the power-angle spectrum which is computed for each range bin. The covariance matrix inversion computation is done through cholesky decomposition. The normalized beamforming weights can also be calculated as

$$W_a = \frac{R_{xx,n}^{-1} a_a}{a_a^H R_{xx,n}^{-1} a_a} \quad (3.6)$$

3.2.3 CFAR Detection Algorithm

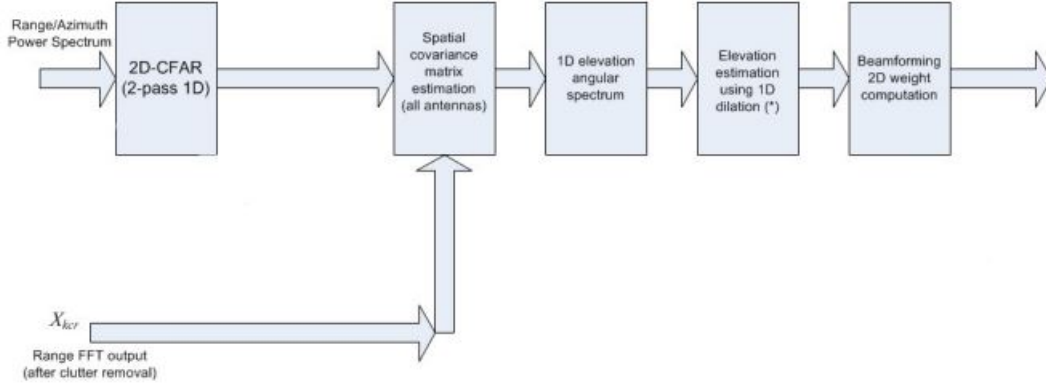


Figure 3.12: CFAR Detection and elevation estimation. From [25].

After the Range-Azimuth Heatmap is formed, a 2D CFAR algorithm is applied to detect the peaks in the spectrum. The two dimensions CFAR means a first SOCA-CFAR is applied in the range domain for each angle step and then, each detected point in the range domain is confirmed by also a SOCA-CFAR in the angle domain [9]. This procedure is explained in flowchart 3.13 and a visual explanation can be seen if fig. 3.14.

The output of this algorithm is a vector containing the detected peaks in the range-azimuth heatmap with each detected point having a 2D polar coordinate ($range, \theta$).

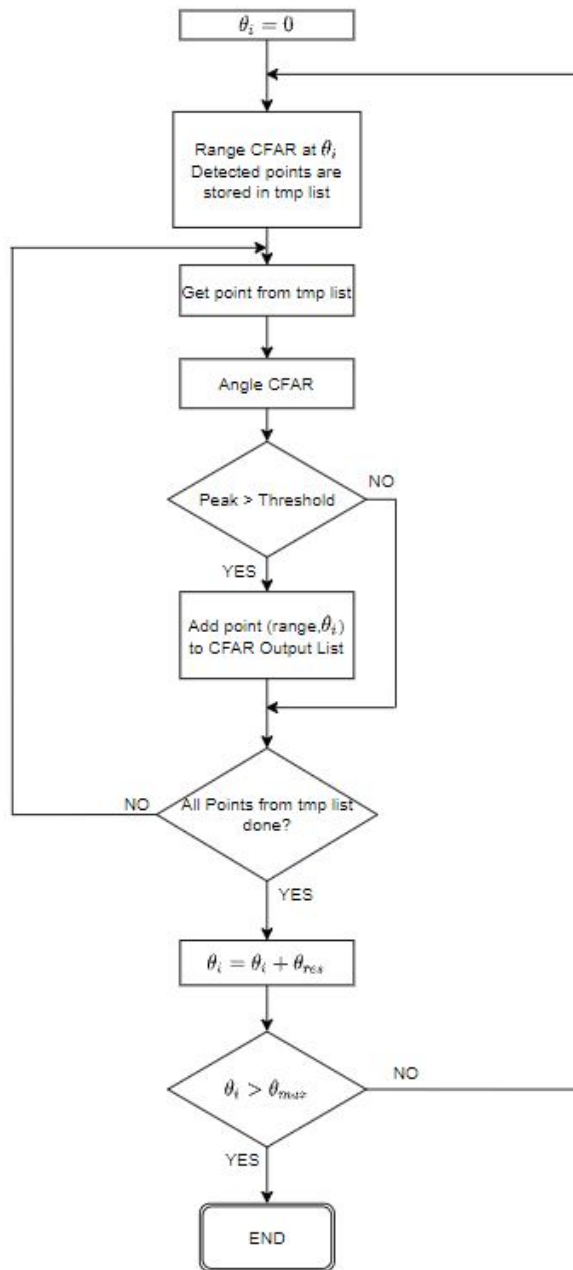


Figure 3.13: 2D SOCA-CFAR flowchart. Adapted from [9].

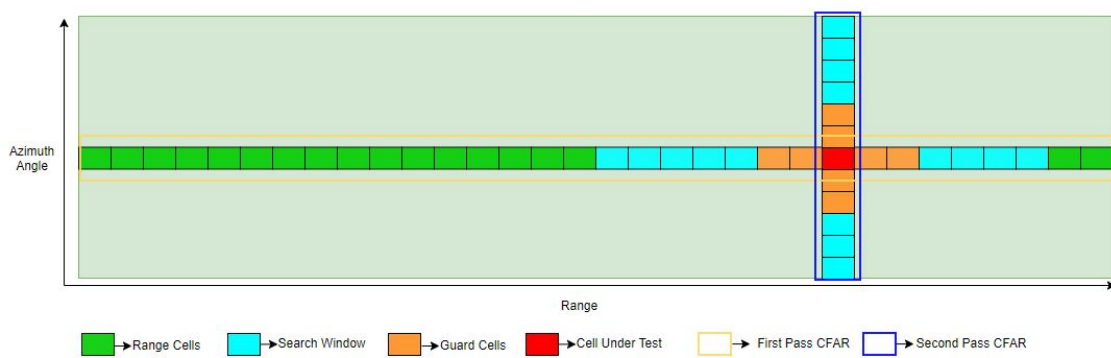


Figure 3.14: 2D CFAR Input. Range-Angle heatmap where the represented cells are a Range-FFT at a θ angle. Adapted from [9].

3.2.4 Elevation Beamforming

The elevation beamforming is performed at the azimuth of each detected point after 2D-CFAR. The process is done following the same steps as the generation of a range-azimuth heatmap, first, a spatial covariance matrix is computed and then the Power-Elevation spectrum is generated but only for the range detected at 2D-CFAR [25].

$$R_{yy} = \frac{1}{N_c} \sum_{c=0}^{N_c} Y_c Y_c^H \quad (3.7)$$

The Diagonal loading matrix is added for stability.

$$R_{yy,m} = R_{yy,m} + \gamma \frac{\text{tr}(R_{yy,m})}{N_r} I_{N_r} \quad (3.8)$$

The steering vector elevation spectrum is the kronecker product of the azimuth and elevation steering vector

$$\begin{aligned} a(\mu_m) &= [1, e^{j\mu_m}, \dots, e^{j(N_r-1)\mu_m}]^T \\ a(v_m) &= [1, e^{jv_m}, \dots, e^{j(N_r-1)v_m}]^T \\ a_m &= a(\mu_m, v_m) = a(\mu_m) \otimes a(v_m) \end{aligned} \quad (3.9)$$

Knowing both the covariance matrix and the steering vector, the 1D elevation spectrum is as follows

$$P_m = \frac{1}{a_m^H R_{yy,m}^{-1} a_m} \quad (3.10)$$

3.2.5 Doppler Estimation

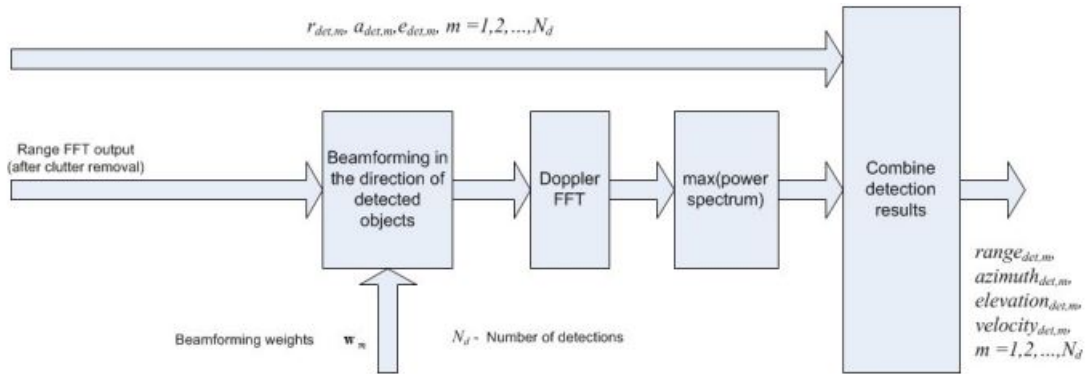


Figure 3.15: Doppler estimation and combination of results with detected points. From [25].

The Radial Velocity estimation is done in the last part after all detected points are determined and each one has a range, azimuth, elevation and SNR information. Before computing the Doppler-FFT seen in the chapter before, the Capon beamforming is applied to all 12 virtual antennas at the detected range to find the sample peak value of the detection across all chirps in the frame. After finding this set of values, the 2D-FFT across all chirps at the detected range and angle is applied which resolves the radial velocity of the target as seen in doppler [25].

3.3 Application Processing Interface

3.3.1 Serial Communication Setup

The communication between radar main sub-system and the PC is done through two UART interfaces, one for the PC \rightarrow Radar and another for Radar \rightarrow PC. The first one is usually used for the initial parameter configuration data transfer and to acknowledge the radar to start working while the second one, the Data Port, is the channel where the radar outputs the point cloud calculated. The figure 3.16 illustrates the setup between Radar and PC. In the figure 3.17 it is possible to see a high-level interaction between the PC, the radar

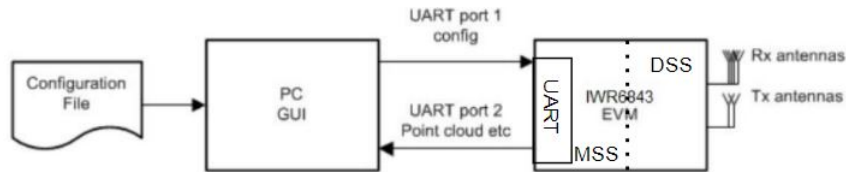


Figure 3.16: IWR6843 UART communication setup with PC. Adapted from [25]

and the sub-systems. The Radar starts by transmitting a frame and doing an inline FFT of the received signals, storing the results in the Radar Cube memory. After the frame ends, the DSP starts to process the data, outputs the detected point cloud and signals the Main Sub-System that the data is ready to be sent through UART. Because of the interconnectivity between sub-systems, this can be done in parallel which means during the data transfer, the radar can transmit another frame and process the respective data while sending through the UART interface the previously detected point cloud.

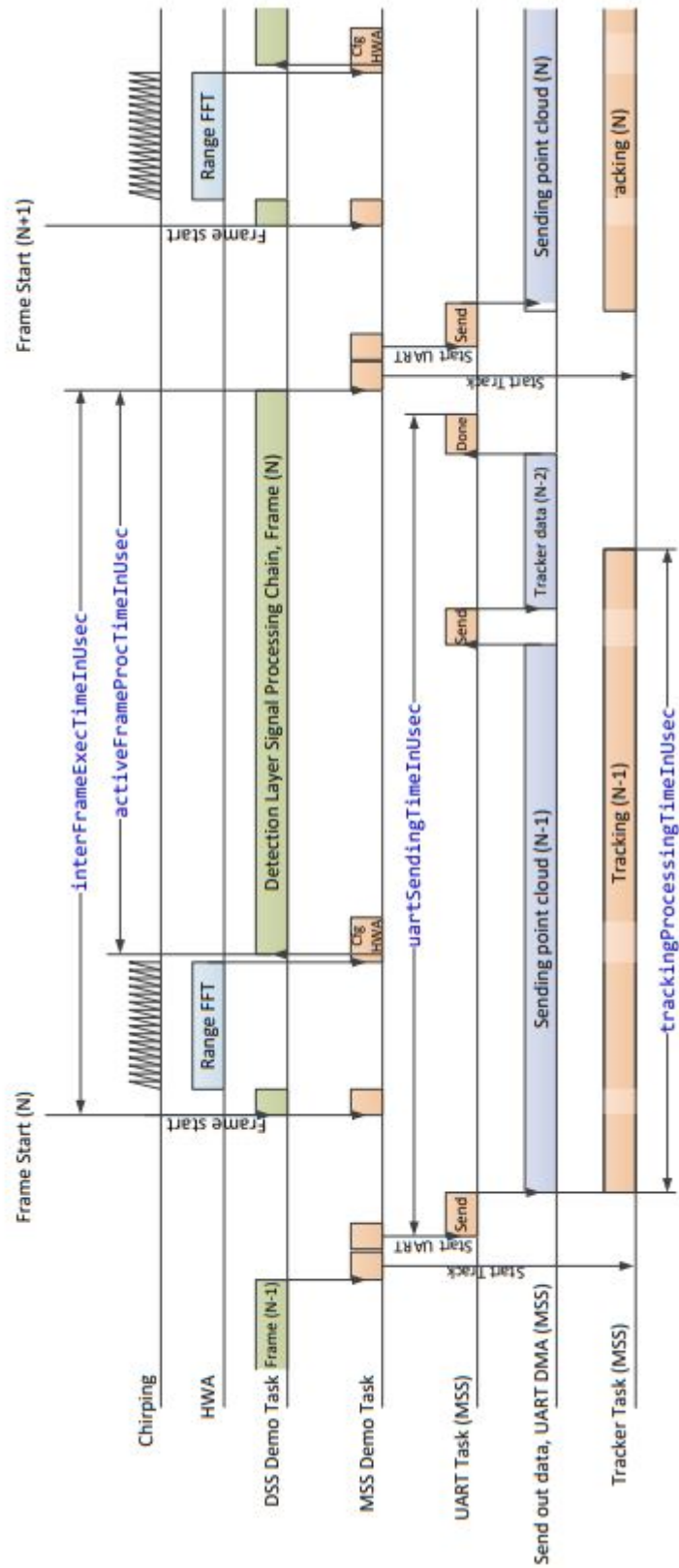


Figure 3.17: Timing diagram of the task activity done between PC and Radar. From [9]

3.3.2 TLV Data Format

The people counting demo outputs one packet of data in each frame, the data is sent through UART configured at 921600 bits-per-second in DMA mode. The data packet is constructed in **Tag Length Value** (TLV) format, which means, each packet sent has a frame header (52 Bytes) with a fixed set of bits at the beginning, in order for the PC to be able to detect the start of the transmission in the UART buffer. In the header, there is also information about the number of TLV blocks, the total size of the packet in bytes and a header checksum to ensure the correct reading [9].



Figure 3.18: Data Packet structure sent to PC with multiple TLV blocks from a single packet. From [9].

```
frameHeaderStructType = struct(...
    'sync',          {'uint64', 8}, ... % Sync Pattern
    'version',       {'uint32', 4}, ... % mmWaveSDK version
    'packetLength',  {'uint32', 4}, ... % In bytes, including header
    'platform',      {'uint32', 4}, ... % 0xA1642 or 0xA1443
    'frameNumber',   {'uint32', 4}, ... % Starting from 1
    'subframeNumber', {'uint32', 4}, ...
    'chirpMargin',   {'uint32', 4}, ... % Chirp Processing margin, in us
    'frameProcTimeInUsec', {'uint32', 4}, ... % Frame Processing time, in us
    'trackProcessTime', {'uint32', 4}, ... % Tracking Processing time, in us
    'uartSentTime',   {'uint32', 4}, ... % Time spent to send data, in us
    'numTLVs',        {'uint16', 2}, ... % Number of TLVs in this frame
    'checksum',       {'uint16', 2}); % Header checksum
```

Figure 3.19: The frame header structure. From [9].

Each TLV block has also a header and a payload, the header indicates what is the expected type of data in the payload and the total length in bytes of the payload including TLV header.

```
pointCloudUnitStruct = struct(...
    'elevationUnit', {'float', 4}, ... % unit resolution of elevation report, in rad
    'azimuthUnit',   {'float', 4}, ... % unit resolution of azimuth report, in rad
    'dopplerUnit',   {'float', 4}, ... % unit resolution of Doppler report, in m/s
    'rangeUnit',     {'float', 4}, ... % unit resolution of Range report, in m
    'snrUnit',       {'float', 4}); % unit resolution of SNR report, ratio
```

Figure 3.20: Point Cloud unit structure. From [9].

```
pointStruct = struct(...
    'elevation',      {'int8', 1}, ... % Elevation report, in number of elevationUnit
    'azimuth',        {'int8', 1}, ... % Azimuth report, in number of azimuthUnit
    'doppler',        {'int16', 1}, ... % Doppler, in number of dopplerUnit
    'range',          {'uint16', 2}, ... % Range, in number of rangeUnit
    'snr',            {'uint16', 2}); % SNR, in number of snrUnit
```

Figure 3.21: Point detected Structure. From [9].

Since the radar is only sending point clouds, there is only one type of TLVs expected which is the point cloud TLV which means each packet transmitted by UART contains only a single TLV. The payload contains a point cloud unit structure (fig 3.20), first data transmitted in the payload followed by all the detected points where each point has a specific structure (fig 3.21). The purpose of the first part sent is to set the units resolution in floating-point, this is done only a single time per TLV, and then, all the detected points are transmitted as integers. To obtain the real value of the detected point, the integer values in the detected points must be multiplied by the unit resolution at the PC side. The length of this type of TLV is $sizeof(tlvHeader) + sizeof(PointCloudUnitStruct) + sizeof(PointCloudStruct) \times numberOfPoints$.

The next chapter will present the work done with the data provided by the radar, explained in this chapter. The work consists in parsing the data retrieved from the UART into point clouds and applying a processing chain that is able to count and track targets on point clouds. After the processing is done, the data is projected onto a real-time camera image.

Chapter 4

People Counting and Tracking

One of the objectives of this project was to build an interface capable of displaying the data sent from the radar and the results of post-processing. So a GUI was built based on the PyQt5 engine which allows creating multiple threads and a 3D plot based on OpenGL where the point clouds and the tracking can be shown in a live environment. This chapter is here just to show the capabilities of the application developed and how the point clouds are processed by the PC, so in this section, all the images and data presented are built with data emulating the point clouds detected by the radar. The TI IWR6843ISK-ODS board is designed for ceil-mount applications but in this dissertation will be used in a wall-mount scenario, the trade-off is the maximum detectable range which is significantly reduced in comparison to the TI IWR6843ISK, but the capabilities and potential remains the same.

4.1 Application Flowchart

The Radar is constantly transmitting point clouds through UART which are stored in the UART Buffer. The tracking module reads the buffer and parses the binary data into point clouds as specified at the end of chapter 3, and then applies a processing chain that detects and tracks point cloud clusters. The tracking module outputs the point clouds, the targets detected and the spatial location to the main thread which plots the points and the targets in a 3D environment. The main thread computes the respective projections into the camera and feeds the data into another thread running the live camera image. The image 4.1 depicts the application flowchart developed in this dissertation, with the representation of the communication between different interfaces. The radar fills the UART buffer with point clouds, the PC reads the buffer and processes the data to detect people and do the tracking. Then outputs the targets, the tracking and the point clouds to the GUI which shows in a 3D view.

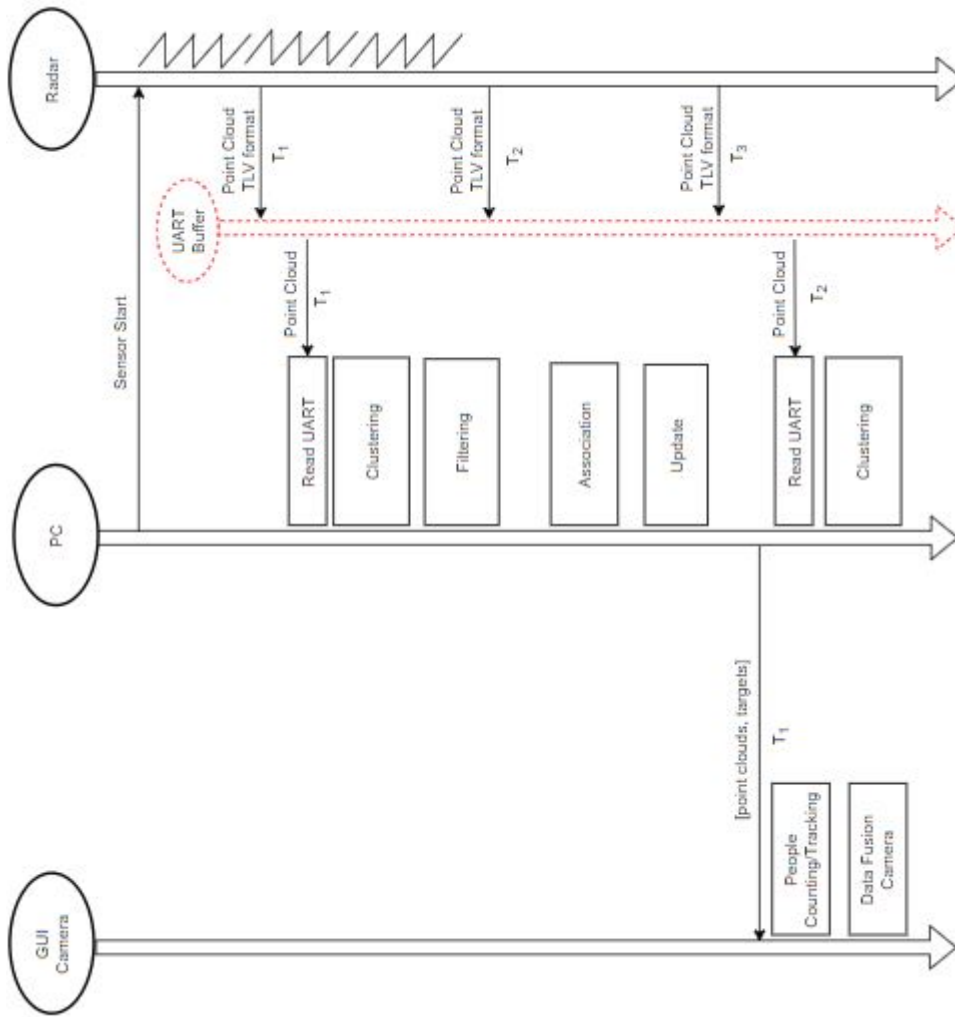


Figure 4.1: Developed application flowchart.

4.2 Point Cloud Processing

After the PC reads the buffer and retrieves a point cloud the first step is to find detections in the data. The easiest way to do this is to apply a clustering algorithm that outputs the detected clusters where each one can be a person or clutter. After the cluster is identified, a filter layer is applied to reduce the impact of the multipath clutter and increase the accuracy of the detections. The next step is to associate the cluster with existent targets and update the new position detected, this is done with the help of a Kalman Filter which helps estimate the state vector of a target.

4.2.1 Clustering with DBSCAN

Each point clouds given by the radar contains the target of interest detections and some clutter and static points. So it is important to be able to separate these false detections and ensure only the points from the target of interest are sent forward for further processing. In

order to do so, clustering methods are able to identify a cluster of points in the spatial domain with an arbitrary shape which is the case of data retrieved from the radar. The method chosen was **DBSCAN** (Density-based spatial clustering of applications with noise) [27]. This method relies only on the minimum number of points ($minPts$) and the maximum distance between them (Eps). In the figure 4.2 three clusters are represented with each color being the label detected. The intuitive way for a human to visually detect a cluster is to find points

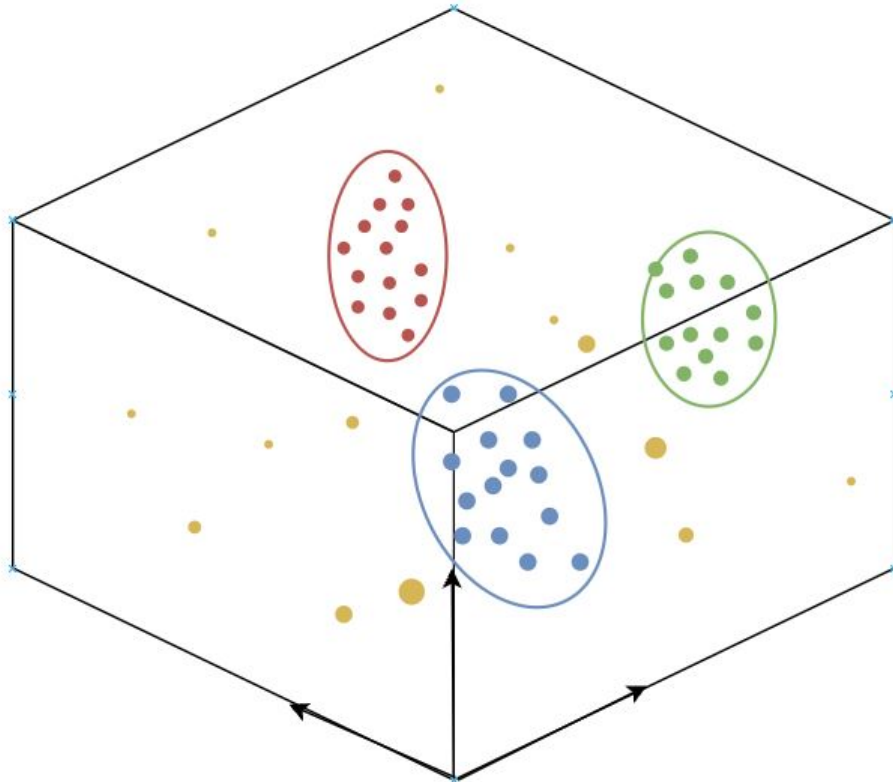


Figure 4.2: Clustering in 3D. Each color represents a label, the yellow color is the noise which are points with no associated cluster.

that are closely spaced between them which is the key idea behind DBSCAN. For each point detected, in a given Eps radius, the neighborhood has to have a minimum number of points belonging to the circle area.

4.2.1.1 Choosing Parameters Value

The approach to assigning the best values of the two parameters of the DBSCAN is empirical, a set of experiments was made in order to find the values that suited the system. The minimum number of points ($minPts$) was chosen based on the sensitivity of the CFAR algorithm, if the threshold was too low a high number of points would be detected and by analysis, a person moving would generate between [20,40] points so a minimum of 20 could be

used. Because the CFAR threshold in the radar configuration was set to a high value, a low number of points would be detected by the radar which means the *minPts* was set to 5.

The *Eps* value was chosen by using rationality, if the *Eps* is the maximum distance for a point to be considered inside the neighborhood and the radar has a range resolution of around *5cm* in this setup which is the minimum distance between points resolved in range domain, then a *Eps* of *12cm* was chosen to accommodate the range resolution and allow for multiple points to be inside the neighborhood.

4.2.1.2 Calculating Centroids of the Cluster

The center of mass of the people detected is used for tracking purposes which can be translated to the central point of the cluster. This can be obtained by the calculation shown in equation 4.1, the centroid is $(mean_x, mean_y, mean_z)$. In the figure 4.3, the centroid calculated is visible as the green dot in the middle of the red dots which are the cluster detected.

$$\begin{aligned} mean_x &= \frac{1}{nPoints} \sum_{n=0}^{nPoints} x_n \\ mean_y &= \frac{1}{nPoints} \sum_{n=0}^{nPoints} y_n \\ mean_z &= \frac{1}{nPoints} \sum_{n=0}^{nPoints} z_n \end{aligned} \quad (4.1)$$

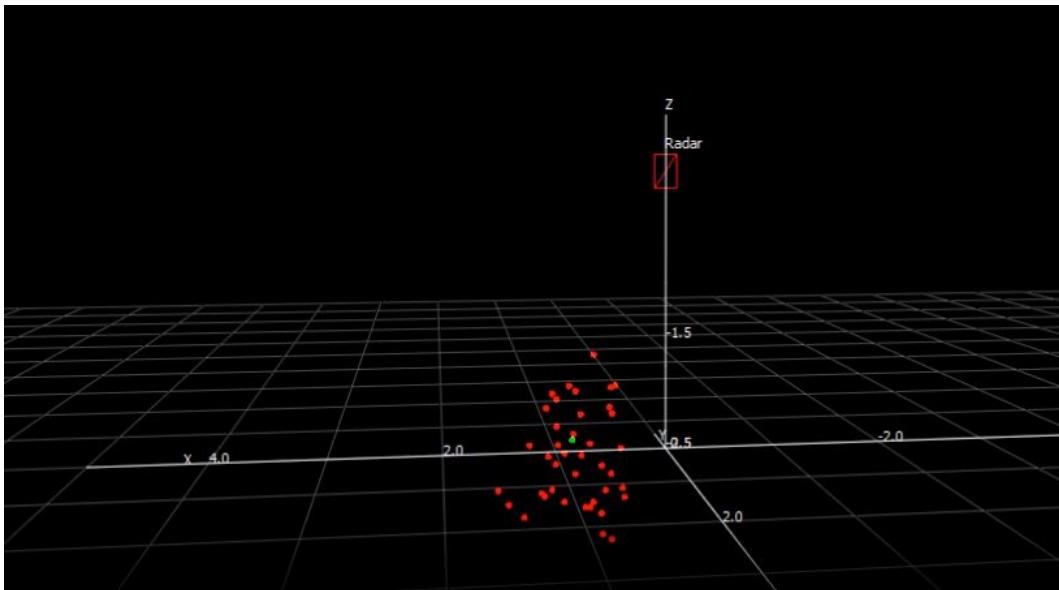


Figure 4.3: Centroid (Green dot) of the cluster (Red).

4.2.2 Point Cloud Filtering

The clutter originated from multipath reflections create clusters of points in different spatial coordinates, so a single person moving in a room can originate multiple clusters. To evaluate the fidelity of cluster identification, a two-layer filter is applied to each cluster detected by the DBSCAN.

4.2.2.1 Two-Step Identification

The first time a cluster is identified, it is flagged as a "possible person". For a cluster to be considered a person, a second one must appear in the neighborhood region of the first identification. Because in a second frame, a real person will originate a detection in the same region of the previous frame since the radar is sending multiple frames in a second, but the multipath effect generates detections in "random" locations.

4.2.2.2 SNR Evaluation

Typically, a multipath reflection means a longer distance traveled by the wave, as seen in chapter 2 the power received by the antennas is inversely proportional to the distance which means the clutter will have a much smaller power. Because the multipath distance is usually longer than the LOS distance, the average SNR of a cluster originated by a real target will be bigger than the respective clutter SNR, so in order for a cluster to be considered a target, it will need to have an average SNR bigger than a certain threshold. This threshold is empirically defined as the value that better detects targets.

4.2.3 Tracking With Kalman Filter

The implementation of the Kalman filter was done with the help of a python library called "*FilterPy*" [28]. The Kalman filter is a mathematical model which uses measurements over time that may contain noise and produce a prediction about the real values of the state vector which contains information about the spatial position, velocity and acceleration in a 3D Cartesian space. These predictions are based on previous measurements and assume a constant acceleration model between each discrete time step [29]. The state transition Matrix F is used to describe the dynamics of the system for a constant acceleration model with T time between steps.

$$F = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 & 0.5T^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 & 0 & 0.5T^2 & 0 \\ 0 & 0 & 1 & 0 & 0 & T & 0 & 0 & 0.5T^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

Because the movement of a person can be non-linear, the Linear Kalman filter performs poorly, so a **Extended Kalman Filter** (EKF) was implemented to handle this problem. The state of the Extended Kalman filter at the instant k is

$$X_k = Fx_{k-1} + w_k \quad (4.3)$$

$$Z_k = Hx_{k-1} + e_k \quad (4.4)$$

Where X_k is the state vector in the Cartesian coordinates for a constant acceleration model, $w(n)$ is the process noise with covariance \mathbf{Q} which represents the deviation between the actual state and the motion model, Z_k is the measurement of the system and e_k is the measurement noise. The state vector can be defined as

$$X_k = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}]^T \quad (4.5)$$

4.2.3.1 Predict Stage

The first step is to predict an a-priori state estimate \hat{x}_k and covariance P_k

$$\begin{aligned} \hat{x}_k &= F\hat{x}_{k-1} + w_{k-1} \\ P_k &= F_k P_{k-1} F_k^T + Q_K \end{aligned} \quad (4.6)$$

4.2.3.2 Update Stage

Then the measurement residual is calculated

$$y_k = z_k - H\hat{x}_k \quad (4.7)$$

The residual covariance is

$$S_k = H_k P_k H_k^T + R_k \quad (4.8)$$

Then the kalman gain

$$K_k = P_k H_k^T S_k^{-1} \quad (4.9)$$

After the kalman gain is calculated, the updated state and covariance estimation

$$\begin{aligned} x_k &= \hat{x}_{k-1} + K_k^T y_k \\ P_k &= (I - K_k H_k) P_k \end{aligned} \quad (4.10)$$

The state transition F_k and observation matrices are defined as the following Jacobians

$$\begin{aligned} F_k &= \left. \frac{\partial f}{\partial x} \right|_{x_{k-1}, w_k} \\ H_k &= \left. \frac{\partial h}{\partial x} \right|_{x_k} \end{aligned} \quad (4.11)$$

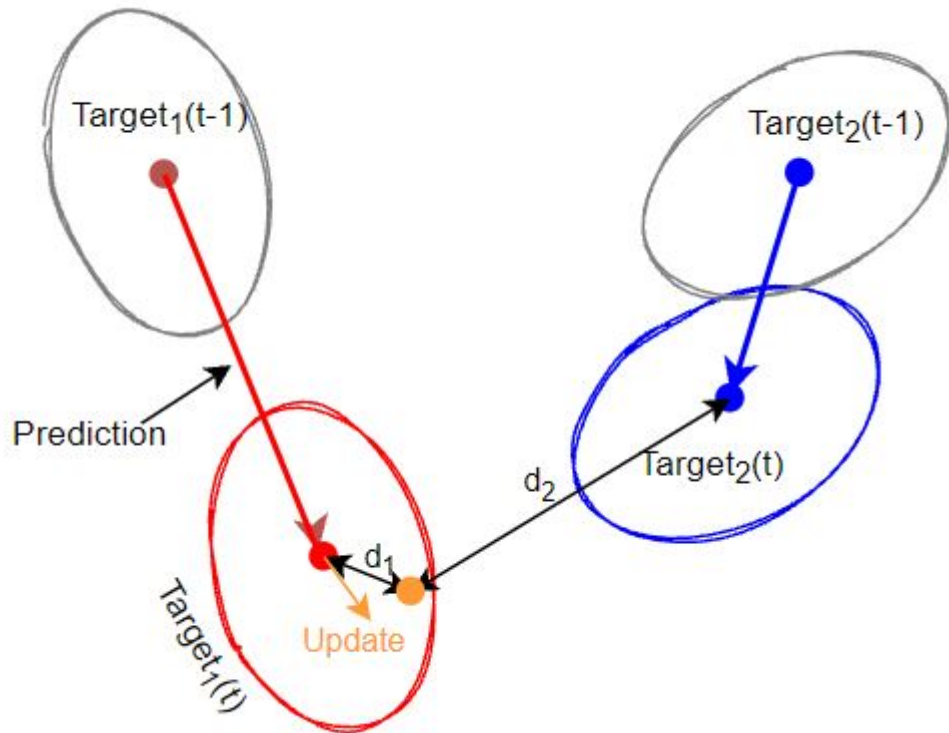


Figure 4.4: Diagram of how the algorithm predict, allocate and update the calculated cluster centroid into existent targets.

4.2.3.3 Predict Target Movement

The first stage of the EKF is able to get a prediction of the state vector based on the previous dynamics. With this in mind, this prediction is used to associate a detected cluster centroid with the most likely target existent. Prediction arrow in figure 4.4.

4.2.3.4 Allocate to the Closest Target

After the prediction is done, the distance between the cluster centroid and each target prediction is computed. The target prediction with the smallest distance to the cluster is the associated one and is used for the next stages. Distance between existent target ellipsoid and cluster centroid in figure 4.4.

4.2.3.5 Update Target Position

After the cluster is associated with an existent target, the cluster centroid is used to update the new state vector by feeding as a measurement to EKF. The state vector after the update stage is used as the new target location. Update arrow in figure 4.4.

Figure 4.5 shows an example of the tracking of a cluster moving in a circular pattern in a noisy environment. The bounding box is the target position computed after the Kalman

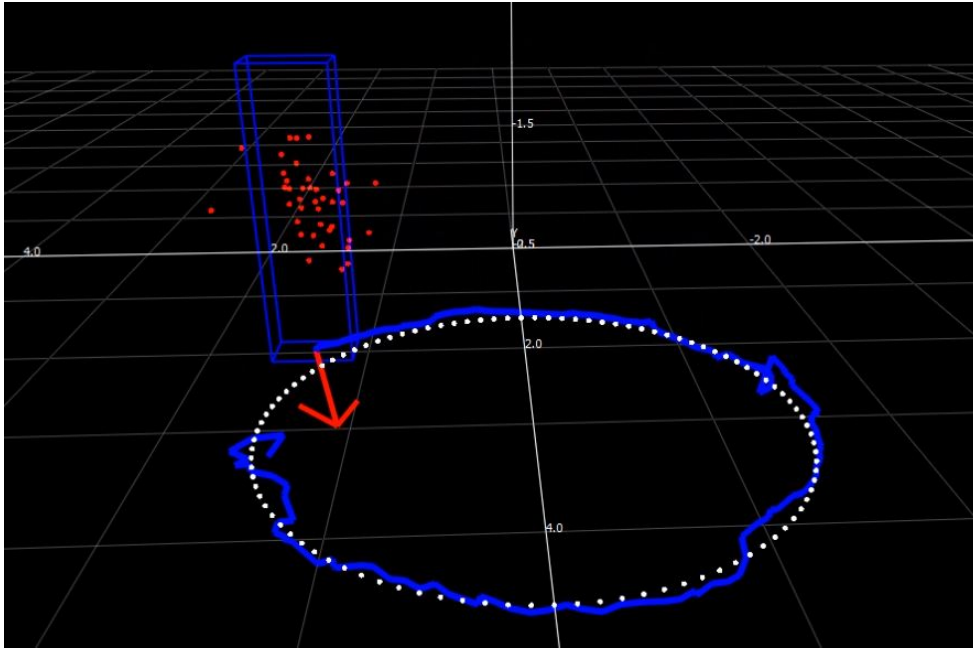


Figure 4.5: Target Detected (blue bounding box) and tracking (blue line) with movement prediction (red arrow) in a circular movement (white dots).

update stage that can be obtained from the state vector, the blue line is the tracking of the target positions and the red arrow is the movement prediction in the X-Y plane.

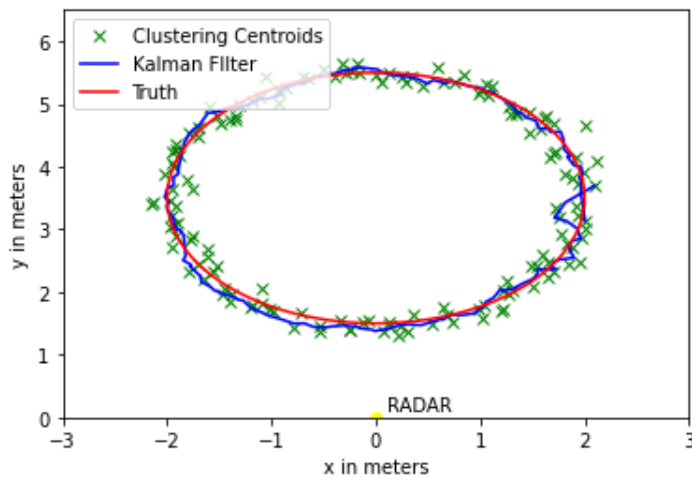


Figure 4.6: Plot of the top view of the target tracking. The red line is the ground truth, the green markers are the calculated cluster centroids and the blue line is the updated state vector.

Figure 4.6 is the respective X-Y measurements done in figure 4.5. The ground truth is the red circle in which the clusters with Gaussian noise are created and used for simulation. The green crosses are the centroid of the detected clusters by the DBSCAN algorithm and the spatial location is fed into EKF. The blue line is the updated state vector spatial coordinates $[x,y]$. The simulations show the EKF can perform well in the presence of Gaussian noise, the performance can be evaluated through the RMSE graphic present in the figure 4.7 The RMSE

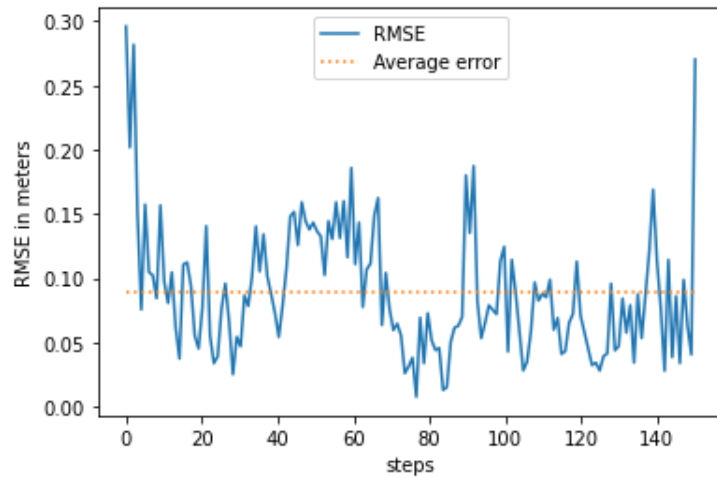


Figure 4.7: The RMSE plot. Difference between Kalman and ground truth in meters across 150 steps.

by definition is the Root Mean Square Error which in this case is the average error distance between the computed Kalman state vector and the ground truth, this means the application is capable of tracking with an accuracy of fewer than 0.1 meters.

A second simulation was made with a different path, this time with a more complex scenario caused by a sudden change in the direction which means a non-linear movement depicted by the red line. This is just used to test the capabilities of the EKF algorithm which was chosen because of the ability to track more complex scenarios.

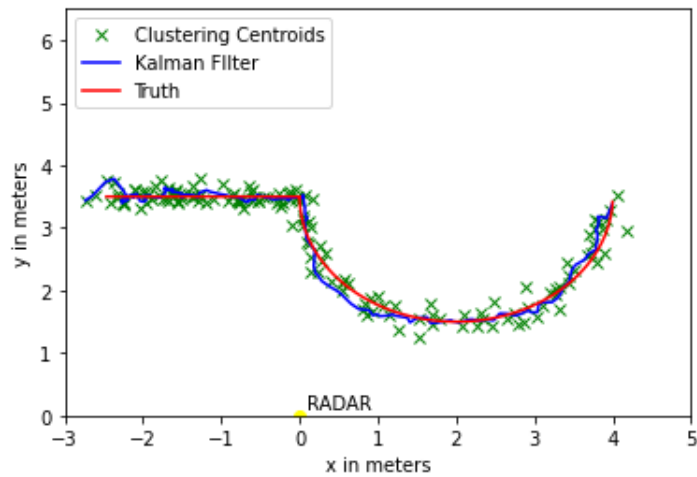


Figure 4.8: Top-view plot of a non-linear movement and the output of the tracking stage.

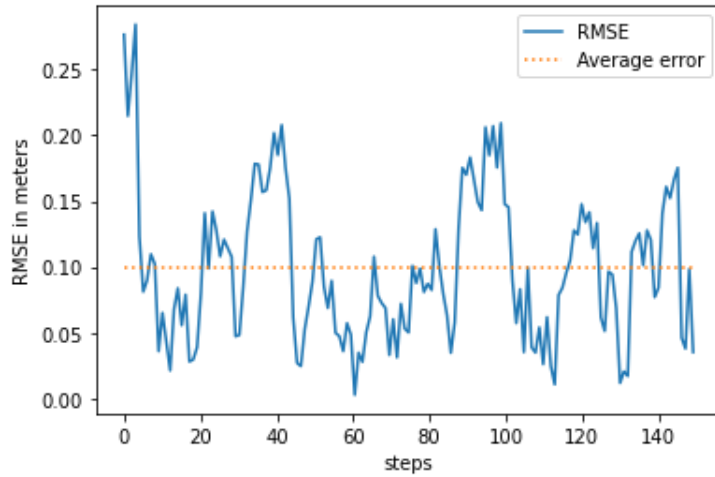


Figure 4.9: Respective RMSE from figure 4.8.

4.2.4 Sensor Fusion with Camera

4.2.4.1 Cartesian Coordinates System into Image Pixels

The conversion of 3D coordinates to 2D image pixels is done through a technique called perspective projection [30]. In short, this works by projecting a point into the surface of a plane as seen in figure 4.10. Based on triangular similarities, the coordinates of the projection

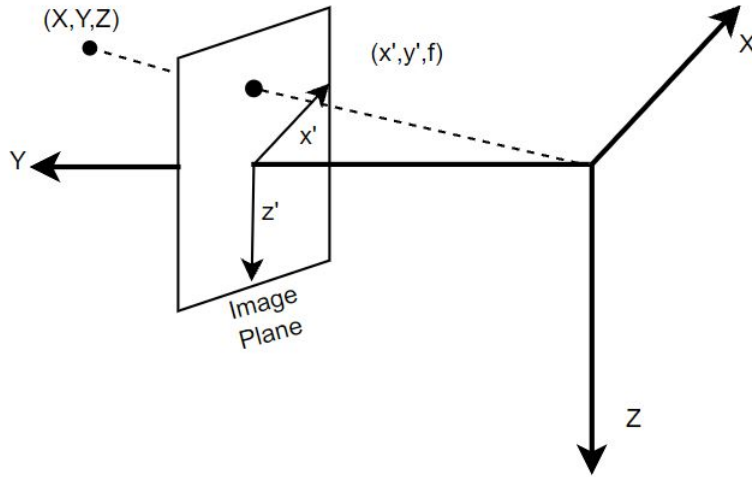


Figure 4.10: Perspective projection of a point located at (X, Y, Z) onto a plane located at distance f from the origin.

on the plane can be computed as

$$\frac{x'}{f} = \frac{X}{Y} \implies x' = f \frac{X}{Y} \quad (4.12)$$

with x' the coordinate of the projection onto the plane, X and Y the 3D Cartesian coordinates of the target and f the distance of the plane. The coordinates of the z' can also be estimated

the same way as the above

$$\frac{z'}{f} = \frac{Z}{Y} \implies z' = f \frac{Z}{Y} \quad (4.13)$$

Then each point must be converted into the pixel location by knowing the image limits spatial coordinates, in other words, the width pixel 0 is located at around 1.5 meters to the right at a focal distance of 2 meters. The same can be done to obtain the height pixels.

4.2.4.2 Image Enhancement with Radar Information

The Camera used in this dissertation has an FoV of 60° in elevation and 60° in azimuth while the radar application has 70° in azimuth and 20° in elevation, this means the radar has additional information in the x-axis as seen in figure 4.11.

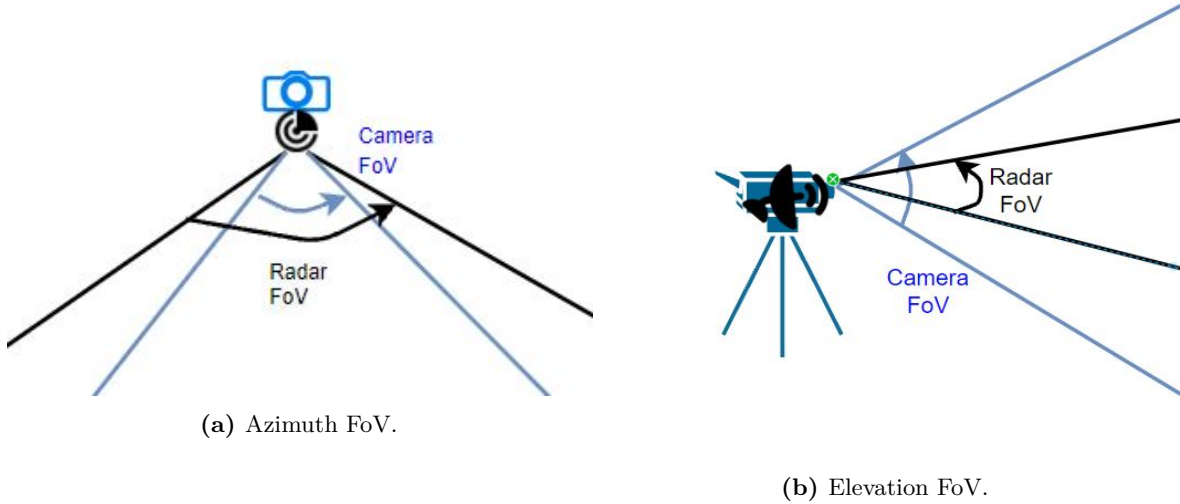


Figure 4.11: FoV enhancement in Azimuth FoV.

With this in mind, the image captured by the camera can be enhanced with extra information provided by the radar along the azimuth axis, henceforth, the image after the projection stage has an increased size in width with two lateral columns of black pixels to be filled with projections from the radar.

4.3 People Counting GUI

The GUI interface shown in figure 4.12 was developed with the help of the PyQt5 library. At the left panel, there are some control boxes, such as UART ports to setup the connection, two buttons for selecting and sending the configuration file, some parameters to control the point cloud processing and plot, and a table layout with some statistics. At the center, there are two tabs, one for the 3D plot based on OpenGL and another for the live camera view with the projection of the targets and point cloud.

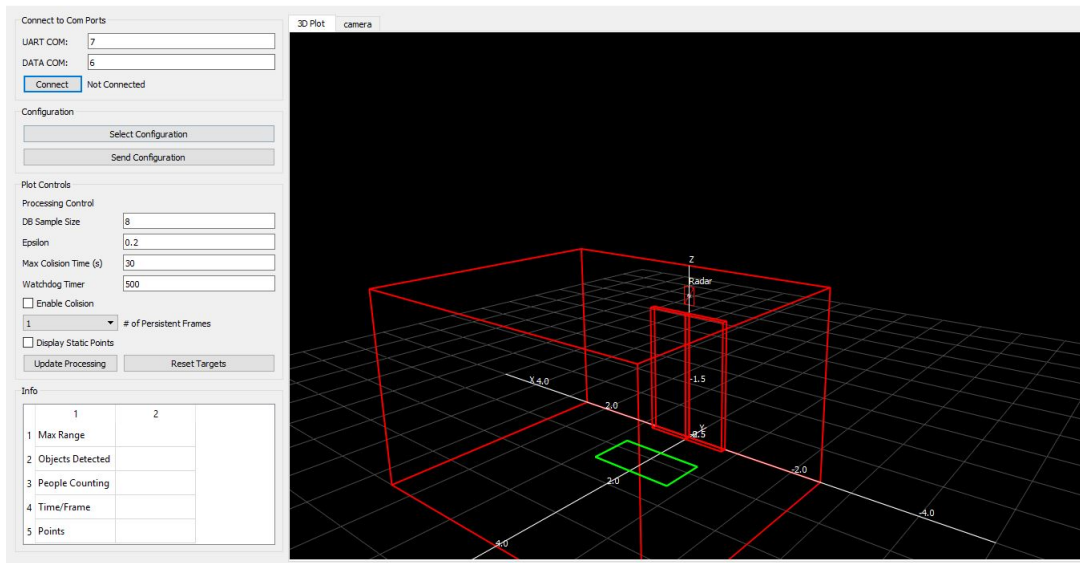


Figure 4.12: People Counting and Tracking GUI.

4.3.1 3D Target Visualizer

This shows the capabilities of the 3D visualizer, two targets were simulated moving in opposite directions with the respective tracking, clusters used for the detection, the arrow with the movement prediction and the respective coordinates of the centroid from the bounding boxes.

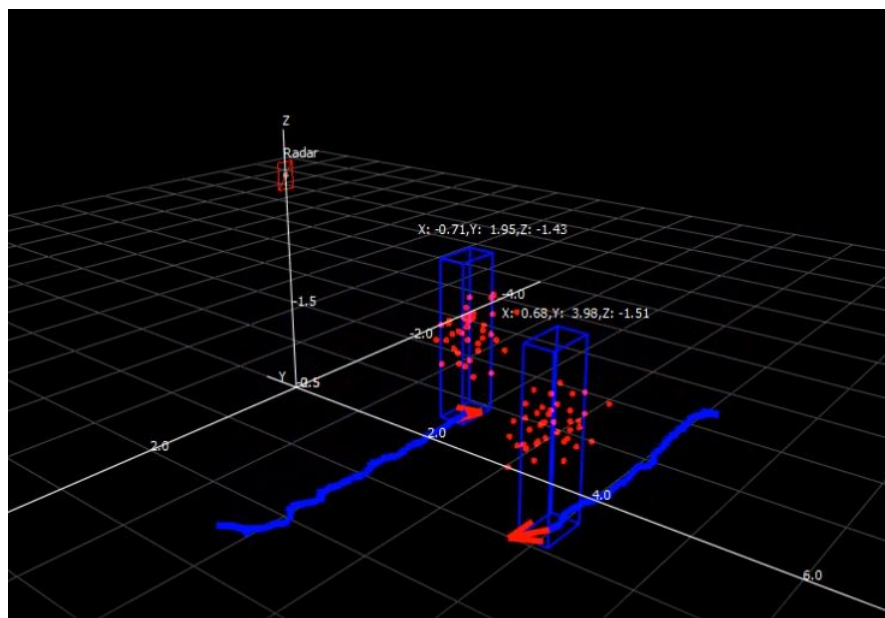
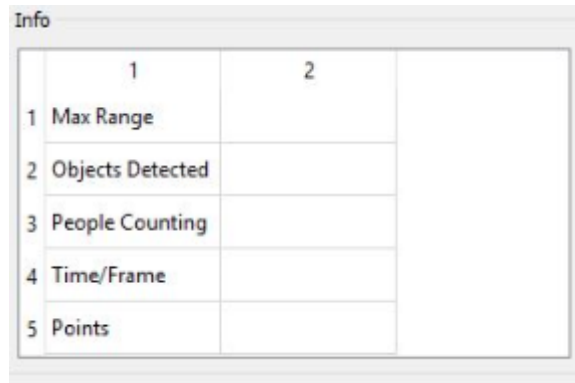


Figure 4.13: Multi-tracking of two targets.

4.3.1.1 Live Statistics

This table shows the maximum range configured, the number of detected people, the number of possible people that still need a second identification pass, the processing time between the moment point cloud is read from the buffer and the 3D plot of the output and the number of points detected by the radar in each frame.



	1	2
1 Max Range		
2 Objects Detected		
3 People Counting		
4 Time/Frame		
5 Points		

Figure 4.14: Statistics table

Chapter 5

Results

In the previous chapter, it has been shown the algorithm implemented on the received point clouds by the radar and discussed the possible results based on simulations. This chapter presents the results with real data provided by the radar in multiple scenarios. The results provided are divided into three parts: people counting, people tracking and sensor fusion. The first section, people counting, is the ability for the radar to count the number of people present in the FoV and the number of false detections, the second part is the tracking the movement of a target following multiple routes and the last section is sensor fusion with the camera.

5.1 Chirp Parameters

The configured parameters of the FMCW radar are presented in the table 5.1. The selected configuration was the common ground found that was able to detect multiple persons with big enough clusters so the DBSCAN algorithm was able to separate the clusters into different targets and label them. The maximum range selected is only 5.4 meters because the radar antennas were designed for a ceil-mount configuration where the common distances between the ground and ceil are around 3 meters. With this constraint, the tests performed are done in a close space to obtain reliable detections.

Initial Frequency	60.20	(GHz)
Ramp Slope	74.95	(MHz/ μ s)
Bandwidth	2974	(MHz)
Range resolution	0.05	(m)
Maximum range	5.4	(m)
Velocity resolution	0.18	(km/h)
Maximum velocity	26.9	(km/h)
Sampling Frequency	2.95	(Msps)

Table 5.1: Parameters used for the tests above derived from chapter 2 (table 2.1).

5.2 People Counting

The people counting detection is based on the tracking module, it counts the number of existent tracks and records the location measured. The tests performed were done in limited conditions because there were some security policies regarding the minimum distance between people in these extraordinary times.

5.2.1 Scenario Setup

To test the people detection capability of the application developed, multiple people were asked to stay idle in specific positions marked on the ground which were physically measured to determine the ground truth.

5.2.2 Tests Performed

The metrics developed for performance evaluation are application counting which is a measure of how much time the application was able to measure correctly. The positional accuracy is the distance from the detection average position during the test time to the ground truth markers.

Real People Counting	Application Counting	Positional Accuracy
1	98%	0.06 meters
2	88%	0.11 meters
3	73%	0.11 meters
5	64%	0.14 meters

Table 5.2: Tests performed and results obtained for people counting.

5.2.3 People Counting Accuracy

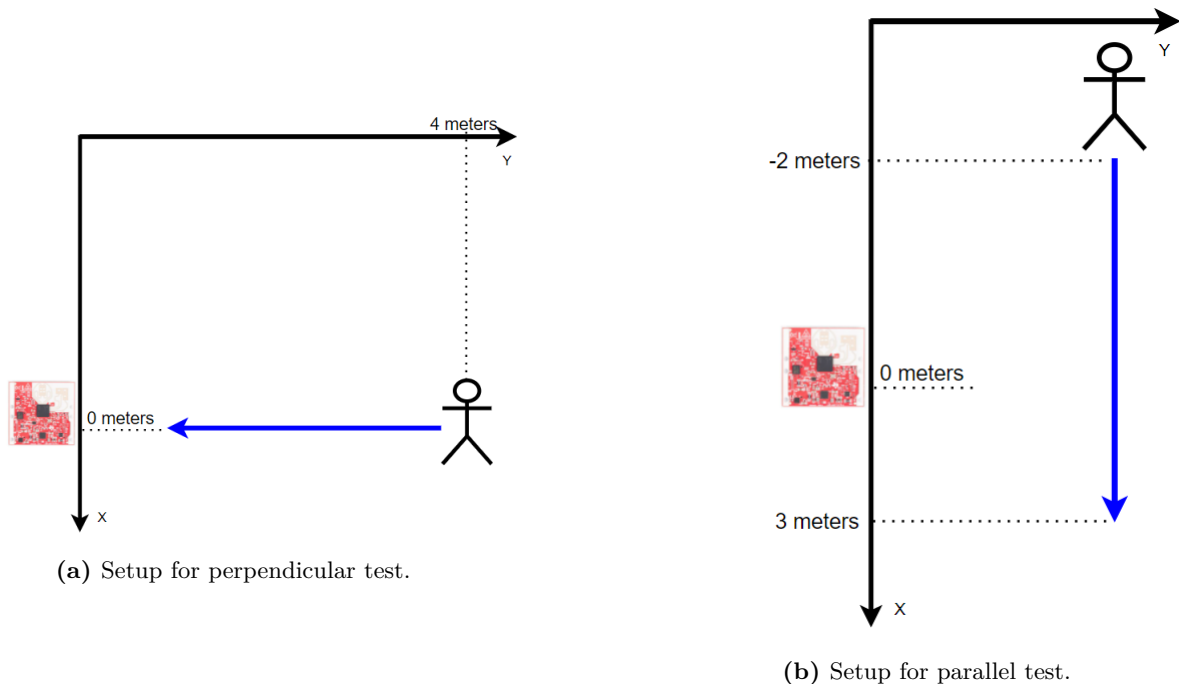
Looking to the results on Table 5.2, one can infer that the counting algorithm performs well if the targets are generating small movements like moving arms or shaking the head. Since the static clutter removal algorithm (described in chapter 3) removes the static points, the point cloud retrieved by the radar contains only information from humans moving. When the number of people increases, the people counting performance decreases because of the complexity introduced by multiple people interacting with each other.

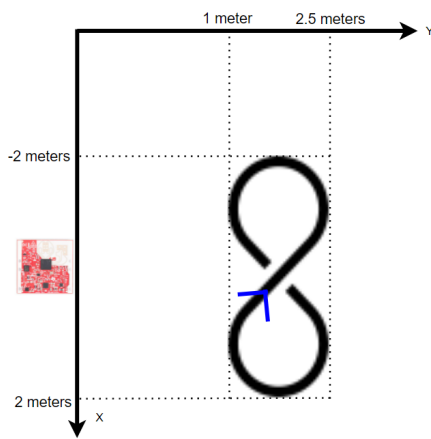
5.3 Real Time People Tracking

The tests performed for tracking were done with a person following predefined routes and measuring the results returned by the people counting and tracking module presented in chapter 4.

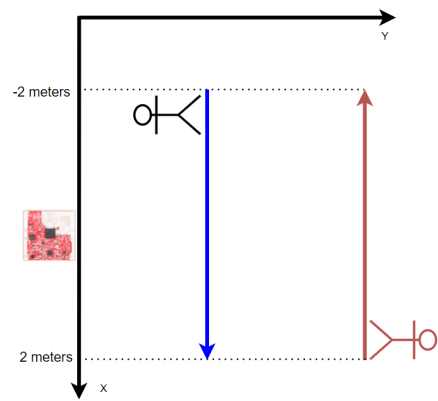
5.3.1 Scenario Setup

The section shows the different scenarios tested for the tracking module. The first three scenarios are done with a single person walking in front of the radar following different patterns with constant speed. The first test (fig. 5.3) is a person walking towards the radar trying to maintain the same angle of arrival, the second one (fig. 5.4) is walking across the FoV (crossing all azimuth angles), the third test (fig. 5.5) is doing a more complex path to show the capabilities of tracking azimuth and range variations. The final test (fig. 5.6) is done with the help of a second person moving in the opposite direction.





(a) Setup for complex route test.



(b) Setup for two targets moving in opposite direction test.

Figure 5.2: Tests performed for People Tracking.

5.3.2 Tests Results

The results from the tests presented before are shown here, the plots are the top view (X-Y axis) of the 3D space created by the GUI. The green points are the centroids detected by the clustering algorithm and the blue line is the EKF state vector..

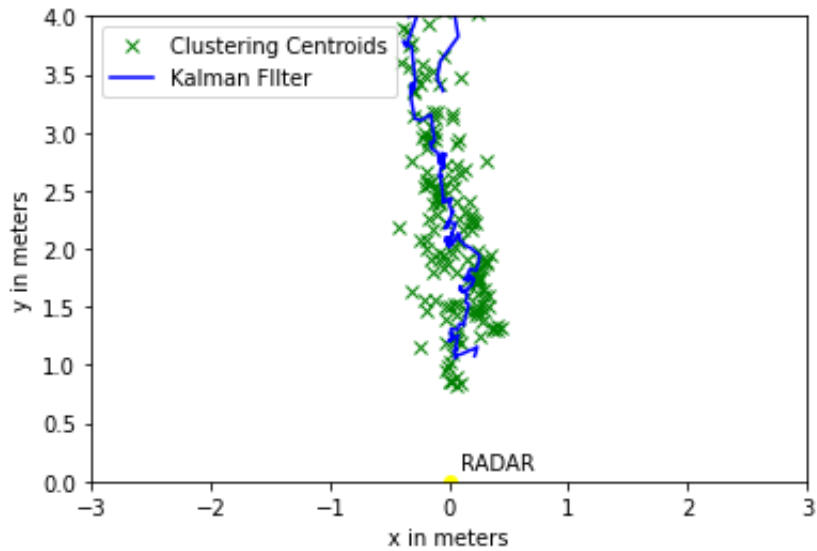


Figure 5.3: Results for perpendicular walking pattern of a person with a slow speed.

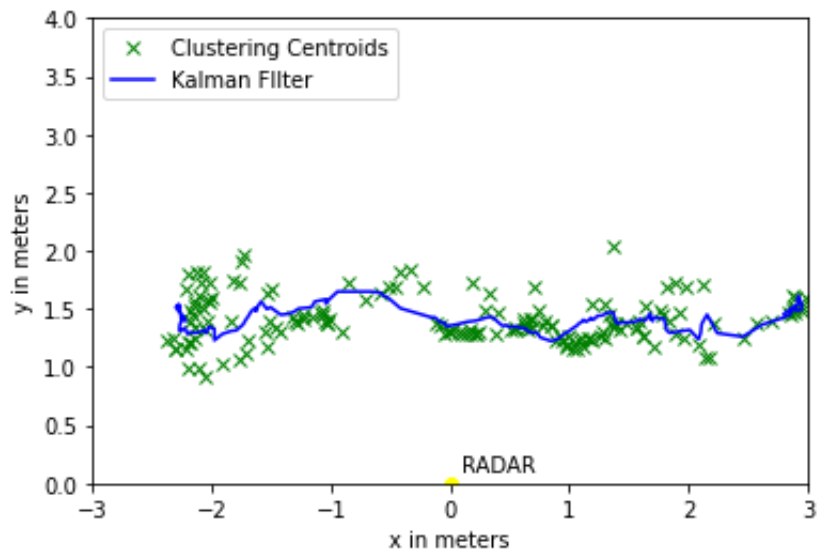


Figure 5.4: Results for the parallel test.

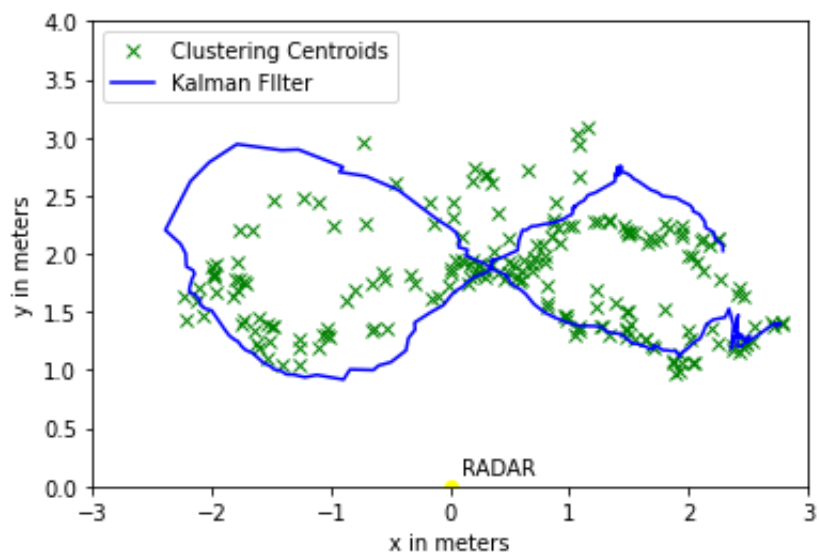


Figure 5.5: Results for a complex walking pattern.

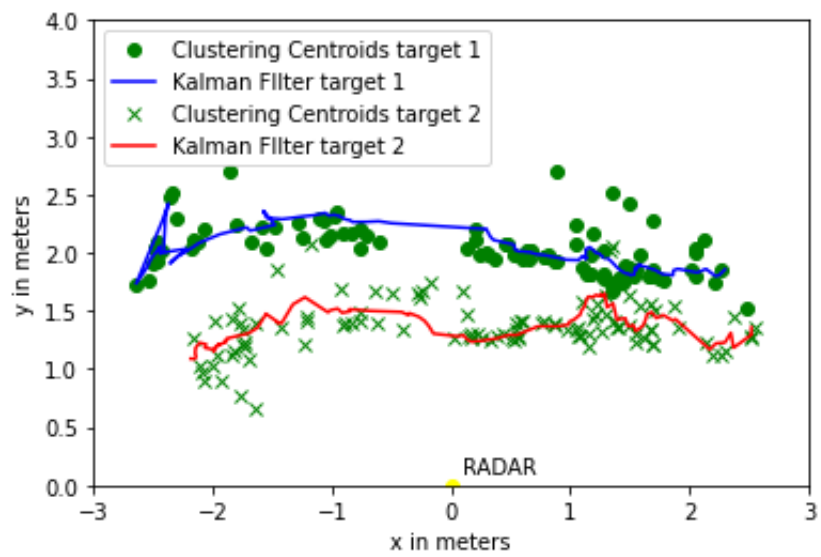


Figure 5.6: Results for two targets moving parallel to each other in opposite directions.

5.3.3 Tracking Analysis

By observation of the results presented above, the tracking module is capable of following the measurements and in the case of multiple persons, the tracking can follow the right person. From the graphics (5.3, 5.4, 5.5 and 5.6) it is possible to see one of the major concerns regarding the development board used, there is an azimuth region which the radar cannot originate a uniform density of detections. This region was found in multiple tests and can be seen in figures 5.4,5.5 and 5.6, the azimuth angles region between $[-10,-2]$ degrees have a much smaller density of detections in comparison to other regions of the FoV. This raised some problems which will later be addressed in the chapter 6.

5.4 Sensor Fusion

The camera used for this test was a webcam from the PC HP Probook 450G3. Each image contains the bounding box relative to the detected target location projected into the image and the points returned by the radar. Each point color represents the respective frame where the points come from because the application adds persistence to the image to fill in the image with more data.

5.4.1 Image with Radar Data

In this test, a person is moving towards the radar from the negative azimuth side of the radar. As stated before, this is the region where the radar fails to detect a great density of points and can be seen by the number of detected points in that frame which is less than ten.

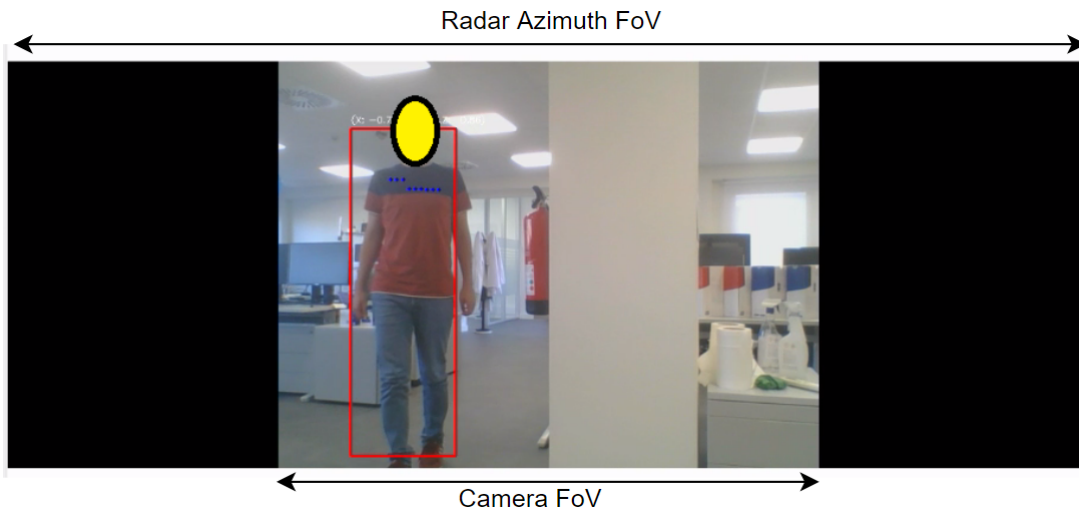


Figure 5.7: Point cloud plus the target location projection onto the camera.

5.4.2 Extending the FoV

The Radar and the camera have different FoV with the radar extending the azimuth view, the black columns represent the camera blind side which is fulfilled by the radar vision. In the image (fig. 5.8) below, a person is walking out of the camera FoV but still inside the Radar vision area.



Figure 5.8: Target moving out of camera vision but still in radar FoV.

5.4.3 Tracking in Low Visibility Conditions

Because the radar is insensitive to weather conditions, the data provided remains viable and the image can still be built with detections from the application as seen in figure 5.9.



Figure 5.9: Tracking in extreme visibility conditions.

5.5 Analysis of the Results Obtained

The counting performance of the radar was found successful in controlled environments with the counting application able to locate people with high accuracy and had good results counting people most of the time when people were spatial separated and stationary with just the small movements people do without thinking of them. Even when the correct counting failed, it was capable of partially detecting people present in the FoV which is of major importance if we consider the Radar as a complementary sensor to provide information when other sensors fail.

The tracking performance is heavily influenced by the behavior of the Extended Kalman Filter which requires some tuning on some matrices parameters, the results show the tracking module is able to follow the path a person is taking. The accuracy of the measurements done through the Kalman filter in relation to the ground truth shows the parameters selection was not the best because the path could not track huge changes of directions as shown in figure 5.5, where the Kalman update stage failed to follow during a fast "U" turn.

Enhancing the camera vision with radar data allowed the creation of a security system capable of detections without a significant impact on the performance of the live video provided by the webcam. The projection of points and objects tracked into the image did not have a major impact on the video FPS, adding just a slight delay between the moment the tracking module outputs the results and the moment they are displayed in the image.

Chapter 6

Conclusion and Future Work

This chapter describes the conclusions drawn after the execution of this dissertation and some of the future work that can be developed with the use of the Radar data.

6.1 Conclusion

As can be seen from the results presented, the FMCW radar has the potential for people detection and tracking without the use of sensitive data that could raise concerns with GDPR (General Data Protection Regulation). The work developed in this dissertation showed the tracking performance achieves good results when people are physically separated so that the clustering does not label multiple clusters as the same cluster and also because the tracking algorithm requires that people walk through paths that are spatially distant from existent tracks or the tracking module will allocate the moving cluster to the wrong track.

The radar of choice was TI IWR6843ISK-ODS which is a radar designed for ceil-mount applications, the maximum range distance is short and makes so that the paths designed and tested were limited to a small room. Because of that the people counting tests were done in limited conditions which translated to a small number of people in the FoV not violating the personal space of each person involved in the tests. The main concern is the development of applications based on the radar is the clutter which if we wanted to distinguish from a real detection, a new degree of complexity would be added to the system. In the people counting and tracking module, the clutter is tackled by measuring the power from the clusters originated and assuming a clutter cluster will have a much smaller power since it traveled a longer distance. This is effective in reducing the clutter-induced detections, but could also mean a loss of valuable information from real targets at a longer distance which would originate smaller power clusters.

The detection and tracking of multiple persons raised another problem such as the maximum number of points detected by the radar, which is a feature limited by the memory available on the board. The maximum number of points is software-defined at the CFAR algorithm with a maximum of 150 possible points, if the CFAR threshold has a low value, a high number of points are detected and if there is a limit, some points with valuable

information about a target may not be considered. This comes into consideration since the main obstacle regarding this dissertation project was the selected board which is not capable of detecting points uniformly across the FoV, more specifically at the $[-10,0]^\circ$ azimuth, due to most likely manufacture flaws, which generated a much smaller density of points in that region. To detect points crossing that region, a smaller sensitivity at the CFAR algorithm had to be implemented so the clustering had enough points to group them, consequently, as stated before, some points would be lost because of the limit on the detection.

6.2 Future Work

The next step to make this technology ready for a real-life scenario, like in a street or even in a room is to move to another FMCW radar board with an antenna design capable of detecting objects at a larger distance at the expense of FoV and resolution such as TI IWR6843ISK. Algorithm-wise, one of the improvements to be done is to add a function in the association stage of the tracking module, which would return the most likely target for a cluster, based on the velocity provided in the state vector and the velocity measured by the radar. This would improve the performance in resolving and tracking targets that are close to each other and moving in different directions.

For the future, a people counting and tracking application powered by Machine Learning is in thought. With the help of computer vision digital processing techniques which nowadays are able to detect and track people on an image with good accuracy, a method of supervised learning could be applied to Radar ML. The Radar ML will use a different set of inputs in comparison to ML applied on images, this means there is a possibility of the Radar outperforming the image techniques in some cases. An obvious case that deteriorates the performance of image ML is when the light conditions change, in the case of total blackout on the illumination system these techniques would fail to detect people where the Radar would still be able to detect since the data is unaffected by light. This work enables the possibility of other applications using the radar, such as a live plant of a building with multiple FMCW radars in a mesh grid, detecting and tracking all people in the building in real-time. This would be useful and non-invasive since it could be installed inside the walls.

Bibliography

- [1] M. Skolnik, *Radar Handbook, Third Edition*. McGraw-Hill Education, 2008.
- [2] J. C. Maxwell, “A dynamical theory of the electromagnetic field,” *Nature*, vol. 119, pp. 125–127,
- [3] H. Hertz, “Electric waves: Being researches on the propagation of electric action with finite velocity through space,” 1962.
- [4] J. A. Scheer and W. Holm, “Introduction and radar overview,” 2010.
- [5] I. Technologies. (2018). “Industrial radar sensing.” Accessed on 25/10/2021, [Online]. Available: https://www.infineon.com/dgdl/Infineon-Presentation_24GHz+Sensing-PPT-v01_00-EN.pdf?fileId=5546d4625debb399015e0a4773e042e7.
- [6] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, “Millimeter-wave technology for automotive radar sensors in the 77 ghz frequency band,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, pp. 845–860, 2012.
- [7] H. Zhou, P. Cao, and S. Chen, “A novel waveform design for multi-target detection in automotive fmcw radar,” *2016 IEEE Radar Conference (RadarConf)*, pp. 1–5, 2016.
- [8] K. Ramasubramanian, “Mmwave radar for automotive and industrial applications,” Texas Instruments, Tech. Rep., 2017.
- [9] T. Instruments. (Feb. 2021). “3d people counting demo software implementation guide.” Accessed on 20/09/2021, [Online]. Available: https://dev.ti.com/tirex/explore/content/mmwave_industrial_toolbox_4_7_0/labs/people_counting/docs/3D_people_counting_demo_implementation_guide.pdf.
- [10] S. Rao, “Introduction to mmwave sensing: Fmcw radars,” Texas Instruments, Tech. Rep., 2018.
- [11] J. W. Cooley, P. Lewis, and P. D. Welch, “The fast fourier transform and its applications,” *IEEE Transactions on Education*, vol. 12, pp. 27–34, 1969.
- [12] J. Li and P. Stoica, “Mimo radar with colocated antennas,” *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 106–114, 2007. DOI: 10.1109/MSP.2007.904812.
- [13] R. Feger, C. Wagner, S. Schuster, S. Scheiblhofer, H. Jager, and A. Stelzer, “A 77-ghz fmcw mimo radar based on an sige single-chip transceiver,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 57, pp. 1020–1035, 2009.
- [14] V. Krishnaveni, T. Kesavamurthy, and B. Aparna, “Beamforming for direction-of-arrival (doa) estimation—a survey,” *International Journal of Computer Applications*, vol. 61, pp. 4–11, 2013.
- [15] L. L. Scharf and A. Pezeshki, “Virtual array processing for active radar and sonar sensing,” *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pp. 740–744, 2006.
- [16] J. Capon, “High-resolution frequency-wavenumber spectrum analysis,” 1969.
- [17] T. S. D. (Shendkar), D. Simunic, and R. Zentner, “Comparison of doa estimation algorithms in sdma system,” *Automatika*, vol. 54, pp. 199–209, 2013.
- [18] M. Wang, *Bartlett beamformer and mvdr beamformer in the browser*, Accessed on 5/10/2021, Jun. 2017. [Online]. Available: <https://research.wmz.ninja/articles/2017/06/bartlett-mvdr-beamformer-in-the-browser.html>.

- [19] EETimes, *Radar basics - part 2: Pulse doppler radar*, Accessed on 20/09/2021, May 2011. [Online]. Available: https://www.eetimes.com/radar-basics-part-2-pulse-doppler-radar/?utm_source=eetimes&-utm_medium=relatedcontent.
- [20] M. Miacci and M. C. Rezende, “Basics on radar cross section reduction measurements of simple and complex targets using microwave absorbers,” 2012.
- [21] M. A. Richards, “Fundamentals of radar signal processing,” 2005.
- [22] T. Instruments. (Oct. 2018). “60ghz mmwave sensor evms.” Accessed on 25/09/2021, [Online]. Available: <https://www.ti.com/lit/ug/swru546d/swru546d.pdf>.
- [23] *Iwr6843, iwr6443 single-chip 60- to 64-ghz mmwave sensor*, SWRS219E, Texas Instruments, 2018. [Online]. Available: <https://www.ti.com/lit/ds/swrs219e/swrs219e.pdf>.
- [24] K. Ramasubramanian, “Using a complex-baseband architecture in fmcw radar systems,” Texas Instruments, Tech. Rep., 2017.
- [25] T. Instruments. (Mar. 2018). “People tracking and counting reference design using mmwave radar sensor.” Accessed 20/02/2021, [Online]. Available: <https://www.ti.com/lit/ug/tidue71d/tidue71d.pdf?ts=1634261115695>.
- [26] B. D. Carlson, “Covariance matrix estimation errors and diagonal loading in adaptive arrays,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, pp. 397–401, 1988.
- [27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *KDD*, 1996.
- [28] R. Labbe, *Kalman and bayesian filters in python*, Accessed on 22/08/2021, May 2020. [Online]. Available: <https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python>.
- [29] T. Instruments, *Tracking radar targets with multiple reflection points*, 2017.
- [30] *The geometry of perspective projection*, Accessed on 26/10/2021. [Online]. Available: <https://www.cse.unr.edu/~bebis/CS791E/Notes/PerspectiveProjection.pdf>.

Appendix

Abstract Presented at Conference URSI GASS 2021

FMCW Millimeter-Wave Radar-Augmented People Tracking Solution for Video Surveillance Systems

Bruno Santos⁽¹⁾, Arnaldo S. R. Oliveira⁽¹⁾, Nuno Borges Carvalho⁽¹⁾, Rui Fernandes⁽²⁾, Andrea Cannizzaro⁽²⁾
and Pedro Miguel Cruz^{*(2)}

(1) Instituto de Telecomunicações and Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, Aveiro, Portugal; e-mail: b.santos@ua.pt; arnaldo.oliveira@ua.pt; nbcarvalho@ua.pt

(2) Bosch Security Systems S.A., Ovar, Portugal; e-mail: ruifilipe.fernandes@pt.bosch.com; andrea.cannizzaro@pt.bosch.com; pedro.cruz4@pt.bosch.com

People tracking in video surveillance systems (e.g. camera) requires clear images to detect a person or object in a reliable manner. Extreme conditions like zero-lux, presence of smoke or difficult outdoor weather conditions create challenges to artificial intelligence/machine learning (AI/ML) image processing algorithms.

Multi-sensory platforms composed of different sensing typologies act as possible solutions to circumvent such challenging situations, bringing complementary characteristics that when fused together, increase system performance (e.g. better accuracy in people/object detection) and aggregate outcomes that are more informative (e.g. velocity and direction) [1].

In this abstract, we introduce the development and test of a frequency-modulated continuous wave (FMCW) radar working on the 60 GHz millimeter ISM band, which allows complementing surveillance camera real-time video streaming with intelligent insights [2] from the scene (e.g. distance of an object, motion speed and/or angle of arrival), even when operated in tough scenarios. Due to the spatial diversity of the antenna pattern, one can create 3D point clouds, which are further processed to apply tracking algorithms to pinpoint relative location of the identified objects and may even permit the objects' classification as human or non-human by considering the physical dimensions of the point cluster.

The piece of hardware employed in the experiments, shown in Figure 1, is a 60 GHz Radar model IWR6843ISK-ODS [3] from Texas Instruments. It has a built-in RF system and digital signal processing (DSP) unit that outputs through UART a point cloud with static and Doppler-affected points. With the help of a clustering algorithm (DBSCAN) [4] and by feeding only the points with motion, one can accurately pinpoint the location of a moving object/person; calculate its speed and identify the direction of movement.

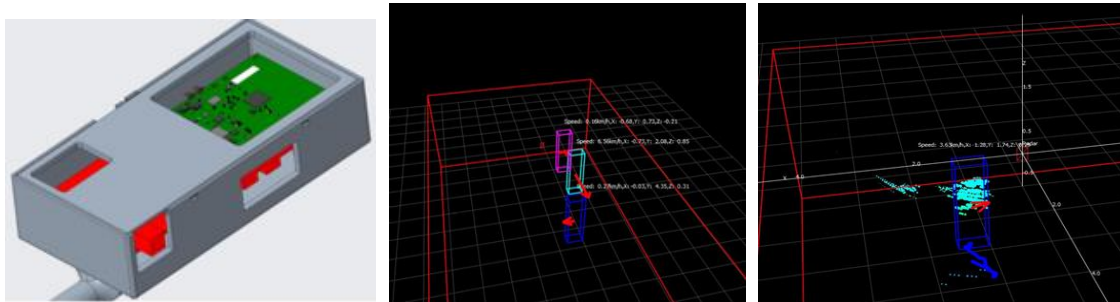


Figure 1. (left) Developed Radar package, (center) post-processed point cloud with three objects: a person sitting close to the radar positioned in (0, 0, 0) (magenta), another person stepping away from radar position at 6 km/h (cyan) and a small object with a right-turn movement (dark blue) and (right) the point cloud with all moving reflections (cyan) and their trajectory in blue.

References

- [1] Elmenreich, Wilfried. "An introduction to sensor fusion." Vienna University of Technology, Austria 502 (2002): 1-28.
- [2] Bosch Safety and Security Systems, "Intelligent Insights: Making data visible and usable", 2021. Online: <https://www.boschsecurity.com/xc/en/solutions/management-software/intelligent-insights/>.
- [3] Texas Instruments Inc., "IWR6843 intelligent mmWave overhead detection sensor (ODS) antenna plug-in module", Technical Datasheet, 2021. Online: <https://www.ti.com/tool/IWR6843ISK-ODS>.
- [4] Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." Kdd. Vol. 96. No. 34. 1996.